PhD Thesis

FAÇADE PROCESSING IN STREETSIDE IMAGES

Automated Framework for Streetside Dataset Processing using Context and Multi-View

Michal Recky

Graz University of Technology
Institute for Computer Graphics and Vision

Thesis reviewers

Prof. Dr. Franz Leberl
Doc. RNDr. Andrej Ferko, PhD

Graz, July 2012

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

_____	_____	_____
Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

_____	_____	_____
Ort	Datum	Unterschrift

Acknowledgments

First of all, I would like to thank my advisor Franz Leberl, who permitted me to work on my PhD in Graz and provided important directions in the course of my studies and my second advisor Horst Bischof for guidance and many advices. I would like to thank my colleagues in ICG for inspiration and professional advises. Special thanks go to the Learning, Recognition and Surveillance discussion group for many fruitful ideas. Among other colleagues, I would like to thank Arnold, Andreas and Hayko for providing insights into my research topic. Furthermore, I would like to thank the administrative staff of ICG, especially Renate, for helping me acclimate in ICG and Austria. Finally, I'm very thankful towards my parents for support and encouragement.

My work has been supported by the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

Abstract

In this work we introduce the approach for an interpretation of a large variety of streetside image datasets. We suggest the complete workflow, starting with a single image, or with large datasets of aligned and matched images and finishing with semantic information about the scene, sufficient to reconstruct façades of buildings and other principal areas. We present a set of state-of-the-art algorithms which tackle different problems with streetside image processing. Each algorithm presents innovations and focus on the two primary aspects – context and multi-view. We suggest that our dataset domain (streetside images) present an opportunity to understand the context and its impact on object detection/recognition. Urban scenes adhere to an inherent organization of man-made objects; therefore contextual cues contribute in an analysis effort. In several different algorithms described in this work we utilize these contextual relationships using global graph models. We examine different Random Fields models, present several applications of the context and in the final chapter we introduce a new global model for a multi-view scenario.

Our second focus of research is a multi-view, or « redundant » dataset. We not only extend algorithms (semantic segmentation, window detection, façade separation) into a multi-view scenario but also examine different organizations of datasets, consider different sources of multi images (industrial datasets, crowd sourcing) and suggest specific approaches for different multi-view scenarios. We compare single and multi-view results for each algorithm and different scenarios of multi-view datasets processing.

Problems of streetside image processing and analysis are currently a focus of many research teams [Simon et al., 2011], [Agarwal et al., 2010], [Müller et al., 2007]. Our work is specific in providing a complete workflow, addressing many problems that are present in this domain, but also our focus on two specific aspects provided us with an opportunity to contribute into the field with new ideas. We achieve approx. 93-97% precision in semantic segmentation of principal areas in streetside photos (building, roof, sky, road, vegetation...), up to 97.1% precision is segmenting specific façades and approx. 96% detection rate for façade elements in average.

Kurzfassung

In dieser Arbeit stellen wir den Ansatz für eine Interpretation von einer Vielzahl von Datensätzen von Straßenbildern. Wir schlagen den kompletten Workflow vor, beginnend mit einem einzigen Bild, oder mit großen Datenmengen von Bildern ausgerichtet und aufeinander abgestimmt und endend mit semantischen Informationen über die Szene, die ausreicht, um Fassaden von Gebäuden und anderen wesentlichen Bereichen zu rekonstruieren. Wir präsentieren eine Reihe von topmodernen Algorithmen, die unterschiedliche Probleme mit Bildverarbeitung angehen. Jeder Algorithmus präsentiert Innovationen und konzentriert sich auf beide primären Aspekte - Kontext und Multi-View. Wir schlagen vor, dass unser Datenbestand-Domain (Straßebildern) präsentiert die Möglichkeit, den Kontext und dessen Auswirkungen auf Objekterkennung / Anerkennung zu verstehen. Stadtbilder halten die inhärenten Organisation von Menschen geschaffenen Gegenständen ein und helfen den kontextuellen Regeln bei der Analyse. In dieser Arbeit beschreiben wir in verschiedenen Algorithmen diese inhaltlichen Zusammenhänge mit globalen Graphen-Modelle. Wir prüfen verschiedene Random Fields Modelle, präsentieren mehrere Anwendungen des Kontexts und im letzten Kapitel führen wir ein neues globales Modell für ein Multi-View-Szenario ein.

Unser zweiter Schwerpunkt der Forschung ist ein Multi-View oder «Redundanter» Datensatz. Wir Erweitern nicht nur Algorithmen (semantische Segmentierung, Erkennung von Fenstern, Fassaden-Trennung) in ein Multi-View-Szenario, sondern prüfen auch verschiedene Organisationen von Datensätzen, betrachten verschiedene Quellen von mehreren Bildern (Industrie-Datensätzen, Crowd Sourcing) und schlagen spezifische Ansätze für verschiedene Multi- Szenarien vor. Wir vergleichen Einzel- und Multi-View-Ergebnisse in jedem Algorithmus und verschiedene Szenarien der Multi-View-Datensatz-Verarbeitung.

Probleme der Straßenbildverarbeitung und analyse sind derzeit ein Schwerpunkt vieler Forscherteams [Simon et al., 2011], [Agarwal et al., 2010], [Müller et al., 2007]. Unsere Arbeit ist spezifisch bei der Bereitstellung eines kompletten Workflows, Adressierung an viele Aspekte, die gegenwärtig in diesem Bereich sind und fokussiert sich auf zwei spezifische Aspekte, die uns die Gelegenheit bieten, in das Feld mit neuen Ideen beizutragen. Wir erreichen ca. 93-97% Präzision in semantischer

Segmentierung der wichtigsten Bereiche in Strassenbildern (Gebäude, Dach, Himmel, Straßen, Vegetation ...), bis zu 97,1% Präzision bei Segmentierung von spezifischen Fassaden und ca. 96% Erkennungsrate für Fassadenelemente im Durchschnitt.

Contents

Chapter 1

Introduction

1.1 Motivation

In the modern world, the urban environment is a place where majority of human population spends most of its time and where the Internet has the greater impact. “Virtual Earth”, the virtual habitat and virtual cities are concepts that have a very short history of approximately 10 years and imply the evolution of 3D models of urban environments [Kimchi, 2009], [Leberl, 2003]. Their creation would be exceedingly costly were it not possible to automate. Scene understanding and object recognition in such an environment has become essential for many computer vision algorithms applied in this domain. However, even rapid progress in computer vision has left recognition task a challenging problem. The best algorithms today cannot compete with a human vision in a scene understanding. A possible reason for this is that in a human vision, object recognition is a global process. In computer vision, many algorithms are focused on a specific object class and tend to neglect overall context information in the image. Background information around such object is considered ineffective and gets removed. But in a human vision, background and contextual information play a major role in a recognition task [Oliva and Torralba, 2007]. It is therefore suggested that the context is a basic improvement of a successful recognition algorithms [Heitz and Koller, 2008], [Rabinovich et al., 2007].

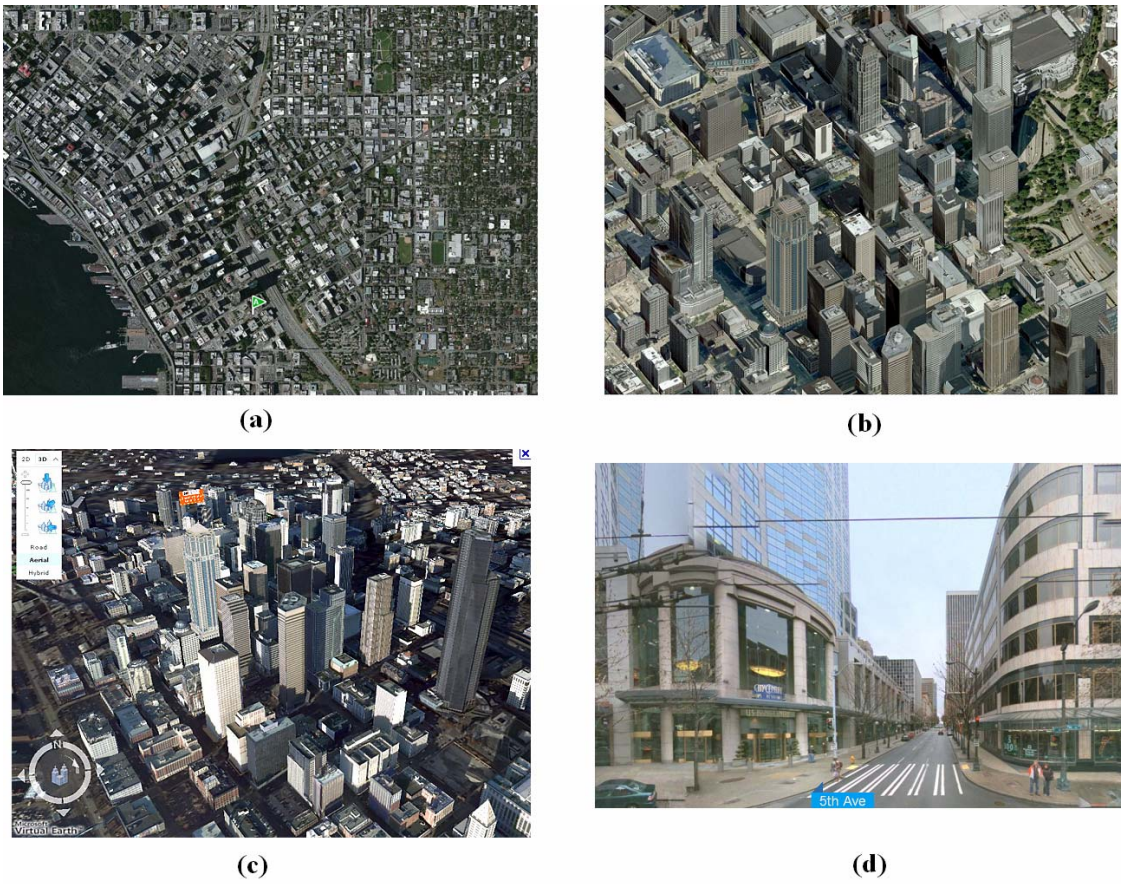


Figure 1.1: Evolution of Microsoft's environment mapping projects. (a) satellite image (b) bird's-eye view (c) virtual earth (d) streetside view. In each of iteration a level of detail and data interpretation has increased. Currently, Microsoft abandoned the 3D modeling of cities and uses 3D only in the street level presentation. Aerial view has evolved from 2D (a) into 3D (c), however the streetside view is still composed of geo-registered 2D images (textures) arranged in a 3D structure of the scene and not 3D models (images show New York center) [Bing Maps, 2011].

For example, many pedestrian detection algorithms do not account for the fact that all pedestrians are walking on a ground plane [Liping and Wentao, 2009], [Huang et al., 2010]. This approach is based on the assumption that objects are defined primary by their visual features. But for example, windows can be defined as visual objects (frame, reflective surface, rectangular shape), or as objects in a context of urban environment (objects at a façade plane, arranged in well defined, repetitive patterns). In visual-based recognition algorithms, it is sufficient to know the visual features for

successful recognition, which is often an easier approach. However, in a context-based approach, additional information about the scene needs to be extracted from an image (e.g. spatial/geometric relations, background classes…). The motivation to use contextual information came from the fact, that the urban environment has very strong geometric and utility organization. This organization was developed as a natural requirement during habitats construction and can provide effective cues for any computer vision algorithm working within this environment [Lee and Nevatia, 2004]. From the application point of view, our motivation is based on two computer vision efforts – Urban Modeling and Scene Understanding

Urban Modeling

An effort to map human surroundings has been present long before a digital photography was introduced. With the application of a digital image processing and computer vision come automation of the task with superior precision and utilization. Combination of terrain data and additional (e.g. geographic, municipal, utility…) information is known as the Geographic Information System (GIS) [Pidwirny, 2006]. An introduction of GIS into internet-hosted environmental models have been presented in geospatial internet mapping platforms like Microsoft Bing Maps [Bing Maps, 2011] (see Figure 1.1), or Google Maps [Google, 2012(III)]. In first iterations of such projects satellite/aerial images have been implemented and basic data interpretation has been introduced (e.g. roads, businesses). These platforms are now undergoing the transformation into 3D, human-scale environments – urban models. A demand for improved levels of detail, a quality of visualization and an introduction of novel applications lead to the need to interpret urban environments even further. Such interpreted data cannot only be stored in a compact form and thereby reduce the amount of data needed to be transferred through internet connections, but also represent the opportunity for users to build applications that would not be possible if the Internet maps were merely images. In the next iteration, mapping platforms utilized street-view models to cope with requirements for human scale details and an immersive experience of the internet user [Google, 2012(II)]. Subsequently, the demand for an interpretation at this level has arisen. In our work, we examine streetside data for the purpose of interpretation and extraction of information relevant for modeling.

(a) **(b)** **(c)** **(d)**

Figure 1.2: Demonstration of the workflow described in Simon et al. for single-view 3D modeling of building. (a) building façade, (b) shape grammar, (c) façade model, (d) street model [Simon et al., 2011].

Given the predefined geometry of urban objects, semantic information can also be used directly in a reconstruction process. [Hoeim et al., 2005] and more recently, [Simon et al., 2011] show how 3D models can be constructed from single image using a semantic segmentation or shape grammars (see Figure 1.2). In our work, we focus a large portion of the research on the interpretation of façades and façade's elements, but we also provide general information about most of the street level image's areas. Building façades are objects of major importance in streetside view scenes and as such, the primary targets of many urban modeling efforts. Due to a large amount and variety of data, interpretation process has to be fully automated and robust as the results would be unaffordable.

Scene Understanding

Many applications of computer vision algorithms working in an urban environment could benefit from the scene understanding. When the context information about a scene is available as an a-priori knowledge, surveillance, traffic control, or tracking algorithms could better cope with problems like occlusions or false positive detections, even if a 3D model of a scene is not available, or implemented [Perko and Leonardis, 2008]. Interpreted areas can remove false correspondences (e.g. from occlusions, sky, moving objects) using only context information. Scene understanding is essential in

most automatic navigation algorithms (robot navigation, car driving assistance, automatic driving) [Correa, 2009], [Micusik et al., 2012]. In recent years, the demand for computer vision algorithms working in urban environments has greatly increased with the introduction of new generations of smart phones with digital camera and increased computational power. Many mobile applications such as photo processing, augmented reality or geo-location require a certain level of image understanding [Hays and Efros, 2008]. Such applications require fast and reliable computing with limited data. To meet these requirements, a context in an urban environment can play a crucial role.

When considering such requirements, we decided not to focus on a specific element of streetside scenes, but instead introduce novel methods and approaches in a computer vision that work with a specific geometry and datasets of urban scenes. Such methods provide means to interpret streetside images in different detail levels - from general surfaces to façade elements.

1.2 Urban Environment

We can divide human habitats into rural areas and urban areas by the properties they exhibit [Census, 2012]. Rural areas are defined as low population density regions with specific type of buildings. Most villages and hamlets are considered rural areas. On contrary, urban areas exhibit large population density and are considered core population and economic activity centers within a larger metropolitan area. Urban areas can be further divided into residential, commercial and industrial zones (see Figure 1.3). Zoning has been introduced as an architectural paradigm during the end of 19th century in Europe [Wikipedia, 2012]. Before this date, only naturally evoked zoning can be observed.

Residential zones are areas, where most of population housing is concentrated. The most common building type is a residential building. We can identify different types of residential building as single family housing, multiple family housing (apartments, duplexes, town homes, condominiums) and mobile homes. As residential zones cover a majority of the urban space, it is important to perform more detailed analysis of this type of environment.

(a) **(b)** **(c)**

Figure 1.3: Examples of images from (a) residential zone, (b) commercial zone, (c) industrial zone. Notice different styles of buildings and different sets of objects (vegetation, pedestrian concentration, etc.) in each zone. Images obtained from Google search engine [Google, 2012(II)].

Commercial zones are areas, which provide a community with economic resources. The majority of buildings in the commercial zones are business oriented. We can identify retail business buildings, wholesale/distribution oriented buildings, financial establishments and offices. Commercial zones generally cover approximately 5% of an urban area.

Industrial zones are areas with concentrated industry infrastructure. Factories, manufacturing plants, storing depots, light industry buildings and offices are commonly located in industrial zones.

Additionally, specific building types may vary due to a historical context, when the building was build. Many European cities have a historical centre, where most of buildings were preserved from several centuries ago. Compared to this, modern architecture buildings, even with the same function as historical ones, may exhibit very different visual features. Therefore, it might by useful to establish some kind of historical and location context before we will attempt any kind of façade element detection or recognition.

(a) **(b)**

Figure 1.4: The examples of building façades. Each "façade" is marked with a different color. In the first picture (a) building is composed of several façades – each façade is limited by (projected) corners of the building. In the second image (b) several buildings are present, each displaying one façade. In both examples, there are other façades present that are not clearly visible, thus are unmarked. In both images, occlusions (from vegetation, car…) to the façades are disregarded. Such occlusions are not considered to be parts of façades from the definition.

1.2.1 Definitions

In our work we refer to several specific terms extensively. In this section, we present the definitions for such terms.

Streetside Image – A digital image obtained from a street level in an urban environment. In general, such image is obtained by a human agent, or an automated system, using a digital camera with optical axis roughly parallel to the ground (± 30 degrees). We also assume such image is properly aligned (sky on the top, ground on the bottom). Streetside images are our primary application domain.

Façade – In the general definition, a building façade refers to one side of a building (usually a front side). Given such definition, we apply this term for a section of building bordered by two vertical building corners (see Figure 1.4 (a)). If buildings are connected to each other, we consider a corner to be located at the border between two

buildings (see Figure 1.4 (b)). Therefore, buildings can be sectioned into several façades vertically, but we do not consider horizontal sections – in all cases façades stretch form the ground to the roof, or to the building tops. A roof is not part of a façade and is considered a different object class. In digital images, the term "façade" refers to an area of an image, where a building façade (as defined before) is projected. We often refer to such area as a "planar façade" (or a façade plane), where the term "planar" refers to the shape of a building – as such area represents roughly a line in a floor plan. Therefore façades can be referred to as planar even if the actual building façade is not and does contain non-planar reliefs (e.g. pillars) or sections (e.g. opened windows). Such non-planar elements are usually approximated in a final application, if the 3D information is available.

This definition of "façade" can be extended in our work, if the application requires it. For example, if the façades are required to be rectified, this requirement is presented in the introduction for such application and from that point on the term "façade" refers to a rectified façade.

Note that the term "façade" is different from the term "façade class", which refers to all areas in the image projected from any façade. Therefore there can be several façades in the image, but only one façade class.

Façade elements – This term refers to any coherent object that is located on a façade and is part of it (window, ornament, relief…) or was added to a façade (shop sign, paintings…). In general, façade elements are considered parts of a façade in our work. All façade elements except windows are considered as parts of the façade class (this is due to the specifics in hand labeling of our ground truth). As this is not always the case in other methods and applications, we refer to this problem in discussion sections for specific methods. Complementary, parts of the façade that are not façade elements (such as an areas between windows) are referred to as "façade area".

Circulation space – Refers to a section of ground in urban environment that is not vegetation and serves a transportation purpose in general (e.g. roads, pavements). The term "Circulation space elements" refers to integral parts of such areas, such as traffic lights, poles, sidewalks, parking lots, etc.

Context – A high level, non-visual information about a projected scene is denoted as context. Contextual information describes relations (semantic, geometric, temporal, etc.) between objects in the scene. In our work, we use either geometric relations between objects to help in classification (e.g. windows are arranged in rows and columns), or semantic relations to limit search area for detection algorithm (e.g. windows are located on building façades). The term "local context" indicates that only a limited section of an image was examined for contextual cues, whereas if the term "global context" is used, entire image area was considered for contextual cues. For a more detailed context definition, see Section 3.1.

Multi-View – Is a notion that point towards a presence of a dataset containing a number of overlapping and matched images of the same scene. A "multi-view scenario" indicates that such dataset was used in a process, whereas a "single-view scenario" indicates a dataset with unmatched images. In our work, matching in a multi-view scenario was achieved either through 3D point clouds, LiDAR, or a manual labeling.

Industrial System Dataset (IS) – A dataset created by a professional camera setup, designed for an urban environment mapping. Such system is generally mounted on a vehicle and supplemented by additional sensor data, such as LiDAR, or a GPS. Several camera sensors with a fixed geometry are used for taking images while the vehicle is moving. The example of such system is given in Section 3.5.

Crowd Sourced Dataset (CS) – A dataset composed of images from different users, usually equipped with different, amateur, hand-held cameras. Images were collected without the intention for an urban mapping application. Such datasets are located at open online image hosting sites e.g. [Flickr, 2012], [Photobucket, 2012]. Images are unorganized; geo-tagging, camera calibration, or image labeling can be missing.

1.3 Framework for Urban Modeling

In this section we propose a framework for processing streetside image data. Our goal is to start with a single image or a stack of multiple images and end up with a semantic

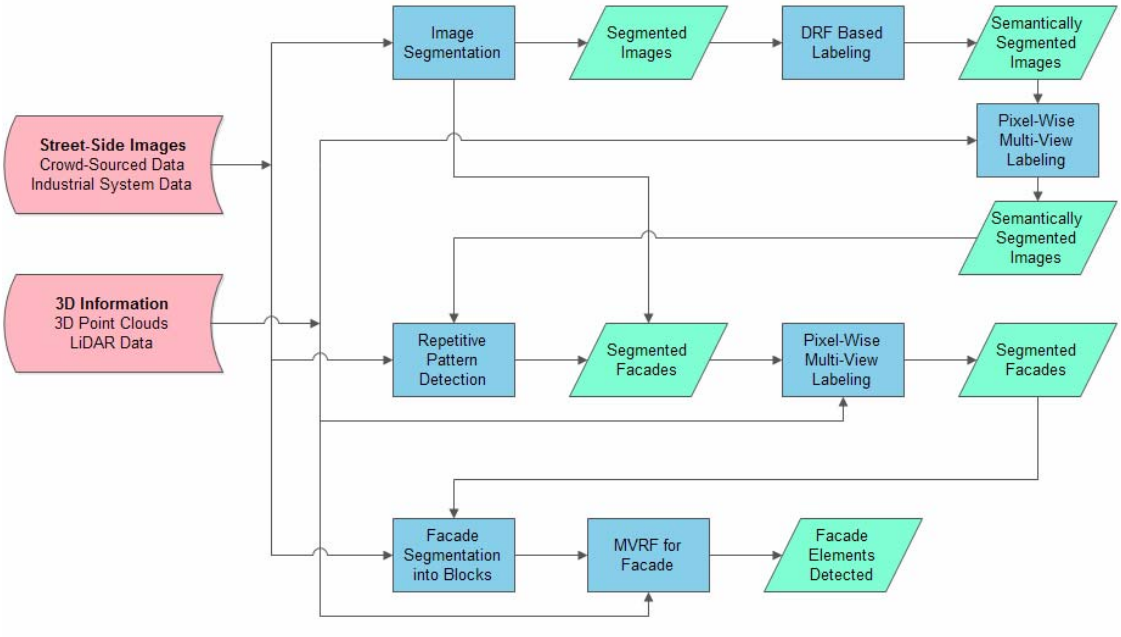


Figure 1.5: Framework diagram describing the workflow in more details. Red – input data, blue – involved methods, green – output data.

model of a scene. Our focus is to provide not just data for 3D modeling, but also labeling and descriptions of the scene and objects involved. The workflow is visualized in Figure 1.5. Three main lines of method-output chains represent tree main steps of framework which is described in separate chapters of this work – semantic segmentation (Chapter 4), façade separation (Chapter 5) and façade elements detections (Chapters 6 and 7).
Workflow can be described in following steps:

Algorithm 1.1

Input: single streetside image, or a stack of multiple images with additional data for matching;

1. Identify principal areas in the image and label them into classes – façade class (building), roof, vegetation, ground, grass area, sky, cloud, shadow and unidentified;

2. Identify separate façade in the image;
3. Detect and label façade elements;

Output: Principal areas labeled, separate façades identified, façade elements detected.

In each step, we consider separate cases for single and multi-view scenarios. We present novel methods, addressing existing problems in current applications. Our primary focus of research aims at the following:

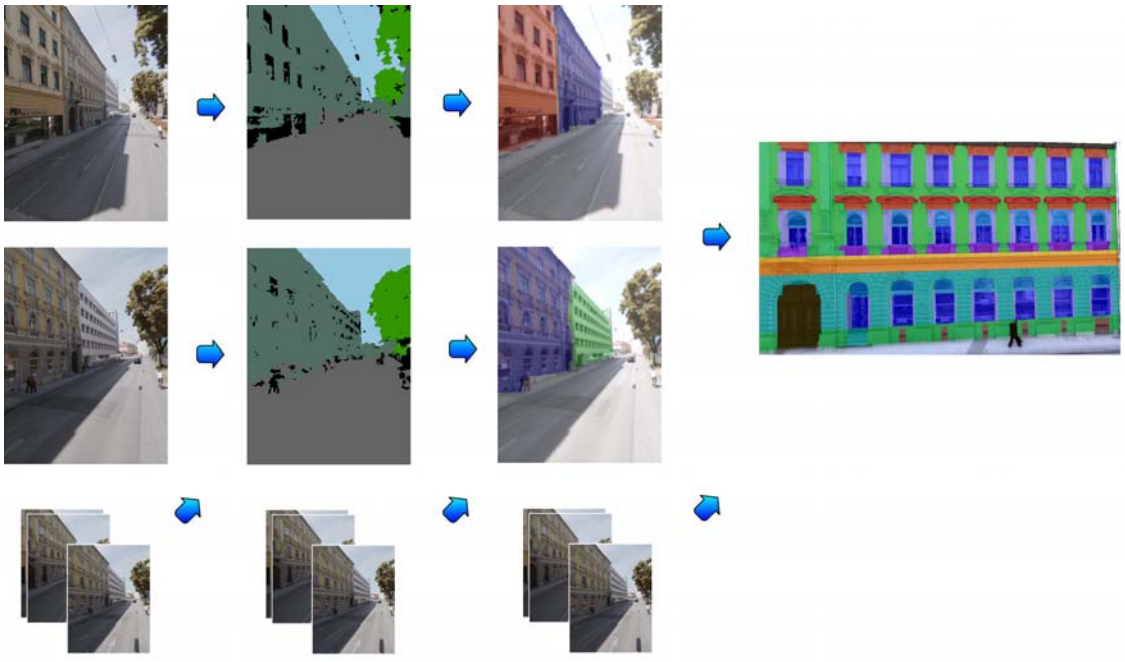
- Involvement of context in each step of the framework. We aim to address global rather then local context, giving us the advantage of superior input data, as we examine context between real objects, not artificial ones (like pixels or superpixels).

- We examine the effect of redundancy (multi-view) for each algorithm we describe. We consider different means of image matching and different precisions of matching. The evaluations of the transition effect between single and multi-view is presented for each step.

- We address two primary image acquisition methods for streetside imaging available – crowd sourced (community photos) datasets and industrial system datasets. We work with general images provided by arbitrary user with hand-held camera as well as with datasets created by industrial environment mapping setups. We provide detailed comparisons and evaluations for both approaches.

- We examine principal problems with each state-of-the-art algorithms involved in our research (e.g. locality of context in Random Fields model, limitations of Gradient Projection methods in complex façades) and provide solutions.

We tested given framework on real-world data (see Figure 1.6), namely on three streets and one square of city Graz. Two of the streets were complemented with laser scanner data as a part of an industrial system setup. Images from the third street and the square were matched using a 3D point cloud obtained by visual matching methods.



(a) **(b)** **(c)** **(d)**

Figure 1.6: Framework for a streetside dataset processing presented in the example. (a) original images from a streetside dataset, (b) semantic segmentation of principal areas (each area label is coded with color, e.g. dark green for the façade class), (c) separate façades identified in each image, (d) labeling of façade elements for each façade. Element's class is coded with color (for explanation of color codes, see Figure 7.4). Each step uses results from a previous one as an input. Note that in each step, other images from a dataset are involved in a process to establish a multi-view scenario. In specific steps, other information can be involved in the process, such as 3D point clouds, laser scanner data or GPS data for image matching and registration. Final façade elements' interpretation is performed for each façade in each image, however involve information from all matched images.

Altogether, we involved around 600 images in this workflow. The size of this dataset is largely limited by the need of hand-labeled ground truth involvement in the testing process. For a comparison with automated methods, appropriate parts of images had to be hand-labeled – a process that is unfeasible for larger, city scale dataset. However, there is no reason, why the workflow could not be considered in city scale situations. Required results can be provided primary for historical city centers of European cities,

as such buildings spot repetitive patterns in a form of multi store levels and columns of windows. The workflow is also suitable for stand-alone single houses, even they do not spot repetitive patterns. Such houses are not connected to different façades from either side, thus don't need to be separated from other objects.
Results given by workflow are semantic information about the scene. Pixel-wise labeling of images is achieved, where major areas are identified – namely façades, roofs, ground, vegetation, sky, clouds and grass areas. Subsequently, a façade class is further processed – individual façades are identified and façade elements are detected. Such results provide all necessary data for a procedural modeling of urban spaces and construction of shape grammas models, such as [Simon et al., 2011].

1.4 Challenges

Objects in urban environments can be divided into two sets – temporal (pedestrians, vehicles, animals…) and permanent (buildings, vegetation, circulation spaces…). Urban modeling algorithms are focused primarily on permanent objects, as these provide essential information about an environment structure [Simon et al., 2011], [Müller et al., 2007]. Temporal objects in this case are mostly ignored and when present are considered occlusions (e.g. pedestrians occluding façades). This is due to a requirement for urban modeling to provide general information about the scene not bound to a specific timeframe. On the other hand, most scene understanding algorithms (surveillance, traffic control, tracking…) are focused on specific temporal objects and can use permanent objects only as contextual information [Perko and Leonardis, 2008]. As our focus is on the urban modeling applications, temporal objects are not considered our objects of interest.
The problem of data interpretation from streetside images can be compared to the data interpretation from aerial images, as both works in a same urban environment and to the same goal. But the different points of view influence the scale, level of detail and overall composition of the scene. In recent years, there has been a fast progress in automated image understanding and reconstruction methods from aerial photographs [Zebedin, 2010], [Kluckner, 2011]. The main advantage of aerial data as an input for computer vision algorithms is that the objects of interests (e.g. roofs, façades) play

dominant role in a composition of an image and are usually less occluded by temporal objects than in a streetside view.
The disadvantage is that at this scale, the level of detail of such objects is lower than in streetside images. For example, aerial cameras are limited by minimal flying altitude and resolution [Leberl et al., 2010(II)] (currently allowing to take images with resolution up to 3 cm/pixel), but streetside resolution can be arbitrary increased by closing the distance to the object to a required level. However, both the advantage and disadvantage make data processing in the domain of aerial data easier when compared to streetside data. In streetside images, the compositions of a scene are often more variable and the objects of interests are often occluded by temporal objects or vegetation. The viewing angle at the objects of interests and camera positions are more variable. The increased level of detail provides more challenge for visual algorithms.
Because of these factors, a simple application of visual classifiers to process streetside data is insufficient and additional cues are included in the process:

- In addition to present state-of-the-art methods, we also examine different sources of streetside images and evaluate the methods according to the organization of datasets (variation of viewpoints, additional data, etc.). Such evaluation provides information, which type and organization of dataset is most suitable for a specific urban environment.

- We employ context to provide an additional source of information as a prior knowledge in computer vision algorithms. This approach (in contrast to purely visual-based methods) benefits from the increased level of detail in improved robustness -more details provide more context to work with.

When processing data from urban environments, one must consider a different value (historical, cultural, social…) of different locations. Most urban modeling efforts are focused on generic urban locations (e.g. generic houses, shops, offices), but some specific locations, like monuments, historical buildings or landmarks require a special attention.

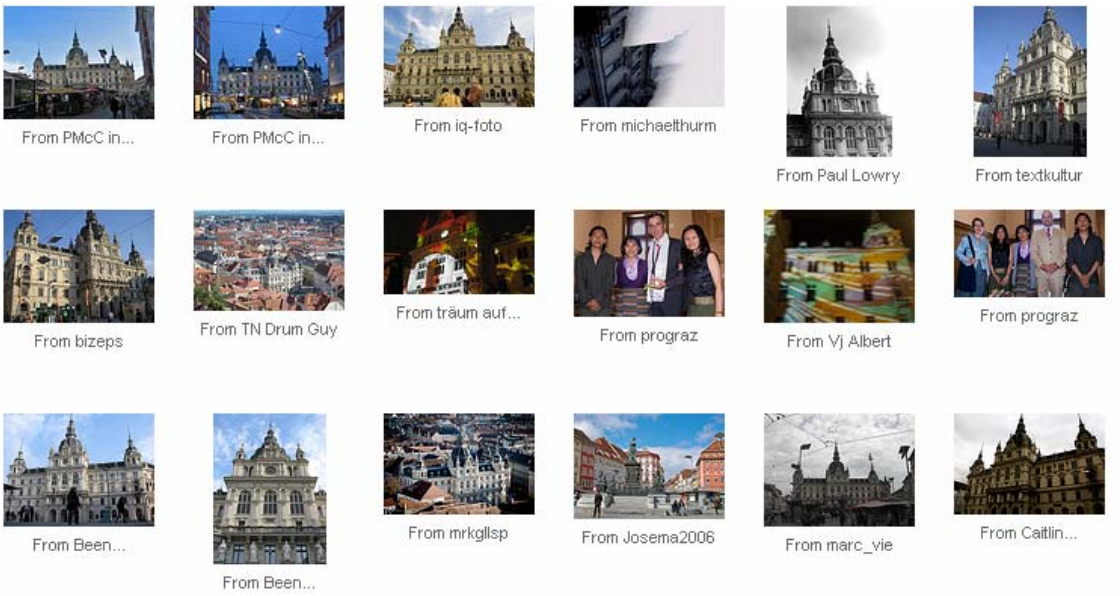


Figure 1.7: A response to a "Rathaus Graz" query in Flickr – an open source image database. Notice several images with a good point of view at the actual Rathaus building, providing a good source of information, but also some mislabeled images in the set (e.g. group of people) [Flickr, 2012].

Therefore, for generic locations, the industrial systems designed for general urban modeling and mapping provide sufficient information, but for special locations, we must search for additional sources of data. As the possible source of information, we examine the crowd sourced datasets. For important urban locations, there usually exists a large number of images located on online image hosting sites [Flickr, 2012], [Photobucket, 2012], [Picasa, 2012]. The primary problem with such datasets is the lack of organization (see Figure 1.7). Therefore, when working with multi-view scenarios, we examine two different strategies, each applicable for different levels of organization:

- *Interpretation first, matching second*. In this strategy, we first apply computer vision algorithm in each image in a stack separately (as if working in single-view mode). Subsequently, we match the interpreted images, and improve interpretation according to the matches. This strategy is useful, when the matching data is insufficient to match point-to-point (sparse 3D point cloud) or

inaccurate. In this case we can match object-to-object or segment-to-segment, disregarding small inaccuracies in matching

- *Matching first, interpretation second* is the strategy applicable, when the point-to-point matching is available. In this case, we match images in a stack first. For each image pixel, we obtain a multi-dimensional vector consisting of data from corresponding pixels in other images. Subsequently, we apply vision algorithms on such vectors to interpret the data. The interpretation is enhanced by information from other images.

Considering these two strategies, we describe which is more suitable for a specific case (input data), apply it in our algorithm and provide results. However, our final goal in this work is to introduce more universal methods, which will be applicable in generic situations. Therefore, we modify the Conditional Random Field approach [Lafferty et al., 2001] (as a standard method of context application in most vision algorithms) to be applicable directly in a multi-view scenarios in Chapter 7. This method can not only transfer visual data between images in a stack, but also examine the context information between images and transfer context information from one image to another. Such approach is especially useful for images with objects in different scales. For example, in one image an object is projected from close distance, providing good visual cues for interpretation, but lack the context from other objects. Another image projects the same object from a longer distance, giving good contextual information, but low level of details. Transferring context/visual information between images can lead to improvements in interpretation in such cases.

Chapter 2

State of the Art in Urban Modeling

2.1 Introduction

Our work is focused on the interpretation of urban scenes; however from the application point of view it is necessary to understand different aspects of urban modeling first. The ultimate goal of urban modeling is to approximate real world data as precise as possible. To this end, we must examine the collection of real world data, it's representation in a digital format and applications that use such data. In this chapter, we describe the state-of-the-art in acquisition of digital data used for a construction of Geographic Information Systems (GIS) and 3D models of urban spaces. We describe parameters and methods for processing of satellite and aerial imagery, terrestrial vehicle-based industrial systems, crowd sourced open data collections, video feeds, micro aerial vehicle systems and laser rangefinder scanners (LiDAR). Subsequently, we describe how data is processed into a 3D representation and interpreted.

However, before we start with the data processing methods, we introduce a brief history and trends in a GIS modeling as an exemplary application in Section 2.2. This will allow us to better explain what data are relevant for urban modeling and why a 3D representation and data interpretation is important.

Section 2.3 describes different data sources and Section 2.4 presents methods for 3D information extraction from such sources in a form of 3D point clouds.
In the Section 2.5 we show, why the interpretation is useful for visualization, data handling and implementation of user applications. We describe standards in modeling (defined as level of details- LOD) and basics for procedural modeling in a form of shape grammars. We also give the set of objects that are relevant for interpretation in an urban environment. We describe, why terminal and non-terminal objects require different approach for interpretation and what are the most common methods in use. An introduction to semantic segmentation is presented and different area representations are described. In the subsequent section, we also provide a more detailed overview for façade processing methods (with emphasis on gradient projection methods), as they are the most relevant to our research.
In the last section, we provide the overview of the research community in the urban modeling and GIS construction field. We list and describe several commercial companies, research centers/groups and influential research journals that contribute to the field.

2.2 GIS – History and Trends

Advances in information technologies enabled and inspired the development of software for an analysis, storage and display of geographical data, currently known as Geographic Information System (GIS) [Pidwirny, 2006]. GIS can be broadly defined through its function:

- *The measurement of natural and human made phenomena and processes from a spatial perspective. These measurements emphasize three types of properties commonly associated with these types of systems: elements, attributes, and relationships.*

- *The storage of measurements in digital form in a computer database. These measurements are often linked to features on a digital map. The features can be of three types: points, lines, or areas (polygons).*

- *The analysis of collected measurements to produce more data and to discover new relationships by numerically manipulating and modeling different pieces of data. The depiction of the measured or analyzed data in some type of display - maps, graphs, lists, or summary statistics.*

The first predecessor to a modern GIS is considered to be a method of Photozincography [James, 1806], introduced in the nineteen century. Using this method, maps of the Earth surface were separated into several layers, each containing different sets of objects (roads, vegetation, water…). However this method is not considered GIS, since it did not provide any additional native functionality.
First true GIS in operation was the Canada Geographic Information System (CGIS), developed in 1964 as the project of Rehabilitation and Development Agency Program. System was designed to regulate the land use and for resource management monitoring [Tomlinson, 1967], [Fisher, 1972]. In 1964, the Harvard Lab for Computer Graphics was established by Howard Fisher, where a research on GIS was centered. Several systems were developed, including SYMAP (Synagraphic Mapping System), CALFORM, SYMVU, GRID, POLYVRT, and ODYSSEY [Pidwirny, 2006], providing the basis for further industrial and government projects. In these early stages, two data models were considered in a competitive manner – a vector data model that represents stored data as a set of lines (useful for representing boundaries, roads…) and a raster data model where a grid is placed over a terrain and data are represented as part of each cell (useful as area descriptors). Later systems implemented both models for different data structures. Subsequently, a methodology was developed for the geospatial data handling and the standardization was introduced. In particular, standardization was presented by the Open Geospatial Consortium, in a form of OpenGIS specification [OGC, 2012], enabling geo-information on the internet. Such standards allowed GIS to evolve into its modern form – internet hosted large scale GIS applications. The functionality of GIS was also extended, from a simple descriptive query to spatial map analysis. This was allowed due to a numerical representation of spatial information and the introduction of a new mapping theory in a form of spatial statistics and spatial analysis [Godchild, 2002].

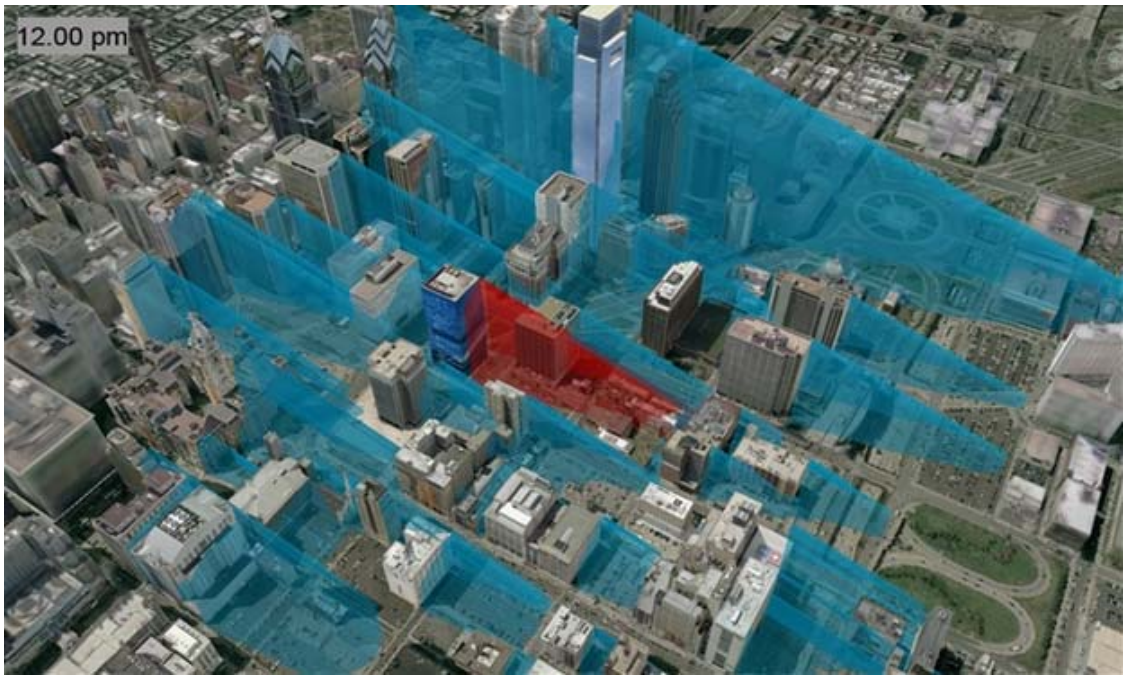


Figure 2.1: The example of volumetric data in the 3D GIS – shadow maps of buildings, that allows to visualize for the shadow effect of proposed building on neighboring buildings. This image is presented by the ArcGIS 10 application [ArcGIS, 2012].

First GIS models were represented in 2D and as such had some limitation in visualization as well as in applications. Notably, noise prediction, water flood, air pollution and geological models require the third dimension for computation. Other fields that could benefit from introducing GIS into 3D are urban and landscape planning, environment monitoring, telecommunications or real estate market [Stoter and Zlatanova, 2003]. The rapid progress in development of 3D GIS circa decade ago was initiated from the one side by such market requirements and from the other side by improvements in 3D data collection techniques and sensors (aerial and close range photogrammetry, laser scanners, GPS…) as well as new hardware technologies (increase in storage space, faster processors and GPU computing).

The first step into the third dimension represents concept of 2.5D GIS, where the height information has been added for each terrain point. This is generally known as Digital Elevation Model (DEM) [Zhilin et al., 2005]. However, it is still a general model of a surface, as no objects are usually semantically identified. Even the semantic

interpretation is not present, additional functionality can be implemented, such as increased range of measurements and new topological models. Height (elevation) information can be represented as a surface model (see Section 2.4.1 for more details), or when triangulation, such as Delaunay triangulation [Delaunay, 1934] is applied, a wire-frame model [Koch and Heipke, 2005].

The transition of a GIS into 3D requires identification of objects as volumetric models. For this purpose, an urban modeling research field was developed. 3D modeling of real urban objects can be done in several ways, such as binary raster (voxels either belong to the object, or not), subdivision of space with an octree, or using constructive solid geometry in a form of geometric primitives. Such representations allow for volumetric measurements such as hydrogeological simulations and others (see Figure 2.1). Given an urban modeling, 3D GIS allows for interpretation of urban structures, such as buildings, or more detailed circulation spaces.

In current models, time parameter is commonly represented as a set of map layers that can be animated to display changes, e.g. in terrain data. Extending this concept, time can be added directly into a model as a fourth dimension in a 4D GIS application. This would allow for additional functionality, such as predictive modeling [Van Ruymbeke et al., 2008].

However, most of first 3D GIS platforms required manual work for construction. It soon become clear, that given large volumes of data and the extension of urban reconstruction into a global level would raise the cost of projects significantly, due to a costly (financial and temporal) manual input. The need for automation has risen as an answer to this problem. Difficulties with automated workflows were tackled with the increase in data redundancy on one side and the improvement of data processing hardware on the other. These circumstances enabled current interest in urban modeling research in computer vision and graphics communities and led to an expansion of GIS applications to many aspects of our personal life.

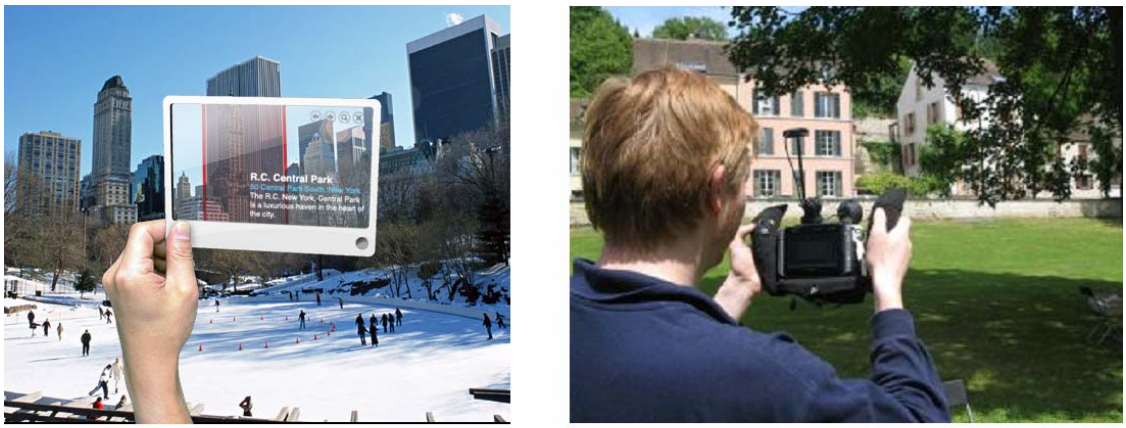


Figure 2.2: Two examples of augmented reality applications in an urban environment. Left – commercial apps for smartphone or tablet users use augmented reality for navigation, information retrieval or assistance (the image is a design concept by Mac Funamizu [Funamizu, 2012]). Right – industrial use of augmented reality in a form of AR scout device [ICG TUGRAZ, 2012].

2.2.1 Trends in GIS and Urban Modeling

In recent years, GIS has moved from several highly specialized industrial professionals to a common every-day user - to our smartphones, tablets and laptops. Such rapid development opened new fields of research and greatly stimulated and increased GIS research. For example, U.S. Department of Labor identified "Geotechnology" as one of the three "mega-technologies" of the 21st century, together with Biotechnology and Nanotechnology [Berry, 2012]. New and exciting developments in several related technological fields show us some possible trends in GIS and Urban Modeling. The boom of camera phones and the reality of broadband internet connection worldwide introduced the phenomena of an internet hosted media. Currently, around 22% of mobile internet traffic is taken by YouTube videos [IBTimes, 2012]. Similar development is observed in digital photography in a form of open community photo collection, such as Flickr, where around 5 million digital photographs are uploaded daily [Scribbal, 2012], many containing useful urban data. It has become clear that such data source cannot be ignored and research in data acquisition and processing from community photo collections is currently under intensive research [Gösele et al.,

2010]. A common requirement is a fully automated workflow, as the sheer volume of data would make any manual input unfeasible [Leberl and Gruber, 2009].

However, this process is also working the other way around. Current smartphones with enhanced computational capacity and internet connection are able to provide additional services in urban navigation for a user. Coupled with a GPS system, such combination represents a navigation device, which can present users with required information about surroundings (e.g. close restaurants, offices, tourist attractions…). This moved GIS applications to mobile platforms, with different paradigms of visualization, processing, and input formats.

Such development is also important for professional applications in urban environments, such as urban planning, municipal community services or critical situation management, where professionals can visualize important location-based data directly in the field (see Figure 2.2). These requirements are pushing towards the augmented reality in mobile devices that allows visualizing hidden data, such as underground pipelines in a real urban background.

Combination of mobile smartphones with GPS devices, internet hosted services and the availability of information about local businesses has led to the introduction of term Location-Aware Internet. This concept makes location based search function available and allows displaying user relevant data [Leberl, 2008]. Such transformation of GIS applications from professionals to everyday user is fueling more research into human-scale urban models and risen the need for indoor models. However, it had also changed the paradigm for general GIS construction and use. Up to this point, GIS were used strictly in a professional environment and as such were designed to work on an optimal "scientific solution". Opening a GIS application to a new, different market – everyday users raised a need for implementation of different parameters to find "social solution". This has extended a set of measurements from strictly physical parameters to more indefinable variables, such as human values, attitudes, trust, etc. It has also opened a question, how to present users with relevant data such that they will be accepted. In general, a user has to fell like a part of the model – be provided with an immersive experience.

This model can transform even further into the concept of "Internet-of-Things" [Ashton, 2009], where important objects in real world will be catalogued, using wireless technology, such as RFID chips [Weis, 2007] and the location awareness for

human surroundings will be established. Cataloguing has to be in a human-scale, in 3 dimensions and with accuracy ± 10 cm range [Leberl, 2010]. In such model, computing, sensing and connecting will extend to all aspects of human presence. This model can be denoted as an *Ambient Intelligence*.

Given the requirement to model urban environment at the worldwide level, one has to consider the amount of data to be processed. When the standard redundancy of an aerial photography is set to 10 images per object, streetside mapping at 40 images per structure and 100 images per interior, the final volume of data would be around one exabyte [Leberl, 2010] and additional data have to be collected for upkeep. Such volumes would put strong constrains on work automation and time efficiency of processing.

2.3 Image Data Sources

We divide relevant data sources into three groups:

- Above ground group for satellite (orbital platforms) imagery and aerial data (aerial vehicles with the altitude in order of hundred meters). Such data sources are used to generate orthophotos, surface models, or basic object models for urban spaces;
- Streetside group for user collected photos, vehicle based industrial systems, micro-aerial vehicles (even factually an aerial platform, data output is more related to the streetside group);
- Underground mapping for a below-ground data acquisition

2.3.1 Satellite/Aerial Imagery

Satellites with digital camera apparatus present fast mean to cover large areas of Earth's surface and it was initially available quickly and globally.

Figure 2.3: The aerial eight-lens UltraCam with storage and processing unit. Four panchromatic sensors are arranged linearly along the flight direction (left). Method of stitching the input from different sensors triggered sequentially into one large image (right) [Leberl et al., 2010(II)].

However, the subsequent effort to interpret and reconstruct objects of interest in a human scale from such digital data proved to be problematic. This is due to a resolution of satellite images at a pixel size in a 50 cm range – a resolution that is insufficient for a modeling task. This limitation is largely present due to the government restrictions on satellite imagery. Plans for a new – 30 cm resolution sensors are scheduled at mid 2014 to be put into service as a part of WorldView-3 project [SatImagingCorp, 2012]. The 3D modeling effort is also hindered by a low geometric variety of the projections, as the optical axis of a satellite cameras are nearly orthogonal to the Earth's surface, preventing disparity measurements for corresponding points, but also limiting side views at objects like building façades. However, satellite images (as well as aerial images) are used to generate an orthophoto by warping the input images on a reference surface [Oda et al., 2004], [Gruber, 2011]. This type of image is geometrically rectified such that it can be superimposed on the planar map of the region which can provide labels for objects such as roads, or landmarks. The combination of such information represents a GIS generated by an automatic interpretation of geographic data. To this day, satellite imagery for orthophoto generation and other user application is still feasible in the areas of globe, where aerial imagery is restricted or denied.

Problems observed with satellite images are mostly avoided in a modern, high resolution aerial photography. Most common aerial platforms today are propeller aircraft flying at the low altitude, equipped with a camera system. These were traditionally equipped with film cameras, providing a stereo view of the earth surface. However the low level of redundancy and the quality of film made the automation of work problematic. This resulted in a high (linear) cost per photo processing as the manual processing was involved and made large scale reconstruction efforts unfeasible. The situation changed with the introduction of modern large format aerial digital cameras and low cost, high capacity data storage system in 2003. Current aerial cameras, such as UltraCam (see Figure 2.3) provide a pixel size at 3 to 15 cm (considered a human scale), multiple area charge coupled devices (CCD) and up to eight separate lens [Leberl et al., 2003]. The data is parsed in RGB color and infrared channels at 13 bits per pixel. Such data provide superior resolution and depth when compared to satellite images. For 3D modeling and reconstruction efforts, the increase in redundancy and more variable point of views are similarly important than the increase in resolution.

In a current typical aerial dataset provided by such camera, an object point is commonly located at ten or more images. When compared to traditional film photogrammetry maps (where object was located in a stereo pair), such increase in redundancy is shown to improve the accuracy of measurements by a factor of six as demonstrated by systematic error measurements of UltraCam dataset compared to a traditional stereo pair [Ladstätter and Gruber, 2008]. This level of redundancy is achieved by a high overlap present in a dataset – 80% in a direction of flight and 60% sideways, practically at no additional cost. Due to the lower attitude of a plane, each object is projected from several different view points, providing views at the sides and diminishing the effect of occlusions. The increase in data output (from 260 Mbytes per traditional film photograph to 1.6 Gbytes per digital photo) and redundancy level resulted in large volumes of data [Leberl et al., 2010(II)]. However the processing of such volumes comes to no additional cost in an automatic workflow, but the time cost. To this end, current applications use parallel GPU processors with embedded tolls for computer vision tasks that can process such quantities in a reasonable time.

Figure 2.4: R7 camera system currently in development by Google. System is composed of 15 small, 5 Mpix CMOS sensors [Anguelov et al., 2010].

In the future, aerial imagery resolution may increase up to the range of 2-3 cm per pixel. This requirement may be motivated by advanced data collection application, to read signs on the façades, traffic signs, or suspended wires. Such resolution can be achieved with current technology, using 100 mm focal length camera mounted on a plane flying at 600 m altitude. Aerial dataset can be complemented with additional data from remote sensing devices, such as LiDAR (see Section 2.4.3), or Interferometric synthetic aperture radar (InSAR). Such devices provide additional height information in a form of a digital elevation and can be considered as an input for building an urban model [Bolter and Leberl, 2002].

2.3.2 Street-Level Imagery

Since the requirement for systematic mapping of an urban environment has been presented, camera systems based on a terrestrial vehicle (commonly van or car) has been became a commercial commodities as a platform for environment mapping. Such "industrial systems" (IS) are designed for collection of panoramic views, employing multiple centrally perspective cameras, sometimes fish-eyed imaging (catadioptric camera).

Figure 2.5: MAV system used for mapping of an urban environment (left) and the semi-dense point model obtained from refined sparse reconstruction from MAV's data with original image from the scene (right) [Wendel et al., 2011].

Industrial systems are usually complemented with additional sensors, such as LiDAR and GPS positioning systems and are designed to map "street canyons" – sections of the urban environment closed from both sides by buildings. The structure of the camera system and the moving pattern of a vehicle provide high overlaps and view variation for building façades. Figure 2.4 displays an example of such camera system, currently in development by Google. This system is designed to capture wider sections of a scene, including more detailed street (pavement) and road sections, however it lacks fish-eyed sensors [Anguelov et al., 2010].

Comparatively to a systematic approach to an urban mapping, a significant source of streetside image data is present in online open image hosting sites and databases. Primary challenges in utilizing such data are the organization of crowd sourced dataset (missing camera calibration data and relative poses), selection of relevant images (missing geotagging, mislabeling of photos), photo quality and a volume of data [Snavely et al., 2006], [Frahm et al., 2010]. However, crowd sourced images are usually well focused on object details and abundant, mainly for landmark objects and city centers, for which can be seen as a significant source of digital data. As both

industrial system and crowd sourced datasets are primary sources of data in our work, we will present more details and examples in subsequent chapters.
Alternative to a human agent, or vehicle based industrial systems, Micro Aerial Vehicles (MAV) (see Figure 2.5) have been proposed as a source of digital data for mapping of urban canyons and reconstruction for example in [Wendel et al., 2011]. Primary advantages are that MAV can provide more variable points of view of objects and access areas not previously accessible by a land based sources. Current MAVs can carry up to 1 kg of equipment (camera system, data storage unit, data processing) and stay in the air up to 15 minutes before landing for a recharge. In such a session, about 5 GB of data can be collected for a scene reconstruction [Lionel et al., 2011].

2.3.3 Underground mapping

Technology for an underground data acquisition has found important utilization in an urban environment. Human cities are supported by extensive underground infrastructure which is also relevant for many GIS-related applications, such as urban planning, risk control, etc. Several possibilities for underground mapping are available, such as Ground-penetrating radar (GPR). GPR is a nondestructive method that works in the microwave band of the radio spectrum, and detects the reflected signals from subsurface structures. GPR can be used in a variety of media, including rock, soil, ice, fresh water, pavements and structures. It can detect objects, changes in material, and voids and cracks. [Daniels, 2004]. GPR can collect a profile view of the subsurface in a one run, thus it can provide a 3D image composed of connected “slices” if used methodically. Other methods for underground mapping are for example - Electrical resistivity tomography, Induced polarization, or Seismic tomography [Loke and Barker, 1996], [Dziewonski, 2004].

2.4 3D Point Clouds

In this section, we present methods for 3D point clouds acquisition from aerial and surface sources. We describe how sparse/dense reconstruction can be achieved. We also include the description of LiDAR as a method for a 3D point cloud acquisition.

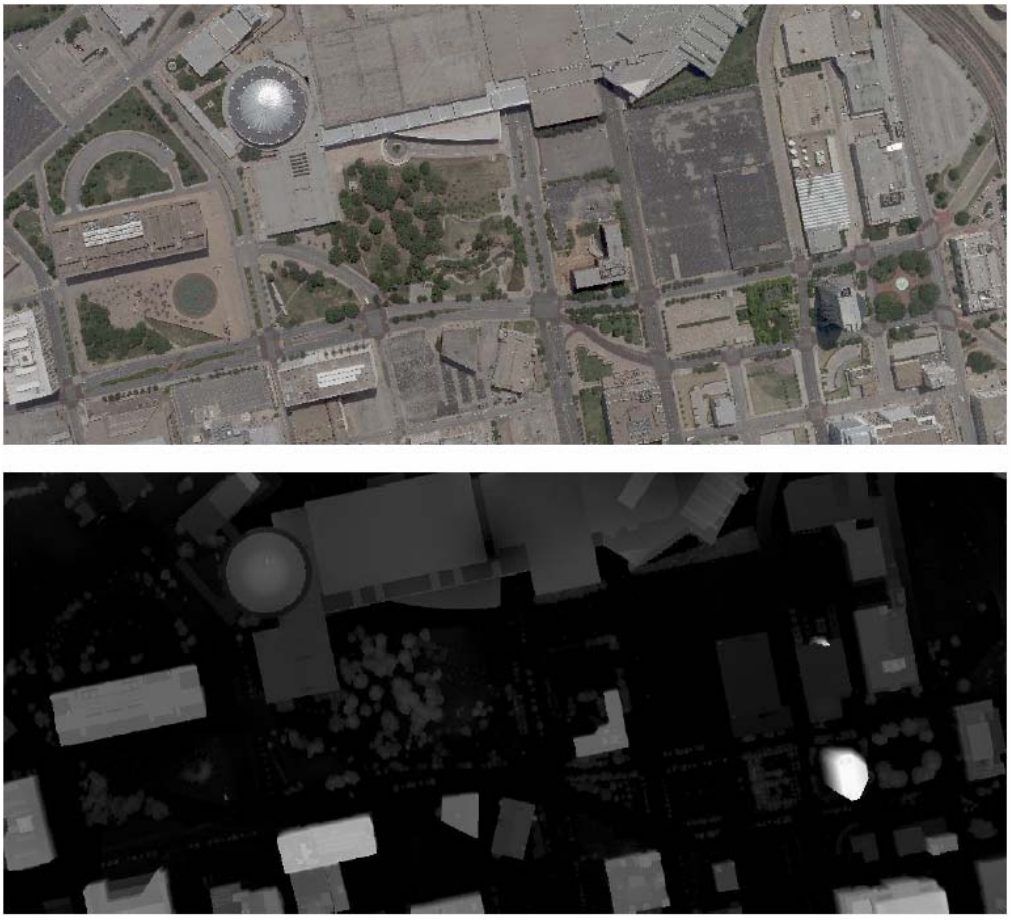


Figure 2.6: Elevation measures for an aerial image. Dark pixels indicate ground level, while bright color defines height of objects [Kluckner, 2011].

2.4.1 Surface Models from Aerial Photography

The extraction of surface models (range images) form redundant datasets of aerial images is established on a principle of multiple view geometry [Hartley and Zisserman, 2004], thus based on the detection and matching of corresponding points. In an aerial imagery domain, the problems with repetitive patterns, occlusions and low-textured areas are addressed through the effect of redundancy.

If each object is located within at least 10 images, high resolution of data and camera calibration established an improved estimation of the underlying scene geometry is possible [Kluckner et al., 2011]. Dense scene geometry estimation can be achieved through two processing steps:

– Structure from Motion (SfM) recovers the camera parameters and the extraction of a 3D sparse point cloud determines pixels correspondences [Irschara, 2011]. This is performed in an automaton of a manual process of triangulation, where measurement points are identify in overlapping images and camera positions and orientations are refined. Given that GPS and Inertial Measuring Units (IMU) measurements of aerial cameras are known, this process can be seen as refinement of camera parameters into a sub-pixel accuracy.

– For many applications, the sparse point cloud provided by SfM is not sufficient. When required, the dense matching techniques are used to estimate depth information for every pixel. These can be divided into three groups, according to the set of pixels on which the optimization is performed: local methods [Yoon and Kweon, 2006], semi-global methods [Hirschmüller, 2006] and global optimization methods [Pock et al., 2008]. Advanced techniques consider also occlusions and matting at depth continuities [Bleyer et al., 2009]. Many stereo and multi-image dense matching methods are based on a plane-sweep technique, as this concept also allows for accumulation of matching costs through multi-image datasets [Hirschmüller and Sacherstein, 2009].

Scene geometry is represented in a form of range image, where the depth information is computed for each pixel (see Figure 2.6). A range image is computed for each aerial image, forming a surface model when transferred into a 3D world coordinates. The merging of range information from multiple redundant images into one surface model can provide additional refinement of depth information [Kluckner, 2011]. Surface models can be directly used to recover the 3D structure of objects (e.g. wireframe models). Several techniques have been presented, such as 3D point segmentation algorithms [Dorninger and Nothegger, 2007]. In this work, the initial clustering of the parameter space allows to identify segments in a high resolution dataset (20 or more points per square meter).

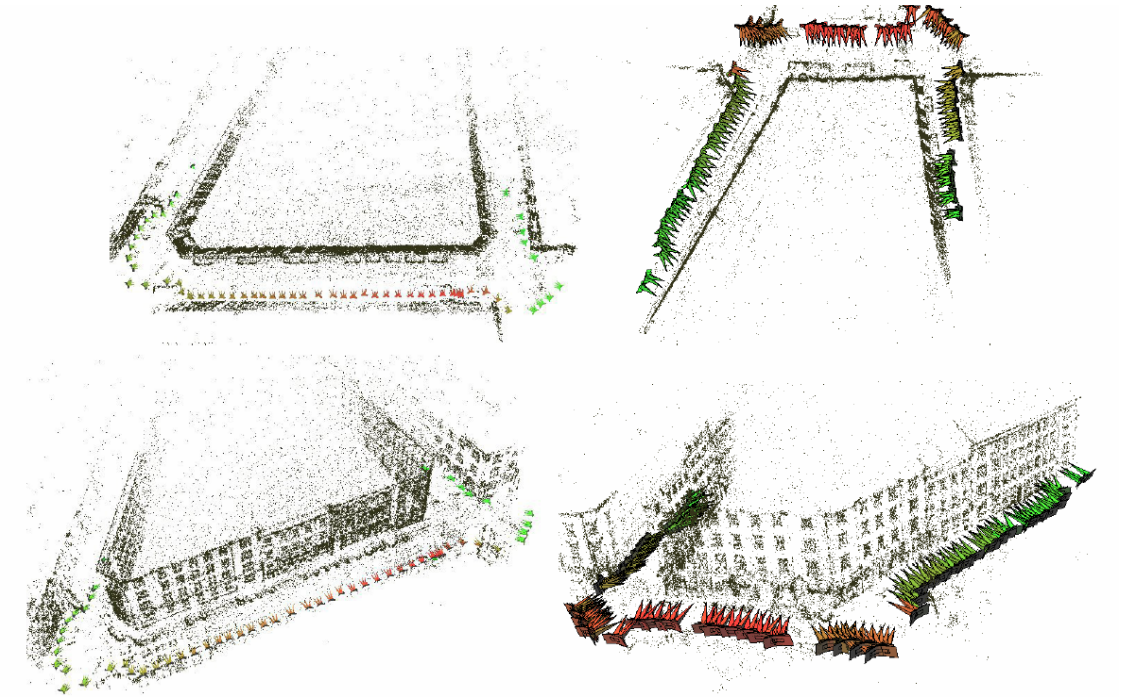


Figure 2.7: Reconstruction of city block from streetside images. Projections positions and directions are shown in color. Note that most detected corresponding points are in the areas with high texture (windows, other façade elements) [Klopschitz et al., 2010].

Such resolution can be achieved by a modern image matching algorithm. Segmentation was also designed to identify vertical planes of 3D structures, such as façades. Another approach are shape recognition methods, such as [Zebedin, 2010], based on energy optimization and geometry hypotheses sets. In this work, multiple data terms (second order regularization) impose the regularization of the shape. The method is based on a probabilistic approach and in a final step, an evaluation of multiple hypotheses is used to derive labeled footprints of urban structures.

2.4.2 3D Modeling from Streetside Images

Image matching based 3D structure modeling from digital streetside images is derived from the same principle as 3D modeling in aerial images domain [Hartley and Zisserman, 2004].

Figure 2.8: Workflow for geometry reconstruction from a community photo collection, as presented by S. Agarwal. Examples of user provided images with variable parameters (left), sparse 3D point cloud reconstruction acquired through SfM (middle) and the dense reconstruction using multi-view stereo algorithm (right) [Agarwal et al., 2010].

The construction of 3D point clouds is based on a detection of corresponding points which is the primary problem in reconstruction efforts and still provides challenge for reconstruction algorithms. This is mostly due to the presence of repetitive textures in an image and unorganized datasets. The greater variability in viewing positions and directions, increased occlusion problems (pedestrians, vehicles, vegetation…) and imaging equipment often not dedicated for reconstruction efforts leads to a greater need for dataset organization.

Therefore, advanced methods for dataset alignment, such as incremental structure from motion [Klopschitz et al., 2010] have been proposed. Such methods rely on more stable parts of a dataset to build the initial structure and incrementally improve results with additional digital data (see Figure 2.7). In 2006, N. Snavely introduced his work on reconstruction of urban objects from an unorganized crowd-sourced dataset [Snavely et al., 2006]. Initially named Photo Tourism, the project evolved into the online application that allows for image triangulation and point cloud generation from community photo collection – Photosynth. This research is continued by Microsoft Research as the part of the Bing project. In a work of [Agarwal et al., 2009] a 3D reconstruction from a large crowd-sourced dataset (approx. 150K images) was demonstrated. The method is implemented on a highly distributed parallel system and experiments with various algorithms in each step of the pipeline provide comparison for current state-of-the-art methods. A sparse 3D points reconstruction is provided as a result. This can be achieved with the application of a Structure from Motion method on

a detected visual word – clustered SIFT features in a photo collection. Subsequently, dense reconstruction can be recovered, using a multi-view stereo algorithm. This method measures depth values for corresponding points in all matched images and selects the depth value with the highest consistency through a photo collection as a value for a 3D point in world coordinates (see Figure 2.8). Thus the reliability of correct depth estimation is increased with redundancy present in a dataset.

This approach is however unfeasible with the current technology at the city scale and a clustering method has to be introduced in a photo collection dataset to reduce the problem into a smaller scale, therefore in [Agarwal et al., 2010], dense reconstruction is extracted in each cluster separately and fused in a final model. Following this work, [Frahm et al., 2010] introduced a system of dense 3D reconstruction from 3 millions unregistered photos. This method is implemented on a single PC with high-end graphic processors and is able to process all images in a single day. Method is based on four steps – appearance based clustering of gist features to obtain canonical views, geometric cluster verification based on RANSAC [Fischer and Bolles, 1981], local iconic scene graph reconstruction and dense computation.

Methods of organizations, data mining and data processing in open online datasets were summarized in a work of [Gösele et al., 2010].

[Pollefeys et al., 2008] introduced a real-time 3D reconstruction system, using a mobile camera system for video acquisition with GPS and inertia sensors to place data into geo-registered coordinates. The 3D model of the environment is generated online from a video stream and is based on tracking 2D feature points in video frames. Dense 3D reconstruction is achieved by plane sweeping and optimized in a fusion step. In this step depth maps from related frames are combined into one depth map, thus correcting possible errors generated in depth computation for individual maps.

2.4.3 Light Detection and Ranging in 3D Reconstruction

In an urban reconstruction effort (aerial as well as streetside), Light Detection and Ranging (LiDAR) has been proposed as an alternative to sparse and dense 3D point clouds acquisition via image based solutions. The success of LiDAR application is based on the ability to provide instant 3D point clouds without need to process additional data [Lato, 2010]. The density of the point cloud provided by LiDAR is

commonly in a scale of 50 pts/m^2 up to 500 pts/ m^2 in terrestrial setup [Glennie, 2009], [Lynx, 2012]. A measurement precision of between 0.029 m and 0.031 m had been achieved in elevation for a mobile LiDAR setup [Barbera et al., 2008]. 3D modeling of urban environments with a help of laser scanner observe similar problem with the alignment and registration of scanner data and visual data, primary due to the inaccuracy of GPS registration [Haala et al., 2008]. This problem can be addressed by the involvement of 2D image features in a process of registration, as demonstrated by [Yang et al., 2011]. It can be also addressed with the introduction of new positioning devices, such as the European positioning system Galileo [Galileo, 2012]. Current methods based on the image matching can provide similar or better results than LiDAR, however the use of laser scanners have gained momentum, due to a large number of laser sensors currently in use and the ability to display results directly in the field. The viability of laser scanner data in the future will depend on the advances in positioning techniques and improvements in scanning resolution [Leberl et al., 2010]. However the greatest challenge for laser data will provide possible paradigm change from scanning technologies to image-based solutions, such as SfM and multi-view stereo methods.

2.5 Data Interpretation

The interpretation of urban data is important for additional functionality (e.g. navigation, geo-locations, urban planning or risk control), improved user experience, visualization and augmented reality [Leonardis et al., 2000]. In this chapter, we describe an interpretation from aerial and streetside sources. We show how the interpretation of urban scene elements can provide ground for urban modeling based on shape grammars [Stiny and Gips, 1972]. In this setup, object surfaces are not described by their texture, but are visualized as elements of shape grammars. Such parsing of objects into smaller elements resulted in the formulation of modeling standards. [Kolbe et al., 2009] provide the CityGML standard in a form of five levels of details for multi-scale modeling – from a simple 2.5D model (LOD0), building block (LOD1) with roof shape (LOD2), details of a façade (LOD3) to a detailed semantic model including interior rooms (LOD4) (see Figure 2.9).

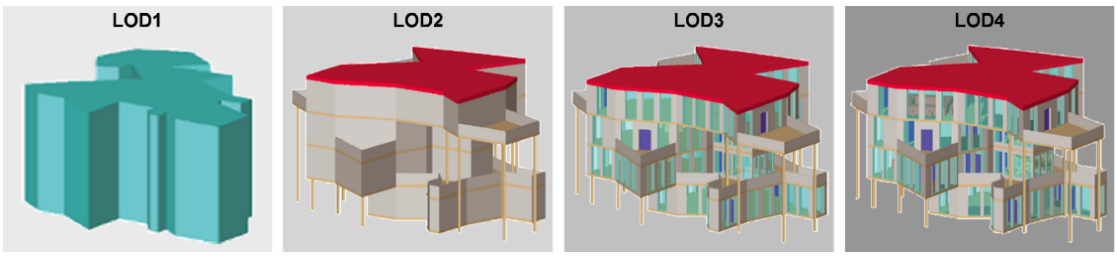


Figure 2.9: Different Levels of Details visualized in the example [Kolbe et al., 2009].

This standardization can provide bases for large scale urban modeling, as the results from different platforms can be interconnected. In the last subsection, we describe how the selection of image areas representation in a form of pixels, superpixels and segments can influence the process of interpretation if the context is involved. As our work is focused on interpretation from streetside images, the distinction between interpretation from pixels and from segments is the key idea in our semantic segmentation method described in Chapter 4.

2.5.1 Interpretation from Aerial Images

With the requirement for utilization of aerial images (e.g. planning, traffic control, land value estimation), methods for data interpretation in this domain were introduced. For example, interpretation of tall structures in aerial photos is applied in tracking methods (vehicle, pedestrian tracking), to detect occlusions [Prokaj and Medioni, 2011].
In general, interpretation is performed in a pixel-wise semantic classification, either in dual-class (buildings-background, cars-background), or multi class (buildings, vegetation, roads, grass, water…). Interpretation techniques utilize visual cues, 3D information (surface models) and other information obtained during the image requisition, such as laser scanner data. In the work of S. Kluckner (see Figure 2.10), pixel-wise semantic interpretation is achieved through combination of low level visual features (true color, edge responses) and 3D information. Both data sources are combined in the region descriptors, where boundaries are refined through unsupervised segmentation and energy minimization. [Kluckner, 2011].

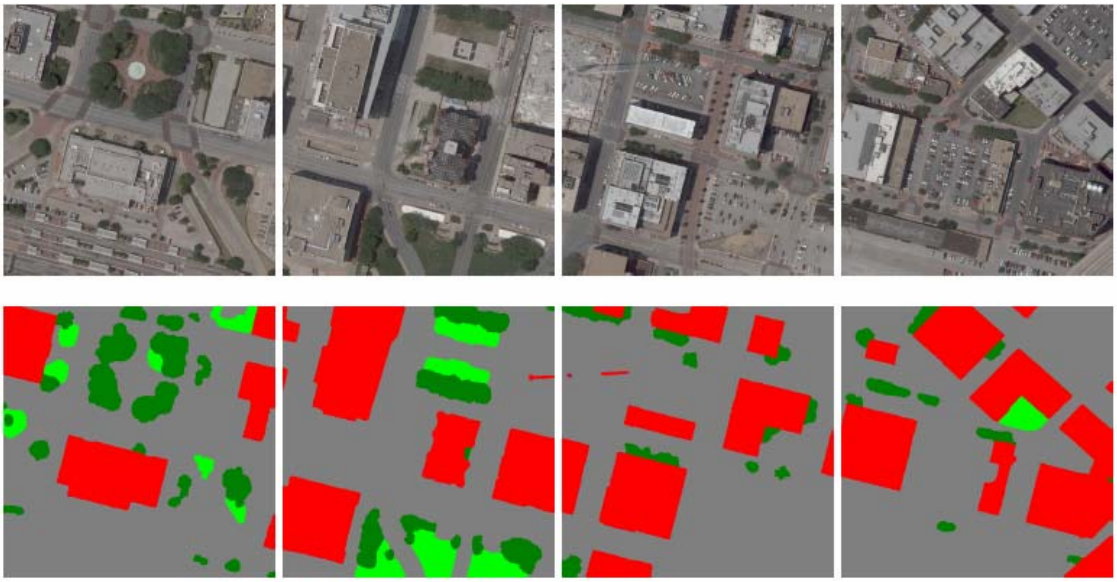


Figure 2.10: Interpretation of aerial photos. Detected classes are color coded – buildings (red), vegetation (dark green), roads (gray) and grass (light green). Labeling is based on combination of visual cues and surface models [Kluckner, 2011].

Other methods for semantic interpretation use only visual data without any height values initially assigned to images [Grubner et al., 2005] or only height information in a form of surface models [Lafarge et al., 2008]. Similary to surface models, LiDAR data has been applied to identify structures such as buildings in aerial images [Toshev et al., 2010]. Aerial pixel-wise data interpretation is based on a high redundancy of aerial datasets [Kluckner and Bischof, 2010]. In a work of Meixner et al. [Meixner et al., 2011] a method for separate façade detection from aerial images and the semantic segmentation of the roof elements are presented. Results of the interpretation are applied in building 3D models of urban environments, or directly in customer's applications, such as a land value estimation [Meixner and Leberl, 2010].
In a work [Ferreira and Bernardino, 2006], a method for 3D urban modeling and interpretation from satellite images is presented. Based on these data a simple prismatic model of an urban surface is created. The method also uses context knowledge of a segmented satellite image to simplify the final model.

2.5.2 Interpretation from Streetside Images

In a streetside dataset, the complexity and variety of data provides challenge for interpretation not only in a single image scenario, but also additional problems in image matching and interpretation merging in a multi-view. For these reasons, several methods working with such datasets have been semi-automatic, thus requiring human interaction (in a façade labeling for example). In a work of [Fabritius et al., 2008] on urban data modeling, a fusion of aerial data and streetside images is presented to a provide framework in a multi-view image environment. However, the streetside data require manual labeling.

Similar to aerial datasets, the basic goal of many methods is to detect and identify building structures for further processing (texture extraction, occlusion detection…). In a work of C. Fruh a method for interpretation of building structures is presented [Fruh et al., 2005]. Input data consist of vertical surface scans and camera images. Processed data are divided into segments which represent individual building façades, or blocks of buildings. Segments are bordered by gaps between buildings, or corners and are consequently transformed into depth images. Building structures are detected in a depth field and missing information is interpolated. This method presents a common approach for identification of structural parts of an urban scene for 3D urban modeling. A large number of sensors provide enough data for precise building identification and representation. Problems of laser scanners with windows and occlusions are solved by using camera data in these areas. Regions of the scene are divided between the background layer and foreground layer. Most of the processing is then applied to the background layer, containing building façades. Due to the precision of a laser scanner, most of the details are preserved. It also employs basic segmentation in 3D data.

Methods of specialized object recognition and 3D urban modeling are combined in a work [Cornelis et al., 2006] to achieve better visual results. During the urban scene modeling, a semantic level is used to enhance the final 3D model. As an example, detection of cars is used to insert car models into an urban scene. Also the results of a scene modeling are used to increase the reliability of car detection, therefore the algorithm process is modeled as a cognitive loop.

Additional works on data interpretation from 3D information provide methods of segmentation directly in 3D data or combination of 3D and visual information [Kim et

al., 2008], [Sithole and Vosselman, 2003]. In these examples, image interpretation has been enhanced by multi-view approaches and/or additional data sources.

2.5.3 Shape Grammars

To define objects, which are relevant in the 3D modeling of massive urban spaces, we must first consider an effective method to describe and visualize them. Detailed levels of representation for a building façade can be provided by shape grammars. This method is very effective in describing the repeating patterns of a façade [Fruh et al., 2005]. For a procedural modeling of computer graphics architecture, a novel variation of shape grammar – CGA shape has been proposed by Pascal Müller [Müller et al., 2006]. This approach allows modeling of large urban spaces with a high level of detail. Also, the problem with modeling of volumetric shapes with arbitrary orientation is solved by this method.

Shape grammars are based on a hierarchical representation. Objects located at the building façade can be represented as non-terminal and terminal symbols. Non-terminal symbols (floors, oriels, risalits…) are at the higher stages of the hierarchy and can be further subdivided.

Terminal symbols (Figure 2.11) are the smallest (defined in application) achievable details (windows, arches, ornaments…). Relations between symbols can be defined as replacement rules (floor -> row of windows). The level of details for the shape grammar visualization is defined by the terminal symbols definition. In the CityFit project [Hohmann et al., 2008] the goal is to detect elements >50cm in the façades as terminal symbols. As proposed by Müller, shape grammars can be also used to describe and model a large variety of elements in urban spaces, besides building façades. In his work, Müller used 190 rules to model a complete city, including roads, vegetation, buildings and other urban objects [Müller et al., 2007].

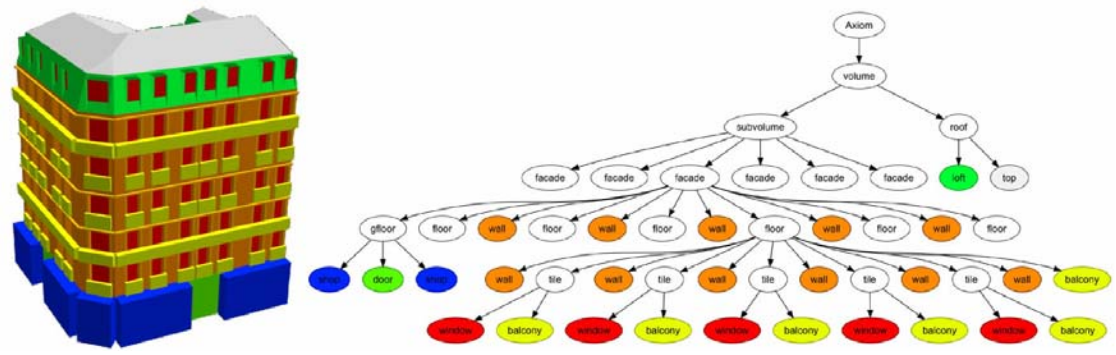


Figure 2.11: Representation of façade elements as non-terminal (inner nodes) and terminal (leaves) symbols as described in a work of Simon et al.. On the left is one possible building model generated from such grammar [Simon et al., 2011].

Recent work on building modeling presented by [Simon et al., 2011] present shape grammars as a method of 3D building representation (see Figure 2.11).
Urban objects can be defined as a set of terminal and non-terminal symbols in a shape grammar. We exclude temporal objects (e.g. pedestrians, vehicles) from the consideration.

Building

Non-terminal:

- building façade
- floor, oriel, risalit, stair shaft, ground floor section, external elevator shaft, glass section, garage section, access staircase, fire staircase

Terminal:

- window, arch, door, pillar, garage door, balcony, column, ledge, cornicle, decorative element, statue, pillar, stairs, shop windows, shop signs, painted ornaments, gutters

Roof

Non-terminal:

- roof section

Terminal:

- attic window, satellite dish, antenna, handrails, chimney

Circulation space

Non-terminal:

- road, bridge, road ramp, traffic sign, traffic light, railway, square, parking space
- lane, pavement, cycleway

Terminal:

- horizontal signalization, crosswalk, crossing island, bus-station, handrails, noise barrier, overpass, roadworks, pole, sign plate, ramp, monument, manhole
-

Vegetation/Nature

Non-terminal:

- tree, bush, grass area, river, lake

Terminal:

- trunk, branch, leaf section, flower section, river bank

To model a real urban scene, using procedural modeling, an appropriate level of interpretation is required. Principal information about the presence and properties of objects must be known. These can be subsequently assigned to specific terminal/non-terminal symbols.

2.5.4 Pixels, Superpixels and Segments in Semantic Segmentation

In computer vision methods, several types of contextual information are usually considered at a different level and organization of visual data. The spatial smoothness of labels can be examined on the lowest level in a digital image – between neighboring pixels. In this case we can assume that the labels of neighboring pixels will not change rapidly. As this is usually a simplest type of context, one can consider in the image, it has been applied in a number of computer vision applications [Korč and Förstner, 2008], [Jiten and Merialdo, 2006]. However, this type of context does not model spatial relations between real objects in the image. The size of the neighborhood considered for a context examination is usually too small in comparison with real objects in an image. Random fields working with such neighborhood cannot encode context between two or more real objects in the image.

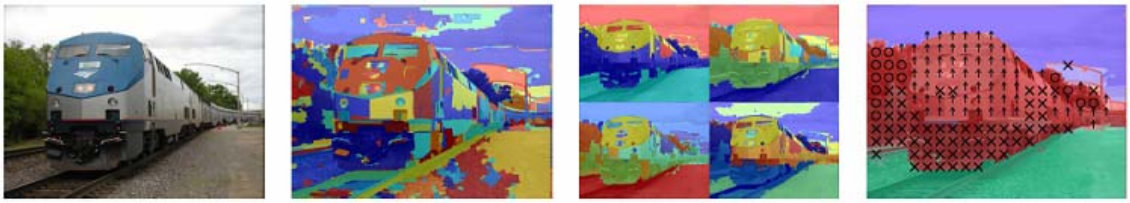


Figure 2.12: Workflow introduced in a work of Derek Hoiem build the structural knowledge from pixels (a), to superpixels (b), to multiple potential grouping of superpixels (c), to the final geometric labeling of the image (d) [Hoeim et al., 2005].

Therefore researchers working with context in a projected scene are beginning to consider larger areas. In a work [Felzenszwalb and Huttenlocher, 2004], the efficient graph-based segmentation has been introduced. The segmentation of an image into regions is provided by this method, while the number and the properties of regions can be easily modified by segmentation parameters. Also, the ability to preserve details in low-variability image regions while ignoring details in high-variability regions is an important feature of this method.

This segmentation has become popular in recent years, as a basis for different over-segmentation techniques [Hoeim et al., 2005], [Daure, 2006], [Chari et al., 2008]. In general, the over-segmentation method uses the graph-based segmentation to create a large number of small segments over the image. In a work of Derek Hoiem, labeling is performed on an over-segmented image and the regions are subsequently merged, based on their labeling results (see Figure 2.12).

This approach introduces a notion of "superpixels", as the labeling is performed not on the standard image pixels, but on the small patches (segments). While the patches naturally provide more information for labeling then standard pixels, it is not obvious that the visual properties of patches allow correct labeling.

Also, the superpixels are not the correct representation of objects located in the scene therefore (semantic, geometric) contextual information between objects cannot be used in the labeling process correctly.

Figure 2.13: An example of segmentation provided by D. Hoiem. Three main classes are supported (green), vertical (red) and sky (blue). Subclasses are indicated by symbols in the vertical class and represent the orientation of surfaces [Hoiem, 2007].

[Hoiem, 2007] and later [Xiao et al., 2009] applied his research on segmentation, MRF and CRF to extract spatial layout for 3D scene understanding. The basic idea of his work is that 3D information can be obtained directly from a single image using only visual cues. Given the visual features of detected superpixels and prior knowledge from trained random fields, one can estimate the geometric labels of detected areas in the image. The set of classes presented by Hoiem in his work are representations of surface layout (orientation of the plane).

Thus Hoiem presents geometric, rather than semantic interpretation of the scene (see Figure 2.13). This indicates a different use of context, for which the superpixels are a suitable representation. In his work, the geometric labels are uniform in areas with similar texture; however this approach is problematic in semantic labeling. Semantically coherent objects (such as a building façade) can contain areas with different texture features. Another approach for detection of man-made structures was presented in a work [Kumar and Herbert, 2006], where an image was divided into a regular grid and a context was examined between the neighboring cells in a grid. Context is represented as pairwise potentials in the DRF.

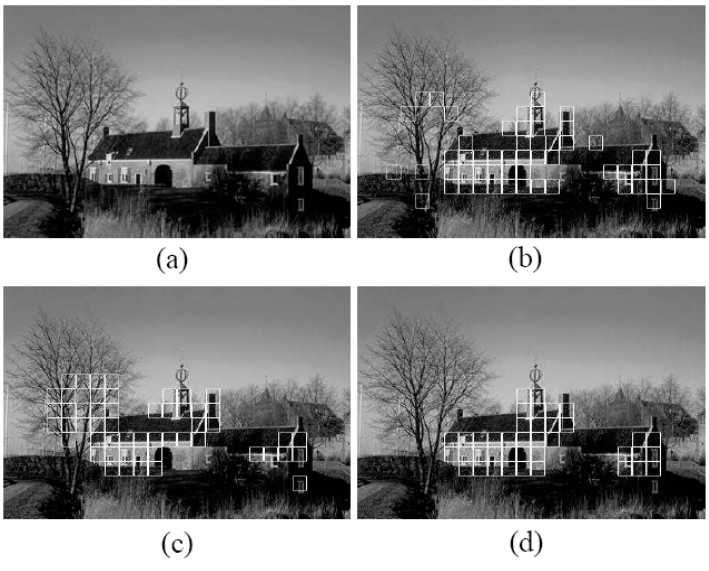


Figure 2.14: An example from the work of S. Kumar on DRF [Kumar and Herbert, 2006]. (a) original image, (b) man-made objects identified by visual features, (c) labeling with context in a form of MRF, (d) labeling with DRF.

In this case, larger objects are also considered for contextual interactions, but cells still do not represent real objects in an image. Such approach does present robust results, but the borders of objects of interests are approximated only roughly, indicating application of the method in a detection task, but not in the modeling (see Figure 2.14). The approach of context in-between objects is also explained in the work of Daniel Heesch, but he does not provide any method for solving the segmentation problem in his work.

Heesch suggests using a Non-Gibbsian Markov random field model to approximate spatial relations between classes in streetside datasets [Heesch et al., 2008]. He suggests a global context in a form of graph of segments located in the image. However, he does not provide the segmentation method, but involves manual segmentation to test spatial rules.

(a) **(b)**

Figure 2.15: Horizontal (a) and vertical (b) projection profiles as described by Lee, Nevatia [Lee and Nevatia, 2004]. Spikes in the projections indicate horizontal/vertical gradients e.g. window frames.

2.6 Façade Interpretation

In this section, we present current trends in recovering a building façade's structure (façade elements), as these represent objects of interests for several methods described in this work. Modeling façades poses several task specific problems. When working with streeside images, occlusions from pedestrians, vegetation, traffic structures and others are common. Windows exhibit reflections and transparency. Façade elements have a high in-class variety. However, one can also take advantage from domain specific features to help in processing. Many façade elements have a specific shape (e.g. windows, doors) and are organized in a predefined geometric style (e.g. rows of windows).

Early works on façade processing [Debevec et al., 1996] utilized such features, but still relied on a manual input. Some geometric and contextual features of façades are projected into its 3D structure. Subsequently, several methods have been developed to exploit such 3D information [Stamos and Allen, 2001], [Fruh and Zakhor, 2001] and the combination of 3D structure and visual information [Coorg and Teller, 1999], [Taillandier, 2000]. T

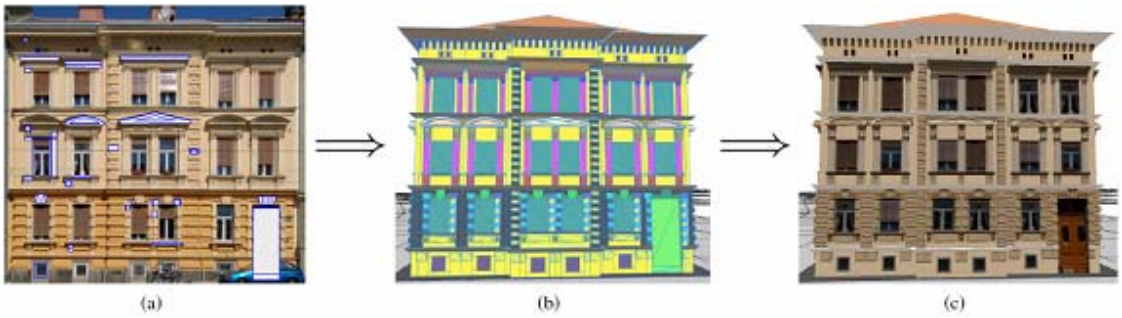


Figure 2.16: (a) Digital photo of building façade with terminal symbols, (b) Shape grammar representation, (c) Shape grammar with applied textures [Hohmann et al., 2008].

hese methods provided only limited semantic information and used strong assumptions (such us specific type of windows) in the process. In a different approach, methods for processing building façades from visual information have been introduced [Werner and Zisserman, 2002], [Lee and Nevatia, 2004]. In their work Lee and Nevatia presented a gradient projection based method for window detection as illustrated in Figure 2.15.

Their method used a single, rectified building façade as an input and relied on the assumption that window frames are oriented almost exclusively horizontally/vertically. Single façade textures were obtained from a wire-frame model of the scene composed from aerial data and façades were considered projections into such frames.

In a rectified façade texture, the horizontal projection profile of horizontal edges would indicate bottom/top frames of windows and vertical projection profiles of vertical edges left/right frames of windows (see Figure 2.16). Combination of both profiles indicated window location in a place where each profile exhibits spikes. Subsequently, a bounding box with variable borders is placed at the location and the borders of windows are refined more precisely. This approach provides a robust method for window detection as it is applicable for any type of windows with a rectangular (semi-rectangular as a special case) shape as long as the windows form a regular pattern. However, it may fail for more complex façade types with rectangular elements other than windows.

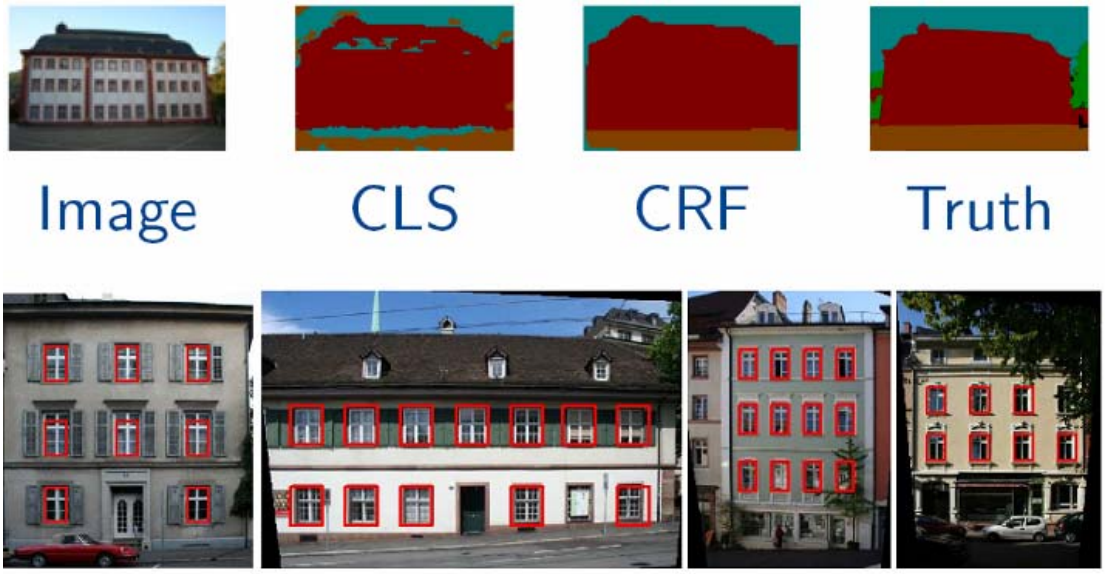


Figure 2.17: Results from the eTRIMS group research. Top row – CRF based image interpretation with pixelwise approach [Korč and Förstner, 2008]. Bottom row – window detection based on AdaBoost [Šochman, 2006].

It is also designed specifically for window detection and it is not suitable for detection of different façade elements.

Different top-down approach for façade modeling is based on a procedural generation. Architectural elements are modeled through shape grammars which are represented by a shape dictionary and a set of derivation rules.The recovery problem is formulated as a search in a space of shapes that is generated by grammar rules based on the input axiom (see Figure 2.16). For each configuration, an image-based score is computed based on trained classifiers to approximate the visual appearance of a building. As a result, such methods provide both texture and geometry descriptions of examined buildings [Wonka et al., 2003], [Müller et al., 2006].

As demonstrated by Simon et al. this approach is robust enough to process large a variety of complex architectural styles [Simon et al., 2011].

The basic problem with the approach is that the space of shapes and rules is potentially infinite and there is no straightforward relationship between intermediate grammar levels and the image.

Another group of researchers working on a streetside image processing and façade analysis is the eTRIMS group. Its focus is on a structural learning, where relations

between components and compositional hierarchies play a central role in object categorization. Such learning is particularly relevant for the interpretation of man-made objects, hence the project uses the recognition of buildings in outdoor scenes as its exemplary application domain. In general, the eTRIMS group views streetside datasets as a suitable testing domain for the research of context, giving more weight on methods and less on applications for dataset processing.

Their research focus is at:

- Application of MRF and CRF for image interpretation in a domain of man made structures at the pixel level [Korč and Förstner, 2008]. The focus is on a parameter learning of a MRF model. The examined method used for parameter approximation is pseudo-likelihood (see Figure 2.17). This research is centered at Bonn University.

- Façade image parsing focused on a window detection in rectified façades. In this research, a regular structure of window array is assumed. AdaBoost is used as a classifier and applied in scale space to detect seed façade structures. Detected seed are processed with high level information to detect (or estimate) the location of repeated structures [Šochman, 2006], [Čech and Šára, 2008], [Zara, 2004] (see Figure 2.17). Group is centered at the Czech Technical University in Prague.

- Bottom up interpretation of man-made scenes that uses blob detector algorithm, segmentation and MRF contextual classifier to extract semantic segmentation from digital photos [Jahangiri and Petrou, 2008]. Research is performed at the Imperial College London.

2.7 Centers of Excellence

In current research community, several commercial companies and academic research centers focus on urban modeling, GIS and related topics. In previous section we gave the detailed example of one of them – eTRIMS. This section lists several others that provide complete or partial solution for specific problems in urban modeling and

describe most prominent result from each group. We divide subjects into a commercial and academic groups according to their status as companies, it they offer commercial product to markets, or a status as academic research centers.

2.7.1 Commercial

Esri

Esri is the leading U.S. based software company with the focus on GIS development [Esri, 2012]. It is estimated that Esri products compose a 30% of the global market in the field, more than any of the competitors. The leading product of the company is an ArcGIS platform for creating, managing, analyzing and displaying any forms of referenced information. ArcGIS uses CityEngine software to model 3D data in an urban environment [Esri, 2012(II)]. It is based on a procedural modeling and GIS geometry. CityEngine software was developed by a company Procedural (spin-off of the Computer Vision Lab, ETH Zurich) and acquired by Esri in 2011. The research group of Procedural is centered around Dr. Pascal Müller with the research focus on shape grammars.

3C Technologies

This was a Swedish company acquired by Apple in 2011 and now reportedly working as a group under the name Sputnik [iMore, 2012]. 3C Technologies implemented a 3D mapping application originally developed as a military application by SAAB. This method is based on an aerial mapping with the resolution of 10 cm per pixel, but allows for integration of streetside images and user provided data. Terrain surface and objects are represented as a wire-frame model with applied textures. Basic interpretation of data is also performed in a form of terrain labeling. Workflow from digital data to surface models is fully automated. It is assumed that this technology will be fused with iOS Maps for 3D experience, as the terrain visualization is superior to currently used applications provided by Google.

Autodesk

Is a multinational company focused on the development of 3D design software [Autodesk, 2012]. Leading product is the AutoCAD application for computer-aided design. With the acquisition of the 3D Geo GmbH company in 2008, Autodesk entered the GIS market with the LandXplorer Studio and the current version - Autodesk® Infrastructure Modeler product [Autodesk, 2012(II)]. This platform allows for integration of data from other Autodesk applications, for example models created in AutoCAD to be used in GIS environment.

Interactive Visual Media Group – Microsoft Research

Is a group focused primary on digital image and video processing [Microsoft, 2012]. The group formed around prof. Richard Szeliski work on street view as a part of Bing Maps. Part of the research is the development of the Photosynth. In particular, Photosynth application allows for 3D experience in a form of "Synth" and Panoramas. Synth methods use a set of 2D images capturing the details of specific objects to be stitched together through the extraction of basic object geometry. Subsequently, details of an object can be viewed as individual images arranged in a 3D comprehensive way. This method also allows for integration of large number of images from different sources.

Google Research

Is the Google's equivalent to Microsoft for the research in this field [Google, 2012]. This research group is known mostly for the development of vehicle based sensors used for streetside city mapping. Group is formed around researchers, such as Richard F. Lyon.

2.7.2 Academic

The Graphics and Imaging Laboratory of the University of Washington's, Department of Computer Science and Engineering (GRAIL)

A group with a wide research field including image processing, scene reconstruction and mobile imaging [GRAIL, 2012]. Relevant research is primary in a field of urban modeling from a set of photos and video, and modeling of building interiors. Research of urban modeling is centered on researchers such as Seven Seitz and Linda Shapiro.

ETH Computer Vision and Geometry Group (CVG)

A group located at the ETH Zurich, focused on the geometric aspect of a scene reconstruction, such as calibration, extraction of shape and motion [CVG, 2012]. Relevant research is also in a modeling of large scale scenes.

eTRIMS

A group of several researchers from different universities (Bonn, Hamburg, Prague, London) with the focus on learning and semantic interpretation of urban images [eTRIMS, 2012]. Group works with aerial as well as streetside images.

Institute for Computer Graphics and Vision (ICG)

A group located at the Technical University Graz, with the focus on general computer graphics and vision fields [ICG, 2012]. Several researchers focus on the aspect of urban modeling, including aerial data processing, modeling and interpretation of streetside images.

Laboratoire MATIS

Focus of this research group is in the reconstruction of urban scenes, sensors and laser range data [MATIS, 2012]. Group also provides several open source application for researchers in this field.

There are several other groups and individuals that contribute to the research in this field. We selected those listed above as an exemplary groups because they product/research is either relevant to our research or unique in the field.

2.7.3 Journals and Conferences

Relevant research can be also found in dedicated journals and conferences, such as:

Journal Name	Impact factor
Remote Sensing of Environment	3.951
IEEE Geoscience and Remote Sensing	2.470
ISPRS Journal of Photogrammetry and Remote Sensing	2.158

Photogrammetric Engineering and Remote Sensing	0.926
IEEE Geoscience and Remote Sensing Letters	1.420
International Journal of Remote Sensing	1.182
The Photogrammetric Record	0.925
GIScience & Remote Sensing	1.000
Journal of Applied Remote Sensing	n/a
Remote Sensing Letters	n/a
IEEE Applied Earth Observations and Remote Sensing	1.140
Remote Sensing	n/a
International Journal of 3-D Information Modeling	n/a

Table 2.1: The list of research journals relevant to urban modeling and their impact factor [SCImago, 2012].

Conference Name	**Acronym**
Computer Vision and Pattern Recognition	CVPR
International Conference on Computer Vision	ICCV
European Conference on Computer Vision	ECCV
British Machine Vision Conference	BMVC
Asian Conference on Computer Vision	ACCV
International Conference on 3-D Imaging and Modeling	3DIM
3D Data Processing Visualization and Transmission	3DPVT
International Conference on Pattern Recognition	ICPR
Computer Vision, Imaging and Computer Graphics Theory and Applications	VISIGRAPP
International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision	WSCG
SIGGPRAH	SIGGPRAH
IASTED International Conference on Computer Graphics and Imaging	CGIM
International Conference on Image and Signal Processing	ICISP

Table 2.2: Several conferences in computer vision research field relevant to urban modeling [Iris, 2012], [Academic, 2012].

Chapter 3

Background

3.1 Context in a Streetside Urban Environment

Much computer vision research in general has been focused on the problem of recognizing specific objects. However, until recently the problem of understanding and formulating general object recognition as a task of properly isolating and identifying classes of objects in an agent's environment has been examined only marginally [Carbonetto et al., 2004]. This problem can be formulated as a presence of context in the image environment. Context plays important role in a human vision [Oliva and Torralba, 2007], [Halgren, 2006] and in general the application of context improves the performance in object recognition tasks, as was demonstrated by several authors [Carbonetto et al., 2004], [Singhal et al., 2003], [Heitz and Koller, 2008].

Many researchers in computer vision claim to use a context in their approaches, but the term itself has no clear definition. In a broad sense, the context is understood as "*any and all information that may influence the way a scene and the objects within it are perceived*" [Strat, 1993].

A term "local pixel context" refers to a most simple and common application of context in computer vision [Santosh et al., 2009]. It is built upon the assumption that pixels around the examined region can provide additional information for a vision task.

Figure 3.1: Different levels in streetside images. Ground and building level separated by yellow lines, building and sky level separated by red lines. Notice different sets of objects located in each level. The building level contains most of the objects of interest for environment mapping algorithms. Note that the lines present only approximate separation, as some objects extend beyond lines to different level.

This information can be accessed by simply extending the scanning window [Dalal and Triggs, 2005], [Wolf and Bileschi, 2006], or application of a Random Fields method in local neighborhood [Carbonetto et al., 2004], [Kumar and Hebert, 2005], [Shotton et al., 2006]. In general, the application of context at this level can identify misdetections caused by outliers, thus eliminate some false positives, but does not help in the recognition task itself significantly. More complex applications of context require detection of specific objects (segments, boundaries, shapes…). This approach will be discussed in more details in Section 2.5.4. Before the application of a specific method, we must examine what type of context can be used in our application domain.
A general urban scene has a very specific composition and geometry. When compared to natural scenes, the presence of man made objects set in a well defined design can provide strong contextual information. For example, an urban scene can be divided into several horizontally oriented levels – ground, building and sky levels – (see Figure 3.1) each containing very specific objects set in context with each other:

- *ground level* contains often the largest variety of objects. It is composed mostly of circulation spaces (roads, walkways), transportation devices (cars, trams, bicycles), pedestrians, vegetation, animals, traffic control devices, and other objects. Buildings

(at the building level) close to a ground level have often different appearance than the rest of a building and contain objects such as doors, shopping windows, signs, cellar windows, stairs, and others. If the detection of any such object is the focus of a vision algorithm, the identification of the ground level in digital images can be of major importance in terms of context. Most of the temporal objects at the ground level present occlusion for façades at the building level (depending on the point of view). Correct recognition of permanent objects at the ground level (circulation space elements, bottom levels of building) can be very difficult in overcrowded areas or areas with high traffic due to such occlusions.

- *building level* is where façades are located. As this work is largely focused at building façades, the detection of this level is a major task for our approach. Most of the façades are highly regular with many repetitive patterns. When the composition of a façade is considered, objects like windows, arches, ledges, are in strong contextual relationships. Occlusion from temporal objects is less significant at this level, but other sources can be present. Building façades are often occluded by vegetation or street elements (lamp poles, traffic control parts…). Many façades are also too large to fit whole into a photograph frame from a close point of view or are occluded by another façade, therefore only part of them are often present in the image. Roofs are also considered as part of a building level, however we do not include them in the façade class and are detected as a separate class.

- *sky level* usually does not contain any object of interest, but its identification can be useful for removing false positives of streetside objects detections. Detection of the sky level is not a trivial task mainly because of several special cases – presence of specifically shaped clouds, different illuminations (sunset, sunrise), or specific object in the sky (birds, planes). When the orientation of the image is not certain, a sky area can provide significant cues.

Many authors of computer vision methods use a only general term “context” when referencing the application of other-than-visual data. This led to very broad definition of what context is and what type of data is actually used. Subsequently, several researchers have tried to define different domains that are considered to provide

context for computer vision. Santosh in his work on empirical study of context [Santosh et al., 2009] defines several types of context. For our work, the most relevant are the semantic context and the geometric context:

Semantic context *indicates the presence and location of objects or materials in the scene. It can be also used to identify the scene category* [Oliva and Torralba, 2001]. Semantic context is a more general term and as such, it is used regularly in our work. For example, we identify general surfaces in an image and continue with identification of the specific object only in semantically correct areas e.g. windows in façades. We also apply semantic context in the façade separation process, where the results are presented for the method without the context and with the context, giving the exact value of context application. In general, semantic context limit the area of image, where the object recognition is performed, thus eliminating false positives of objects that would be detected outside such area.

Geometric context *aims to capture the coarse 3D geometric structure of a scene, or the "surface layout", which can be used to reason about supporting surfaces, occlusions, contact points, etc.* This type of context plays an important role in human vision, as was demonstrated in [Bar and Aminoff, 2003]. The application of spatial (geometric) context as an early facilitator of object recognition in human vision is provided by an activation of cortical context neurons that appear to store spatial relationships. These spatial relationships can be determined without high frequency information, thus provide for a very early stage visual recognition [Bar et al., 2006]. Objects inside urban scenes have very strong geometric context e.g. windows are arranged in rows, arches above windows e.g. An urban scene itself is organized in a predefined geometric setup, but when projected into 2D image, geometric relation can be distorted. For example in a real scene clouds are always above buildings, but in a digital image, this may not be a case. For this reason, a probability method has to be applied when considering geometric context in streetside images. In a section 4.1.3 we examine a method of extracting spatial relations between classes in street-side images.

[Santosh et al., 2009] also define a temporal context as a context from temporally proximal information e.g. nearby frames of video, images taken before or after the

given image. This type of context is similar to what we examine as an effect of redundancy (see Section 3.3), however we assume that the camera is moving and/or changing direction of optical axis.
An example of context application is given in a work of [Perko and Leonardis, 2008]. Authors use context to enhance a standard task of computer vision in an urban environment – pedestrian detection. In processed images, a specific area is selected with a help of visual and geometric cues as a "focus of attention", where pedestrians are likely to be located. This is a principal example of semantic context, which is also extensively applied in our work.

3.2 Random Fields

In many computer vision applications, *graphical models* have been presented as a method to introduce context information into processes. Graphical models are explained as a combination of two areas – graph theory and probability theory [Murphy, 1998]. This combination provides several advantages:

- When working with real world images, one must deal with special cases, in-class variation, occlusions or image noise. The solution is in probabilistic modeling, giving required flexibility to the task.

- Even the image should be considered a global model (and for a human vision, this is the case), in computer vision, the raw amount of data makes this approach often computationally intractable. Reducing context interaction to only a local level on the other hand does not often provide sufficient results. The graph theory can introduce required balance between local/global approaches.

Graphical models in a vision framework are further differentiated as causal/directed and noncausal/undirected. The causal models are mostly used in segmentation problems [Cheng and Bouman, 2001], [Feng et al., 2002]. The most common noncausal graphical models in computer vision are **Markov Random Fields** (MRF) [Kindermann and Snell, 1980].

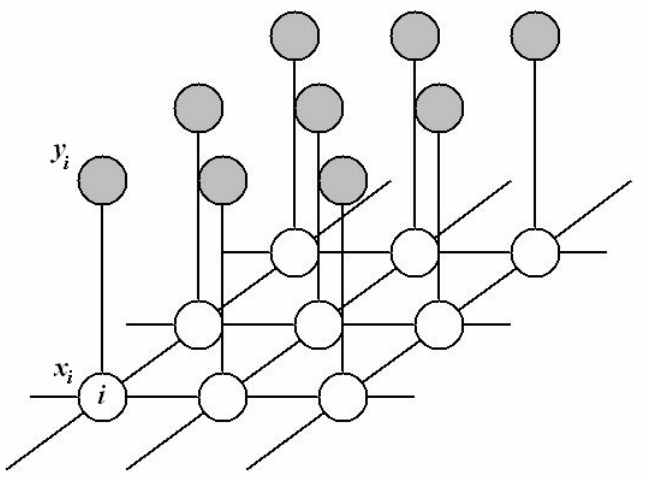


Figure 3.2: A typical application of Markov Random Field (MRF) in computer vision. At each node i, the observed data is denoted as $\boldsymbol{y}_i$ and the corresponding label as x_i. For each node, only local observations are possible.

MRF have been used extensively in labeling problems for classification tasks from early works in computer vision [Cross and Jain, 1983], [Besag, 1972] and for image synthesis problems [Zhu and Wu, 1998]. In a labeling task, MRFs are considered to be probabilistic functions of observed data in measured sites of the image and labels assigned to each site. Given the observed data $\mathbf{y} = \{\mathbf{y}_i\}_{i \in S}$ from the image, and corresponding labels $\mathbf{x} = \{x_i\}_{i \in S}$, where S is the set of sites, the posterior distribution over labels for MRF can be written as

$$P(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z_m} \exp\left(\sum_{i \in S} \log p(\mathbf{y}_i \mid x_i) + \sum_{i \in S} \sum_{j \in N_i} \beta_m x_i x_j \right), \qquad (3.1)$$

where Z_m is the normalizing constant, β_m is the interaction parameter of the MRF and N_i is the set of neighbors of site i. The pairwise term $\beta_m x_i x_j$ in MRF can be seen as a smoothing factor. Notice that the pairwise term in MRF uses only labels as variables, but not the observed data from the image. In this arrangement, the context in a form of MRF is limited to be a function of labels, thus allowing for semantic context and limiting geometric context to a structure of MRF graph (see Figure 3.2). Any relations between sites observable from the image are disregarded. This makes the MRF applicable mainly for simpler forms of context.

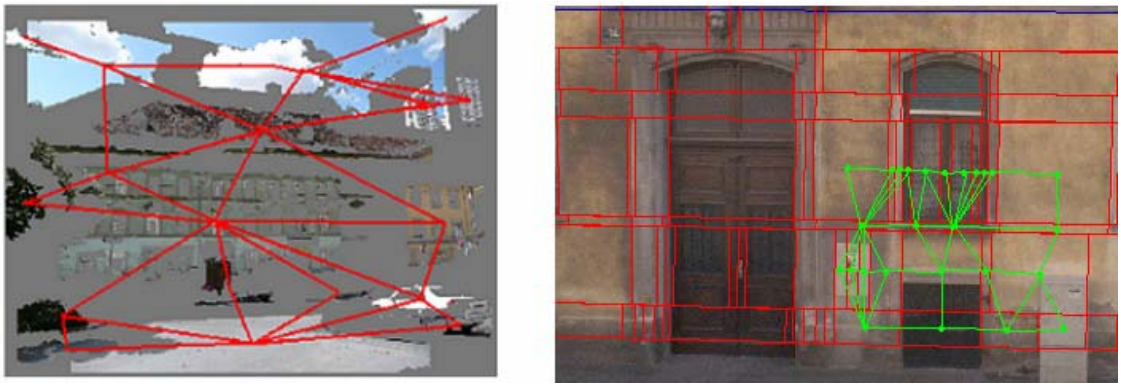


Figure 3.3: The implementation of graphical models in our dataset. In both images, only some relations are visualized for better orientation. In left image a set of segments is identified in the image and a graph structure is placed over it such that each segment (site) is considered a node of the graph and an edge is placed between segments with common borders. In second image a façade is segmented into blocks. Each block is than considered a site, thus a graph node is placed in each block and two blocks are connected if they are neighboring.

In our work we use the **Discriminative Random Fields** (DRF) [Kumar and Herbert, 2006] to cope with such limitations. DRF are based on the concept of Conditional Random Fields (CRF) proposed by [Lafferty et al., 2001] for the segmentation and labeling a text sequence. In the Figure 3.3 we can see the application of RF in our methods. In a first case (left image) a graph structure is constructed over segments of an image. The number of nodes varies between 50 to 100 in general. In a second case (right image) a graph structure is placed over a set of blocks that segments a façade. In this case, a number of nodes can range from around 300 to 1000, depending on the complexity of façade (very low for simple façades, but rises significantly for more complex ones). In both cases, the visual features of segments/blocks and their spatial relations are considered as observations. Details of feature descriptors are located in relevant sections (4.1.2, 4.1.3 and 7.3.1). In subsequent text we will describe the concept of DRF in more detail, as the notation of this method is used extensively in our work. Our goal in this section is to make a clear distinction between visual feature, which are applied in RFs model as a unary potential and contextual features applied as a pairwise potential. This distinction is referred to a number of times in the text of this work and in this section we provide mathematical implementation of this idea. We

provide a simplified explanation of both potentials to demonstrate how the involvement of context in our work is used for classification. The CRF are discriminative models that represent the conditional distribution over labels.

Definition 1. CRF: *Let G = (S, E) be a graph such that* $\mathbf{x}$ *in indexed by the vertices of G. Then (*$\mathbf{x}$*,* $\mathbf{y}$*) is a conditional random field if, when conditioned on* $\mathbf{y}$*, the random variables* x_i *obey the Markov property with respect to the graph:* $P(x_i|\mathbf{y}, \mathbf{x}_{S-\{i\}}) = P(x_i|\mathbf{y}, \mathbf{x}_{\mathcal{N}_i})$*, where S-{i} is the set of all the nodes in the graph except the node i,* $\mathcal{N}_i$ *is the set of neighbors of the node i in G and* $\mathbf{x}_K$ *represents the set of labels at he nodes in set K.*

Using the Hammersley-Clifford theorem [Hammersley and Clifford, 1971], assuming only pairwise cliques potentials to be nonzero, the conditional distribution in DRF over all labels **x** given the observation **y** can be written as,

$$P(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A_i(x_i, \mathbf{y}) + \sum_{i \in s} \sum_{j \in N_i} I_{ij}(x_i, x_j, \mathbf{y}) \right), \tag{3.2}$$

where Z is the normalizing constant, $-A_i$ is the unary and $-I_{ij}$ pairwise potential. The principal differences between the conditional model (2) and MRF distribution (1) are that the unary potential $A_i(x_i, \mathbf{y})$ is a function of all observations instead of only observation $\mathbf{y}_i$ in specific site i and the pairwise potential in (2) is also the function of observation, not only labels as in MRF. In an example from our work (see Figure 3.3), this would mean that for each node/segment in the image, we have also visual and spatial information from all other segments. Given this information, we can involve classifiers based on global features, such as a position matrix described in Section 4.1.2. Furthermore, in DRF as an extension of CRF, both unary and pairwise potential are designed using arbitrary local discriminative classifiers. This feature allows algorithms based on DRF to be specifically designed to work with particular structure of input data and is useful when applied to high-dimensional complex visual data. Also DRF are generally defined over 2-D lattices and allow graphs with loops. Even though this makes DRF more easily applicable to visual data, it makes parameter learning and inference significantly harder task.

In DRF, the unary potential $A_i(x_i, \mathbf{y})$ is considered to be a measure of how likely a site i

will take label x_i given the observation in image **y**. The unary potential is modeled as local discriminative model that label site i as class x_i as

$$A_i(x_i, \mathbf{y}) = \log P'(x_i \mid \boldsymbol{f}_i(\mathbf{y})), \tag{3.3}$$

where $P'(x_i \mid \boldsymbol{f}_i(\mathbf{y}))$ is the local class conditional at site i and can be any probabilistic discriminative classifier. In a work of Kumar [Kumar and Herbert, 2006] Generalized Linear Models are suggested as one possible option. In this case, the classifier function can be compactly expressed as

$$P'(x_i \mid \mathbf{y}) = \sigma(x_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{h}_i(\mathbf{y})), \tag{3.4}$$

where $\boldsymbol{w} = \{w_0, \boldsymbol{w}_1\}$are the model parameters (w_0 is a bias parameter) and $\boldsymbol{h}_i(\mathbf{y})$ is the transformed feature vector at site i composed of a image feature vectors kernel mapped into a high dimensional space. This classifier function ensures the linearity of unary potential and can be seen as a discriminative counterpart of a generative unary function of MRF.

In DRF, the pairwise term is considered to be a measure of how the labels at neighboring sites i and j should interact given the observed image **y**. The pairwise term is defined as:

$$I_{ij}\left(x_i, x_j, \mathbf{y}\right) = \beta\left(Kx_i x_j + \left(1-K\right)\left(2\sigma\left(x_i x_j \mathbf{v}^T \boldsymbol{\mu}_{ij}\left(\mathbf{y}\right)\right)-1\right)\right), \tag{3.5}$$

(where $0 \le K \le 1$, **v** and β are the model parameters) and $\boldsymbol{\mu}_{\mathrm{ij}}(\mathbf{y})$ is a feature vector. This formulation can be seen as an extension of Markov's pair wise term (if $K = 1$ in DRF, pair wise terms are identical to Markov's), but allows us to apply more complex context as a feature vector $\boldsymbol{\mu}_{\mathrm{ij}}(\mathbf{y})$. Parameter K determines the contributions of two terms present in the formula. The first term $x_i x_j$ is data independent and provides a label smoothing, while the second term map the pairwise logistic function. The value of K can determine what type of context is applied. For semantic context, high K value ensures, the relations between labels is examined. When K values is low, relationships between sites are examined, thus geometric context can be applied.

Parameter Estimation and Inference

Given the equations (4) and (5) the parameters of a DRF model are $\theta = \{\boldsymbol{w}, \boldsymbol{v}, \beta, K\}$. From the definition of DRF in [Kumar and Herbert, 2006], the parameters of class generative models $p(\boldsymbol{y}_i|x_i)$ and of the prior random field on labels $P(\boldsymbol{x})$ are not learned separately in contrast to the MRF framework. In a work of Kumar, the standard maximum-likelihood method is applied to learn the parameters. This is a NP-hard problem due to evaluation of normalizing constant Z. For this reason, an estimation of parameters based on the pseudolikelihood is used and defined as

$$\hat{\theta}^{ML} \approx \arg\max_{\theta} \prod_{m=1}^{M} \prod_{i \in S} P\left(x_i^m \mid \boldsymbol{x}_{N_i}^m, \boldsymbol{y}^m, \theta\right), \qquad (3.6)$$

where m are indexes over training images and M is the total number of training images. The formula for single image $P\left(x_i \mid \boldsymbol{x}_{N_i}, \boldsymbol{y}, \theta\right)$ is evaluated based on parameters from equation (2). As the pseudolikelihood is not a convex function, a good initialization is necessary to avoid local maxima. This can be achieved through the computation of standard maximum on log-likelihood in training data.

In the inference process, our aim is to find the optimal label configuration $\boldsymbol{x}$ over the image sites, given the observation $\boldsymbol{y}$. Maximum A Posteriori method is suggested as a solution to the optimization problem in [Kumar and Herbert, 2006]. The cost function for optimization is defined as $C(\boldsymbol{x}, \boldsymbol{x}^*) = 1 - \delta(\boldsymbol{x} - \boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ is the true label configuration. Exact solution can be computed if $K \geq 0.5$ and $\beta \geq 0$. Experiments shows, the Maximum A Posteriori method performs poorly when β takes large values. Other suggested methods for parameters interference are the Maximum Posterior Marginal and Iterated Conditional Modes [Besag, 1986]. In our work on semantic segmentation, we use Belief Propagation method to estimate DRF probability.

3.3 Redundancy

Even though the majority of computer vision algorithms is focused on a single image as an input dataset, the presence of multi-view datasets is a common reality in current settings. Automatic processing of real world digital image data continues to present

challenges for researchers and many computer vision tasks are considered unsolved. “Hard problems” may become more tractable if one generalizes the input data to consist not of a single image, but of a stack of multiple images. We can denote this as “redundant” or “multi-view” input data. Therefore in our image databases, we usually want to employ multiple images of any given scene. Before application of a computer vision algorithm in a multi-view scenario can be examined, two principal problems have to be addressed – the organization of a multi-view dataset and the image matching.

3.3.1 Organization of Multi-View Dataset

Depending on the source of an image stack, different strategies can be used to exploit redundant information and different results can be expected. Some datasets are specifically designed to provide redundant information (e.g. organized datasets, industrial systems), but there is also a vast amount of unorganized sources (e.g. open online datasets, crowd sourcing). In our work, we focus on what type of redundant information can these sources provide. We can identify several types of redundancy in regard to the position of cameras:

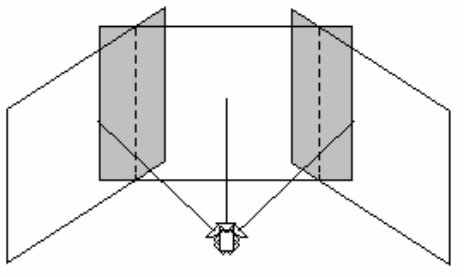


Figure 3.4: Camera setup in star formation. Gray areas denote overlapping regions in images taken form one single position.

(a) Multiple views from a single position using rotation

This type of redundancy is usually present from industrial systems, where the multiple cameras are aligned in a ”star formation” (see Figure 3.4). In this case, the rotation between images is well established and calibrated; the overlap areas between images provide “redundancy” in precisely defined manner. Also,

this kind of setup may occur in crowd-sourced type datasets, when a user (photographer) makes different images from one single position. The rotation parameters will not be known in this case and the redundant areas must be established through a search for correspondences in the images. There are no geometric differences for a given object, but the context may change in multi-view (due to new objects in different views), as well as the visual features of the objects.

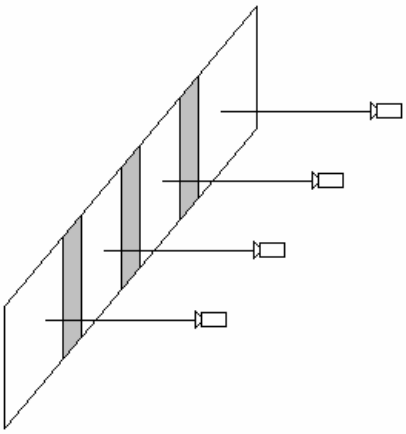


Figure 3.5: Camera setup with parallel axes. Gray areas denote overlapping regions, typical for images acquired from a moving platform like a car.

(b) Multiple views from varying position using translation

This type of redundancy is generally present in systematic environment mapping (see Figure 3.5). It will result from industrial systems or from hand-held cameras if a purposeful "strip" of images is being collected, often this is the case in planning for a 3D reconstruction. The translation of the pose between the views is more or less regular, but in a natural environment, the high level of regularity is sometimes difficult to achieve.

(c) Multiple views from varying position

This type of redundancy is present in an unorganized dataset, usually obtained from hand held cameras. It can be observed in a crowd-sourced database that provides a large volume of data. The lack of organization causes difficulty with image alignment and matching. Even the state of the art block adjustment

algorithms today need dozens of views of the single object to correctly establish matches. Several approaches have been designed to create some kind of organization structure in this type of datasets. Usually, some number of correspondences have to be established first [Nister, 2004], [Bujnak et al., 2008] and the camera parameters have to be determined [Irschara et al., 2007]. It is therefore assumed that the overlaps and thus the "redundant information" can be obtained from this type of data.

We can also consider different viewing directions, when examining the redundancy. Digital images are usually taken with the intention to capture some specific information about the object. The viewing direction is subsequently set with this purpose. In a human held camera setup, view direction is usually set directly towards the object. But the position of the camera is often arbitrary, therefore the inclination and perspective distortion of the object is hard to establish. In the industrial system setup, the process of image capturing is usually designed to achieve desired viewing directions for each camera. In this case, we can process this information as a prior knowledge.

When we are considering building façades, best data can be obtained, if the camera view direction is set directly towards a façade plane thus is orthogonal to this plane. In this case, the level of detail is dependent only from the distance between camera and the façade plane. Any redundant image taken from the same distance is likely to contain less information than the orthogonal view. This can be observed in the process of façade rectification, when the perspective distorted façades exhibit a large amount of blurring and distortion when rectified. Compared to this, façades captured from orthogonal view contain more accurate information about the edges and details [Liebowitz and Zisserman, 1998]. Usefulness of different view directions in the presence of an orthogonal view can be increased, when some part of the object is obscured in the orthogonal view, but visible in a non-orthogonal view.

(a) **(b)**

Figure 3.6: Original image with marked point (a) and paired image (b) with corresponding point (green) located at the epipolar line (light blue). False epipolar line (dark violet) introduced in [Recky, 2006] is applied to pinpoint the location of corresponding point.

3.3.2 Image Matching

Before the redundant information can be applied in a computer vision algorithm, images have to be matched (usually by the means of corresponding points).

To establish the redundancy and image overlaps in a stack of images, multiple view geometry methods can be applied. This approach is explained in the book [Hartley and Zisserman, 2004]. A connection between two images can be established through epipolar geometry. This geometry is independent of scene structure and only depends on camera parameters and relative pose. These parameters can be used to obtain a fundamental matrix. Given the fundamental matrix F, an epipolar line defined by the equation F**x** = 0 for an arbitrary point **x** from the first image can be computed. A point **x'** in the second image, which is corresponding to the point **x** in first image is then located always on the epipolar line. We can assume from the definition of the fundamental matrix, that given the camera parameters, or enough correspondences in the images, we can estimate the relative pose of cameras, one to each other [Horn, 1990]. If camera parameters are unknown, a fundamental matrix can be approximated using the normalized 8-point algorithm described in [Hartley, 1997]. Given arbitrary

eight corresponding points (from which neither three are co-linear), parameters of fundamental matrix can be estimated with the application of SVD decomposition of point matrix.

In our previous work [Recky, 2006] we used this approach for fundamental matrix estimation and introduced a False Epipolar Constraint (see Figure 3.6) to increase the speed and precision of corresponding points detection. As shown by [Wang et al., 2007], this approach can decrease the time consumption of a matching algorithm to almost 1/50 of the original without reducing accuracy. This approach can also provide us with the information about the type of redundancy, we are dealing with.

In a controlled situation (e.g. a dataset taken by an industrial system, robot or by users for environment mapping/reconstruction) the camera parameters are known beforehand. A more complicated situation is when an unorganized image dataset is considered. In the work on PhotoTourism and subsequent online application PhotoSynth, [Snavely et al., 2006] demonstrated how such datasets can be automatically aligned and calibration can be extracted. This method allows construction of a sparse 3D point cloud from such datasets and estimates the camera positions to establish dataset alignment.

When considering the problem of image matching for classification, the ideal situation is to apply pixel-by-pixel matching e.g. dense point clouds. However for current application (primarily in a crowd sourcing scenario, when calibration is not available), generally only some points are matched and a sparse 3D point cloud is created. When working with building façades as our objects of interest, we can resolve this problem by considering façades to be planar. This approximation allows us to interpolate between corresponding points in the image and compute correspondences also for points where the matching was not performed. Another approach is to consider segments as basic units of the image instead of pixels and perform segment-to-segment matching. This can be achieved when one or more matched corresponding points are located inside segments from matched images. However several problems have to be solved in this approach, namely when one segment is matched to several different segments in the corresponding image, when there are no matched points inside a segment and when segments are covering multiple classes. For these reasons, segment-to-segment matching is practically feasible only when segmentation is more robust and precise then point matching.

3.4 Objects of Interests

In a Section 2.5.3, we discussed shape grammars as the tool for suitable representation of objects in an urban scene. In this model, non-terminal symbols in streetside images generally cover large areas and are composed of several different object classes (terminal symbols). Such large areas do not have a specific shape – even stable shaped non-terminal symbols (like façades) are mostly trimmed or partially obscured. Because of high visual variability, non-stable shape and area size, local descriptors are not suitable for detecting non-terminal symbols. For these reason, non-terminal symbols can be identified in the image by the mean of image segmentation (see Section 4.1).
However, for the recognition of the small elements and details (most of the terminal symbols) general segmentation is usually not suitable. As we want a detailed definition of terminal symbols for better visualization, other methods than segmentation present better results, as they provide more precise borders. For this purpose, we apply gradient projection methods, which focus on a specific geometry and shape of façade elements. We combine our work on segmentation, context and redundancy with such gradient projection method to achieve state-of-the-art results in semantic segmentation and façade element detection.
In our work, we primarily focus on building façades and objects contained within. We also identify other objects in street-side images, but only as non-terminal symbols e.g. one class for all circulation spaces, or vegetation, but not terminal symbols in these classes (e.g. tree branches, circulation spaces elements). There are two reasons for choosing building façades as our objects of research:

- Façades are considered primary objects of interest for many environment mapping algorithms. They are usually the primary focus of reconstruction algorithms in urban environments, as they represent a large volume of streetside image data. The automatic and robust processing of building façades from a single streetside image or set of images would present a major contribution for urban environment reconstruction effort [Becker, 2011], [Simon et al., 2011].

- For the research of application of context in computer vision, façades present unique properties. The highly regular composition of façades suggests that context can play a crucial role in the recognition of a façade's elements yet there is a high variety in visual appearance of these elements. This setup provides a challenge for visual recognition algorithms and an opportunity for context-based algorithm to contribute results. Similarly, façades present unique properties for our other focus of research – redundancy. As façades are static objects with very little moving parts (doors, windows…) the appearance and geometry of the façade would not change significantly in different timeframes. This allows examination of the effect of redundancy (multi-view) in a relative stable environment.

In our work, we propose a workflow in which façades and subsequently façade elements are identified in a top-down process, from more general non-terminal objects to specific terminal objects. We first identify primary classes in streetside images, including the building class. In this class, we identify specific façades. In each façade we identify levels (street level and window levels) and in each level, terminal symbols, e.g. windows. In this process, each step provides semantic context information for subsequent steps, limiting the area of images where the objects of interest are located. This approach can be used also for other non-terminal objects in the image, with few exceptions. In a step from specific façade to façade levels, we rectify the façade. In this case, we make an assumption that the façade is planar. If it is not, the rectification will cause distortion in non-planar areas, but the identification of non-terminal symbols can still be performed in a case they are not distorted. This is due to our modifications to the gradient projection algorithm which makes it possible to process textured façades (in this case, a distortion is processed as a texture). For the identification of façade levels and elements, we rely heavily on the specific the geometry of building façades. In most other-than-façade non-terminal symbols (e.g. circulation spaces), these heuristics cannot be applied and a different approach has to be used.

3.5 Datasets

Our primary source of input is a color digital photograph taken from the street level. As we focus our work to be as robust as possible, we use multiple sources with different photograph specifications. The resolutions range from 520x390 pixels to 2576x1932 pixels. Lower resolution images would not display significant objects in a sufficient size for processing methods. Top resolution is limited by the depth of recursive procedure during segmentation (memory allocation limit in Visual C++). Methods were tested also on higher resolutions images, but the limitation of a recursive segmentation procedure caused unnatural segment borders and the subsequent application of contextual information was less effective. We use lossless raster images (BMP, PNG, TIFF) and lossy compressed images (JPEG) in our experiments. We observed minor issues with artifacts caused by compression in segmentation results (primary in low resolution, highly compressed images), but the impact on labeling was insignificant.

We assume that all photographs are oriented correctly (ground level at the bottom, sky at the top). If this is not the case, orientation can be corrected. If calibration is available, rotation parameters of the camera shows how the photograph is oriented. If no calibration is available, visual cues can be used to estimate the orientation e.g. sky should be located on the top and its saturation should increase upwards. In our work, we use three different datasets:

General Images

A sataset of images was acquired from the internet and inside sources of ICG TUGraz. This dataset was applied to test special cases and increase the variation of test data. Parts of this dataset were acquired from the LabelMe database [LabelMe, 2011] and the eTRIMS [Korč and Förstner, 2008] database. These sections are partially labeled and were used as test and training data. The rest of the dataset was downloaded using an image search engine with queries “building”, “façade”, “urban”, “street side” or provided by colleagues at ICG TUGraz. No calibration information was used in this dataset nor was there 3D information and image matching involved during tests on images.

Figure 3.7: An example of images from *General Images* dataset. This dataset was designed to cover a large variety of different scenarios, including variations in façade types, occlusions, viewing angles and weather conditions.

This dataset was used exclusively in single-view scenarios. Most images in ground truth for learning are from the General Images dataset, as it provides the largest variety of objects, visual features and scene compositions and consist of 250 images (see Figure 3.7).

Tummelplatz Dataset

A dataset was created specifically to simulate an open, multi-user image source. All images were acquired at Tummelplatz Graz, using two different cameras, in different weather conditions and times of a day. To achieve a required variety in illumination, images were taken between 10:00 AM and 17:00 PM, on three days with different cloud coverage (sunny, partially clouded, full cloud cover). Tummelplatz was selected as it provides a high variety of objects, scenarios and viewpoints. The dataset contains five primary building façades, each shown in approximately 60 different images (see Figure 3.8(b)). Façade types wary from historical building to modern architecture. Occlusions are caused by trees and pedestrians. Part of the dataset was calibrated and a 3D point cloud for the purpose of image matching was created by Arnold Irschara [Irschara et al., 2007]. This dataset was used in both single and multi-view scenarios and include 290 images in total.

(a) **(b)**

Figure 3.8: An Example of the image type from the Industrial System database (a) with the parallel optical axes providing a high level of redundancy from sequential exposures in a moving vehicle; in this example the optical axes are pointing halfway forward. In the Tummelplatz-Graz database (b) the viewpoints and viewing directions of manually collected images can differ significantly for each object.

Industrial system (CityFit) Dataset

In the Industrial System dataset images are taken by a calibrated multi-camera apparatus mounted on a car (see Figure 3.9). This setup creates overlapping images with a rigorous and calibrated geometry from a single image-taking position, and delivering for each object point multiple images from that single sensor position.

By moving the car and repeating the image collection, the level of redundancy gets further increased. Carrying along a scanning laser arrangement with the imaging sensors provides one with additional range information and means to match the images. Figure 3.8(a) is an example of a data set that consists of 250 images from each camera on the car platform. In our work, we used the input of only two cameras – one sideways and one frontal-sideways tilted camera (see Figure 3.8(a)). The data base supports investigations into the issues of the types of redundancies, namely multiple images, all taken with parallel optical axes from different camera positions; or multiple images all taken from a single position but with different directions for the optical axes, and various hybrids between these two concepts.

Figure 3.9: An example of a camera system mounted on a car. It is designed to cover wide viewing range.

The dataset is complemented with LiDAR scanner data and a method for image matching was developed based on this data. In general, this kind of matching can be considered equivalent to a 3D point cloud obtained from vision based photogrammetry. In both cases, the quality and precision of matching highly depends on specific input data – in case of photogrammetry, the precision of camera calibration, in case of laser data, the GPS positioning precision. Given the precision of such inputs, both methods can achieve geometric accuracies in global coordinates up to ±1-2 cm. Both approaches are examined in more detail in a work of [Leberl et al., 2010].

3.5.1 3D information

For part of the *Tummelplatz* dataset and for the *Industrial System* dataset we have 3D information about the scene in a form of either sparse 3D point cloud or LiDAR data. We use this information for image matching in a multi-view scenario; however we do not describe any use of 3D data in recognition process. This decision is based on two reasons:

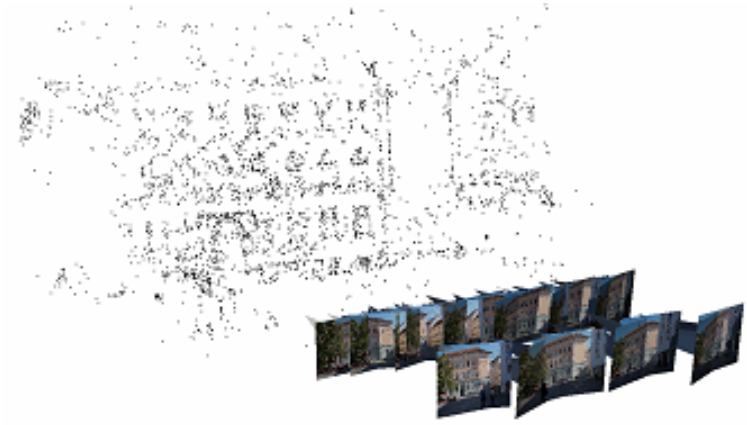


Figure 3.10: An example of 3D point cloud in the Tummelplatz dataset. This point cloud was created from 28 views and consists of 3498 points (thus of 125 points per image in average). Of a given façade one has 2623 points to work with. Provided by the method designed by [Irschara et al., 2007].

- Due to a robustness requirement, we aim to provide results for very general input data. We do not consider any specific requirements for the input dataset and we designed our methods to work in single-view scenario (without image matching) as well as with multi-view scenario. This decision was based on the observation, that it is still difficult for current methods to obtain sparse point clouds, even when there is a good number of images of the scene in the dataset. This is especially the case in unorganized datasets, where calibration is missing or is not precise.
- Even we do not apply 3D point cloud in the recognition process in this chapter, we use methods that allow for such application. For example, in a DRF model, 3D data can be used in an observation vector $\boldsymbol{y}$. We do not use such application in Chapters 4, 5 and 6 of this work as our primary focus is to examine the difference between single and multi-view scenarios and introduction of new data would not meet the back-compatibility requirement for results evaluation. However, we examine the involvement of 3D data for the recognition task in more details in Chapter 7.

Figure 3.10 presents the example of a 3D point cloud for a single façade in the *Tummelplatz* dataset. We can observe that detected corresponding points are clustered

around objects of the façade, such as windows, door or ornaments. In smooth, planar areas, the detection of points is lacking. This type of clustering is useful in our approach, where we can interpolate between three closest corresponding points, to estimate correspondences, which are not detected automatically. In areas which are non-planar, the high density of detected correspondences assures good approximation and in planar areas, we can still approximate from far away points. However this type of clustering is only present in image-based sparse 3D point clouds and not in the laser scanner data in our IS dataset.

We can also observe from Figure 3.10 that the sparse point cloud was created from 28 views, however the dataset contain 64 images, where this façade is located (at least 20% of façade area). The rest of the images were not matched automatically by the algorithm, thus cannot be matched through a point cloud. This omission was caused by automated matching algorithm, as it was not able to correctly identify interest points in other 36 mages. For this reason, we marked several façades manually, to increase the test dataset. We mark four or more points on the façade borders and place them in a global coordinates and for all points inside, we interpolate between three closest points the same way as with the point cloud. In this approach, the entire façade is considered planar and small reliefs are disregarded. In both cases (point cloud and manual marking), the interpolation is not linear, as we must consider perspective distortion for the façade. Thus we must compute the inclination of top and bottom borders. Subsequently, the interpolation is the function of perspective.

3.5.2 Annotation

We developed several annotation tools to test various aspects of streetside image processing. For semantic segmentation testing, the pixel-by-pixel labeling has to be used as ground truth. We achieve this form of annotation by automatic segmentation and subsequent manual classification of segments. In locations where segmentation was not successful to separate different classes into different segments or the borders of objects were not approximated correctly, the error was corrected manually. We also used an open eTRIMS labeled database, as it provided ground truth data with sufficient accuracy. The eTRIMS dataset does not include “roof” and “cloud” classes, so these were labeled manually (see Figure 3.11, second row). The annotation of

LabelMe database proved not to be precise enough for our purpose. This is primarly due to insufficiently labeled borders of the objects, as occlusions are often neglected (and included in the object area), or vegetation class borders labeled only roughly. For this reason, we used only specific objects from this dataset, or made our own annotation. It should be also noted, that in our ground truth dataset, borders of some specific objects are not well defined. This is primary the case for borders between clouds and sky and borders for vegetation as it is a hard task to label such objects pixel-by-pixel manually. When manual labeling is involved in ground truth, we must consider errors in labeling to contribute in overall testing results.

For our work with building façades, we developed another annotation tool capable of labeling specific façades in the image. In this tool, a façade is identified by its borders as a set of at least four points connected by lines. Four points are used, if the rectangular façade is fully visible in the image. More points can be used, if façade has different shape, or is only partially visible (see Figure 3.11, two bottom rows). In this annotation, façades are considered planar objects located inside marked borders. When two sides of a building are visible in the image, each side is labeled as a separate façade. Façades, which have large parts occluded, trimmed from the image or are under severe perspective distortion are not labeled. The database with labeled façades is used as a ground truth for a façade separation method, but also as an input for façade elements processing methods. In the automatic workflow (from streetside images to scene descriptions) façade separation algorithms would provide the input for these methods, but for the testing, we used hand-labeled datasets as the form of model input (see Figure 3.11).

Figure 3.11: Examples of ground truth annotation. Top row – semantic segmentation ground truth created to learn classifiers (each class is color coded). Second row – ground truth created from eTRIMS dataset (roof and cloud classes added). Below –

façade annotations ground truth created for methods focused on façade processing (each façade is marked with different color).

We developed ground truth datasets for testing window detection algorithms. This is based on the testing dataset for segmentation, except windows are no longer labeled as unidentified, but a class label is assigned to them. Several more images from the *Tummelplatz* dataset had been labeled exclusively (only windows) to test multi view for window detection. Windows are marked by a rectangular bounding box and include the window frames.

3.5.3 Software Implementation

The implementation of methods was done in C++ in Microsoft Visual Studio 2008. All methods were processed on a CPU. Beside standard C++ libraries, we use OpenCV library[1] and Jpeg library[2]. Tests were performed with a following hardware setup: Intel 2660Mhz, 2 GB RAM, GeForce 8800. For 3D visualization we use OpenGL toolset[3]. All further open code solutions used in this work are marked as footnotes in relevant sections.

[1] http://opencv.org/
[2] http://gnuwin32.sourceforge.net/packages/jpeg.htm
[3] http://www.opengl.org/

Chapter 4

Semantic Segmentation of Streetside Images

4.1 Context-Based Semantic Segmentation

The goal of a semantic segmentation algorithm is to retrieve the image content information through classification of each region in the image into some predefined classes (usually definable for an application domain and type of image). In this section, we consider single street-side images and introduce a semantic segmentation method for this specific domain [Recky and Leberl, 2009]. Subsequently, we extend the problem into multi-view [Recky and Leberl, 2010]. Common classes in street-side scenes are the building façades, sections of road (ground level) and sections of sky. We also consider vegetation, clouds, building roofs and grass areas. Following algorithm is presented in this chapter:

Algorithm 4.1

Input: Single streetsite image, or multi-view dataset of matched images

1. Run low threshold Watershed Segmentation on an image to segment it into patches
2. Merge patches into larger segments using the Visual Similarity measure
3. Connect segments into graph and establish DRF model

4. Use Belief propagation to estimate optimal solution for DRF
5. Establish final labeling of image's areas
6. if (multi-view is present) merge labeling from matched images

Output: Semantically segmented images (image areas are labeled as specific classes)

We introduce several innovations in this approach

- Merging of patches into segments with a goal to represent real objects as a focus of context application. For this, we use our own criterion – Visual Similarity measure.
- A global model of DRF where entire image is connected into a graphical model instead of local models
- We examine several multi-view scenarios with regard to image overlaps and camera positions

This workflow can also be described in more details as follows:
At first, an image is segmented into large, logically coherent regions created from small, merged patches. It is assumed that only one object class is associated with one region. During the subsequent classification, only regions are considered as the objects of classification. To improve the result of classification, spatial relations between the segments are examined. An example of spatial rules is the fact that in a majority of properly oriented street-side images, building façades are located below sections of the sky. If in the classification output this probability rule is not met, it may indicate an error in classification. These spatial rules get represented as a Discriminative Random Field (DRF) as they represent a context between objects in the image. Visual classifiers and spatial relations in DRF are learned in a supervised process. As it is described in [Hoeim et al., 2005], only a small number of training pictures is required to train the classifiers. For this purpose, we use the hand-labeled ground truth database (see Figure 3.11(Top)). The same database is used for training of the DRF [Wallach, 2002].

4.1.1 Segmentation

Localization of the region borders and position is formulated as a segmentation problem. Several requirements must be met by the segmentation to cope with the

problems presented in street-side scenery. At first, regions have to be logically coherent. It means for example, a single region should contain only one building façade (or part of it) and should not extend to façades of different buildings, or to the sky or ground regions (for a definition of "façade", see Section 1.2.1). But also, segmented regions should be as large as possible.

When examining pictures of street-side scenery, it is obvious that texture covariance can change rapidly through a logically coherent region. As an example, in one building façade, regions with low covariance alternate with regions containing ornaments or pillars, where covariance is high. But both of these regions may still be part of the same building façade, so we would like it to be considered as one region. This requirement is not easily met, because segmentations are usually designed to distinguish between such regions. Also, borders between two regions can be well-defined in street-side images, but they may also be very smooth (for example, between clouds and sky regions).

To meet these requirements, we use a non-standard segmentation approach with a novel variation. Our segmentation process consists of two steps. First, we compute the *over-segmentation* of the image. In this step, our goal is to find suitable objects borders. We tested two approaches for the over-segmentation: the watershed-type segmentation and the graph-based segmentation [Felzenszwalb and Huttenlocher, 2004]. Both approaches are parameter type segmentations and can be set such that different levels of segmentation precision and in-segment covariance can be achieved. During the testing, the graph-based segmentation proved to be more robust, as in some special cases the watershed-type did not approximate object borders correctly (primary when the borders were not sharp enough, or the quality of a picture was low and artifacts were present). However, the graph-based segmentation often over-segmented borders of the objects into multiple layers, which are shown to cause systematic errors in the next step of the method. Therefore we decided to run experiments and evaluate results with watershed-type segmentation.

Figure 4.1: Segmentation examples represented in three steps described in this section. From the left: original image, over-segmentation from the watershed process that use texture as image primitive and the final segmentation after segment merging (segments are represented by a random color). Notice that in the final segmentation, the façades are represented only by one, or few large segments (even the covariance of the original façades varies greatly).

Watershed Segmentation

We use the Meyer Flooding Algorithm to perform the segmentation [Meyer, 1991]. The process is initialized on a grayscale gradient image, provided by a Robert's Edge Detector [Roberts, 1965][4]. The gradient image is modified such that low gradients are truncated to zero by a threshold T_1 to present large areas for flooding. This truncating allows areas with fine texture (such as façades or roofs) to be considered uniform. In

[4] http://www.codeforge.com/article/19293

each zero-gradient area a seed is placed from where a flooding is performed in a recursive process.

Threshold T_2 is set to represent a limit for flooding. In each direction of flooding, the gradients contribute to a height value. If this value reaches the threshold T_2, the flooding in corresponding direction stops. After this process, segments are detected in the image; however some pixels (with high gradient value) might not be included in any segment. Such pixels are subsequently attached to the nearest existing segment. T_1 and T_2 are the parameters of segmentation and represent the final uniformity and size of segments. Threshold T_2 is deliberately set to a low value such that the image is over-segmented (see Figure 4.1). After this step, the borders of small objects are detected often within one pixel precision but large coherent areas of the image (such as ground or façade) are segmented into a large number of regions. This kind of segmentation can also be considered as segmentation into superpixels; however as we aim to examine context between larger areas, we proceed with a merging step.

Segment Merging

In this step, segments which are geometrically close to each other and are visually similar (their color looks similar to a human expert/trainer) are joined into larger regions. Our goal is that final regions approximate real class objects (in this work – cloud, sky, roof, façade, ground, vegetation, grass) as precisely as possible. This is necessary for subsequent context application at the object level. For example, we want to define context of the façade as a location between roof/sky and a ground. This can be achieved best when only one region is located between roof and ground and such region is considered in context evaluation. Detection of such uniform segment is not an easy task for a gradient-based segmentation, as many façades are not uniform, e.g. façades can have slightly different color at ground level, or ledge between levels. Even when the façade is uniform, shadows can provide regions with different color/illumination or occlusion (e.g. wires, lamp posts) can dissect façade's projection into several regions. In these cases the segmentation detects multiple segments in one class object. In this step we want to identify such cases and correct them.

We define "Visual similarity" as a floating point value between 0 and 1 expressing how similar two color values look like (what is their visual difference). This value is considered a probability value, how likely two segments belong to one class object.

Alternatively, visual similarity can be computed in the CIE-Lab color space, as a Euclidean distance of Lab values. However, the implementation revealed that this approach is not suitable in the current application. The main reason is that in CIE-Lab space, hue and saturation have approximately the same weights in computing similarity. In street-side images, most building façades can be distinguished by their hue, but nearly all façades have a rather low saturation. Therefore, to differentiate between two buildings, a large weight must be put on hue, and smaller on saturation. To achieve this, visual similarity is computed through a specific formula in HSV color space:

$$\varphi(C_1, C_2) = |h_1 - h_2| \cdot \min\left(f_1(\max(s_1, s_2)), f_2(avg(b_1, b_2))\right), \tag{4.1}$$

where $C_1 = [h_1, s_1, b_1]$ and $C_2 = [h_2, s_2, b_2]$ are colors in HSV color space and f_1, f_2 are logarithmic functions:

$$f_1(x) = \frac{1}{Z_1}\log(k_1 x + 1), \tag{4.2}$$

$$f_2(x) = \min\left(\frac{1}{Z_2}\log(k_2 x + 1), 1 - \left(\frac{1}{Z_3}\log(k_3 x + 1)\right)\right), \tag{4.3}$$

where Z_1, Z_2, Z_3 are normalizing constants (normalizing f_1 and f_2 into <0,1>).
Similar modifications are used for differences of saturation and brightness. A final visual similarity value is computed as maximum of the differences of hue, saturation and brightness. In this approach, several variable coefficients ($k_1, k_2, k_3, \ldots$) are used (in logarithmic functions). To achieve best results, these coefficients have been optimized in a supervised learning process. Hand-labeled validation dataset (with each façade marked as different object) was used, and for each set of coefficients, segmentation was performed. Coefficients that achieved the best results are subsequently used in segmentation. In this approach, it is not necessary to compute transformations between CIE-Lab and HSV color space and still compute similarity values with modifiable weights on hue, saturation and brightness. The logarithmic functions were chosen to simulate the requirements on HSV parameters, as these

functions can narrowly modify weights when required (close to zero) and still remain nearly constant in higher values.

Merging of segments into regions is an iterative process. In the first step, only segments larger then 0.2% of the image and visually similar are merged into a composite region (more then two segments are allowed to merge in one step). Subsequently, smaller segments are merged into existing regions. Also, visual similarity is computed and required for merging, but the similarity threshold is reduced with each step. The representative color of the region is recomputed after each step. In this approach, it is assumed that large segments are more important for the subsequent classification, as they are usually representing some coherent areas in the image. On the other hand, small segments may represent some small objects, or texture elements. Therefore, large segments have the priority in the merging step, but the requirements for their merging are high. We allow merging segments that are geometrically close to each other, but not necessary connected. In each iteration step a scanning window is used to identify segments that can be merged. The size of the scanning window defines how far away the segments can be from each other and still be merged. Such approach allows final regions to be discontinuous. We use this method to prevent façades segmented into several regions when they are sectioned in projection. This is especially useful in urban areas, where building façades or other logically coherent areas are often dissected by wires, traffic lamps, poles, or other objects in the image. However, this approach also often causes two or more separate façades segmented into one region if their colors are similar. In this stage, it is not considered an error as we have only one class for buildings (thus it does not have significant effect on context application).

This approach for image segmentation has several advantages over the non-parametric methods [Andreetto et al., 2007]. As described before, segmentation can be easily modified by adjusting the coefficients, obtained from ground truth data. By over-segmenting the image in watershed segmentation, most details are preserved, so in the final output, borders of the regions are well-defined (see Figure 4.2).

Figure 4.2: Examples of image segmentation. Each segment is represented by it's color. Borders of segments larger then 10% of the image are marked for better overview. For this reason, smaller segments are displayed without borders and some borders may look incomplete (due to location of small segments at that position).

4.1.2 Segment Classification

In a segmentation step, several large regions are usually detected in the image. These regions are subsequently classified into building façades, sky, cloud, roof, ground, vegetation and grass classes. Regions with intensity <0, 1> lower then 0.1 are marked as dark/unclassified, as in our database they lack any features necessary for the classification (due to camera quality). Such regions are often parts of scenery in strong shadows and it is not possible to classify them based on their visual features. Only regions larger then 1% of the image are classified. As described in the previous section, smaller segments are merged into regions with increased visual similarity tolerance. Therefore, if the region smaller than 1% still exists in the image (was not merged into larger neighboring segment), it is unlikely it belong to one of the major classes. Such small regions are generally some small objects (pedestrian, animal, bicycle, street accessories…) and are marked as unidentified.

In a classification step, we use Discriminative Random Fields to define the probability distribution over classes. In this definition, both visual features and spatial relations (representing context) are applied in one framework. Let us represent the conditional distribution $P(\mathbf{x} \mid \mathbf{y})$ over classes ($\mathbf{x}$ is a vector representing classes and $\mathbf{y}$ are the observations) as a conditional distribution described in Section 3.2

$$P(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A_i(x_i, \mathbf{y}) + \sum_{i \in s} \sum_{j \in N_i} I_{ij}(x_i, x_j, \mathbf{y}) \right), \tag{4.4}$$

where Z is normalizing constant, S is the set of nodes, N_i is the set of neighbors of node $i \in S$. -A_i is the unary potential and -I_{ij} is the pairwise potential.

The unary potential A_i represents the measure of how likely node i belongs to class x_i, given the observation vector $\mathbf{y}$ (and disregarding the neighborhood). In our approach, this potential is directly computed in a visual classification step.

Visual classification is based on a decision tree. We use 30 hand labeled images as a training set, 100 images remain for testing purposes.

In the process of classification, each region is considered a coherent object (only one class can be assigned to each region). Classification is based on color, position in the

image, size and texture (confidence scores are computed for each feature by comparing the feature vectors):

- A single representative color value is computed for a region as an average of color of the pixels inside the region. In a learning process, a color histogram is created for each category. In a classification process, the color of the region is compared with a class histogram.
- The position of the region in an image is represented in a position matrix. The image gets divided into a regular grid; each cell in the grid represents a coefficient in a matrix. It is computed if the region (or part of it) belongs to the cell. The same process is applied during the position classifier training. In the classification step, the position matrix of the region and the position matrix of the class are compared. The score value for position feature is computed as a sum of overlaps over position matrix.
- The texture of the region gets expressed as a histogram of gradient values over the region area. This representation of the texture is sufficient to distinguish smooth regions from textured regions. In the process of image over-segmentation, textured areas get segmented. As described in the previous section, these areas may be subsequently joined, so the insides of the regions may contain high gradient values. Therefore, classes like building roofs, or vegetation areas that contain some texture information relevant for classification, can be recognized thanks to this feature.

In the decision tree used for classification, the last level contains the confidence values for each class computed as a joint probability of the classifiers located in the path from the root to the leaves. These values may be considered as the classification result, but as described in [Hoeim et al., 2005], features presented in this section may not be sufficient to discriminate between all classes. For example, regions of the sky and regions of façade windows (mirrored reflections of sky) can be very similar in color and texture and they may be located in similar positions in the image. Therefore, it is necessary to use some additional constraints in the classification. In the next section, we present spatial rules for verification of the classification.

4.1.3 Spatial rules in classification

Pairwise potentials I_{ij} in (4.4) represents the measure of interaction between two neighboring nodes i and j given the observation vector $\mathbf{y}$. Pairwise potentials are derived from the training set during the learning process. Let us assume, that M is the set of training images, $\mathbf{x}_k$ is the classification of the k-th image and $\mathbf{y}_k$ is the observation of the spatial relation in the k-th image (see Table 4.1). We can represent the set of classified regions neighboring the region i in image $k \in \mathrm{M}$ as $\mathbf{x}_k^{N_i}$. Then the probability that region i in the k-th image is classified into class x_i is $P\left(x_{ik} \mid \mathbf{x}_k^{N_i}, \mathbf{y}_k\right)$. This value can be computed directly from the training set. Inserting this value into equation (4.4) gives us the parameters for pairwise potential I_{ij}, as described in [Kumar and Herbert, 2006]

Real objects in street-side scenes are in specific spatial relations to each other. For example, sky and clouds are usually above the buildings, roofs are above the façades and ground is below the buildings. It is assumed that some of these rules are transferable into digital images as a central projection of the real scene. Using these rules may be valuable as constraints in the classification.

Spatial rules are encoded as a probability of spatial relations between two different classes. To extract the spatial rules that are commonly valid in street-side images, we must have a labeled ground truth database, with all objects classified.

Spatial rules are implemented as a DRF's pairwise potential, representing every region in the image as a graph node. Regions close to one another are neighbors in the graph. In a classical approach, where each node represents a pixel, or a grid element in the image, spatial relations are implicit in the position of such element in the picture [Kumar, 2005]. In our model, regions are not assembled in any predictable fashion and they vary in shape and size. To extract the spatial relation, the graph structure is assembled with the image regions as the nodes (see Figure 4.3).

Figure 4.3: The example of the graph structure placed over the segmented image. Regions are represented as a graph node. An edge is placed between each two neighboring regions. This graph is the basic data structure for DRF and defines node areas as well as their neighborhood relationships. We can observe 19 nodes in the graph, each with 1-8 neighbors. In this example, only areas large than 15% of the image are displayed, as they represent significant objects in the image and are primary contributors for contextual relations.

Edges of the graph are assigned between two neighboring regions. In this approach, only one graph structure exists for each image and the context information from the entire image is considered for the classification of each region. This makes the contextual classification a global process, yet the computation is very time effective. When compared to local methods, we have one graph for one image, instead one graph for each pixel, we compute context from the entire image instead of context from close neighborhoods and we represent context between real classes. However we make a strong assumption that our regions represent real objects. Our experiments will show that this assumption is reasonable.

Region i	Relation	Region j
Bounding box	*Inside* *Enveloping* *partially above* *partially below* *fully above* *fully below*	*Bounding box*
Region centre	*Inside* *Above* *Below*	*Bounding box*
Region centre	*Above* *Below*	*Region centre*

Table 4.1: Spatial relations are described based on relations between bounding boxes and centers of two regions.

In the case of street-side images, mostly vertical spatial relations are relevant. This observation is based on a vertical division of the image, in which the classes relevant for this method are assembled into vertical levels. Other types of relations we examine are *inside/enveloping*, as these are often present in class relations such as cloud-sky, or vegetation-façade. Relations that are examined between the regions are described in Table 4.1.

4.1.4 Evaluation of DRF

We use belief propagation to estimate the conditional probability distribution of DRF. To speed up the process and achieve better results we limit the set of possible classes for each node based on the visual feature. This approach comes from observation, that in certain classes visual cues are too decisive in final labeling. For example, the possible color of the sky is too limiting for a vegetation to be considered into sky class (and vice versa). Therefore, before the evaluation of DRF, we exclude classes with too low a visual score from consideration in specific nodes. This step is also based on the assumption, that given an arbitrary high score from a pairwise potential, such score should not overcome a low unary potential score.

Belief propagation [Pearl, 1982] is computed in an iterative process. In each step, marginal probabilities (beliefs) $P_b(x_i, m_{ji}(x_i, \boldsymbol{y}))$ for each possible label x_i are computed for each node i of the graph. Variable $m_{ji}(x_i, \boldsymbol{y})$ is a message from node j to node i, how likely it is that the label of node i is x_i given the observation in the image $\boldsymbol{y}$. In each step, the class with top score of P_b is selected in each node and the messages $m_{ji}(x_i, \boldsymbol{y})$ directed from that node to neighborhood nodes are computed according to such class. In the initial step, only a visual score contribute to P_b (messages are considered as null). After several iteration steps over all nodes, the set of classes with top P_b in each node stabilize (if not, a threshold for the number of iterations is applied) and winning classes are taken as labels for the nodes. Such solution is implemented in C++ open source library[5].

4.1.5 Results

For testing purposes, 230 images with different weather and lighting conditions were selected from the *General Images* and *Tummelplatz* database. These images contain a large variety of objects from historical buildings, standard city blocks, residential apartments and modern architecture.

Thirty of these 230 images were used in a supervised training process as the hand-labeled ground truth data. Segmentation and classification of an image (640x480) takes approximately 2 seconds on un-optimized single CPU implementation.

As a first experiment, we demonstrate the segmentation performance by using the visual similarity calculation described in Section 4.1.1. To test the performance of only segmentation (without any classification process), 50 testing images were selected and the precision of façade segmenting was tested. In this set of images, each building façade was manually labeled as separate area. The segmentation of images, based on visual similarity and CIE-Lab distance was computed and compared to manual labeling. For each image and each building façade, the area of the façade region extending to other than original coherent area was computed (thus marking the error).

[5] http://cs.ru.nl/~jorism/libDAI/

	façade (%)	roof (%)	ground (%)	sky (%)	vegetation (%)	grass (%)	cloud (%)
clas	89,3	76,5	92,4	97,6	80,4	93,5	57,5
with DRF	93,7	85.2	94,3	98,1	83,7	95,4	62,3

Table 4.2: Results of the classification. In the first row, only visual features were used for the classification. In the second row classification was reinforced by Discriminative Random Fields.

In the case of CIE-Lab distance, this was approximately 5.7% of the region (average of all façade regions in all testing images). In case of visual similarity, this area was reduced to 3.2%. The second experiment tested the performance of a classification. All 200 images were semantically segmented (segmented and classified) and errors were manually marked. In Table 4.2 we can see the correct classification rates for each class in the testing database. The percentage numbers express the value of correctly classified pixels of each class presented in the image. When computing the average over all testing images, contribution of each image was weighted by a size of area covered by a class. Classification is performing worst in the cloud area. This is due to the weak visual differences between the sky and clouds and difficult segmentation of the area. Using DRF for verification provided best results in roof and façade areas, as these have strong contextual relations to other classes, but their appearance-based classification is difficult as there is a large variation in texture and color (see Figure 4.4). For our future studies, the number for façade classification is most important, as this result will be used for the subsequent façade separation method. With the application of context in the form of DRF, the overall precision of façade pixels classification is 93,7%. The errors in classification are observed at the border of the façade, entire façades classified incorrectly and from windows labeled as façades (in ground truth, windows are mostly labeled as unidentified). The improvement in results between classification without context and with context mostly comes from the misclassification of entire façades. In the context-free approach, entire façades can be labeled as different class, when the visual properties are met (e.g. white buildings at the top of photos). This error can be corrected with the application of context (when

such white building has a roof, or a window, the label is changed from “cloud” to “façade”).

In the final labeling, windows are often labeled as either unidentified or façade. In the ground truth we label windows as unidentified, unless they are too small (see Figure 4.4). If the method labels such window as façade an error is produced by such labeling. In a Figure 4.4, it can be observed that most errors in a façade labeling are produced by mislabeled windows (for example, façades in third row have most windows labeled as façade class). However in our subsequent methods of façade separation, we consider both façade and unidentified classes to be parts of potential separate façades, so this kind of error is largely neglected. Because of this, the result for façade classification can be considered to have a higher value that given in Table 4.2 in some specific applications.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

Figure 4.4: Examples of image context. Each class is marked in different color. Dark green (building façade), brown (roof), gray (ground), green (vegetation), blue (sky), light green (grass), dark gray (shadow), black (unclassified). We can see in these examples that most problems are in false positives for the vegetation class (mostly façade areas labeled as vegetation due to similar color or low illumination) – images (b), (c), (j), (k) and false negatives for cloud class (clouds labeled as sky) – images (a), (d), (k). Unidentified areas are mostly small, visually distinctive regions, like windows, or pedestrians – images (a), (c), (d), (g). In several images, a building color is closely similar to a color of the sky/cloud – images (d), (i), (j), or to a color of ground – images (e), (f); however the building class was labeled correctly due to an involvement of context relations between classes. Notice a variety of architectural styles and scene compositions.

4.2 Multi-view Streetside Scenario

We examine the effect of multi-views in the semantic segmentation method. The task is to present experiments on how the semantic segmentation performs in different images that can be matched to one another [Recky and Leberl, 2010]. As the visual information generally differs only slightly in matching images, this experiment shows how the different context present in images has influence on a classification. For this purpose we do not modify the classification process for multi-view. Instead we run classification (semantic segmentation) for each image in the stack and decide the global classification of the scene based on the partial classifications from images. We apply the method in two datasets. First dataset is the *Industrial System*. In this section, we used the input of only two cameras – one sideways and one frontal-sideways tilted camera. The data base supports investigations into the issues of the types of redundancies, namely multiple images, all taken with parallel optical axes from different camera positions; or multiple images all taken from a single position but with different directions for the optical axes, and various hybrids between these two concepts.

Our second dataset is the *Tummelplatz* set. In this data set we also have sufficient images to be able to group them by similarity of their optical axes by dissimilarity due to differences in position and orientation of the optical axes, and by geometric

resolution. The object range of these two datasets does not permit us to study the results as a function of various object types. For this to be possible, we need to increase the variation of objects and scenes being studied.

4.2.1 Multi-Image Semantic Segmentation

In the general case of urban imaging, a block of images would be triangulated in today's typical workflows as illustrated by Photo-tourism and Photosynth [Kumar, 2005]. We also employed this approach and created a sparse 3D point cloud from the subset of the *Thummelplatz* dataset. The algorithm described in [Irschara et al., 2007] was used to extract the point cloud. However in our case, we used a calibrated camera in the process. This allows us to work with more precise and reliable data for image matching. When working with crowd sourced datasets, one must account for errors caused by lack of calibration data. As our goal in this section is to match the building façades between two images pixel-by-pixel, a sparse point cloud does not provide us with enough data for this. It is necessary to develop a dense point cloud. If we can assume that the sparse point cloud do describe the 3D shape with sufficient detail, we ca interpolate the positions of pixels between the points belonging to the sparse point cloud inside the façade. We can operate with a simple assumption, that the area between two façade points is planar. In perspective imaging, a planar object is mapped into the image plane by a projective transformation. Establishing the parameters of this transformation can provide image matching even for pixels not belonging to a point cloud. We merely need to identify at least 4 façade points in each image. We can use four non-collinear points from the point cloud, or when the point cloud is not present, we can mark these points manually and assign world coordinates to them. The perspective transformation matrix can be defined uniquely if image and object coordinates of at least four points are measured. The relation between the point in the image plane $\mathbf{x}$ and the point in the world plane $\mathbf{x'}$ can be defined as $\mathbf{x'} = H\mathbf{x}$, where H is the projective transformation matrix. The parameters of matrix H can be computed from 4 corresponding point coordinate sets, or alternately can be derived from the certain metric properties such as length ratios and angles, as described in the work of [Liebowitz and Zisserman, 1998].

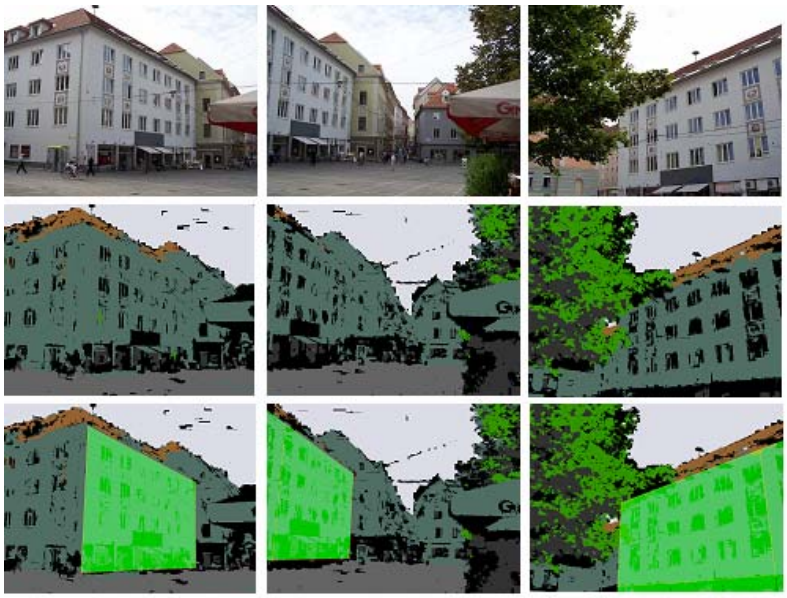


Figure 4.5: Image segmentation and matching. Top row – original images of the same objects from different view point. Middle row – semantic segmentation of the individual images. Dark green – façade, brown – roof, light green – vegetation, white – clouds, gray – ground. Bottom row – building façade is marked and matched.

For the purpose of testing the segmentation results, the borders and the inner area of the building façade were manually labeled. By associating with each façade in object space a unique identifier (number), we can automate the matching task for each group of images of the same façade (see Figure 4.5). In an Industrial System database, we used the laser scanner data in similar way as a point cloud. Given that each image is geo-tagged, the position of a laser scanner point on the building façade in the world coordinate system can be projected back into each image. This will provide us with image and object coordinates of a sufficient number of object points so that the image-object relationship is fully defined. We use this simple method to relate the overlaps of the images to one-another and to then study the differences in the segmentation from image to image in the overlaps.

Figure 4.6: The segmentation of three different views of the same object. Each of the segmentations show errors (in a red box) different from the others.

4.2.2 Simple application of multiple views

For the purpose of testing the multiple-view image interpretation, we use an annotated ground truth. This allows us to identify façades and the correspondences between the planar objects in the images. For each façade, the perspective distortion is computed from the combinations of four non-collinear points from a point cloud (if present) or from points manually marked.

The position in world coordinates for each point of the façade is computed through interpolation between three closest point cloud points. The identification number for each façade helps in automating the work with multiple images. We also identify objects that generate occlusions such as vegetation, pedestrians or cars. This type of annotation can provide us with pixel-by-pixel correspondences between the images.

For each image, a semantic segmentation is being performed in accordance with Section 4.1.1. Our task in this experiment is to assess how the results of the façade segmentation differ between images, and the identical object areas do get defined by means of image matching as previously discussed.

The framework can be described in the following steps:

1. For each façade object, identify the group I of images, where it is located and annotated.
2. Compute the perspective transformation matrix H_i for each image $i \in I$
3. For each point $x_{ij} \in F_{ij}$, where F is the façade in the image i with the identification number j, transform $x_{i,j}$ into the world plane coordinates $x'_{i,j} = H_i x_{i,j}$
4. $\forall x_{ij}$ compute the new classification as $s_{ij} = \frac{1}{Z} \sum_{i \in I} w(x_{ij}).c(x_{ij})$

   where $c()$ is the classification of façade pixel x_j as façade in image I and $w()$ is the weight function. Z is the normalizing factor, setting s_{ij} into <0,1> interval
5. Compute the new classification as a result of s_{ij} for each pixel of the façade.

We designed several scenarios according to the concept of redundancy described in Section 3.3. Three scenarios are identified for the *Tummelplatz* database as follows:

a) single position, rotated optical axes (SPRA)
b) varying position, parallel optical axes (VPPA)
c) varying position, varying axes (VPVA)

The industrial system (IS) is considered as a separate case with a varying position, parallel optical axes and a high level of redundancy.

For a testing purpose, we used 5 façades (3 hand-labeled and 2 with automatic matching) façades from *Thummelplatz* dataset, each on approx. 30-60 images (253 image total) for SPRA, VPPA and VPVA scenarios and 4 façades from *Industrial System* dataset for IS scenario. In our first experiment, the weights $w(x_{ij})$ are set to 1 for each image. This approach was chosen to demonstrate that even the simple summing of classification through all images can provide improved results over a single image. Pixel x_{ij} is then classified as a façade, if $s_{ij} > 0.5$ (see Figure 4.6).

Results from this experiment are summarized in Table 4.3. For each of the 4 overlap cases, we produce three numbers. "# img" is the number of images used in the scenario; "Single img" is an average result of classification for each image in the scenario separately. This number is expressed in a percentage of all façade pixels that were correctly classified as a façade class. The row "Multi img" is the result of multiple view approach, as described in this section.

Scen.	SPRA	VPPV	VPVA	IS
# img	24	22	55	250
# img/obj	8	11	6	27
Single img (%)	93.9	94.2	93.4	89.2
Multi img (%)	96.2	96.9	95.7	93.3

Table 4.3: Area labeling in different scenarios. "# img" is the number of images; "# img/obj" is the average number of views of a given object point. "Single img" is the average value of correct classification of pixels into a "façade" class in single image approach (in percentage); "Multi img" is the value of correct pixel classification in multiple views approach (in percentage).

From the results of this experiment we can observe that the improvement in a multiple views approach can be achieved in all examined scenarios. The single image approach has the highest error rate in the industrial system dataset. This is probably due to lower quality of the images (lower resolution and lens quality). But the improvement in multiple views approach is also higher in this scenario. It is assumed, that the high level of redundancy may be the contributing factor in this result. It is therefore assumed that this scenario can benefit the most from the redundancy in a dataset.

4.2.3 Classification consistency as a function of distance from a camera

A second experiment examines the effect of redundancy in regard to the distance of the objects from the camera. It is assumed that distant objects are more difficult to classify, as they contain larger pixels and thus less information about the object, but the relationship between the distance and the classification result is unclear. In this scenario, we use the industrial system database with laser range data to classify and match objects. We are comparing the classification of areas of building façades at different distances from the camera. The area of the façade is considered consistently classified if it is labeled as a façade, or unidentified.

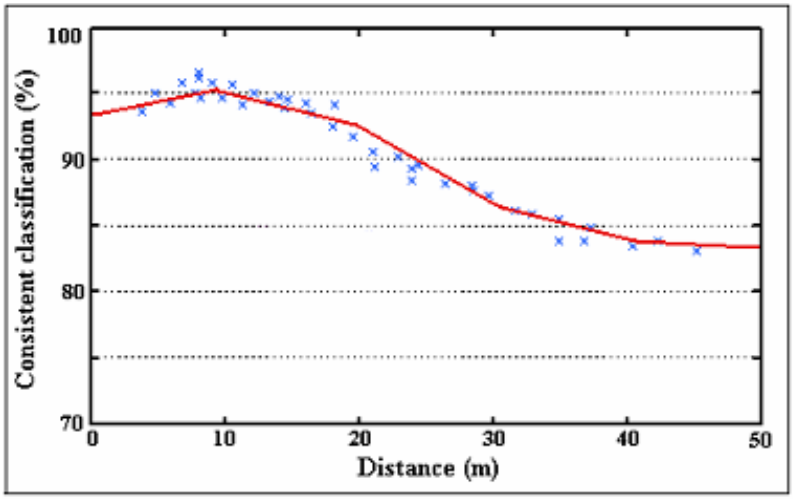


Figure 4.7: Relationship between the distance from the camera and the consistency of a façade classification. Values are plotted in blue for objects at various distances from the camera; the red line is an average value.

The results can be read from Figure 4.7. We see that at a distance of 10 meters, 95% of the façade pixels are consistently being classified as "façade" or unidentified. Going further way to 40 m, this reduces to a level of 84%.
This result can be used in the further experiments, to derive a distance dependent weight for the classification in image overlaps or redundant databases. The classification of an object closer to the camera is at a higher confidence than that of an object that is further away.

4.2.4 Multi-view classification based on distance

In this experiment, we apply our previously extracted function of distance based classification consistency to improve the algorithm described in Section 4.2.2. We set the weight function $w(x_{ij})$ as a function of distance from a camera. This will provide the weighting for each pixel, when the distance information is available. We used a subset of Tummelplatz dataset, for which the sparse 3D point cloud is available and the Industrial System dataset with laser range data for this experiment. The results can be observed in Table 4.4.

Scen.	VPVA	IS
# img	28	250
Single img (%)	93.5	89.2
Multi img (%)	96.1	95.7

Table 4.4: Area labeling based on distance. "# img" is the number of images; "Single img" is the average value of correct classification of pixels in single image approach (in percentage); "Multi img" is the value of correct pixel classification in multiple views approach using the distance as a weight function (in percentage).

In this experiment, we can conclude that selecting a more appropriate weight function $w(x_{ij})$ for the classification in multiple views scenario can add some improvement. The selection of weight function is dependent on additional data, in this case, the presence of distance information. This will need to use the calibration of the camera (preprocessed or automatic), or some other source of data (laser scanner, for example).

4.3 Discussion on Semantic Segmentation

In the previous section, we introduced a semantic segmentation method for streetside images and extended it into a multi-view scenario. Our method is applied to detect only non-terminal symbols in the image, limiting the class set to {façade, roof, ground, vegetation, grass area, sky, cloud, shadow, unidentified}. Terminal symbols are labeled as part of the non-terminal class, or are left unidentified. This approach is based on our hierarchical framework for streetside processing; however additional classes of terminal objects can be added directly into the set of classes if necessary. For example, adding the "window", or "cross-walk" class into the process would provide relevant results and the detection would benefit from the context of such objects.
For further processing, our primary class of interest is the "façade" class. In a single-view scenario, we achieved 93,7% pixel-wise precision of labeling for this class. This precision is further increased up to 96,1% in a multi-view scenario. In a single-view, the segmentation process (defining the borders of segments) does not depend on the context, thus the increase in precision between non-context approach and the context application comes from the changes of entire segments labeling. In this way, the

context helps primarily in systematic error cases, where entire façade (or large portions of the façade) are mislabeled (usually into ground, vegetation, or roof class). This could happen when the visual features are not discriminatory enough to distinguish between such classes. The application of context helps in these cases. This is in contrast with a local context application, where the correction of error is present mostly at the borders between objects, but it cannot change entire object mislabeling. In a single-view scenario, our method is more reliable in labeling entire objects (façades, roofs…), but the object borders are detected only in non-context segmentation. Therefore, if one only considers a single-image scenario, the additional application of local context, for example in a form of MRF in semantically segmented images would increase labeling precision further. The same effect is however achieved by the application of multi-view, as the borders of objects can be detected more reliably in corresponding images. In addition to this, the multi-view scenario can provide further correction of mislabeling at the object level.

Another principal difference between our approach and a local context method is the computational complexity. In a local pixel-wise method, a random field is applied for each pixel and in a super-pixel method, for several sub-areas of the image. In our approach, only one DRF is computed for the entire image. The number of nodes of such DRF is usually in the order of ten and the number of neighbors for each node is usually up to ten. There are cliques in the DRF graph in general, so the evaluation methods usually lead to an approximation of solutions. Another method how to approximate a solution is the application of brute force for evaluation with some heuristics. In such approach, we can vary labeling in each node according to top values from visual classifiers and compute a set of hypotheses for image classification. Subsequently, we compute the conditional distribution of DRF for each hypothesis, given the fixed label in each node. This provides a very fast solution, but the optimality is hard to assess. The success of such approach is based on the assumptions that many classes are visually very different (for example, the colors of sky, vegetation and ground are exclusive), therefore, hypotheses provided from visual classifiers are very stable in many areas of the image.

We can observe in a multi-view scenario, that the dataset based on the “crowd sourced” paradigm (CS) achieved better results than the industrial system dataset (IS). The possible explanation for this is that in the CS dataset, images are not organized in

a random fashion, but according to the expectation of the human agent. The users, when taking pictures were intentionally selecting a good viewing position, avoiding obstacles and were focusing on the details of objects. This results in a better "quality" of projected scenes. On the contrary, the IS was designed to provide results in general situations and was not intended to cope with specific types of scenes. Images for IS do not always provide a best viewpoint and as the camera positions are not very varying, the visual and contextual redundancy level in a multi-view is high.

We described the application of the method in a multi-view scenario, where each image was interpreted separately and the interpretation was finally refined through image matching. The selection of this method was based on the fact that only a sparse point cloud was available in our dataset. Observed precision of matching was ±2 pixels. In case, where a more precise, dense point cloud is available another approach can be chosen. For each pixel in the image, we can look for corresponding pixels and transfer visual information from them. This will provide a high dimensional feature vector in each pixel, which provides superior visual cues for segmentation and classification. However in this case, the geometry of the image is not influenced.

We also described two methods of merging interpreted information for the images. First is the simple application of multi-view, when all interpreted information from each image has the same weight, second is when the information is weighted based on the distance from camera. Other possible weighting factors may include the quality of photos in the stack (especially useful for crowd sourced dataset from multiple users), geometry of the scene (inclination of planes, presence of occlusions), or variation of classes in the projected scene (when less classes are present, the probability of mislabeling is decreased).

Chapter 5

Façade Separation

5.1 Separation of Façades in Single Images

In previous sections, we introduced a method to detect major non-terminal symbols in a street-side image. This represents the basis for image understanding. However, in the semantic segmentation, all building façades are considered to belong into one class. Many applications (e.g. 3D building modeling methods) require them to be labeled as separate objects. In general, street-side images present complex urban scenes and a building landscape with many façades. In this section, we introduce a robust approach to solve this problem and thereby increasing the likelihood of success of 3D modeling from street-side images [Recky et al., 2011]. In the most basic case, we consider a single street-side image as an input. First, we label principal areas into classes (sky, vegetation, ground, buildings). Subsequently, we proceed with the segmentation of building fronts into separate façades based on the detection of repetitive patterns built by windows and architectural styles. This segmentation subsequently can be used to interpret the details in each façade. We present following algorithm:

Algorithm 5.1

Input: Streetside images and their semantic segmentations

1. Detect Repetitive Patterns in areas labeled as "building façade" class
2. Find separation areas in Repetitive Patterns
3. Label all segments between each two separation areas as unique façade
4. if (image matching is available) match façades between images and merge results

Output: Separate façades identified as areas in images

We present following innovations over the previous state-of-the-art work:

- Semantic context is used to limit repetitive patterns only to areas where it is relevant
- Results are enhanced in a multi-view scenario by labeling from matched images

Our approach is based on a convolution of two methods. First we apply the semantic segmentation method described in Chapter 4 to extract contextual information from the image. We use this information as prior knowledge for façade segmentation [Wendel et al., 2010]. We extend our method into a multi-view scenario, where a redundant dataset with overlapping street-level images exists. For this application, we use the *Industrial System* dataset, as it provides a large variety of different façades in each image. We select a challenging subset of images taken from a frontal-sideway camera. In this subset, most façades are under perspective distortion, testing the limits of detecting repeated patterns and of segmentation. The dataset is complemented with laser scanner data to help in matching overlapping images within pixel accuracy. We thus combine results from a previously described façade segmentation with a context-based image segmentation and achieve an improvement of the segmentation towards 97% effectiveness.

5.1.1 Detection of Repetitive Patterns

We build on the method of [Wendel et al., 2010] for finding repetitive patterns within a single image. First, the Harris corners [Harris and Stephens, 1988]

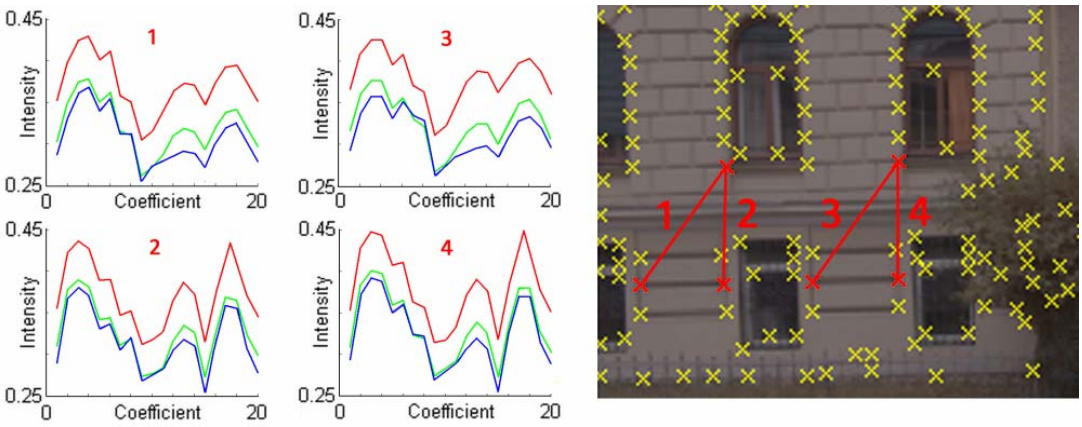
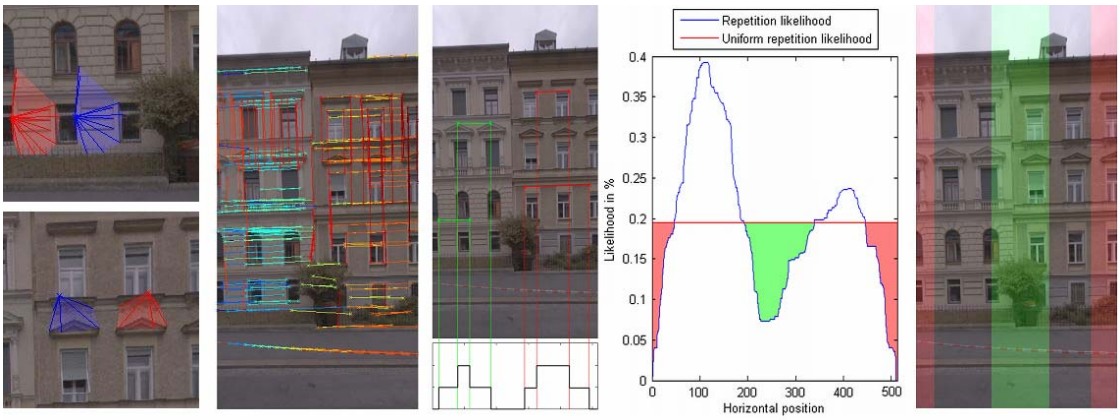


Figure 5.1: We describe the image content between Harris corners by extracting intensity profiles with 20 values for every RGB color channel [Wendel et al., 2010].

are detected as interest points. Subsequently, color intensity profiles are extracted on straight lines between each of them. The resulting complete graph structure has interest points in nodes and each edge corresponds to a color profile. We limit the complexity of the graph by connecting only the 30 closest neighbors to each node. The color profiles are constructed as 60-dimensional normalized descriptor: for every color channel of RGB, we compute a 20-dimensional descriptor sampled by linear interpolation along the line. In this approach, the scale invariance is achieved by setting the number of coefficients for interpolation as a constant.

For an example of intensity profiles, see Figure 5.1. For matching of the descriptors, we use a kd-tree method [Friedman et al., 1977]. The tolerated threshold was set to 5% deviation off the descriptor prototype for finding repetitive patterns. Matches with more than 10 descriptors involved are ignored, as they showed to be insufficiently discriminative. To achieve the required robustness, an additional voting step has been included in the process. All matching profiles vote for their corresponding interest points. The vote is counted only if the descriptor has not contributed to a correspondence so far. This method is described in [Tell and Carlsson, 2000], [Tell and Carlsson, 2002] and removes the bias in a voting matrix.

We finally locate the repetitive patterns in the voting matrix by thresholding the votes a correspondence received. Interest point correspondences are established if at least 3 of 30 intensity profiles were matched. An example for arbitrarily shaped repetitive image areas obtained by this voting process can be observed in Figure 5.2(a).



(a) **(b)** **(c)** **(d)** **(e)**

Figure 5.2: From streetside data to separation (best viewed in color): (a) Matching of arbitrary areas (b) Detected repetitive patterns (color-coded lines) (c) Projection results in a match count along the horizontal axis (d) Thresholding the repetition likelihood with the uniform repetition likelihood (e) Resulting repetitive areas, separation areas (green), and unknown areas (red) [Wendel et al., 2010].

We eliminate outliers by the assumption that repetitive patterns are not likely to occur across an entire image, and are not likely to be very small. Therefore, we restrict the horizontal and vertical distances of accepted matches. The result of this approach can be observed in Figure 5.2(b) as color-coded lines.

5.1.2 Façade Segmentation

Several methods for processing a single façade have been introduced recently [Simon et al., 2011], [eTRIMS, 2012]. However, façade separation or segmentation itself has not received much attention. P. Müller introduced an algorithm which is able to summarize redundant parts of a façade and thus subdivide images into floors and tiles [Müller et al., 2006]. A major limitation is the dependency on single façade images, and automatic processing fails for scenarios with blurry texture, low contrast, chaotic ground floors, and occlusions caused by vegetation. Other works on façade separation [Hernandez and Marcotegui, 2009], [Xiao et al., 2009] are based on the evaluation of directional gradients, which only works for highly regular façades.

The previous section summarized a method for detecting areas of repetitive patterns. Due to the natural setting of objects in street-side images, we assume that the repetitive patterns are located along the horizontal direction and separations between façades occur in a vertical direction. Subsequently, we project lines between matched interest points into the horizontal axis, constructing the histogram of match counts (see Figure 5.2(c)). We compute the repetition likelihood as a percentage of all matches in a given interval of the histogram.

The next step is the detection of separation "areas", as an extended interval between repetitive areas (marking the positions where one façade ends and another starts). Computing the separation areas from minima on the repetition likelihood is not sufficient, as the global minimum does not account for images with more than two connected façades and local minima can be detected in common false positive cases, such as between rows of windows. If all parts of the façade contribute equally, we would get uniform repetition likelihood. Setting this value as a threshold, areas with low likelihood are defined as separation areas and areas with higher likelihood as repetitive areas (see Figure 5.2(d)).

To cope with narrow fields of view, where the location of repetitive areas is not detected, we define the areas on borders of an image as "unknown". These areas start at the image boundary and end at the first repetitive area. An example of repetitive, separation and unknown areas can be seen in Figure 5.2(e).

5.1.3 Façade Identification

In this section, we enhance the previously presented method of façade separation using prior knowledge. Given the semantic segmentation described in Chapter 4, we consider all areas in the image labeled as building or unidentified to potentially be part of repetitive areas, defining a separate façade. Subsequently, we apply the repetitive pattern detection only to those pixels. The resulting set of Harris corners gives us a better basis for interest point matching (see Figure 5.3).

Each repetitive pattern area identifies a unique façade. To detect the entire façade area, we use the results of the segmentation described in Section 4.1.

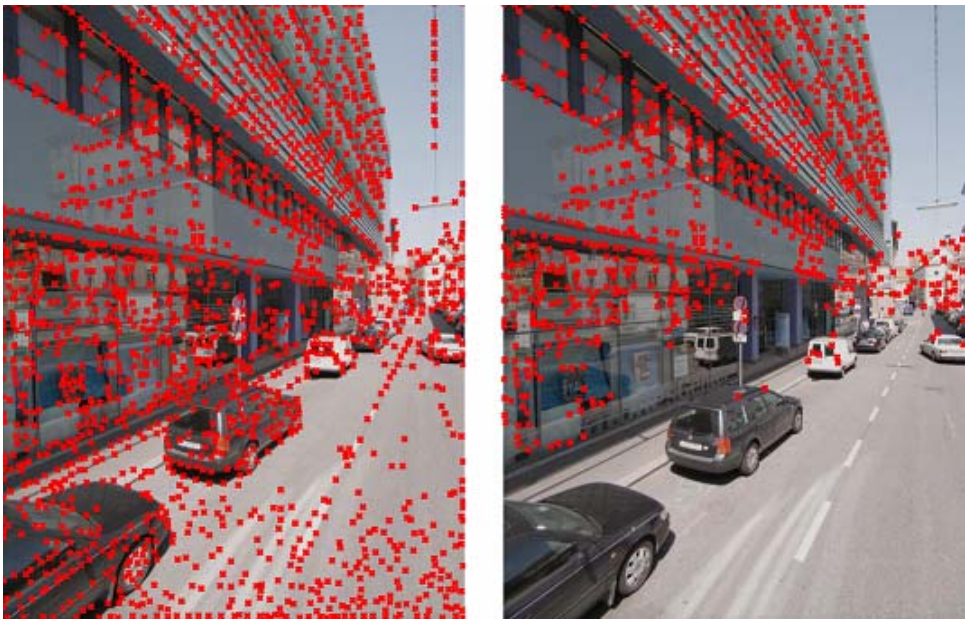
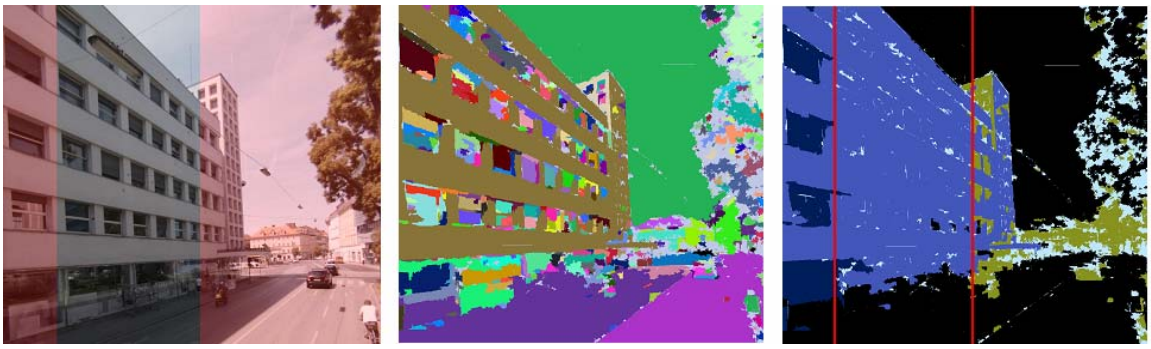


Figure 5.3: Set of Harris corners without prior knowledge of façade class (left) and with prior knowledge from semantic segmentation (right).

As described, the segmentation was designed to detect natural objects in the scene such as a building façade. Therefore, façades are usually represented as one or a small number of large segments, labeled into a façade class.
Subsequently, we compute the ratio of pixels of segments which belong into a repetitive pattern area to the number of pixels outside of that area. If the ratio is larger than one, segments are labeled as separate façade (see Figure 5.4). It is a matter of definition if objects like windows, doors or shop signs are considered part of the façade or separate objects. In the semantic segmentation, these objects are segmented separately and as there is no class for them, they are usually labeled as unidentified. However, in our ground truth, only the border of a façade is labeled, so these objects are included in the definition of a façade. To cope with this problem, we consider all unidentified segments as façade segments and we include them in the evaluation by repetitive areas. This solution sometimes gives unwanted results, as other objects, like cars, or pedestrians originally labeled “unidentified” are often included into façades.



(a) **(b)** **(c)**

Figure 5.4: Façade identification in a single image: (a) Original image with repetitive area (b) Segmentation of the image (notice the façade segmented as one, brown segment) (c) Identified façade (original repetitive area is marked by red lines).

5.2 Multi-View Scenario

In this section, we examine a multi-view scenario for façade separation. We use the *Industrial System* dataset of 250 images from each camera on the car platform complemented with laser scanner data. In our work, we use the input of only one, frontal-sideways tilted camera. The data base supports investigations into the issues of the types of redundancies, namely multiple images, all taken with parallel optical axes from different camera positions [Recky and Leberl, 2010].

In the dataset, we have identified 9 separate building façades, each located in approximately 20-50 images. 7 of the façades are under severe perspective distortion, which present the worst case scenario for a repeated pattern algorithm (as the repetition is less evident in the perspective) and provides a challenge for segmentation.

5.2.1 Image Matching

Matching the images of the stack is based on LiDAR laser scanner data, provided as a supplement to the Industrial System dataset. In each position of a car, a set of LiDAR points has been measured. Given the geo-position of the camera system, a set of global 3D points of the scene is available from the LiDAR in the same coordinate system. The projection of these 3D points into individual images gives us an instrument for

superimposing the separate images. In this case, the set of LiDAR points can be considered an equivalent to an image-based sparse 3D point cloud.
As the sparse distribution of points does not provide us with enough data to match pixel-by pixel, we proceed with matching segments. The precision of matching depends on the number of LiDAR points located in the segment. In the worst case scenario, there are no LiDAR points that can be projected into a segment. In this case, we use the assumption of planarity of the façade. As most of the façades are nearly planar objects (or are mostly composed of planar regions), we consider the interpolation between points to provide a valid approximation with sufficient accuracy. In the case of no LiDAR points projected into a segment, we find the two closest points and create a new point by interpolation. Considering the planarity, the corresponding interpolation between 3D coordinates of such points will provide the means of matching. This solution is necessary only for small segments, as major ones have usually a large number of LiDAR points projected into them.

5.2.2 Labeling of Segments in a Multi-View Scenario

Given the set of LiDAR points projected into a segment, we have the means of matching a segment to the points in other images. When the façade identification has been performed in these images, we can gather data from multiple images about the labeling of segments as part of separate façades. Subsequently, we can make the decision for the segment based on the following criteria:

- If the segment was classified as a *building* in the semantic segmentation and was labeled as a part of separate façade *in at least one image*, we consider the segment to be part of that façade.

- If the segment was classified as *unidentified* in the semantic segmentation and was labeled as part of a separate façade *in the majority of images*, we consider it to be a part of that façade. Segment classification can then be projected from corresponding images.

This distinction between façade class and unidentified class removes most of the problems with the labeling of unidentified objects like cars or pedestrians as façades, but still keeps windows and doors as parts of a façade.

5.2.3 Results

Our evaluation procedure is similar to that of Wendel et al. [Wendel et al., 2010] to ensure compatibility. We use precision and recall [Rijsbergen, 1979] to evaluate our algorithm and combine these by a harmonic mean to obtain the measure of effectiveness, also called F_1-measure. We obtain ground truth by manually labeling individual façades. We estimate the point matching quality by clustering the matches and assigning them to the ground truth, resulting in a set of inliers and outliers for every segment. We only use the match precision (PRmatch), as it is not possible to estimate the amount of false negatives required to compute the recall. Façade separation quality (Separation F_1) is estimated by checking if the detected repetitive area lies within the ground truth segment. More or fewer splits lower the effectiveness except when they occur in an unknown area. The façade segmentation quality (Segmentation F_1) is estimated using a pixel-wise comparison between the automatically obtained segment and the ground-truth segment. Our approach depends on the parameters of the Harris corner detector: $\sigma_D = 0.7$, $\sigma = 3.0$. All other parameters are defined with respect to the image scale.

In our first experiment, we compare the separation precision using the semantic segmentation as a prior knowledge to the original algorithm proposed by Wendel et al. [Wendel et al., 2010] (see Table 5.1).

	Original	Mask
Matching precision (%)	43,7	45,5
Separation precision (%)	83,6	96,1
Separation recall (%)	75,3	85,3
Separation F_1 (%)	79,2	90.4

Table 5.1: Separation result without (Original) and with the prior knowledge from semantic segmentation (Mask).

In the next experiment, we compare the results of façade segmentation. We tested three different approaches, namely the segmentation method of Wendel et al. as proposed in [Wendel et al., 2010] without prior knowledge, Wendel's segmentation with the prior knowledge, and finally the approach described in this section – repetitive areas acquired with prior knowledge applied on context based segmentation (see Table 5.2).
In these results, we can observe a significant improvement with the application of prior knowledge (more than 15%). The segmentation with prior knowledge (Single) has performed less effectively in segmentation precision, but outperformed the original approach in recall, resulting in better overall performance (by 3%).

	Original	Mask	Single
Segmentation precision (%)	53,2	91,3	90,2
Segmentation recall (%)	79,7	69,5	74,6
Segmentation F_1 (%)	63,8	78,9	81,7

Table 5.2: Segmentation results for three different approaches. Wendel's segmentation without prior knowledge (Original), with prior knowledge (Mask) and the segmentation described in this section (Single).

Our final experiment is focused on the difference between the single-view and multi-view scenarios. In both approaches, we use the prior knowledge from semantic segmentation to extract repetitive areas. The segmentation method for both scenarios is the one described in this section. In this test, we also examine the effect of segmentation post-processing, using morphological operators. As there were some gaps within the segmentation due to patches not completely merged into the segments, or mislabeled areas of the façade, we applied the morphological closing and opening to close the gaps and avoid false negatives (see Table 5.3). Results show that given the laser scanner 3D point cloud, or another robust image matching method, the transition from single-view to multi-view can improve the output by approximately 15%. Also, with the segmentation described in this section, we can achieve better results with larger kernels of morphological operators.

	Single (small)	Single (large)	Multi (small)	Multi (large)
Segmentation precision (%)	92,5	90,2	96,9	97,1
Segmentation recall (%)	68,2	74,6	92,3	96,2
Segmentation F_1 (%)	78,5	81,7	94,6	96,6

Table 5.3: Segmentation result in the single-view and the multi-view scenario. In both scenarios, we tested the effect of morphological operators with kernel diameter of 3 (small) and kernel diameter of 30 (large).

An example for the separation and segmentation of façades is presented in Figure 5.5 for a single-view scenario and in Figure 5.6 for a multi-view scenario (both with large kernels). The visual comparison shows that the multi-view scenario does not only provide more complete façade segments in terms of coverage, it also enables our algorithm to detect more separate façades in a single image. For better visualization, videos are provided online.

Figure 5.5: Separation and segmentation results for different façades in a *single-view scenario* (large kernel setting). While separate façades are found in general, the algorithm in the single-view scenario does not always provide complete façade segments and is not able to detect as many separate façades as in the multi-view scenario.

Figure 5.6: Separation and segmentation results for different façades in a *multi-view scenario* (large kernel setting). Even for façades under perspective distortion, the multi-view scenario provides robust results.

5.3 Discussion on Façade Separation

The necessity to have a single façade representation is essential for further work on streetside image analysis. In many previous works on façade analysis/processing, this problem was either unaddressed (input was provided manually), or some additional data was introduced as prior knowledge [Werner and Zisserman, 2002][Šochman, 2006][Hohmann et al., 2008]. As an example, [Lee and Nevatia, 2004] proposes the use of a wire-frame model obtained from the aerial view to project façades from the streetside image into planes of such model. This is applicable for a single building, but when the buildings (façades) are connected with each other, a wire-frame model cannot distinguish between such façades. We argue that enough information to separate façades properly is directly observable in streetside images themselves. This information must be obtained from visual cues, as geometric features of two different façades are usually not too discriminative.

For testing purposes, we select a left-forward view oriented subset of the *Industrial System* dataset. This decision was made for two reasons:

- Most façades in the subset are under strong perspective distortion. These provide a challenge for both the segmentation and detection of repetitive patterns methods. Results from the subset thus provide a better understanding about the robustness of the algorithms. As it can be observed in several examples from the test set, less perspective distortion would cause fewer errors in both methods. Therefore, we can expect even better results in easier datasets, where façades are projected from more perpendicular angle.
- As the selection of a challenging dataset caused problems in the original repetitive pattern detection algorithm, the introduction of semantic context as a-priori knowledge and the translation of the process into multi-views demonstrated greater success in error correction. Processing of the difficult dataset cases benefit mostly from the workflow described in this section. Based on this observation, we can assume that the benefit of context and multi-view is best demonstrated in hard cases rather than in systematic improvement of original algorithm.

The output of this method is in a form of semantic labeling of separate façades in the image. The shape of the façade's borders is arbitrary; however for the future processing, we still make the assumption of the façade's planarity. In a dataset, where the profiles of the façades are expected to be highly non-planar, we recommend the involvement of 3D information to better approximate actual façade form. As argued before, geometric information is not enough to separate façades (for example of two buildings that are connected), so it has to be combined with visual information, such as with the result of our method. In our framework, we assume the planarity, even in the presence of balconies, pillars or various reliefs as such objects do not represent significant deviation from planarity assumption (as defined in section 1.2.1).
The method of repetitive pattern detection was designed to separate (or identify) façades that spot some repetitive pattern on the surface (usually in a form of windows). If a façade does not contain such pattern, the method would eventually fail. However, in our approach, we can still use semantic segmentation to identify the façade. As we consider "separator lines" to be indicators of façade split, the absence of such indicators will keep façade detected through semantic segmentation intact. This is usually a correct action, as if two connected façades without repetitive patterns would be present in the image, there are no general cues to distinguish between them.
Transition of the method into a multi-view scenario registered significant improvement in results. This improvement is evident mainly in two cases:

- Many images in our dataset contain façades that are too far away, thus are too small and distorted for a repetitive pattern algorithm to detect them. However as the camera system gets closer, such façade is projected visually better and it is detected in another subset of the dataset. This information is transferred into images, where façade is not detected and they score better in test scenario.
- Repetitive patterns are usually located in the middle section of the façade, thus the separator lines do not approximate façade borders well. This is corrected by the segmentation, as we label façades based on segments marked by separator lines. However, the segmentation itself can not often approximate façade borders correctly, especially when two neighboring façades have similar visual features. In a multi-view scenario, errors in approximation of borders are often corrected, when results from multiple images are merged.

From these observations, we can conclude that the multi-view scenario helps detect façades in more images and better approximates vertical façade borders as displayed in Table 5.3. Façades still do not get identified when the façade area is very small/thin (façades with length of less then 50-70 pixels). This is usually the case for façades trimmed by image borders, façades far away from the camera or under a strong perspective distortion (angle of less than 25 degrees between a façade and a camera axis). Such façade can be labeled correctly in the segmentation, but the façade's borders are not identified, as the repetitive pattern algorithm does not provide the response. This is not considered a significant error in our workflow, as such façade would not provide relevant visual cues for a further processing.

Chapter 6

Window Detection in Complex Façades

6.1 Window Detection in Single Façade

With the semantic segmentation method introduced in previous sections, we have general surfaces in the image identified. Subsequently, we can focus on more specific objects of the urban scenes. With the introduction of the façade separation method, we can now consider building façades to be separate objects and can proceed with identification of terminal symbols inside a façade. In this section we introduce a method based on a gradient projection approach to segment a façade area into blocks. We label each block into window and non-window classes and examine how the performance of the method changes in a transition from single to multi-view [Recky and Leberl, 2010(II)]. We selected façades as a representative of non-terminal symbols because we have the façade borders well defined in an automated process and windows as a representative of terminal symbols because they are usually major objects located on each façade. In a subsequent section we introduce a multi-view context into the process and extend a class set.
We present following workflow:

Algorithm 6.1

Input: Separate Façades identified in the images

1. Use vertical gradient projection to identify levels in the façade
2. In each level, apply horizontal gradient projection to segment façade into blocks
3. Use k-means clustering in a CIE-Lab color space to establish descriptors for façade area in an iterative process.
4. Identify windows an non-façade object in block segmentation
5. if (multi-view dataset is present) merge results from matched images

Output: Window areas and Façade areas identified in the images

We introduced following innovations over previously established methods:

- Gradient Projection method is modified such that it can provide more robust detection in complex (highly textured) façades. This is done by segmenting a façade into blocks, instead of detection of windows in gradient projection peaks
- We introduced a k-means clustering in CIE-Lab color space descriptor that can represent areas with multiple colors and is illumination invariant
- Window detection method was extended into a multi-view, where errors from segmentation and labeling can be rectified

We focus our work at ***complex façades*** – façades with a large number of different façade elements. In general, the most common façade element is "window"; however many façades in our datasets contain other elements, such as different ornaments, reliefs, arches or patterns. The measure of "façade complexity" is a gradient value over the façade area, as the edges of such elements increase this value systematically.
The complexity of a building façade provides challenges for window detection algorithms. Especially in the gradient projection approach, the presence of gradients outside window areas significantly reduces the quality of results. We present a modified gradient projection method robust enough to process complex façades of historical buildings.

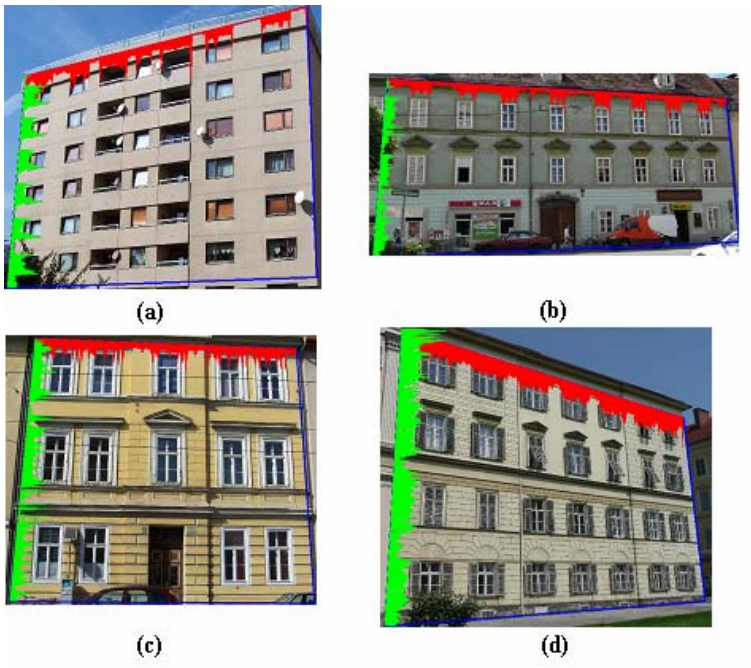


Figure 6.1: Four different types of façades in our database. The vertical gradient projection is marked in a green and the horizontal gradient projection is in a red color. Façade (a) has relative simpler texture and windows can be directly extracted from the projections. Façade (b) has additional horizontal structures, which make the horizontal separation difficult. Façades (c) and (d) have both projections highly non-regular and the extraction of windows is more complex.

In a single image scenario, this method is able to provide results even for façades under severe perspective distortion (for façades projected under sharp angles). Our algorithm is able to detect many different window types and does not require a learning step. In this section, we also extend this method into a multi-view scenario. We examine the results of the method in multi-view and evaluate its benefits.

6.1.1 Gradient Projection for Complex Façades

The description of an algorithm for processing a single façade located in a single image is given in this section. Our work is based on horizontal/vertical gradient projection approaches, primary on a work of [Lee and Nevatia, 2004]. This is a straightforward approach for the façade analysis as it takes advantage of specific

geometry of windows. But in the original form, it is not suitable for complex façades, where the high levels of gradient in vertical/horizontal direction can be located also outside the windows area (see Figure 6.1). We therefore introduce a new method to deal with this problem – Gradient Projection for Complex Façades.

The gradient projection methods are based on observation, that in the simple building façades, the strongest vertical and horizontal gradients are located at the edges of the windows. In more complex façades (with multiple different objects other then windows), this observation is usually not valid. Strong horizontal responses can be generated at the façade rims, shop signs or aches and vertical responses at columns or stone plates. Therefore, we approach this problem in a different way. The general idea is to segment the façade into rectangular areas – blocks. Subsequently, we use visual features to label each block as "façade" or "window".

In addition to standard gradient projection method [Lee and Nevatia, 2004], we introduce several new terms:

Separator lines / Levels

We base our method upon observation that in a single façade, the number of horizontally oriented objects is greater than the number of vertically oriented ones. In a most simple façade, only windows top/bottom rims are horizontally oriented, but as the complexity of the façade increases, more objects divide a façade into horizontally oriented levels (e.g. ledges, arches, brick patterns…). Based on this observation, we use the vertical projection to establish a horizontal division of the façade. For each local peak in the vertical projection a horizontal separator line is created. In this step, a façade is divided into a set of levels (bordered by separator lines) (see Figure 6.3(1)). In simple façades, separator lines will be located on the borders of windows, but for more complex façades, there will be many more separator lines dividing façades into more complex structures. Areas between two separator lines are denoted as *Levels*.

Figure 6.2: Gradient projection in our approach (on the left) and original façades (on the right). Horizontal projections are computed for each area between separator lines (green lines) independently. Value of the horizontal projection is visualized as an intensity of white lines.

Blocks

Subsequently, the horizontal projection of gradients is computed for each level separately. Only gradients in the area between separator lines are considered for projection (see Figure 6.2). In this step, "level" is divided into a set of blocks. The application of thresholds on the horizontal projection in each level will provide the borders for the block. The areas with the overall projected gradient above the threshold and the areas below the threshold are separated into different blocks. Left/right borders of the blocks are also established at the gradient peaks in the projection (see Figure 6.3 (2)). This division will result in blocks with high and low gradient content.

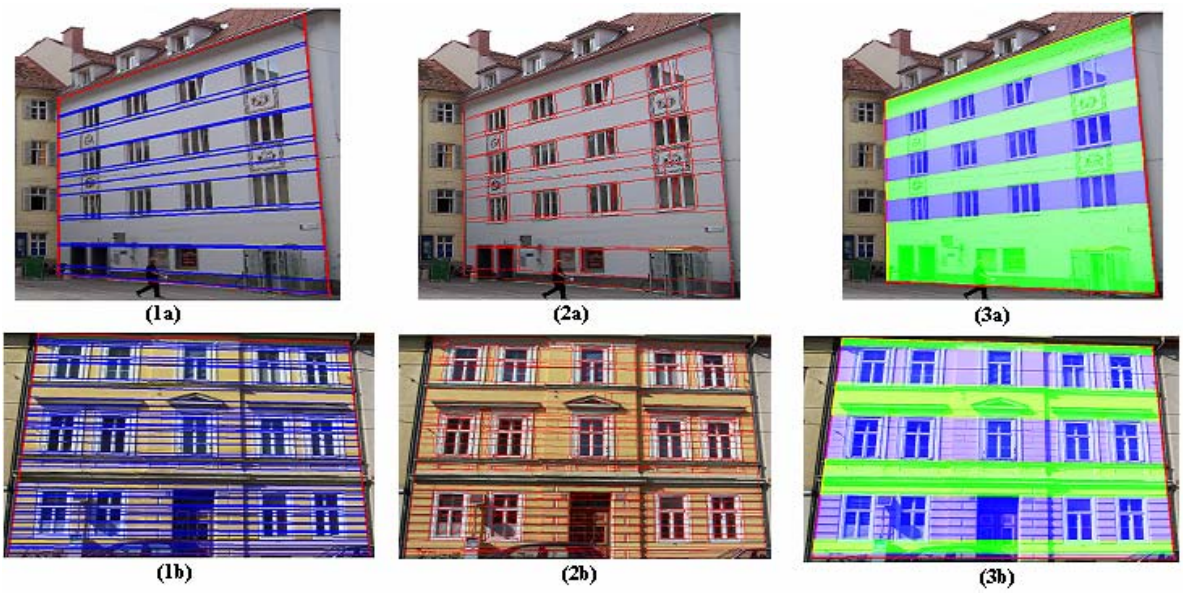


Figure 6.3: Analysis of a simple (a) and a complex (b) façade. In the first column, the separators between the levels are displayed. First image (1a) contains 16 levels, second image (1b) 46 levels. The second column displays the blocks located in the façades – 122 blocks for the first façade (2a), 1021 blocks for the second façade (2b). Third column shows the separation between the window levels and the façade levels.

In this method, "block" can be considered a segment and the above described process a geometry specific segmentation of the façade. As we don't only consider vertical/horizontal gradients, but combined gradient detector for each projection, also objects with non horizontal/vertical edges are segmented into such blocks

Block Descriptor

The goal of the algorithm is to label each block as window or non-window (façade). This decision is computed based on visual descriptors for each block, namely the size of the block, color and the gradient content of the block. The introduction of visual descriptors into the process is the main reason for additional robustness in datasets with complex façades when compared to standard gradient projection methods. These methods apply only gradient information – i.e. shape, disregarding other visual cues.

Figure 6.4: k-means clustering in CIE-Lab color space. Top row – façade with uniform color and it's clustering visualization on the right side. The façade in the bottom row consist of areas with several different colors. This non-uniformity express by itself as several different clusters in a color space.

The color descriptor is especially useful, when shape/gradient information is ambiguous. As the gradient information and method of its processing is designed specifically for the geometry and properties of façades, we can also use the same approach for color information. For this reason we will describe the color descriptor in more detail with the focus on specific façade's properties. Building façades usually consist of large areas of a uniform color, or small number of different colors (in clusters). The changes of illumination (mostly shadows) are often present and the same color may be displayed in different levels of brightness.

As a color descriptor for block areas we tested a novel method: the k-means clustering in CIE-Lab color space (see Figure 6.4). This method can describe areas with different color regions in a 2D space, instead of standard 1D histograms. Selection of CIE-Lab color space can provide two significant advantages:

1. Euclidean distance of two colors in CIE-Lab space is directly proportional to the visual similarity of the colors. This can provide simple metric for a clustering.
2. The clustering can be performed only in "a", "b" space, which represent the color value component. The "L" component in CIE-Lab space represents the

luminosity. A single value of L computed as a mean of each color in the cluster can be used as representative for each cluster to cope with shadows and illumination problems.

We describe façade area as a set of clusters in a color space [Recky and Leberl, 2010(II)]. For the purpose of clustering, we selected k-means method, as it can be built incrementally and we can easily estimate the value of k in O(n) time, where n is the number of pixels. As an input for a k-mean clustering process, the value of k (number of clusters) has to be given. Computing the k value is a method specific problem and the effectiveness of final clustering is largely dependent of computation of k. In our method, the k value is computed as follows:

1. k = 0; S = Ø
2. $\forall c \in \mathrm{F} : if\left(\neg\left(\forall s \in \mathrm{S} : |c - s| \le th\right)\right) c \rightarrow \mathrm{S}$
3. k = |S|

Where k is the number of clusters, F is the set of façade pixel's colors and *th* is threshold for the distance of two colors belonging into the same cluster. K-means clustering process in C++ can be implemented from an open source library, such as [6].
In an iteration process of block labeling (described below) we are provided with blocks labeled as the façade in each iteration step. The color of pixels in these blocks is transformed into a CIE-Lab color space and the process of clustering is performed. The façade color descriptor is built in a labeling process.

Labeling

Our next step is to decide if the block is part of the window, or part of the façade. This is done in an iterative process, where in each loop, the decision for each block is made if it is part of façade, or not.

- In the initial step, the blocks horizontally longer than 1/3 of the façade width are automatically labeled as façade blocks. This step is based on an observation that in uniform areas horizontally longer than 1/3 of the façade a window is unlikely to be located (windows are thinner than one third of the façade). Also

[6] http://www.koders.com/cpp/fid05CA27827355FE202A774065DAB0D4EFD8B0299E.aspx

in a large majority of façades, areas like these can be easily located (between window levels, at ground level below windows…). After this initial step a clustering is performed for each such block to extract first estimation of façade's color descriptor.

- In each subsequent step all blocks are re-labeled (as façade or non-façade) according to the actual color description of the façade. After the re-labeling, a façade's color descriptor is refined with the color information from blocks currently labeled as façade. When all blocks in one level are labeled as the façade, the entire level is excluded from the reclassification, but still contributes to the façade's color descriptor.

After several iterations – usually less then five, depending on the number of blocks – there are no more changes in labeling. After this step, all blocks are labeled as façade, or non-façade. Window blocks are identified as a non-façade blocks with gradient content.

Window Levels

In a simple façade, the methods of horizontal/vertical projection of gradient are able to identify windows and non-windows levels directly from the projections. This is also the case in our approach. Since there is no extensive gradient outside the windows area, the divisions between the levels are located on window frames. However, the presence of different patterns on the façade of more complex historical buildings is the reason why there are many more levels identified in the horizontal projection of these types of objects. There are usually multiple levels covering one windows row and the next step is to group these levels. The problems with grouping can be observed in the blocks bordering windows. Frames of the windows in the historical buildings are often irregular and may contain extensions into the façades, or different ornaments. Also the different types of arch windows are usually divided into several non-similar levels. Therefore, we identify the inside window and façade levels at first as levels containing blocks with strongest response to color and gradient classifiers. Subsequently we move into the in-between levels. The identification of a level as the façade/window is based

on the presence of window blocks, the identification of neighboring level and the height of the level (see Figure 6.3(3)).
In the next step, we proceed with the identification of windows inside the window levels. As the levels are assumed to be located horizontally – parallel to the ground plane, the borders of the windows are vertical objects inside the window levels. As the blocks inside levels are already labeled as window/non-window, the identification of window borders is straightforward. The assumption is that the border is located in the area of intersection between the most window and non-window blocks. In this process, "windows" are defined as blocks with window label clustered together and "façade areas" as remaining non-window blocks. For testing purposes, the window borders are projected into the original image.

6.2 Multi-View Scenario

The focus here is on the crowd-sourced, online open image dataset. The images are contributed by a large number of users and are taken in various lighting and weather conditions. The dataset is natively unorganized and often lacks additional information (camera calibration, geo-tagging…). For the purpose of testing the multi-view scenario, we use the *Tummelplatz* dataset, which simulates the crowd-sourced paradigm. The dataset is complemented with a 3D point cloud.
In the presence of multiple images of the same façade we consider again two approaches:

- Merging the images into a single, rectified façade and performing the window detection on the merged data.

- Applying window detection in each image separately and merging the results in a world coordinate system

Merging multiple images into one rectified façade is trivial when the means of image matching in a form of point cloud are present. We simply reconstruct the façade in the world coordinates by assigning a color into each façade pixel. This color is computed as a median from the hue, saturation and intensity from each corresponding pixel in the

multiple views. The selection of median would provide the elimination of outliers on the façade, like shadows, temporal object occlusions, or specific illumination problems. The façade analysis and window detection algorithm is applied to the rectified façade without any modification. We can consider this approach as providing the best possible input for the algorithm, given the different views and projections of the façade; however results are largely dependent on the accuracy of matching.
When the matching methods (point clouds, laser scanner…) are not precise enough, artifacts such as distorted objects, false edges or blurred borders can occur at the composed façade. For this reason, this method is less reliable at processing open sourced datasets, as the calibration is usually not present and subsequent matching is less precise.

The second approach - application of the method on each image in the multi-view stack of the façade is straightforward except in some special cases. After the window detection is performed for an image in the stack, we have the candidates for the windows in each image located. In case, windows are matched one-to-one in images, the coordinate of the corners are projected into the world coordinates for each window candidate. For each window, the corners are computed as the average of the corners of window candidates. To eliminate inaccuracies caused by matching, we can perform a factorization procedure over the set of windows, uniforming the results over each row. However, there can be a situation, when two or more window detections are matched to single detection in another image (see Figure 6.5). This may happen when façade section between two windows is not labeled and windows are fused in one image, but separated in other image, or when one window is incorrectly labeled into several sections. If this problem with specific windows is present only in small subset of the dataset, it can be considered an outlier and discarded from the merging step (thus it will not contribute to the corner coordinates computation).

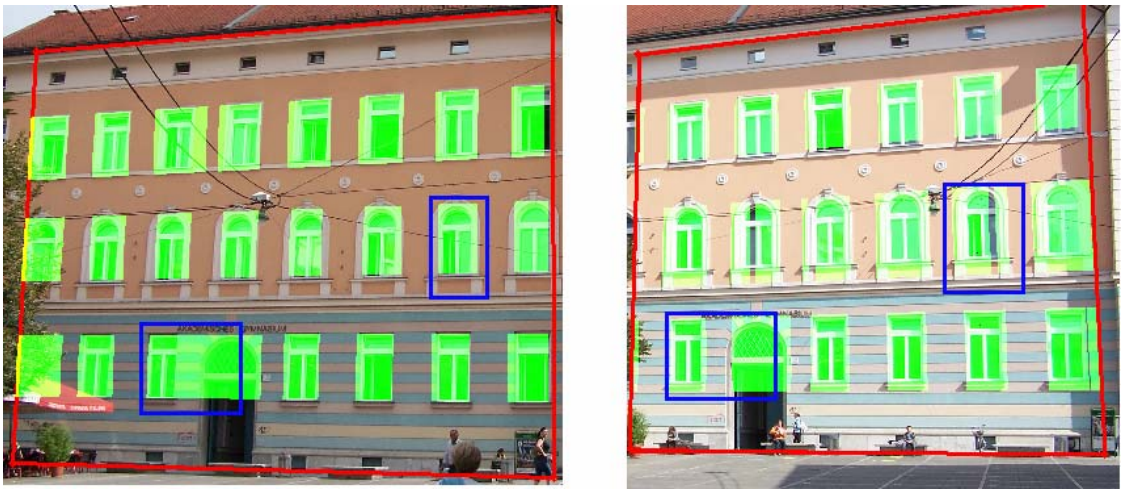


Figure 6.5: Differences in window detections for the same façade in different projections. Notice the fused detection (first image) and split detection (second image) in blue boxes that are corrected in other image.

6.2.1 Results

In our experiments we use 5 façades (each at 20-60 images, 253 images total), located at multiple images and their corresponding point clouds. The average probability of detecting the window is 91.4%. In subsequent experiments, we evaluate the precision of window placement (only for windows that were detected).

We compare our method in a single image scenario with the typical gradient projection method, as described in the paper [Lee and Nevatia, 2004]. For testing purposes, the windows were manually marked in the images. The precision of window placement (in percentage) is computed from pixel-wise comparison between detected window and ground truth as a ratio of mislabeled area (labeled area outside ground truth and area in ground truth not labeled) to the area of ground truth. In a **geometric measure** for a window with dimensions 80x135 cm, thus the area of ground truth of 10800 cm^2, each 1% of precision decrease means that the area of 108 cm^2 was mislabeled. For example in 90% precision of window placement, 1080 cm^2 was mislabeled. This could mean that for example the detected area had a geometric dimension of 88x135 cm, thus the 8 cm of window area width was a false positive. In results, the “Precision” value is computed as an average of precisions of placements for all windows.

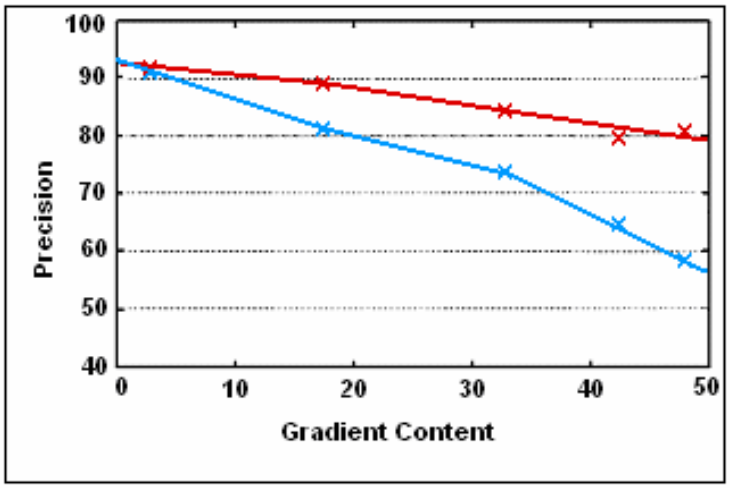


Figure 6.6: Relationship between the gradient content of the façade (excluding windows) and the precision of windows placement (in percentage). The blue line is displaying the relationship for the standard gradient projection method; the red line is for the method described in this section. One point is plotted for each façade and the "Precision" is computed as an average precision placement from all windows located in the façade.

In a first experiment, we examine a relationship between the gradient content, as the measure of façade complexity and the precision of window placement. The gradient content of the façade is computed as an average of gradient value {0,...,255} for each façade pixel (windows pixels are not considered as part of the façade in this case). The results are displayed in the Figure 6.6.

From the results of this experiment we can conclude that method described in this chapter performs significantly better for the façades with high gradient content. Most of historical building in our database (city core in Graz) has a gradient content between 40 and 50. In this group, the precision of window detection can improve up to 22%, using our method.

Our second experiment is focused on an implication of multi-view approach. We examine the dependency between the precision of window detection and the number of different views of the façade. Both approaches described in Section 6.2 have been examined. The results can be observed in the Figure 6.7.

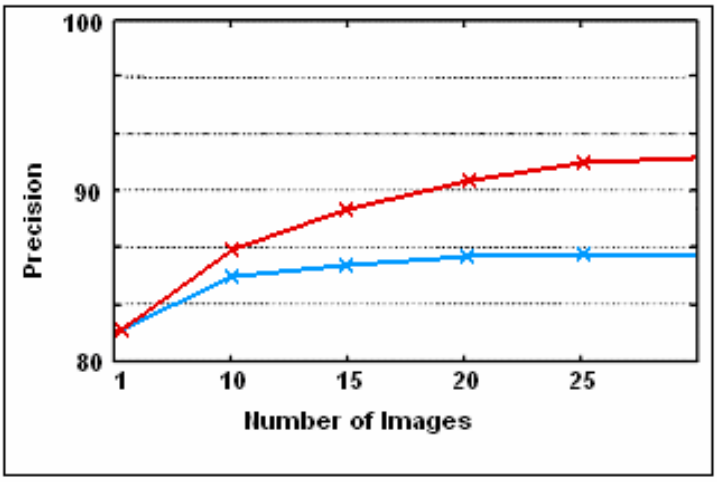


Figure 6.7: The relationship between the number of images in the multi-view scenario and the precision of window detection (compared in a hand-labeled, rectified façade). The blue line is for the method a) (first merging, then detection), the red line is for the method b) (first detection, then merging).

This experiment shows that at the certain number of images, the precision in window detection is reaching the limit for both methods. Also, the method of first detection, then merging provides better improvements in the multi-view scenario, when more images are available. This is considered to be an effect of a more robust error management for this type of approach, as the outliers are averaged and subsequently over-weighted in the merging step.

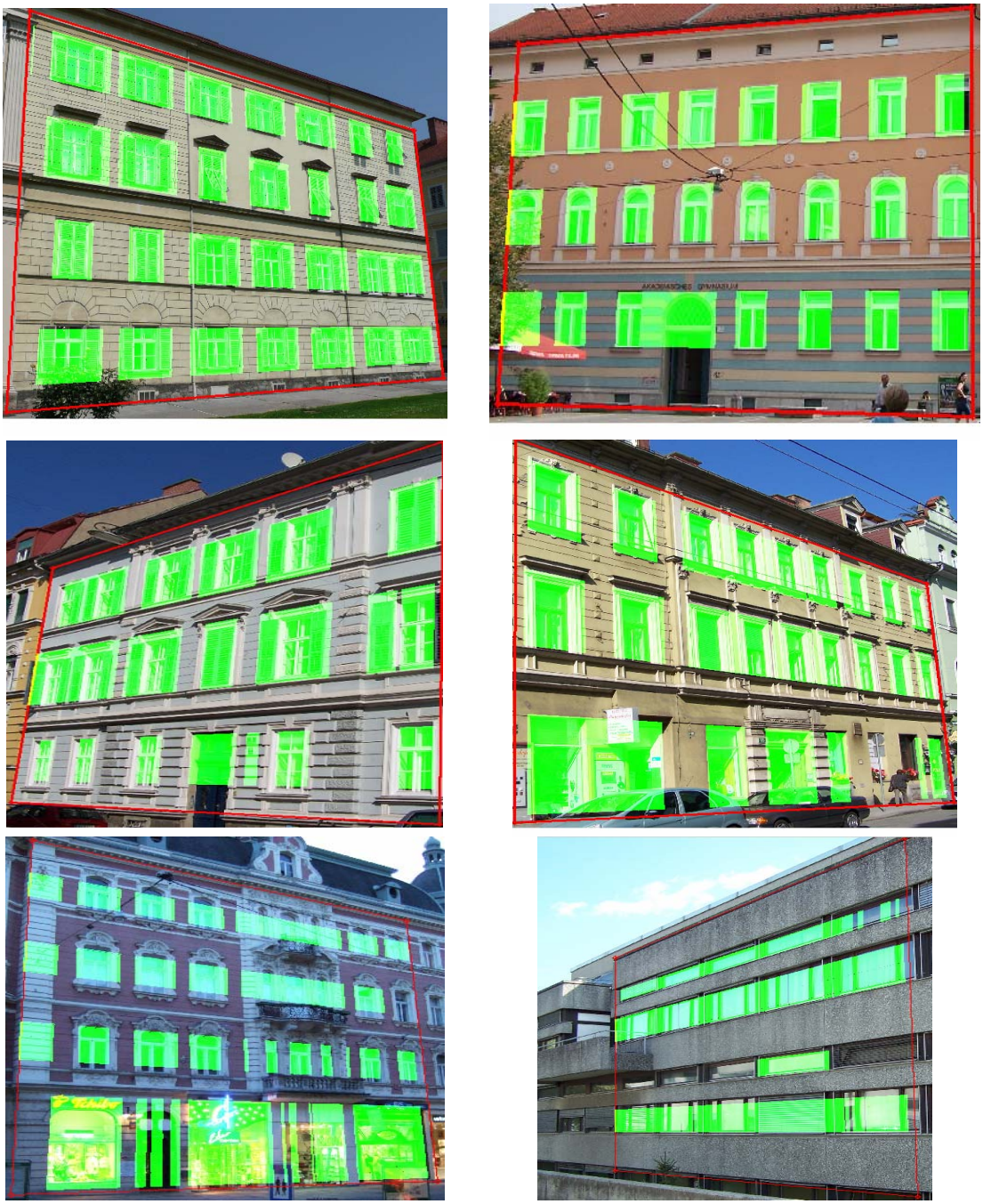


Figure 6.8: Examples of window detection results (single-view scenario). Successful detection has been performed even for the complex façades with other-than-window rectangular objects and patterns. Mislabeling is present mainly in cases, where windows are close to each other and façade section between them is too thin.

6.3 Discussion on Window Detection

When datasets created in European city centers are established, a large number of façades with complicated patterns can be observed. Standard gradient projection methods have problems with such façades, as non-window objects with vertical/horizontal lines provide similar responses in the algorithm than windows. This method was designed to present the same results for simple façades as the standard gradient projection method, but includes visual cues to detect non-window rectangular objects in complex façades. We applied Robert's edge detector that is able to identify directions of gradients. This allows us to identify not only rectangular objects, but also objects with different shapes (e.g. arches above windows) when an inclined gradient (from cross Robert's mask) is extracted, as is shown in Chapter 7.

We also use images with occluded façades for testing; however we identified occlusions by vegetation from the semantic segmentation. When the occlusion is not excluded from the façade areas, the method can fail as color features from vegetation gets included into façade descriptors or high gradients in such areas can cause false positives. Therefore, having the prior information about the occlusion improves results significantly.

As we compute color descriptors only for façade areas, this method can be viewed as façade area detector instead of window detector. We do not compute any visual descriptors for windows, as the reflective properties of window glass can cause windows to have very different color/shape features even in one single façade. Because of this, other objects at the façade can get labeled as windows (e.g. doors, shops, signs), but only if they have a different color than the façade. We considered such cases as false positives, as in our ground truth, only windows are labeled.

As demonstrated before, we allow façade areas to be composed of sections with different colors. These get represented as clusters in color space. However to initiate a cluster in a color space, a section with corresponding color must be present in the façade, that is at least 1/3 of façade length long, without significant gradient inside. If such section is present, a cluster is created in the first step of the color descriptor formation and is subsequently refined in the next steps. If the long section with façade color is not present, areas with such colors will be identified as non-façade, but if they are not enclosed in a rectangular box with high gradient at borders, they will not be

identified as windows. An example of this can be observed in Figure 6.8, left bottom images, where the white sections of the façade are not included into façade color descriptor and are subsequently identified as windows at the window levels.
Even in the final results, we represent windows by their bounding box (to be compatible with ground truth), the shape of the windows is approximated much more precisely in gradient projections. For example, the top section of arch windows (curvature) is usually sectioned into several levels and in each level, the curvature is approximated by start/end gradient. This results in arch shape to be evident in projections. Also in a case of sectioned windows, each section is identified as a separate level. Therefore, additional information about the shape and composition of windows can be extracted directly form the gradient projections and levels if needed.
Extension of the method into multi-view, as described in this work, can provide benefits in two cases:

- Window borders are not precise. If there are other images in the dataset, that contain the same window and the window borders are estimated more precisely in them, this error will be corrected. Such error may be present when gradient at the border is distorted, e.g. window borders are partially obscured, shadow is present at the border, or there are illumination defect in the image. To estimate window borders better in other images, such errors should not be present. Therefore the most significant improvements are in very variational datasets, such as crowd sourced.

- A window is dissected into several detections or two/more windows are merged into one detection. Such errors get corrected, if windows involved are labeled correctly in the majority of images where they are present. If this is not the case, it is difficult to decide, which detection is correct, as there are no additional visual cues to make the decision. Such errors may be present when there is some object between windows. If this object is temporal, the error can get corrected eventually from other images, where the object is not present; however if the object is permanent and part of the façade, it may cause mislabeling in most of the images. An example of this can be observed in Figure 6.8 (right, middle row), where a banner is placed between windows and

causes merged detection and in the image above, where merged detection is caused by distorted gradient at the door border.

The cause of a false negative (when a window is present, but not detected) is not addressed in a multi-view scenario. This decision was made based on the observation that even when the window can be actually present in the scene, it may be occluded. If such window is detected in other images (from different points of view, or at different time, the occlusion may not be present) and the detection would be transferred to the original image, it would case a false positive. Therefore, the extension of the method into multi-view does not solve a missed detection in our approach.

Chapter 7

Multi-View Random Fields

7.1 Context in Multi-View

In previous sections, we described the concept and importance of context in our application domain and described several methods of context application in streetside image processing. However, the great majority of computer vision algorithms is considering context only in a single image [Kumar and Herbert, 2006], [Hoiem, 2007], [Heesch et al., 2008], [Korč and Förstner, 2008]. Exception from this rule is the application of context in algorithms working with video streams, or with 3D point clouds (or other 3D data) composed from multi-view input. But even in these cases, the application of context is based on a principally different approach than in the single-view scenario, making these approaches incompatible [Fruh et al., 2005]. For example, Random Fields (RF) methods are natively constructed for single image data [Lafferty et al., 2001], [Kumar and Herbert, 2006]. One can consider the generalization of a Random Field model to include 3D data as a kind of observation in the image. This will include the context in the process as a form of observation data from other matched images, but this context is not at the same level as single image context encoded in pairwise potential of RF.

In this section, we introduce a model of RF, which is designed to work with context in a multi-view scenario. The basic idea is to consider context in the image as observation

data transferable between images, when matching is available. This idea is based upon the assumption that real 3D objects are assembled in a contextual 3D scene which is unique and unambiguous. Subsequently, each digital photo is only a projection of scene into a 2D plane and the context of such projection is only an incomplete approximation of the scene's context. Given multiple images of the scene, we can improve the approximations of the contexts in separate images to retrieve a context data superior to ones located in each image. For this process a 3D model of the scene is not required, however if available, it can be used as an observation data in RF model.

A motivation behind this approach is in the duality of data used in computer vision methods. In a vision algorithm, we can use visual features (color histogram, texture covariance…) and context features (spatial relations, semantics…) to perform recognition [Recky and Leberl, 2009]. The efficiency of visual and contextual features varies in different images. In general, visual features are most effective, when a detailed view at the object of interest is present (e.g. object is not obscured, it is not located far away from the camera and it is projected from suitable angle). Contextual features are effective when strong context data is present in the image (e.g. more objects are located in the image – scene is projected from longer distance) [Recky and Leberl, 2010]. This observation was also confirmed in our experiment described in Section 4.2.3 (Classification consistency as a function of distance from the camera**).** In one digital image of the scene, the visual features can provide strong cues for recognition, but the context can be inefficient. However, a second image of the same scene can provide stronger contextual features (see Figure 7.1). For these reasons, we developed a method to transfer contextual features between matched images.

7.1.1 Context from Different view positions

Many computer vision works have been focused on the idea, how visual information changes in images of the same scene projected from different view points. Such research is the primary focus of scene reconstruction algorithms. For example, in a work of Hartley and Zisserman, Multiple View Geometry in Computer Vision [Hartley and Zisserman, 2004] a multi-view dataset is used for 3D reconstruction of the scene. For such task, the knowledge of an object's changing geometry between different

views is essential. It has been observed that for local visual information on geometry, affine transformations are sufficient to describe the changes. Therefore, local descriptors used for image matching, such us SIFT [Lowe, 1999] are constructed affine invariant.

However, it is less well understood how context of the scene can change between images taken from different view points [Santosh et al., 2009]. When comparing context as a feature that can change between images in multi-view datasets, we consider what objects are located in the scene, how are they aligned and what spatial relations exist between them. These features can change rapidly, even if images are taken from a single position, rotating optical axes of camera (as new objects can appear in different photos, providing new semantics and spatial relationships – see Figure 7.1). Even though the context information started to play a more important role in computer vision, a focus of a majority of algorithms is on the context in a single view scenario. The reasons for such limitation are:

Local vs. Global Context

Most algorithms claiming to work with context are applying only local information. For example, the most common application of MRF is at the pixel level, extending the patch around an examined area and considering several more pixels as context information [Santosh et al., 2009]. This locality of context information makes the information from different view points highly redundant and non-usable. As it was demonstrated by multi-view geometry methods, one can expect only affine transformed information in such local context. However, when the context of an entire image (we can denote this as *Global Context*) is considered, very different context information can be expected. For example, taking two digital images from the same position, but with different optical axes, objects on such images do not change geometry or their geometric relations to each other, but the context in images can change dramatically, as different sets of objects can be located in both images, providing new spatial relationships. For such reasons, local context change only slightly, or not at all, but global context can be very different (see Figure 7.1).

Figure 7.1: Two examples of façade set in different context in each image (from *Thummelplatz* dataset). Red lines represent the context features (spatial relationships) between objects detected in the images. For each example, the top row represents the façade in context with other objects in the scene (other façades, vegetation, ground, temporal objects…) and at the bottom row, façade is primary set in context with its elements (windows, shops…). For better overview, only segments larger than 15% of the image area are displayed and only for such segments, red edges are visualized, representing the spatial relations between segments. Notice that in closer views, the visual elements are better represented (more details, better angle), however the context with other scene objects is missing (as they are not in the view). Transferring the context in such cases form other image would help in vision task.

Context vs. Reconstructed Scene

Algorithms working with multi-view datasets deal with 3D scene reconstruction. When such reconstruction is achieved (for example in a form of 3D point cloud), one could expect from it to contain most information that would be extracted as contextual features in all involved images. For example, a 3D point cloud would contain all spatial relationships between objects, as the 3D structure of the scene is known.

For this reason, the 3D reconstruction can be considered to provide superior context information about the scene to all partial context information from individual images. However, this is usually not the case. The focus of 3D reconstruction is commonly at one central object, or few such objects [Irschara et al., 2007]. In a final reconstruction, only these objects are actually reconstructed and large areas of the involved images are neglected. This is due to insufficient matching in such areas, as the objects in them are not located on adequate set of images to be reconstructed. For this reason, each image in the stack used for reconstruction can usually provide additional context information, which is not observed in 3D reconstruction.

7.2 Global Context as a Feature of an Image

In a multi-view image stack, each image can be considered a unique unit of information. Even if a large volume of data from the image can be transferable to different images in the stack (redundant information) each image usually contains also exclusive information. For this reason a global context can be considered a feature assignable to each image separately. Let's define context as a relationship (semantic, geometric…) between two objects that occur in an observed scene. Image context is a set of context relations between each two or more objects located in the image. In computer vision methods, objects are usually represented by pixels or segments. In case of pixels, the application of global context would require to establish contextual relations between each two pixels in the image, which is for current computation power considered unfeasible. Therefore we consider segments to be most suitable representations of objects for contextual examination, but other representations could also be possible (e.g. cells of regular grid). We can denote the elemental representation of objects between which context is examined by the more general term "site".

Sites

In a Random Field framework, object representation is denoted in more general term as *site* and the set of all sites in one RF model is denoted as *S*. In a local context approach, all sites in *S* are usually located on a small patch of the image. The global context framework requires the sites from *S* to cover entire image, or at least a large majority of the image. From the application point of view, an area of image is assigned to each site. Areas from different sites are not overlapping and represent specific objects of the scene. As such, a single label is assigned to each site after RF evaluation. Visual features of the area assigned to specific site are denoted as image observation $\boldsymbol{y}_s$ from site s. In a graphical model, if there is an edge between nodes assigned to sites s_1 and s_2, let's denote this relation as $\Phi(s_1, s_2) = 1$ and consequently if there is no edge between s_1 and s_2, denote this as $\Phi(s_1, s_2) = 0$.

Context between sites in multi-view

In a most simple application of context transfer, we can select each site from the image, find corresponding sites from other images in the stack and in every such image replace the corresponding site with the original site from the first image. After evaluation of RFs from all images, we get scores for each class. We can subsequently select the best score or most top scores as winning classification. This approach is roughly equivalent (with the exception of different visual features in transferred site) with the evaluation of each image separately and merging results through the image matching – a method that has been examined in previous chapter. The advantage of such approach is that we can use standard Random Fields models for evaluations, making the application straightforward. However we only examine partial context in each image separately in this way, making no assumptions that some images can provide superior context information than others. This approach is also computational demanding and redundant, as we often examine context between same objects in each image, which usually does not change. Therefore, we can examine different approach that eliminates such disadvantages.

Transferable sites set

Our proposed solution is to transfer sites that are not located in the original images, but are in the spatial relationship with examined site in other images (see Figure 7.1). This is not a trivial task, as matching sites between images is a hard problem. One site can be matched through corresponding points to multiple sites in other image or can even represent different object. However, let's first assume that each site from the first image can be only matched to a single site in the other image. For a single image from the image stack, let's define the transferable set of sites as:

Definition 2: *If $S_k = \{s_1, s_2, \ldots, s_n\}$ is the set of sites for single image $k \in I$, where I is the set of images and correspondences have been established between the images from I such that $s_i' \in S_l$ is a site from image $l \in I\text{-}\{k\}$ corresponding to a site s_i, than the $R_k = \{r_1, r_2, \ldots, r_m\}$ is the set of transferable sites for the image k if $\forall r_j \in R_k \exists s_i \in S_k \mid \Phi(r_j, s_i') = 1$ and $\forall r_j \in R_k \neg \exists r_j' \in S_k$. R_k is constructed such that $\forall r_i, r_j \in R_k$, r_i and r_j are not correspondent to each other in any two images from I.*

Thus the R_k is the set of sites from other images than k, that are in the relationship in graphical model with some corresponding site to sites from S_k, but themselves have no correspondences in S_k. Set of transferable sites can be seen as a context information, that is available in the image stack, but not in the examined image. If sites are the representations of objects, than in a transferable set, there are objects in context with the scene of the image that are currently not located in the projection, thus are occluded, out of the view or in a different timeframe. This also means that the visual

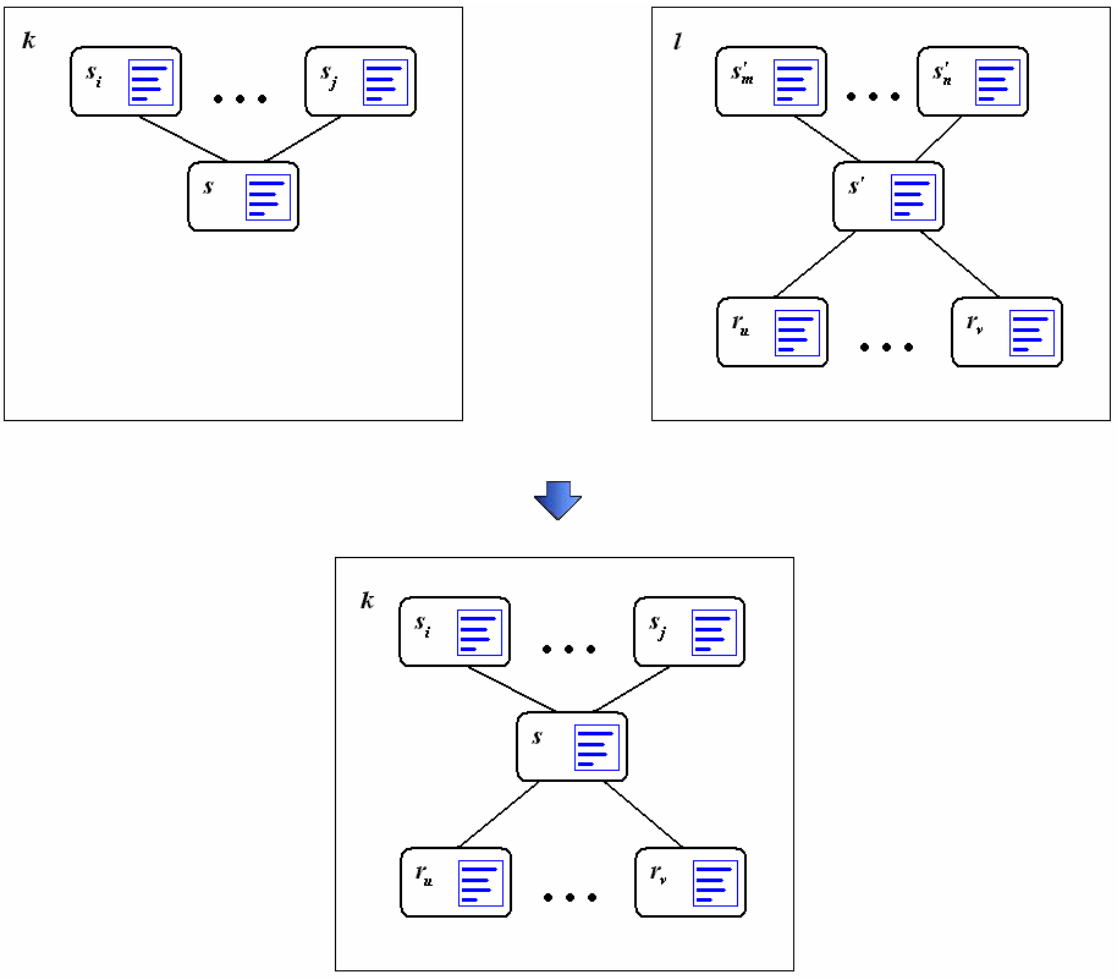


Figure 7.2: Transfer of sites from the image $l \in I$ to the image $k \in I$, as presented in Definition 2. Only sites from l that are not corresponding to any sites from k are transferred. This figure demonstrates only transfer between two images. If more images are involved, the set R_k include sites from all such images. An example of transferable sites in specific application can be observed in Figure 7.7.

information from sites in R_k is not present in the image k. If the sites from R_k are included in vision process, they can provide additional context and visual information that is not originally present in the examined image.

7.2.1 Multi-view Random Fields Definition

Our next step is to examine the compatibility of existing random field models with the application of transferable sites set. The new model, incorporating transferable sites can be denoted as Multi-view Random Field (**MVRF**) [Recky et al., 2012]. Transferable sites have the same set of visual features than sites native to the image and they can be assigned the same set of spatial and contextual relations in a graphical model, however the status of these sites is not equal to native sites. Transferable sites lost all original contextual relationships except for the relationships to the sites they are connected within the examined image. This makes them harder to label. But the labeling of transferable sites is not the aim in the case of an examined image (the goal is to label only native sites), thus transferable sites can contribute information for image labeling, but the labeling of themselves is irrelevant. This makes the concept of transferable sites difficult to use in standard Markov Random Field model [Kindermann and Snell, 1980]. In the posterior distribution of MRF defined as

$$P(\mathbf{x} \mid \mathbf{y}) = \frac{1}{Z_m} \exp\left(\sum_{i \in S} \log p(\mathbf{y}_i \mid x_i) + \sum_{i \in S} \sum_{j \in N_i} \beta_m x_i x_j \right), \tag{7.1}$$

the pairwise potential is independent from observations in the image (only from labeling) and the unary potential is defined only based on observations in a specific site. This model in inherently not able to consider different type of sites (native, transferable), as this observation cannot be considered. The other reason why the standard MRF model in not suitable for the task is that as described in the previous chapter, the MRF model is primarily suitable for a simple formulation of context, thus it is generally used in local context applications. Therefore, in our work, we focus on the Conditional Random Field model. The definition and posterior distribution of CRF is given in Section 3.2 Random Fields. We extend this posterior probability distribution into MVRF model framework.

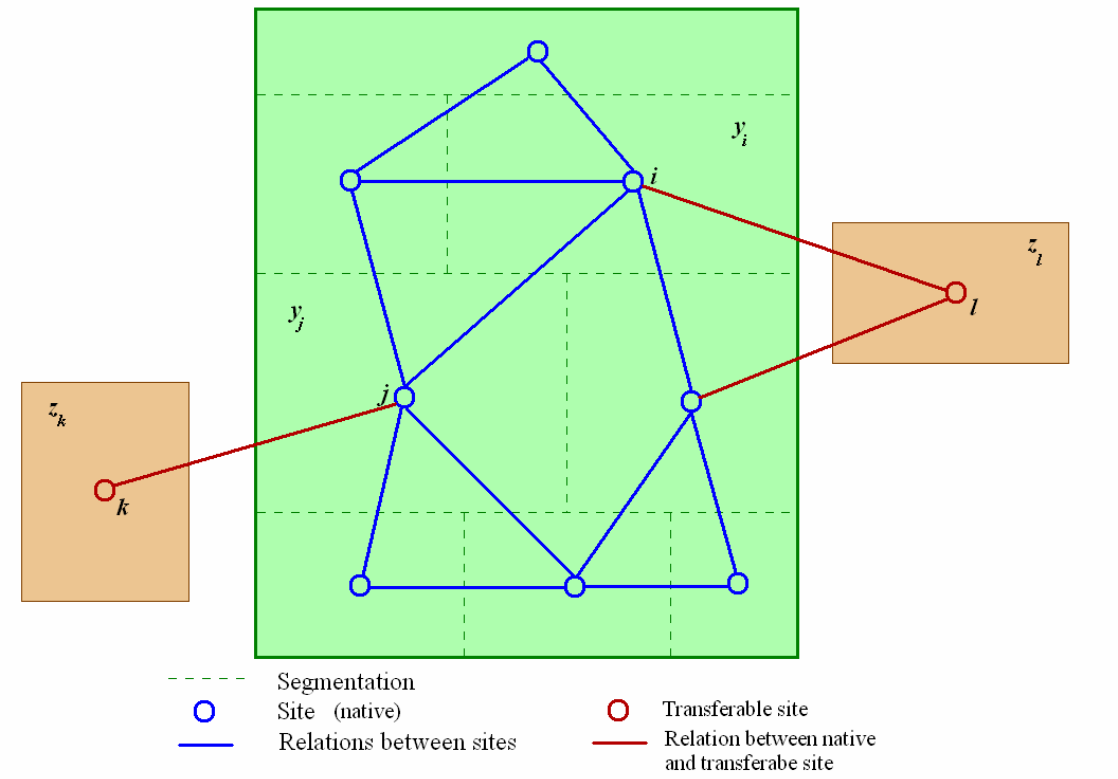


Figure 7.3: A graph structure of MVRF. In this example, segments in the image represent sites and the green area is a native image. We assume that there are other images in a dataset matched to a native image and sites *l* and *k* were detected in such images, but not in a native image. Blue is the set of native sites (nodes) and their relationships (edges). We connect transferable sites into this structure, which will provide additional visual information (z_l, z_k …) and contextual information (edges between *i* and *l*, *j* and *k* …). Observe that a transferable site can be connected to multiple native sites (if such relations are detected in some of matched images), but no to transferable sites can be connected directly in the graph structure.

Given the observed data $\mathbf{y} = \{\mathbf{y}_i\}_{i \in S}$ from the image, corresponding labels $\mathbf{x} = \{x_i\}_{i \in S}$, where S is the set of sites from the image and observations from transferable set $\mathbf{z} = \{\mathbf{z}_i\}_{i \in T}$ (see Figure 7.3) with corresponding labels $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i \in T}$ the posterior distribution over labels is defined as:

$$P(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A_i(x_i, \mathbf{y}) + \sum_{i \in T} A_i'(\tilde{x}_i, \mathbf{z}_i) + \sum_{i \in S} \left(\sum_{j \in N_i} I_{ij}(x_i, x_j, \mathbf{y}) + \sum_{j \in K_i} I_{ij}'(x_i, \tilde{x}_j, \mathbf{y}, \mathbf{z}_j) \right) \right)$$

,(7.2)

where Z is the normalizing constant, N_i is the set of native sites neighboring i and K_i is the set of transferable sites neighboring site i. - A_i and - A_i' are unary potentials, - I_{ij} and - I_{ij}' are pairwise potentials (for native sites and transferable sites respectively). The differences between potentials for transferable sites and for native sites are as follows:

- In the unary potential, only observations from the transferable site itself is considered, instead of observation from the entire image for native sites. This is due to the fact, that transferable site does not to have any connections to the image except for the site it is neighboring. Even if other connections exist (with other sites in the image), it is hard task to establish relationships. In the discriminative model – DRF, discriminative classifiers are allowed to be included directly in the unary potential. Therefore such classifiers has to be modified to accept only local features (from single site), when unary potential for transferable site is considered. For native site, there are no changes to standard conditional model.

- In the pairwise potential, in addition to observation from the image, local observation from the transferable site is considered, when a pairwise relations are examined between native site and transferable site. The inclusion of all image observation grant at least the same level of information in pairwise relations computation as in a standard CRF model pairwise potential and the additional observation from a transferable site represent extended context for native image observation. Because of this, the computation of pairwise potentials must be modified to include such additional information. In the DRF model, this also means the modification of discriminative classifier, as is the case in the unary potential. The pairwise potential for two native sites is the same as in standard CRF model.

- No pairwise relations are considered between two transferable sites. This is based on the construction of a transferable sites set. A site from such set can be neighboring several native sites, but not an other transferable site. This can be seen as a limitation for the model, however without additional high frequency information about the scene (as a prior knowledge), it is virtually impossible to establish relationships for transferable sites.

- The computational complexity of the model is not increased significantly. Pairwise potentials are computed only for native sites, as it is in the standard CRF model. The difference is in the number of neighbors for each site, however even this number should not increase significantly. When considering a global model, each new neighbor (transferable site in relation to the native site) represents a new object in the projection. This is dependent on the differences between projection parameters – camera positions, optical axes…, but even for very different parameters, the number of objects should not differ significantly for the same scene. From the general observation, the number of neighboring transferable sites is notably lower than the number of neighboring native sites.

Unary potential modification

As described in Section 3.2 Random Fields, the unary potential for native image sites in the DRF is a measure of how likely a site i will take label x_i given the observation in image $\mathbf{y}$. Given the model parameter $\mathbf{w}$ and a transformed feature vector at each site $\boldsymbol{h}_i(\mathbf{y})$, the unary potential can be written as:

$$A_i(x_i, \mathbf{y}) = \boldsymbol{log}\left(\sigma\left(x_i \mathbf{w}^T \boldsymbol{h}_i(\mathbf{y})\right)\right), \tag{7.3}$$

Assuming that Generalized Linear Models (GLM) are applied as local class conditional. For the transferable sites, feature vector is limited to the observations from single site. This limitation define a new expression for unary potential, exclusive to transferable sites as

$$A_i'(\tilde{x}_i, \mathbf{z}_i) = \boldsymbol{log}\left(\sigma\left(\tilde{x}_i \mathbf{w}^T \boldsymbol{h}_i(\mathbf{z}_i)\right)\right), \tag{7.4}$$

The feature vector at the transferable site i given the feature space vectors $\boldsymbol{f}_i(\mathbf{z}_i)$ is defined as $\boldsymbol{h}_i(\mathbf{z}_i) = \left[1, \varphi_i(\boldsymbol{f}_i(\mathbf{z}_i)), \ \ldots \ , \varphi_L(\boldsymbol{f}_i(\mathbf{z}_i))\right]^T$, where φ_k are nonlinear functions mapping feature vectors into high dimensional space. Model parameter $\mathbf{w} = \{w_0, \mathbf{w}_1\}$ is composed of bias parameter w_0 and model vector $\mathbf{w}_1$.This model is designed for dual-class labeling problems, where $x_i = \{-1, 1\}$, but can be easily extended into multi-class (as described in [Kumar and Herbert, 2006]) with introduction of step function:

$$A_i'(\tilde{x}_i, \mathbf{z}_i) = \sum_{k=1}^{C} \delta_k(\tilde{x}_i) \boldsymbol{log}\, P'(\tilde{x}_i = k \mid \mathbf{z}_i), \tag{7.5}$$

where $\delta_k(\tilde{x}_i)$ is 1 if $\tilde{x}_i$ = k and 0 otherwise and C is number of classes. In this formulation, separate model parameters $\mathbf{w}_k$ are used for each class k in local class conditional [Lafferty et al., 2001].

Pairwise potential modification

The pairwise potential for two native sites from the image remains the same as described in Section 3.2, given the GLM are applied to compute class conditional:

$$I_{ij}(x_i, x_j, \mathbf{y}) = \beta\left(Kx_i x_j + (1-K)\left(2\sigma\left(x_i x_j \mathbf{v}^T \boldsymbol{\mu}_{ij}(\mathbf{y})\right) - 1\right)\right), \tag{7.6}$$

where $0 \le K \le 1$, $\mathbf{v}$ and β are the model parameters and $\boldsymbol{\mu}_{ij}(\mathbf{y})$ is a feature vector. For the native sites, this formula remains the same as in standard CRF model. For transferable sites, we introduce additional feature vector in a form of observations from specific site:

$$I_{ij}'(x_i, \tilde{x}_j, \mathbf{y}, \mathbf{z}_j) = \beta\left(Kx_i \tilde{x}_j + (1-K)\left(2\sigma\left(x_i \tilde{x}_j \mathbf{v}^T \boldsymbol{\mu}_{ij}(\mathbf{y}, \mathbf{z}_j)\right) - 1\right)\right), \tag{7.7}$$

where $\boldsymbol{\mu}_{ij}(\mathbf{y}, \mathbf{z}_j) = \boldsymbol{\mu}(\boldsymbol{\psi}_i(\mathbf{y}), \boldsymbol{\psi}_j(\mathbf{z}_j))$ is a feature vector defined in domain $\boldsymbol{\mu} : \Re^{\gamma} \times \Re^{\gamma} \to \Re^{q}$ such that functions $\boldsymbol{\psi}_s(.)$ are mapping observations from the image/sites related to site s into a feature vector with dimension γ. Note that the smoothing term $Kx_i\tilde{x}_j$ is the same as in standard DRF definition. Thus if K = 1, the pairwise potential still perform same function, as in a MRF model, however given new

transferable sites, the smoothing function will depend also on their classification $\tilde{x}_j$. In this case, visual information from transferable sites is not involved in pairwise term and is only applied in unary term. If $K<1$ the data-dependent term $2\sigma\left(x_i \tilde{x}_j \mathbf{v}^T \boldsymbol{\mu}_{ij}\left(\mathbf{y}, \mathbf{z}_j\right)\right) - 1$ is included in pairwise potential. Observations from the image related to the examined native site and observation from transferable site are transformed into feature vector and involved in computation.

The pairwise potential for standard DRF in a multi-class case is defined in a work [Kumar and Herbert, 2006]. We use the same definition for native sites. For transferable sites, the pairwise potential in multi-class formulation is defined as:

$$I'_{ij}\left(x_i, \tilde{x}_j, \mathbf{y}, \mathbf{z}_j\right) = \sum_{k=1}^{C} \sum_{l=1}^{C} \mathbf{v}_{kl}^T \boldsymbol{\mu}_{ij}\left(\mathbf{y}, \mathbf{z}_j\right) \delta_k\left(x_i\right) \delta_l\left(\tilde{x}_j\right), \tag{7.8}$$

where C is a number of classes, $\delta_m\left(x\right)$ is a step function defined as 1 if $x=m$ and 0 otherwise. Vector $\mathbf{v}_{kl}$ is a class dependent model parameter. Note that the pairwise potential in this formulation contain only data-dependent term, forcing feature vector $\boldsymbol{\mu}_{ij}$ to encode all relevant contextual relations between classes, including smoothing function if necessary. In this formulation, similar to the binary class DRF, feature vector can be seen as discriminative model, partitioning the feature space into $C(C+1)/2$ regions.

7.2.2 Parameter Learning in Multi-view

For DRF training in multi-view, we use a labeled ground truth dataset that has a means of image matching available, particularly images from an *Industrial System* dataset and parts of *General Images* dataset without matching. Given the assumption, that parameters of unary potential can be learned without image matching, as the visual features does not change for transferable sites, therefore they can be learned from original image. The spatial relations defined for pairwise potential also do not change significantly for the pair native-transferable site. For such reasons, we can assume that multi-view random fields can be learned even directly from single images without dataset matching. This assumption is based on a mechanism, how transferable sites set

is constructed, and how such sites are involved in process, providing no new spatial relationships, or visual features.

Similar to standard DRF model, the parameters of multi-view random field are $\theta = \{\mathbf{w}, \mathbf{v}, \beta, K\}$. The suggested approach in [Kumar and Herbert, 2006] for an estimation of parameters is based on the pseudolikelihood and defined as

$$\hat{\theta}^{ML} \approx \boldsymbol{arg\,max}_{\theta} \prod_{m=1}^{M} \prod_{i \in S \cup T} P\left(x_i^m \mid \mathbf{x}_{N_i}^m, \mathbf{y}^m, \mathbf{z}^m, \theta\right), \tag{7.9}$$

where m are indexes over training images and M is the total number of training images. The formula for single image $P\left(x_i \mid \mathbf{x}_{N_i}, \mathbf{y}, \mathbf{z}, \theta\right)$ is evaluated based on parameters from equation (). As the pseudolikelihood is not a convex function, a good initialization is necessary to avoid local maxima. This can be achieved through the computation of standard maximum on log-likelihood in training data.

In a multi-class model, parameters are $\theta = \left\{\{\mathbf{w}_k\}_{k=1..C-1}, \{\mathbf{v}_{kl}\}_{k,l=1...C}\right\}$. The maximum likelihood estimates of the given parameters are defined as:

$$l(\theta) = \sum_{m=1}^{M} \boldsymbol{log}\, P\left(\mathbf{x}^m \mid \mathbf{y}^m, \mathbf{z}^m, \theta\right), \tag{7.10}$$

where M is the number of training images. The computational complexity of such formulation scale with the number of classes dramatically, therefore the analytical computation is untraceable for larger number of classes. As suggested in [Kumar and Herbert, 2006], computation of pseudo-marginal can be applied to reduce the complexity. A Belief Propagation [Pearl, 1982] approach is applied in a work of Kumar to get a Pseudo-Marginal Approximation. Similar approach can be applied, if the problem get complex due to a large number of transferable sites.

7.2.3 Inference in Multi-view

As described in Section 4.1.4, we used Belief Propagation to infer a standard DRF model for semantic segmentation. It was demonstrated in previous works that using the

same method for parameter approximation and inference minimizes the classification error.

Parameter inference in MVRF can be implemented much the same way as in DRF. In a Belief Propagation framework, messages are exchanged between neighboring nodes in each iteration, until no more changes in classification is observed. Other possible methods for Random Fields parameter inference are Tree-Based Reparameterization and Expectation Propagation [Wainwright et al., 2002], [Kolmogorov and Wainwright, 2005].

7.3 Application of Multi-View Random Fields

The introduction of Multi-View Random Fields (MVRF) in a properly aligned redundant dataset and the involvement of transferable sites in image processing have two primary applications:

- The introduction of new context and visual features from the sites which are not located in the original image. This is mostly the advantage in datasets, where the images contain less redundant information and more novel visual sites between each other. Therefore this application is relevant mostly in crowd sourced datasets, where each image is projected from different view points and angles. In this case, the novel sites are usually located at the borders of images. To involve such information to have impact on classification of more sites, we can increase the size of neighborhoods for each examined site. This approach also requires matching to be available for objects (sites) in the image. This is not an easy task as for example, there is no general matching method for object classes, like “sky”, or “cloud” and the matching of many other classes is difficult (vegetation, temporal objects, etc.). With limited success, we can match such sites directly by their visual features and location in the image. For these reasons, the application of MVRF with aim on introduction of new context information is difficult in street-side scenes. However in different scene settings (for example in indoor scenes), it can be much more straightforward.

- Improvement in robustness of site detection. The method of site detection can occasionally fail for several reasons, such as temporal or spatial illumination anomaly (shadow or specular reflection), occlusion, image quality problem, etc. This is primary the problem, when high order of organization is expected from objects in the image and such misdetection can create gap in contextual arrangement. For example, if arches above windows require window to be detected right below them in a context-based classifier, the misdetection of the window can present problem also for the detection of corresponding arch. The introduction of transferable sites in a MVRF model can largely correct this problem, as the undetected site can be located in other images and can provide appropriate context for native sites. This application of MVRF is mostly useful in highly redundant datasets, where the same object is located in multiple images, therefore primary in industrial system datasets. In such cases, the application of MVRF allows for much stronger application of context-based classification, as in a standard MRF or CRF models. In standard models, the possibility of misdetection has to be accounted for by assigning less weight on contextual cues and more weight on visual cues. This limitation was mostly notable in highly organized scenes, as building façades. With the application of MVRF we can establish stronger contextual relationships between objects in such scenes.

In subsequent sections, we will present the application of MVRF in building façades for the purpose of façade elements detection and classification. This application is based on *Industrial System* dataset, however the image matching is provided by corresponding point detection method instead of LiDAR. We selected left camera subset of dataset, which provides clear view of building façades, not distorted by perspective, that are easy to rectify and provide good visual cues. As described before, this setting will demonstrate the advantages of MVRF in cases, when a site was misdetected and present lost contextual information in standard models. It should be noted, that in most images, building façades are not projected whole and parts of them are located in other images. Therefore in such cases, the MVRF will also provide new contextual and visual information in a form of transferable sites based on the objects that are not located in the original image. However, the usefulness of such information

is in this case diminished by the presence of similar native sites. For example, if the transferable site represents a window in other images which is not located in the native image, the presence of similar windows in a native image decrease the contextual impact of such transferable site.

In MVRF framework, every image is processed separately (with the additional information from transferable sites) and there is no comparison of classification between images afterwards embedded in the model itself. This is the main difference between the multi-view approach described in the previous chapter and MVRF. In previous multi-view approach, results were compared, when classifications were available for all images and errors/inaccuracies were corrected based on all results. This approach is not a part of the MVRF model, therefore it can be assumed that results for separate images will differ and can contain inaccuracies that can be removed when results from other images are compared. For this reason, processing based on MVRF could still benefit from results unification, but this approach will not be described in this chapter, as it has been discussed and described in Chapter 6 and can be applied without any modifications.

7.3.1 Façade Elements Detection

In this section, we describe the application of MVRF for detection and labeling of façade elements. The method is based on the gradient projection approach and segmentation of façade into blocks similar to the method described in Section 6.1.1. However, as our goal is to present a more detailed façade labeling, we introduce several modifications to façade segmentation, establish contextual relationships between blocks (sites) and apply MVRF.

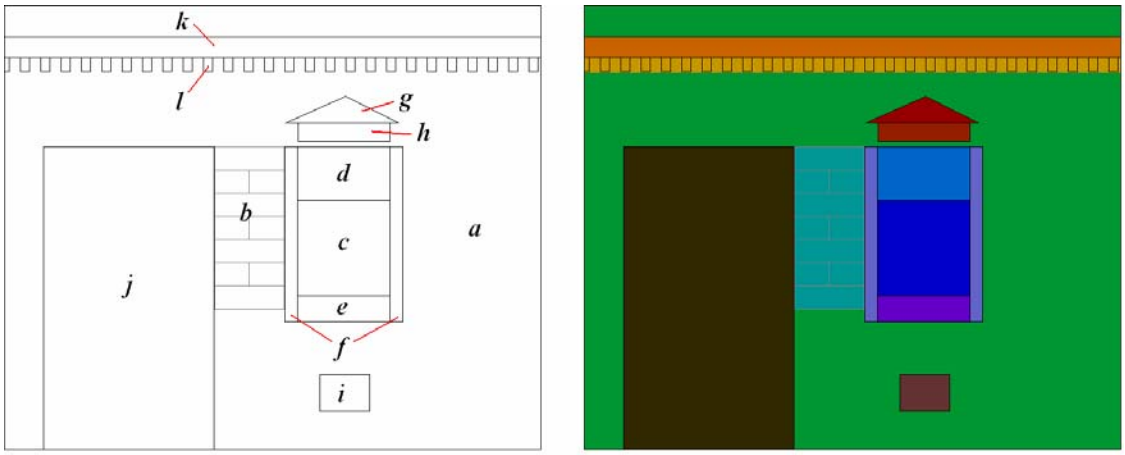


Figure 7.4: Set of classes: *a*) clear façade; *b*) brick façade; *c*) window centre; *d*) window top; *e*) window bottom; *f*) window margins; *g*) arch top; *h*) arch bottom; *i*) basement window; *j*) door; *k*) ledge; *l*) ledge ornament; On the right side, color representation of each class is displayed. In the Result section, processed images have labeled areas overlaid with corresponding class color.

We introduce the following algorithm:

Algorithm 7.1

Input: Separate Façades identified in a multi-view dataset

1. Segment façades into blocks
2. Establish the MVRF model (with blocks as sites) for each façade
3. Project a set of blocks of examined façade from one image to another and mark non-overlapping blocks as transferable
4. Introduce transferable sets of blocks from other images into a MVRF model of an examined façade
5. Approximate optimal solution of MVRF using Belief Propagation
6. Compute final labeling from probability distribution of MVRF

Output: Façade elements detected in a set of blocks

This workflow is novel in using the central method – MVRF in a process. The image matching is established with the help of a sparse 3D point cloud. As our goal is no longer to detect only windows, but also other façade elements, we must account for

differently oriented and non-rectangular objects. We must also consider that not all objects are located in a grid pattern.

Our goal is to label given set of objects (see Figure 7.4). These represent architectural façade elements and may contain several subclasses or variants. We label façade areas into these classes: *clear façade* (façade area with no pattern), *brick façade* (façade area with brick pattern, including rectangular bricks and bricks with inclined edges), *window centre* (central area of window, including window glass, opened window, sectioned window), *window top* (top part of the window, usually un-sectioned area/blinds), *window bottom* (area at the bottom of window, including balcony or ledge), *window margins* (areas on sides of window, included extended window margins, window lesen, or window pilaster), *arch top* (area with inclined edges on top of window, including arch, window pediment, triangular/semicircular ornament), *arch bottom* (area on top of window without inclined edges, usually between window and arch), *basement window* (small windows/doors for basement spaces located below bottom window row), *door* (including doors, portals, door pilasters), *ledge* (uniform area of façade running through entire façade length, such as cornice or molding), *ledge ornament* (ornamented area running through the entire façade length, such as frieze, usually located below ledge)

Horizontal Level Division

Similar to our previous approach, the first step is to divide the façade into horizontal levels with separator lines. To account for different object types, we establish three different gradient images of the façade – horizontal gradient, vertical gradient and inclined gradient image. We use the Roberts Edge detector [Roberts, 1965], but for each gradient image only the corresponding direction of the gradient is computed. Vertical gradient projections are computed for each gradient image – for each line of pixels, a value of projection is computed as a sum of the gradients of all pixels on the horizontal line. Values of all projections from all lines in the façade are considered a vertical gradient projection of the façade. Subsequently, separator lines are established in steps of the vertical projection function. The façade is divided into levels bordered by separator lines. Each level indicates different types of objects based on what directions of gradients are contained within:

Figure 7.5: An example of façade division into set of blocks. Blocks are marked by red lines borders and each represents a uniform patch of the façade. Blue separator lines are detected from horizontal gradient image and indicate the presence of rims. Note that windows are divided into several blocks (window's planes, tops, frames, margins). In this façade, 1231 blocks have been detected, divided between 36 levels.

- Levels with vertical gradient indicate the presence of windows, doors, columns, bricks and other rectangular patterns.
- Levels with horizontal gradient indicate the presence of rims, balconies, strips and other objects horizontally dividing the façade.
- Levels with inclined gradient indicate arches and arched windows

We set the levels to be minimum 4 pixels high, preventing multi responses from the same source (the Roberts edge detector has a 3x3 size kernel). To achieve high precision of projection, we mark façade borders manually for the testing (part of labeled ground truth) and interpolate projection lines between borders. In an automated workflow, this can be achieved by automatic façade rectification and/or vanishing point detection. Subsequently, we process each level separately, without considering information from neighboring levels.

Vertical Division - Blocks

For each level detected in the façade, we perform horizontal gradient projection. For each gradient image (vertical, horizontal, inclined), separate horizontal projection is computed and in steps of horizontal projection function, vertical separator lines are detected. A set of blocks is established such that the top/bottom borders of the blocks are set at the horizontal separator lines of given level and left/right borders are set at any two neighboring vertical separator lines. Blocks thinner than four pixels are removed as noise (due to a 3x3 Robert's kernel) and to disregard multiple responses. In this approach, blocks represent uniform patches of the façade, indicating presence of some façade element. Blocks are also organized into levels, which represent architectural division of the façade (see Figure 7.5). In this approach, significantly larger set of blocks is detected when compared to the method described in Section 6.1.1 (Gradient Projection in a Single Image). This modification is necessary to detect sites for all required façade elements, instead of just detection of windows. This increase in blocks number and the number of classes makes the previous method not applicable, as one color descriptor for the façade area is no longer sufficient. Therefore we use a similar set of descriptors for the blocks as in the previous method, but also include several other features, namely pairwise blocks relations between blocks in a MVRF graph.

Block Descriptors

For a single block, we use similar descriptors as in the Section 6.1.1 based on color, gradient and size. A gradient descriptor is computed from vertical projections inside the block as mean and average values.

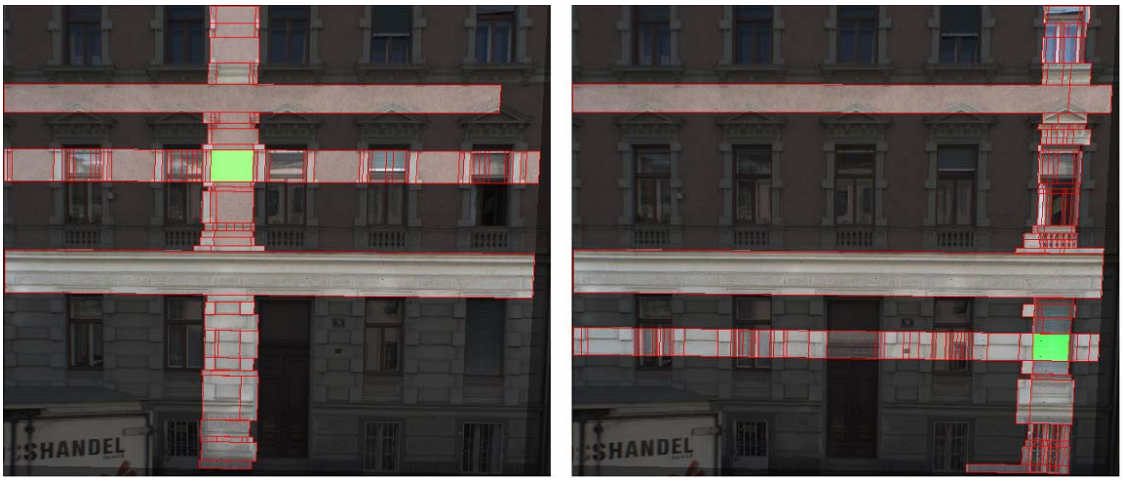


Figure 7.6: Highlighted blocks are the neighborhood of examined block (in green). For the examined block, all blocks in the same level and all blocks at the same vertical position are considered neighborhood. On the left image, a façade block is examined. Notice the row of windows at the level, but no window blocks at vertical position. On the right image, a window block is examined. Notice another window blocks at the vertical positions essential for context examination.

In general, façade blocks contain low values of gradient when compared to a window, or other element blocks. The size of the block is described as ratio of block width/façade width and block height/façade height. Some elements require minimum size (e.g. rims) or maximum size (e.g. window frames) of the blocks. The color descriptor is based on k-means clustering in a CIE-Lab color space [Recky and Leberl, 2010(II)]. Clustering is performed in "a", "b" space. For each cluster, a representative "L" value is computed as a mean of all colors in the cluster. The euclidean distance of CIE-Lab colors is used as a metric.

Pairwise Blocks Relations

To establish pairwise relations between blocks, a graph is created for the block structure. For each block, a graph node is created. Edges are created between each two blocks that are vertically at the same position in the façade (both blocks have pixels at some vertical line going through a façade). Moreover, an edge is also placed between each two blocks at the same level. This setup allows for examinations of relationships between blocks that are at the same vertical position, thus implement relations between

façade elements such as window-arch, but also the column of windows and blocks at the same level that exhibit repetitive patters, such as row of windows.
This graph structure is further used in MVRF model with the same definition of neighborhood (see Figure 7.6).
For each two blocks, that are neighboring each other, we define following descriptors:

Spatial Descriptors

- Vertical position to each other, if the blocks are not in the same level. It is defined if one block is on top of other, or below other.
- Distance relative to façade width, if the blocks are in the same level. The ratio of blocks distance/façade width is computed.
- Distance relative to façade height, if blocks are in different levels. The ratio of blocks distance/façade height is computed
- Number of blocks between them, if they are in the same level
- Number of levels between them, if they are in different levels.

Visual Descriptors

- Color visual similarity based on the CIE-Lab clustering of each block. Visual similarity is a number <0, 1>, where 0 means blocks have no similar clusters in color space and 1 means block have same clusters in color space.
- Gradient visual similarity for the horizontal projection inside the blocks. For inside of each block, an average of horizontal gradient projection is computed. Gradient visual similarity is a number <0, 1> computed from the Euclidean distance of gradient averages in blocks.

Our observation of streetside scenes shows that there is high intra-class variability in many façade elements, even for objects in the same façade. For example, windows in one façade can have different color characteristics due to specular reflections on the glass, open or half open windows, blinds, etc. Also the repetitive patterns can be disrupted by occlusions or incomplete façade projections in the image. These problems can be largely solved by introducing transferable sites (blocks) from other images and

examining inter-class relationships. Both of these solutions can be implemented as a MVRF model.

7.3.2 MVRF Model for Building Façade

Given the graph structure introduced in the previous section, we can establish the MVRF model for the façade element detection. In this model, nodes of the graph (blocks) are considered sites and edges define pairwise relations between sites. Only one graph structure is defined over the entire façade. The neighborhood for each site is defined as all sites at the same level and sites at neighboring levels that have common borders with the examined site. Such parameters define this specific MVRF model as global and the entire examined scene is processed at once.
However, given this definition, only native sites are considered. To complete MVRF model, we have to introduce transferable sites from other images. This require the entire database to be processed for block detection and matched before the classification of specific images can begin. Block detection can be performed in each image separately, as was described in previous sections. This process includes performing vertical and horizontal projections for each façade, detection of blocks and establishing block descriptors. Image matching is performed through spars point cloud. For the points of the façade, where no matched points of point cloud are located, we can perform linear interpolation from three closes matched points.
In this application, we define transferable sites for examined image as blocks in other images, that are located at the same façade, but have no relevant counterpart block detected. This situation can occur in two different cases:

- Transferable blocks are located in a part of façade that is not visible in examined image, i.e. it is either occluded (if occlusion is detected), or trimmed by image border.

- Transferable blocks are at the visible part of the façade, but due to a detection error were not located. This case is harder to identify, as in general borders of the blocks in the same façade are not the same in different images. Division into blocks was designed such that blocks represent uniform patches of the

façade, however edges of the façade elements are seldom sharp enough to define block borders with pixel level precision. This is also the case for horizontal division into levels. Therefore we consider two levels of the same façade that are located in different images to be corresponding to each other if they have at least 2/3 of area composed of corresponding points. This is mostly the case, when one level is at least 2/3 of height of other level and are located approximately at the same position at the façade. Subsequently, we search for transferable sites only in corresponding levels. Likewise, two blocks are considered corresponding to each other if they are at the corresponding level, they intersection is at least 2/3 width of the longer block length. Given this definitions, transferable sites are blocks that are located at some corresponding level, but have no corresponding blocks in the examined image. In general, such blocks are usually patches of the façade, that are somehow bordered by edges (or gradients of edges), but these edges were not detected in examined image.

In both cases, transferable sites import new information into the examined image. In first case, such information is not located anywhere in the examined image, in second case the information was processed differently, resulting in misdetection. Transferring the information from other images provide additional robustness for the MVRF graph model.

In Figure 7.7 (bottom right image), an example of set of transferable sites is displayed. This set represents sections of image that are described by blocks with no equivalent in examined image (top left corner of Figure 7.7). These are either blocks, that are out of view in examined image (blocks on the left side of the façade), or blocks that represents patches not detected in examined image (e.g. frames of windows, parts of arches). Subsequently, set of blocks in examined image (middle left of Figure 7.7) is complemented with the set of transferable blocks.

Figure 7.7: Two images of the same façade. Top row – matched corresponding points, middle row – blocks located in each façade, bottom row – blocks from first façade projected into second façade (on the left) and highlighted set of transferable sites (on the right). Notice the inaccuracies in projection of blocks between image (inclined edges of blocks) in bottom left image caused by imprecise matching of points. However these errors are not significant in computation of transferable sites.

Training and Classification

Our training dataset is composed of mostly single images with no matching available (12 images) and small set of matched images (5 images). As described before, MVRF can be trained on single images, as unary and pairwise features are essentially the same for native and transferable sites. Therefore, single images provide same level of information for the training as matched dataset. The purpose of training is to primary obtain pairwise relations between classes. Visual features are less important, as these are continuously refined in an inference process for the specific façade. From the training set, we obtain representative color, gradient and size signature for each class, but these are applied only for initialization of classification. In a training process we retrieve spatial relations between classes (pairwise feature) for all neighboring blocks (neighborhood defined in MVRF graph). These are described in the section *Spatial Descriptors* and are observed from established graph structure and measurement in examined façade.

The process of classification can be performed on either single image and/or matched images. In case of single images, only native sites are considered and MVRF model is equivalent to DRF model introduced in [Kumar and Herbert, 2006]. If the images are matched and the same façade is located in some subset, we involve transferable sites set in the process of classification, as described in previous sections. The process of classification is performed iteratively. Workflow for classification, together with previously described process of MVRF construction can be described as:

Algorithm 7.2

Input: Separate façades identified and matched in multi-view dataset

1. Segment each façade into a set of blocks
2. For each façade in each image {
   2.1 Define façade as “examined”
   2.2 Project all blocks from examined façade to all matched façades

2.3 Consider any block from matched façades as "transferable" if such block does not have any correspondence in the projected set of blocks

2.4 Use all bocks from examined façade as native sites and all transferable blocks as transferable sites to construct the MVRF model in examined image.

2.5 In the initialization step of iterative process, assign each site (block) initial class based on trained visual features.

2.6 In the iterative step, a belief propagation method is applied to infer MVRF model parameters and establish site labeling. We maximize a site's class posterior probability, based on actual visual descriptors for the class and pairwise relations with the neighboring site's classes. Such relations are propagated as beliefs, exchanged between neighboring sites.

2.7 After each round of belief propagation process, new visual descriptors are computed for each class. This process is similar to one described in Section 3.6.1, subsection *Block Descriptor*, where color clustering was computed for a façade area. We perform similar clustering in the CIE-Lab color space for each class in this step. We also compute a new representative L (illumination) histogram and gradient histogram for each class.

2.8 Steps 3.6. and 3.7 are repeated until there are no more changes in classification for each block or after some number of rounds, to cope with non-convergent situations. Usually, the process converges after 6-10 rounds.

Output: final block classifications are considered labels for areas covered by blocks

In a step 2.6, marginal probabilities (beliefs) $P_b(x_i, m_{ji}(x_i, \boldsymbol{y}))$ are computed for each possible label x_i in each native and transferable site i. Variable $m_{ji}(x_i, \boldsymbol{y})$ is a message from site j to site i, how likely the labeling of site i is x_i given the observation in the image $\boldsymbol{y}$. In case of transferable sites, only local observations from the site are considered. Messages $m_{ji}(x_i, \boldsymbol{y})$ are recomputed after each step according to changes in sites classification. Values m_{ji} are computed from pairwise parameters of the MVRF obtained from training and are based on similarities between neighboring sites. Before the iterative process, we pre-compute color, gradient and positional similarities for neighboring sites and weight beliefs m_{ji} according to these similarities. For example, façade sites in a single row have very similar color features, giving high values of color similarity. Window sites in one row, on the other hand, can have different color

features, but usually have high values of gradient similarity and they are located in uniform spaces, giving high value of position similarity. Application of this observation in MVRF model helps in convergence of iterative process, as detection and classification of one window site reinforce classifications of all neighboring window sites with the same gradient and position similarity. Weights for all similarities are obtained in a training process, as observations between sites. Note that this process is different than the one described in step 3 of iterative process, where the visual descriptors are computed for each class. These visual descriptors are applied in unary classifiers as representatives for each class, giving better visual features for classification.

7.3.3 Results

Our testing dataset consist of 44 matched images. This dataset cover three full building façades and one half façade. A sparse point cloud of 1429 3D points is used to match images. Approximately 800 – 900 points are projected into each image. In the testing process, we compare the number of façade elements to the number of detected elements with applied method. We use overall numbers of elements through entire dataset, as displayed in Table 7.1. For example, total number of 536 "window centre" elements can be observed in all images, that is approximately 12 "window centers" per image. Each façade was processed separately, that is if there were two façades in one image, such image was processed two times (each time for different façade). After running the algorithm, number of detected elements is counted visually. Façade element is defined as detected, if at least 2/3 of its area is labeled with corresponding class.

We examine results for three different scenarios:

- MVRF for single images. In this case, we do not apply image matching and process each image separately. In a MVRF framework, no transferable sites are applied in a process and MVRF model is equivalent to standard DRF.
- MVRF in multi-view without transferable sites in results. This case is standard MVRF approach with images matched and transferable sites applied in a process of classification. However in the final results, only native sites are

considered as area labels. In this case, information from matched images is used in classification process, but only native detections are used for labeling.

- MVRF in multi-view with transferable sites in results. Transferable sites are used for classification in MVRF model as in previous case, however if transferable sites give different labels than native sites in certain image areas, median of labels is preferred in results. This approach solves the problem, when certain façade elements are not detected in native image. If there are two or more transferable sites that give different labels in such areas, these areas are labeled according to the transferable sites instead of native site (this is only applicable for transferable sites that can be projected inside native image area).

Results can be observed in Table 7.1, where each scenario is displayed in separate column and each façade element class in separate row.

Class	**# elements**	**single img.**	**multi/native**	**multi/trans.**
Clear façade	61	61 (100%)	61 (100%)	61 (100%)
Brick façade	54	54 (100%)	54 (100%)	54 (100%)
Window centre	536	485 (90%)	531 (99%)	531 (99%)
Window top	311	270 (87%)	303 (97%)	308 (99%)
Window bottom	300	227 (76%)	273 (91%)	288 (96%)
Window margin	683	572 (83%)	618 (90%)	654 (95%)
Arch top	199	176 (88%)	189 (95%)	192 (96%)
Arch bottom	199	184 (92%)	194 (94%)	194 (94%)
Basement window	121	98 (81%)	115 (95%)	117 (97%)
Door	34	32 (94%)	33 (97%)	33 (97%)
Ledge	90	90 (100%)	90 (100%)	90 (100%)
Ledge ornament	34	32 (100%)	34 (100%)	34 (100%)

Table 7.1: Results for the MVRF application. "# elements" displays the overall number of each class for entire dataset (44 images). "single img." displays detected elements in MVRF single image scenario, "multi/native" displays results for multi-view scenario with only native sites in results and "multi/trans." display results for multi-view scenario with transferable sites labels in results. Numbers displayed are the detected façade elements in all images of dataset.

Elements can be not detected in two cases a) element was not labeled with appropriate class and b) element was not correctly segmented in set of blocks. In first case an error can be corrected in both multi/native and multi/trans scenario, as correct label can be identified through additional context from transferable sites. In second case, only multi/trans scenario can yield correct labeling, as missing site is not located in a set of native sites and the labeled area must be overlaid with transferable site. This is mostly evident for not well defined areas, such as "window bottom" or "window margin" (as it is often uncertain, where is the border between "window centre" and "window bottom/margin"), where the improvement from multi/native to multi/trans is at 5%.

From experiments, we can observe that the transition from a single-view to a multi-view in MVRF model helps in detection of façade elements that have less well defined borders (window margins window bottoms), but also correct some mislabeling of similar classes (window centre/window top). These corrections are particularly effective, when blocks do not cover borders between such classes correctly. For example, blocks that are located at window top also cover parts of window centers. In that case, window top can be labeled as centre (as block have similar visual characteristic and tops/centers are usually in the same neighborhoods in graphs with similar contextual cues). Extending this case in multi-view can provide better neighborhoods (with transferable sites) for such façade element to be labeled correctly.

Difference in multi/native and multi/trans scenario can be observed primary in cases, when class was detected correctly, but blocks does not cover entire objects (blocks with other classes interfere into the area of façade elements). In this case, element is not considered detected if at least 2/3 of its area is not labeled correctly. When transferring labels from transferable sites in multi/trans. scenario, correctly labeled area of façade element may extend over 2/3 object area and change status from not-detected to detected. Two examples of results can be observed in Figure 7.8.

The precision of placement of elements can be evaluated as a ratio of an area of element not correctly labeled (false positive and false negative) to an area of ground truth. As the dataset contained 2622 elements in total, the hand labeling of entire set was not feasible due to time constraint. Instead, we labeled only "window centre" class in one façade. The average precision of placement was 88% in multi/trans scenario. This is consistent with Section 6.2.1, as both methods use the same segmentation of

façade into blocks. The difference is in correction of labeling through transferable sites. In geometric measure, this would mean that for example for a window with 80x135 cm dimensions (10800 cm^2 area), 1296 cm^2 area would be mislabeled in average.

7.4 Discussion on MVRF application

In Figure 7.7, top row display part of projected point cloud. Notice that the point cloud is sparse and there are only few corresponding points located at façade elements (8-15 points per window). This number is sufficient for matching, as we can apply the assumption of façade planarity and interpolate/extrapolate any point of the façade for matching. However the sparsity of point cloud prevents examination of 3D structure for classification. In MVRF model, 3D structure of the scene can be directly used as an observation and provide cues for classification. For example, balconies, window frames or arches have specific signature in 3D façade relief that can be used as a feature. This would however require densier point cloud to provide more precise information about local changes in façade planarity.

The process of classification is constructed such that in each step, it is equivalent with previous applied methods. In the initialization step, where only visual features are applied, the classification is equivalent to a gradient projection method, with no application of context. This can be compared with Nevatia's method described in [Lee and Nevatia, 2004], with the exception that not only gradient descriptors are used, but also color descriptors. As described in previous section, Nevatia's method can be successfully applied for simple façades and does not cope with multi-view datasets. Introduction of graph structure and MVRF model into the process, but for single images is equivalent to Discriminative Random Field (DRF) model, as no transferable sites are present. In this step, the context between classes is considered, but no multi-view data is utilized.

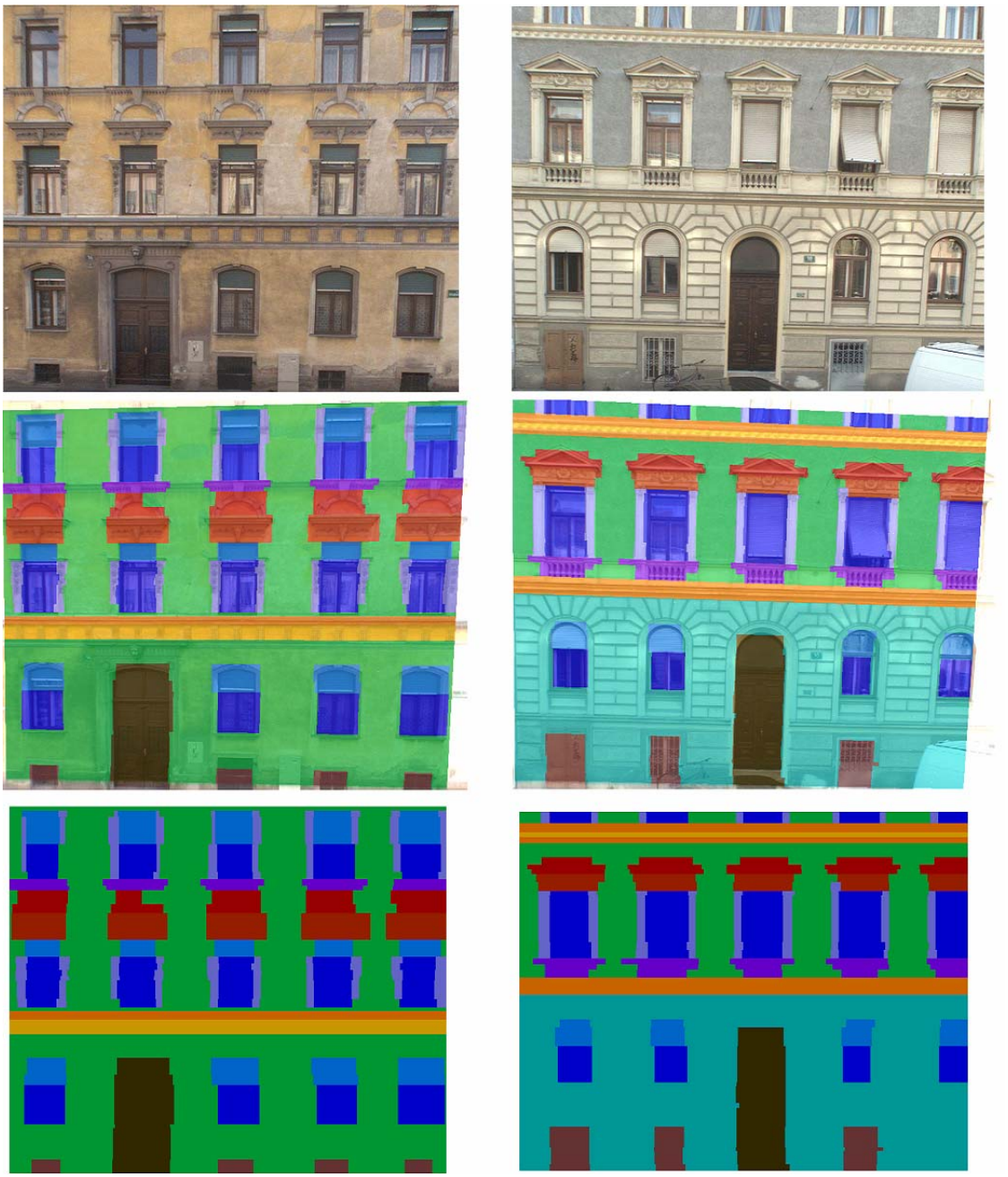


Figure 7.8: Results for two building façades. Top row – original images, middle row – images overlaid with class colors (for color codes, see Figure 7.4), bottom row – rectified façade schemes with class colors. In the classification process, each block was assigned a class label and the corresponding color for each class in the result images represents the detection of the façade element.

This can prove beneficial for single images, as it provides strong contextual cues for façade elements, but misdetections of sites cannot be corrected without multi-view. Also with the introduction of transferable sites in full MVRF model, we get the benefit of additional data that was trimmed of the image, or occluded.

Set of classes defined in this section encompass a large variety of objects per class. For example, class “window margins” may indicate the presence of window pilaster (slightly projecting column build into a façade), extended window frame (area of façade next to the window with different color), ornament located next to the window, etc. In our method, such objects are labeled into one encompassing class and indicated by rectangular bounding box. This type of output presents a façade analysis, but may not by directly suitable for some façade modeling applications. For example, if the method of façade modeling is based on shape grammars, additional detection of shapes has to be performed inside bounding boxes to better identify objects that are present.

Chapter 8

Conclusion

8.1 Façade Processing using Context and Multi-View

In chapters 4, 5 and 6 we described a workflow for analysis of streetside dataset. Every method described can be viewed as a part of complete study on street-side image processing, presenting an automated and robust workflow. We examined the context of a scene projected into an image and how the context influence results in semantic segmentation. This segmentation was designed to identify basic classes in the image and their context using graph model namely Discriminative Random Fields. In our method, the context of an entire image is considered, not only a local context near the examined object. We described a method for façade separation, identifying individual façades in the images. Subsequently, we introduced a method for façade analysis based on a gradient projection.

In addition to introducing several improvements and critical modifications to state-of-the-art methods, we extended each method into a multi-view. For every method in this chapter, we examined single-view and multi-view scenarios separately. This approach makes the application of method more robust, as it is making no assumptions about the availability and the level of image matching. Subsequently, our methods can be applied to single image, multiple images when no matching is available, sparsely matched imaged dataset and finally, densely matched dataset, reflecting different

image matching techniques (manual matching, laser scanner, projections into aerial wire-frames, 3D point clouds obtained through epipolar geometry…). We also examined two principal approaches when dealing with multi-view data interpretation – interpretation first, matching second and matching first, interpretation second, addressing different precision level in matching techniques. We presented novel approaches and methods in each step of the process, examining the effect of context, multi-view and presenting new solutions for existing problems. For each involved method and given result, we present an extensive discussion about the advantages/disadvantages and alternatives for such method.

In Chapters 4, 5, 6 and 7 we introduced a number of novel methods to improve upon existing state-of-the art:

- Visual Similarity: a trainable measure which can give a probability value if two segments belong to a same class, based on their colors. This measure is applied to iteratively segment a real world objects in an image.
- Global DRF model for classification. Instead of number of graph structure as local models, we implemented a single graph structure for entire image, encompassing all detected objects for context examination.
- We implemented a semantic segmentation for a multi-view scenario.
- Semantic segmentation as a prior knowledge for façade separation method. In this case, repetitive patterns are only detected in façade class areas.
- Iterative segmentation with Visual Similarity measure for façade separation
- Refining of individual façades in multi-view scenario.
- Segmentation of façades into a set of blocks using gradient projection method.
- k-means clustering in CIE-Lab color space descriptor.
- Windows detection for complex façades.
- Windows detection in multi-view.
- Multi-View Random Fields

8.2 Multi-View Random Fields

In Chapter 7, we utilized our research on multi-view and context and combined it into a new model in Random Field framework. When considering multi-view scenario in a dataset of images, one must decide between two principally different approaches. As

each image is a separate unit of information, it is possible to process each image separately and after that, merge results through image matching. This is a straightforward process and as such, it is a first approach in multi-view (as it does not require any modification for algorithms designed for single-view scenario). This approach was tested in previous chapters and applied for three different algorithms in streetside datasets.

The second approach is to merge information from images before vision algorithms are applied. However before an application, several problems have to be solved. First, each image is a separate unit and as such, information from the matched images has to be put in context with the native image. Second, an important decision is the volume of information to be transferred between images. One can decide to merge all information from all matched images into one, but in general, this would make computational complexity of the processing algorithm often unfeasible. To cope with this problem, we can assume that most of the information in matched dataset is highly redundant, thus will not provide new data. Therefore, we can transfer only new, or low redundant information to maximize the usefulness of transferred data. In this chapter, we considered both these problem and introduced a Multi-View Random Field model. We used a native graph structure of Random Field models to establish context with transferred data and introduced the term “transferable site”, which represent a unit of information from matched images that is most useful in processing of a native image (as it is missing in the native image structure). We built a MVRF framework on the Discriminative Random Field model, as it allows application of more complex contextual relationships that is applicable on a global level of the image. In this chapter, we introduced mathematical model for MVRF and presented an example of its application. We applied MVRF in a highly contextual system of building façade for detection of façade elements.

Together with methods introduced in previous chapters, we presented a complete workflow for streetside images processing. Starting with single streetside image, or matched dataset, we identify principal areas in the image, separate building façades and process façades into set of façade elements. We introduced new methods and approaches in each step of the workflow, examined different multi-view scenarios and applied context at each level. We compared our approach with related methods in the vision field and improved upon state-of-the-art.

List of Publications

Recky, M. Fast Area-Based Stereo Algorithm. *CESCG, Proceedings of the 10th Central European Seminar on Computer Graphics*, pp. 95–101, 2006

Recky, M., Leberl, F. Semantic Segmentation of Street-Side Images. *Proceedings of the Annual OAGM Workshop*, pp. 271–282, 2009

Recky, M., Leberl, F. Multi-View Image Interpretation for Street-Side Scenes. *Computer Graphics and Imaging (CGIM)*, Innsbruck, Austria, pp. 48-54, 2010

Recky, M., Leberl, F. Windows Detection Using K-means in CIE-Lab Color Space. *Internation Conference on Pattern Recognition (ICPR),* Istanbul, Turkey, pp. 356-360, 2010

Recky, M., Leberl, F. Window Detection in Complex Façades. *European Workshop on Visual Information Processing*, pp. 220-225, 2010

Recky, M,. Wendel, A., Leberl, F. Façade Segmentation in a Multi-View Scenario. *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pp. 358-365, 2011

Recky, M., Leberl, F., Ferko, A. Multi-View Random Fields and Street-Side Imagery. *International Conference in Cetral Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, unpaged DVD, ISSN 1213-6980, 2012

List of Acronyms

CRF	Conditional Random Fields
CS	Crowd Sourced
DRF	Discriminative Random Fields
GIS	Geographic Information System
GPR	Ground-Penetrating Radar
GPS	Global Positioning System
ICG	Institute for Computer Graphics and Vision
IS	Industrial System
LiDAR	Light Detection and Ranging
LOD	Level of Detail
MRF	Markov Random Fields
MVRF	Multi-View Random Fields
MAV	Micro Air Vehicle
RF	Random Fields
SfM	Structure from Motion
SIFT	Scale-Invariant Feature Transform

List of Figures

List of Tables

List of Algorithms

References

[Academic, 2012] Top conferences in Computer Vision [online] http://academic.research.microsoft.com/RankList?entitytype=3&topDomainID=2&subDomainID=11 (Accessed on July 7, 2012)

[Agarwal et al., 2009] Agarwal, S., Snavely, N., Simon, I., Seitz., S. M., and Szeliski, R. Building Rome in a Day. *International Conference on Computer Vision (ICCV)*, pp. 105-112, 2009

[Agarwal et al., 2010] Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S. M., Szeliski, R. Reconstructing Rome. *Computer*, pp. 40-47, June, 2010

[Andreetto et al., 2007] Andreetto, M., Zelnik-Manor, L., Perona, P. Non-Parametric Probabilistic Image Segmentation. *International Conference on Computer Vision (ICCV)*, pp. 1-8, 2007

[Anguelov et al., 2010] Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., Weaver, J. Google Street View: Capturing the World at Street Level. *Computer*, pp. 32-38, June, 2010

[ArcGIS, 2012] ArcGIS Resource Center. Shadow Maps [online] http://blogs.esri.com/esri/arcgis/2011/02/28/shadow-maps/ (Accessed June 7, 2012)

[Ashton, 2009] Ashton, K. That 'Internet of Things' Thing, RFID Journal [online] http://www.rfidjournal.com/article/view/4986 (Accessed on July 3, 2012)

[Autodesk, 2012] Autodesk - 3D Design & Engineering Software for Architecture, Manufacturing, and Entertainment [online] http://usa.autodesk.com/ (Accessed on January 21, 2012)

[Autodesk, 2012(II)] Infrastructure Modeler - Urban Planning Design Software - Autodesk [online] http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=17276659 (Accessed on January 21, 2012)

[Bar and Aminoff, 2003] Bar, M., and Aminoff, E. Cortical analysis of visual context. *Neuron*, pp. 347-358, 2003

[Bar et al., 2006] Bar, M., Kassam, K. S., Ghuman, A. S., Boshyan, J., Schmid, A. M., Dale, A. M., Hämäläinen, M. S., Marinkovic, K., Schacter, D. L., Rosen, B. R., and Halgren, E. Top-down facilitation of visual recognition. Proceedings of the National Academy of Science, pp. 449-454, 2006

[Barbera et al., 2008] Barbera, D., Millsa, J., Smith-Voysey, S. Geometric validation of a ground-based mobile laser scanning system. *Journal of Photogrammetry and Remote Sensing (ISPRS)*, pp. 128–141, 2008

[Becker, 2011] Becker, S. Towards Complete LOD3 Models – Automatic Interpretation of Building Structures. *Photogrammetric Week '11*, pp. 39-56, 2011

[Berry, 2012] Berry, J.K. A Brief History and Probable Future of Geotechnology [online] http://www.innovativegis.com/basis/Papers/Other/Geotechnology/Geotechnology_history_future.htm (Accessed on July 8, 2012)

[Besag, 1972] Besag. J. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Royal Statistical Society B*, pp. 192-236, 1974

[Besag, 1986] Besag, J. On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society, Series B, pp. 259-302, 1986

[Bleyer et al., 2009] Bleyer, M., Gelautz, M., Rother, C., and Rhemann, C. A stereo approach that handles the matting problem via image warping. *Computer Vision and Pattern Recognition (CVPR)*, pp. 501-508, 2009

[Bing Maps, 2011] Bing Maps [online] http://www.bing.com/maps/ (Accessed on November 10, 2011)

[Bolter and Leberl, 2002] Bolter, R., Leberl, F. Virtual Habitat: From Multiple View Interferometric Radar to Building Models. *4th European Conference on Synthetic Aperture Radar (EUSAR)*, pp. 435-437, 2002

[Bujnak et al., 2008] Bujnak, M., Kukelova, Z., Pajdla, T. A general solution to the p4p problem for camera with unknown focal length. *Computer Vision and Pattern Recognition (CVPR),* pp. 1-8. Anchorage, USA, June, 2008

[Carbonetto et al., 2004] P. Carbonetto, N., de Freitas, and Barnard, K. A statistical model for general contextual object recognition. *European Conference on Computer Vision (ECCV)*, pp. 350-362, 2004

[Census, 2012] Urban and Rural Classification. United Stated Census Bureau [online] http://www.census.gov/geo/www/ua/urbanruralclass.html (Accessed on July 4, 2012)

[Chari et al., 2008] Chari, V., Singh, J. M., and Narayanan, P. J. Augmented reality using over-segmentation. *The National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pp. 125-132, 2008

[Cheng and Bouman, 2001] Cheng, H., Bouman, C. A. Multiscale Bayesian Segmentation using a Trainable Context Model. *IEEE Trans. on Image Processing*, pp. 511-525, 2001

[Coorg and Teller, 1999] S. Coorg and S. Teller, Extracting textured vertical façades from controlled close-range imagery. *Computer Vision and Pattern Recognition (CVPR)*, pp. 625-632, 1999

[Correa, 2009] Correa, F.R. Semantic mapping with image segmentation using Conditional Random Fields. *International Conference on Advanced Robotics (ICAR),* pp. 1-6, 2009

[Cornelis et al., 2006] Cornelis, N., Leibe, B., Cornelis, K., Van Gool, L. 3D City Modeling using Cognitive Loops. *3D Data Processing, Visualization and Transmission, Third International Symposium*, pp. 9-16, 2006

[Cross and Jain, 1983] Cross, G. C., Jain, A. K. Markov Random Field Texture Models. *IEEE Trans Pattern Anal. Machine Intelligence*, pp. 525-39. 1983

[CVG, 2012] CVG @ ETHZ [online] http://www.cvg.ethz.ch/ (Accessed on January 21, 2012)

[Čech and Šára, 2008] Čech, J., Šára, R. Windowpane Detection based on Maximum Aposteriori Probability Labeling. *Proceedings of the 12th International Workshop on Combinatorial Image Analysis*, pp. 3-11, 2008

[Dalal and Triggs, 2005] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition (CVPR),* pp. 886-893, 2005

[Daniels, 2004] Daniels, D.J. Ground Penetrating Radar. *Knoval (Institution of Engineering and Technology)*. pp. 1–4. ISBN 978-0-86341-360-5., 2004

[Daure, 2006] Duarte, A., Sánchez A., Fernández, F., Montemayor, A. S. Improving Image Segmentation Quality through Effective Region Merging using a

Hierarchical Social Metaheuristic. *Pattern Recognition Letters*, 27, pp. 1239-1251, 2006

[Debevec et al., 1996] Debevec, P. E., Taylor, C. J., Malik, J. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-based Approach. *In SIGGRAPH*, pp. 11-20, 1996

[Delaunay, 1934] Delaunay, B. Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, pp. 793–800, 1934

[Dorninger and Nothegger, 2007] Dorninger, P., Nothegger, C. 3D segmentation of unstructured point clouds for building modelling. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, pp. 191-196, 2007

[Dziewonski, 2004] Dziewonski, A.M. Global seismic tomography: What we really can say and what we make up. *Geol Soc Am Penrose Conference*, 2012.

[Esri, 2012] Esri - GIS Mapping Software [online] http://www.esri.com/ (Accessed on January 21, 2012)

[Esri, 2012(II)] Esri CityEngine | 3D Modeling Software for Urban Environments [online] http://www.esri.com/software/cityengine (Accessed on January 21, 2012)

[eTRIMS, 2012] eTRIMS [online] http://www.ipb.uni-bonn.de/projects/etrims/index.html (Accessed on January 21, 2012)

[Fabritius et al., 2008] Fabritius, G., Krassnigg, J., Krecklau, L., Manthei, C., Hornung, A., Habbecke, M., Kobbelt, L. City Virtualization. 5. *Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, pp. 243-246, 2008

[Feng et al., 2002] Feng, X., Williams, C. K. I., Felderhof, S. N. Combining belief networks and neural networks for scene segmentation. *IEEE Trans on Pattern Anal. Machine Intelligence*, 244, pp. 467-483, 2002

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P., Huttenlocher, D. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167-181, 2004

[Ferreira and Bernardino, 2006] Ferreira, J., Bernardino, A. Acquisition of 3D regular prismatic models in urban environments from DSM and orthoimages.

Symposium on Computational Modeling of Objects Represented in Images (ISR), 2006

[Fisher, 1972] Fisher, T. An overview of the Canada Geographic Information system (CGIS). *Lands Directorate Environment Canada*, 1972

[Fischer and Bolles, 1981] Fischler, M.A., Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, pp.381-395, 1981

[Flickr, 2012] Flickr - Photo Sharing [online] http://www.flickr.com/ (Accessed on July 2, 2012)

[Frahm et al., 2010] Frahm, J. M., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn1, E., Clipp, B., Lazebnik, S., Pollefeys, M. Building Rome on a Cloudless Day. *European Conference on Computer Vision (ECCV)*, pp. 368-381, 2010

[Friedman et al., 1977] Friedman, J. H., Bentley, J. L., Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software* (3), pp. 209-226, 1977

[Fruh and Zakhor, 2001] Fruh, C., Zakhor, A. 3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans. *Computer Vision and Pattern Recognition (CVPR)*, pp. 31-38, 2001

[Fruh et al., 2005] Fruh, C., Jain, S., Zakhor, A. Data processing algorithms for generating textured 3D building façade meshes from laser scans and camera images. *International Journal of Computer Vision*, 61(2), pp. 159-184, 2005

[Funamizu, 2012] Funamizu, M. Future of Internet Search: Mobile version [online] http://petitinvention.wordpress.com/2008/02/10/future-of-internet-search-mobile-version/ (Accessed June 20, 2012)

[Galileo, 2012] European Space Agency - Galileo Navigation [online] http://www.esa.int/esaNA/galileo.html (Accessed on July 3, 2012)

[Glennie, 2009] Glennie, C. Kinematic Terrestrial Light-Detection and Ranging System for Scanning. *Transportation Research Record: Journal of The Transportation Research Board*, Vol. 2105, pp. 135-141, 2009

[Godchild, 2002] Godchild, M.F. GIS and Spatial Statistics: One World View or Two?, Spatial Statistics: Integrating Statistics, GIS, and Statistical Graphics, *University of Washington*, 2002

[Google, 2012] Research at Google [online] http://research.google.com/index.html (Accessed on January 21, 2012)

[Google, 2012(II)] Street View for Google Maps [online] http://maps.google.com/help/maps/streetview/ (Accessed on July 4, 2012)

[Google, 2012(III)] Google Maps [online] https://maps.google.sk/maps (Accessed on July 4, 2012)

[Gösele et al., 2010] Gösele, M., Ackermann, J., Fuhrmann, S., Klowski, R., Langguth, F., Mücke, P., Ritz, M. Scene Reconstruction from Community Photo Collections. *Computer*, pp. 48-53, June 2010

[GRAIL, 2012] GRAIL: Graphics and Imaging Laboratory [online] http://grail.cs.washington.edu/ (Accessed on January 21, 2012)

[Grubner et al., 2005] Gruber-Geymayer, B. C., Klaus, A., Karner, K. Data Fusion for Classification and Object Extraction. *International Archives of Photogrammetry and Remote Sensing*, XXXVI(3), pp. 125–130, 2005

[Gruber, 2011] Gruber, M. The Evolution of Vexcel's Camera Calibration and Image Processing Technology. *JACIE 2011 Civil Commercial Imagery Evaluation Workshop*, Boulder, Colorado. 2011

[Haala et al., 2008] Haala, N., Peter, M., Cefalu, A., Kremer, J. Mobile lidar mapping for urban data capture. *International Conference on Virtual Systems and Multimedia*, pp. 95-100, 2008

[Hammersley and Clifford, 1971] Hammersley, J. M., Clifford, P. Markov field on finite graph and lattices. Unpublished, 1971

[Harris and Stephens, 1988] Harris, C., Stephens, M. A combined corner and edge detector. *In Proceedings of the Alvey Vision Conference,* Volume 15, pp. 147-151, 1988

[Hartley, 1997] Hartley, R. In Defense of the Eight-Point Algorithm. IEEE Transaction on Pattern Recognition and Machine Intelligence 19 (6), pp. 580–593, 1997

[Hartley and Zisserman, 2004] Hartley, R., Zisserman, A. *Multiple View Geometry in Computer Vision. 2ed*. Cambridge University Press, 2004, ISBN: 0521540518

[Hays and Efros, 2008] Hays, J., Efros, A.A. IM2GPS: estimating geographic information from a single image. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, 2008

[Heesch and Petrou, 2007] Heesch, D., Petrou, M. Non-Gibbsian Markov Random Field Models for Contextual Labelling of Structured Scenes. *British Machine Vision Conference (BMVC),* pp. 930-939, 2007

[Heesch et al., 2008] Heesch, D., Tan, R., Petrou, M. Context First. *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 97-106, 2008

[Heitz and Koller, 2008] Heitz, G., Koller, D. Learning Spatial Context: Using Stuff to Find Things. *European Conference on Computer Vision (ECCV).* pp. 30-43, 2008

[Hernandez and Marcotegui, 2009] Hernandez, J., Marcotegui, B. Morphological segmentation of building façade images. *International Conference on Image Processing (ICIP)*, pp. 4029-4032, 2009

[Hirschmüller, 2006] Hirschmüller, H. Stereo Vision in Structured Environments by Consistent Semi-Global Matching. *Computer Vision and Pattern Recognition (CVPR),* pp. 2386-2393, 2006

[Hirschmüller and Sacherstein, 2009] Hirschmüller, H., Scharstein, D. Evaluation of Stereo Matching Costs on Images with Radiometric Differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9), pp. 1582–1599, 2009

[Hoeim et al., 2005] Hoeim, D., Efros, A. A., Herbert, M. Geometric context from a single image. *In Computer Vision, Tenth IEEE International Conference*, volume 1, pp. 654–661, 2005

[Hoiem, 2007] Hoiem, D. Seeing the World Behind the Image, Spatial Layout for 3D Scene Understanding. *Robotics Institute, Carnegie Mellon University,Ddoctoral dissertation*, 2007

[Hohmann et al., 2008] Hohmann, B., Krispel, U., Havemann, S., Fellner, D. CITYFIT – High-Quality Urban Reconstruction by Fitting Shape Grammars to Image and

derived Textured Point Clouds. *Int. Computer Vision Summer School (ICVSS)*, 2008

[Horn, 1990] Horn, B. K. Recovering baseline and orientation from essential matrix. ISBN 978-0-521-54051-3, 1990

[Huang et al., 2010] Huang, Y., Kaiqi, H., Tieniu, T. A heuristic deformable pedestrian detection method. *10th Asian conference on Computer vision (ACCV)*, pp. 542-553, 2010

[IBTimes, 2012] International Business Times. YouTube, Other Video Sites Taking Over the Internet's Traffic Load [online] http://www.ibtimes.com/articles/187217/20110726/youtube-ott-providers-online-video-mobile-bandwidth.htm (Accessed on July 3, 2012)

[ICG, 2012] Institute for Computer Graphics and Vision (ICG) [online] http://www.icg.tu-graz.ac.at/ (Accessed on January 21, 2012)

[ICG TUGRAZ, 2012] ICG TUGRAZ, AR SCOUTING - AN EXPERT USER INTERFACE FOR OUTDOOR DATA COLLECTION [online] http://studierstube.icg.tugraz.at/ipcity/scouting.php (Accessed June 20, 2012)

[iMore, 2012] Apple acquired 3D mapping company, C3 Technologies? [online] http://www.imore.com/2011/10/29/apple-acquires-3d-mapping-company-3c-technologies/ (Accessed on January 21, 2012)

[Iris, 2012] Conference Calendar for Computer Vision, Image Analysis and Related Topics [online] http://iris.usc.edu/Information/Iris-Conferences.html#deadlines (Accessed on July 7, 2012)

[Irschara et al., 2007] Irschara, A., Zach, C., Bischof, H. Towards wiki-based dense city modeling. *International Conference on Computer Vision (ICCV)*, pp. 1-8, 2007

[Irschara, 2011] Irschara, A. *Image Based 3D Reconstruction for a Virtual Habitat*. PhD thesis, Graz University of Technology, 2011

[Jahangiri and Petrou, 2008] Jahangiri, M., Petrou, M. Fully Bottom-Up Blob Extraction in Building Façades. *9th International Conference on Pattern Recognition and Image Analysis: New Information Technologies*, 2008

[James, 1806] James, H. Photo-zincography, 2nd edition. *Forbes and Bennett*, 1806

[Jiten and Merialdo, 2006] Joakim, J., Bernard, M. Semantic Image Segmentation with a Multidimensional Hidden Markov Model. *Advances in Multimedia Modeling*, pp. 616-624, 2006

[Kim et al., 2008] Kim, G., Huber, D., Hebert, M. Segmentation of Salient Regions in Outdoor Scenes Using Imagery and 3-D Data. *IEEE Workshop on Application of Computer Vision WACV/VISN*, pp. 1-8, 2008

[Kimchi, 2009] Kimchi, G. History and lessons from Microsoft Virtual Earth. *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 2-9, 2009

[Kindermann and Snell, 1980] Kindermann, R., Snell, J.L. Markov Random Fields and Their Applications. *American Mathematical Society*, ISBN 0-8218-5001-6, 1980

[Klopschitz et al., 2010] Klopschitz, M., Irschara, A., Reitmayr, G., Schmalstieg, D. Robust Incremental Structure from Motion. *Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*. 2010

[Kluckner and Bischof, 2010] Kluckner, S., Bischof, H. Large-Scale Aerial Image Interpretation using a Redundant Semantic Classification. *Proceedings International Society for Photogrammetry and Remote Sensing, Photogrammetric Computer Vision and Image Analysis*, pp. 66-74, 2010

[Kluckner et al., 2011] Kluckner, S. , Pock, T., Bischof, H. Exploiting Redundancy for Aerial Image Fusion using Convex Optimization. *Proceedings Annual Symposium German Association for Pattern Recognition*, pp. 303-312, 2010

[Kluckner, 2011] Kluckner, S. *Semantic Interpretation of Digital Aerial Images Utilizing Redundancy, Appearance and 3D Information*. PhD thesis, Graz University of Technology, 2011

[Koch and Heipke, 2005] Koch, A., Heipke, C. Semantically Correct 2.5D GIS Data - the Integration of a DTM and Topographic Vector Data. *Developments in Spatial Data Handling*, pp. 509-526, 2005

[Kolbe et al., 2009] Kolbe, T., Nagel, C., Stadler, A. CityGML-OGC Standard for Photogrammetry? *Proceedings of the Photogrammetric Week '09, Wichmann-Heidelberg Publishers*, ISBN 978-3-879-7-483-9, pp. 265-277, 2009

[Kolmogorov and Wainwright, 2005] Kolmogorov, V. N., Wainwright, M. J. On optimality of tree-reweighted max-product message passing. *In Proc. Uncertainty in Artificial Intelligence*, pp. 480-487, 2005

[Korč and Förstner, 2008] Korč, F., Förstner, W. Approximate Parameter Learning in Conditional Random Fields: An Empirical Investigation Rigoll. *Pattern Recognition*. Springer, DAGM (5096), pp. 11-20, 2008

[Kumar, 2005] Kumar, S. Models for Learning Spatial Interactions in Natural Images for Context-Based Classification. *Carnegie Mellon University*, Pittsburg 2005

[Kumar and Hebert, 2005] Kumar, S., Hebert, M. A hierarchical field framework for unified context-based classification. *International Conference on Computer Vision (ICCV)*, pp. 1284-1291, 2005

[Kumar and Herbert, 2006] Kumar, S., Herbert, M. Discriminative random fields. *International Journal of Computer Vision*, 68(2), pp. 179–201, 2006

[LabelMe, 2011] LabelMe [online] http://labelme.csail.mit.edu/ (Accessed September 6, 2011)

[Ladstätter and Gruber, 2008] Ladstätter, R., Gruber, M. Geometric Aspects Concerning the Photogrammetric Workflow of the Digital Aerial Camera UltraCamX. *International Archives of Photogrammetry and Remote Sensing*, XXXVII(1), pp. 521–525, 2008

[Lafferty et al., 2001] Lafferty, J., McCallum, A., Perreira, F. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. *In Proc. Int. Conf. on Machine Learning* (*ICML*), pp. 282-289, 2001

[Lafarge et al., 2008] Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M. Automatic Building Extraction from DEMs using an Object Approach and Application to the 3D-city Modeling. *International Journal of Photogrammetry and Remote Sensing*, 63(3), pp. 365–381, 2008

[Lato, 2010] Lato, M. GEOTECHNICAL APPLICATIONS OF LIDAR PERTAINING TO GEOMECHANICAL EVALUATION AND HAZARD IDENTIFICATION. *Geological Sciences & Geological Engineering Graduate Theses*, 2010

[Lee and Nevatia, 2004] Lee, S. C., Nevatia, R. Extraction and integration of window in a 3d building model from ground view images. *Computer Vision and Pattern Recognition (CVPR)*, Vol.2, pp. 112-120, 2004

[Leberl, 2003] Leberl, F. Models and visualizations in a virtual habitat. *Spring conference on Computer graphics (SCCG)*, pp. 35-38, 2003

[Leberl et al., 2003] Leberl, F., Gruber, M., Ponticelli, M., Bernoegger, S., Perko, R. The Ultracam Large Format Aerial Digital Camera System. *In Proceedings of the ASPRS Annual Convention*, pp. 5-9, 2003

[Leberl, 2008] Leberl, F. Locational Awareness of the Internet. *Visual Information Systems*, vol. 5188, 2008

[Leberl and Gruber, 2009] Leberl, F., Gruber, M. 3d-Models of the Human Habitat for the Internet. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp)*, pp. 7-15, 2009

[Leberl, 2010] Leberl, F. Time for Neo-Photogrammetry? *GIS Development vol.14*, pp. 22-24, 2010

[Leberl et al., 2010] Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S., Wiechert, A. Point Clouds: Lidar versus 3D Vision. *Photogrammetric Engineering and Remote Sensing*,Vol. 76, No. 10, pp. 1123-1134, 2010

[Leberl et al., 2010(II)] Leberl, F., Bischof, H., Pock, T., Irschara, A., Kluckner, S., Aerial Computer Vision for a 3D Virtual Habitat. *Computer*, pp. 24-31, June, 2010

[Leonardis et al., 2000] Leonardis, A., Solina, F., Bajcsy, R . Confluence of computer vision and computer graphics. *NATO science series, Kluwer Academic Publishers*, ISBN 0-7923-6612-3, 2000

[Liebowitz and Zisserman, 1998] Liebowitz, D., Zisserman, A. Metric rectification for perspective images of planes. *Computer Vision and Pattern Recognition (CVPR)*, pp. 482-488, 1998

[Lionel et al., 2011] Lionel, H., Gim Hee, L., Fraundorfer, F., Pollefeys, Marc. Real-time photo-realistic 3D mapping for micro aerial vehicles. *Intelligent Robots and Systems (IROS)*, pp. 4012-4019, 2011

[Liping and Wentao, 2009] Liping, Y., Wentao, Y. Pedestrian Detection Fusion Method Based on Mean Shift. *Second International Conference on Machine Vision (ICMV)*, pp. 204-207, 2009

[Loke and Barker, 1996] Loke, M.H., Barker, R.D. Practical techniques for 3D resistivity surveys and data inversion. *Geophysical prospecting*, pp. 499-523, 1996

[Lynx, 2012] Lynx Mobile Mapper [online] http://www.optech.ca/lynx.htm (Accessed on July 3, 2012)

[MATIS, 2012] Accueil MATIS [online] http://recherche.ign.fr/labos/matis/accueilMATIS.php (Accessed on January 21, 2012)

[Meixner and Leberl, 2010] Meixner, P., Leberl, F. Framework to automatically characterize real property using high resolution aerial images. *ASPRS (American Society of Photogrammetry and Remote Sensing) Annual Convention*, Unpaginated DVD, 2010

[Meixner et al., 2011] Meixner P., F. Leberl, Bredif, M. Interpretation of 2D and 3D Building Details on Façades and Roof. *Photogrammetric Image Anlysis*, pp. 137-142, 2011

[Meyer, 1991] Meyer, F. Un algorithme optimal pour la ligne de partage des eaux. *Dans 8me congrès de reconnaissance des formes et intelligence artificielle*, Vol. 2, pp. 847–857, 1991

[Microsoft, 2012] Interactive Visual Media - Microsoft Research [online] http://research.microsoft.com/en-us/groups/interactivevisualmedia/ (Accessed on January 21, 2012)

[Micusik et al., 2012] Micusik, B., Kosecka, J., Singh, G. Semantic parsing of street scenes from video. *The International Journal of Robotics Research vol. 31 no. 4*, pp. 484-497, 2012

[Müller et al., 2006] Müller P., Wonka P., Haegler S., Ulmer A., Van Gool L. J. Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3), pp. 614-626, 2006

[Müller et al., 2007] Müller, P., Zeng, G., Wonka, P., Van Gool, L. J. Image-based Procedural Modeling of Façades. *ACM. Transactions on Graphics*, 26(3), pp. 85-97, 2007

[Murphy, 1998] A Brief Introduction to Graphical Models and Bayesian Networks [online] http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html (Accessed on July 3, 2012)

[Nister, 2004] Nister, D. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence*, pp. 195-202, 2004

[Oda et al., 2004] Oda, K., Lu, W., Uchida, O., Doihara, T. Triangle-based visibility analysis and true ortho generation. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, pp. 623-629, 2004

[OGC, 2012] Open Geospatial Consortium [online] http://www.opengeospatial.org/ (Accessed on July 2, 2012)

[Oliva and Torralba, 2001] Oliva, A., Torralba, A. Modeling the shape of the scene: a holistic representation of the spatial envelope. ***International Journal of Computer Vision***, 42(3), pp. 145–175, 2001

[Oliva and Torralba, 2007] Oliva, A., Torralba, A. The role of context in object recognition. *Trends in cognitive sciences,* 11(12), pp. 520-527, 2007

[Pearl, 1982] Pearl, J. Reverend Bayes on inference engines: A distributed hierarchical approach. *Second National Conference on Artificial Intelligence*, pp. 133–136, 1982

[Perko and Leonardis, 2008] Perko, R., Leonardis, A. Context Driven Focus of Attention for Object Detection. *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*. Springer-Verlag, ISBN 978-3-540-77342-9, 2008

[Photobucket, 2012] Image hosting, free photo sharing & video sharing at Photobucket [online] http://photobucket.com/ (Accessed on July 2, 2012)

[Picasa, 2012] Picasa [online] http://picasa.google.com/ (Accessed on July 2, 2012)

[Pidwirny, 2006] Pidwirny, M. *Introduction to Geographic Information Systems*. Fundamentals of Physical Geography, 2nd Edition, 2006

[Pock et al., 2008] Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D. A Convex Formulation of Continuous Multi-Label Problems. *European Conference on Computer Vision (ECCV),* pp. 792-805, 2008

[Pollefeys et al., 2008] Pollefeys, M., Nister, D. et al. Detailed real-time urban 3d reconstruction from video. ***International Journal of Computer Vision***, 78(2), pp.143–167, 2008

[Prokaj and Medioni, 2011] Prokaj, J., Medioni, G. Using 3D Scene Structure to Improve Tracking. *Computer Vision and Pattern Recognition (CVPR)*, pp. 1337-1344, 2011

[Rabinovich et al., 2007] Rabinovich, A., Vedaldi, A., Galeguillos, C., Wiewiora, E., Belongie, S.; Objects in context. *International Conference on Computer Vision (ICCV)*, pp. 1753-1760, 2007

[Recky, 2006] Recky, M. Fast Area-Based Stereo Algorithm. *CESCG, Proceedings of the 10th Central European Seminar on Computer Graphics*, pp. 95–101, 2006

[Recky and Leberl, 2009] Recky, M., Leberl, F. Semantic Segmentation of Street-Side Images. *Proceedings of the Annual OAGM Workshop*, pp. 271–282, 2009

[Recky and Leberl, 2010] Recky, M., Leberl, F. Multi-View Image Interpretation for Street-Side Scenes. *Computer Graphics and Imaging (CGIM)*, Innsbruck, Austria, pp. 48-54, 2010

[Recky and Leberl, 2010(II)] Recky, M., Leberl, F. Windows Detection Using K-means in CIE-Lab Color Space. *Internation Conference on Pattern Recognition (ICPR),* Istanbul, Turkey, pp. 356-360, 2010

[Recky and Leberl, 2010(III)] Recky, M., Leberl, F. Window Detection in Complex Façades. *European Workshop on Visual Information Processing*, pp. 220-225, 2010

[Recky et al., 2011] Recky, M,. Wendel, A., Leberl, F. Façade Segmentation in a Multi-View Scenario. *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pp. 358-365, 2011

[Recky et al., 2012] Recky, M., Leberl, F., Ferko, A. Multi-View Random Fields and Street-Side Imagery. *International Conference in Cetral Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, unpaged DVD, ISSN 1213-6980, 2012

[Rijsbergen, 1979] Rijsbergen, C. J. V. *Information retrieval*. Butterworth-Heinemann Newton, USA, 1979

[Roberts, 1965] Roberts, L.G. Machine perception of three-dimensional solids. *Optical and Electro-Optical Information Processing*, pp. 159-197, 1965

[Santosh et al., 2009] Santosh, K., Divvala, S. K., Hoiem, D., Hays, J. H., Efros, A. A., Hebert, M. An Empirical Study of Context in Object Detection. *Computer Vision and Pattern Recognition (CVPR),* pp. 1271-1278, 2009

[SatImagingCorp, 2012] WorldView-3 Satellite Imagery and Satellite System Specifications [online] http://www.satimagingcorp.com/satellite-sensors/worldview-3.html (Accessed on July 7, 2012)

[SCImago, 2012] Journal Rankings on Computer Vision and Pattern Recognition [online] http://www.scimagojr.com/journalrank.php?category=1707 (Accessed on July 7, 2012)

[Scribbal, 2012] Scribbal. INFOGRAPHIC: How Much Daily Content Is Published To Twitter, Facebook, Flickr? [online] http://www.scribbal.com/2011/04/infographic-how-much-daily-content-is-published-to-twitter-facebook-flickr/ (Accessed June 7, 2012)

[Simon et al., 2011] Simon, L., Teboul, O., Koutsourakis, P., Paragios, N. Random Exploration of the Procedural Space for Single-View 3D Modeling of Buildings. *International Journal of Computer Vision*, vol. 93, pp. 253-271, 2011

[Singhal et al., 2003] Singhal, A., Luo, J., Zhu, W. Probabilistic spatial context models for scene content understanding. *Computer Vision and Pattern Recognition (CVPR),* pp. 235-241, 2003

[Sithole and Vosselman, 2003] Sithole, G., Vosselman, G. Automatic Structure Detection in a Point-Cloud of an Urban Landscape. *Remote Sensing and Data Fusion over Urban Areas*, pp. 67–71, 2003

[Shotton et al., 2006] Shotton, J., Winn, J. M., Rother, C., Criminisi, A. Textonboost: Joint appearance, shape and context modeling for multi-class object. *European Conference on Computer Vision (ECCV)*, pp. 1-15, 2006

[Snavely et al., 2006] Snavely, N., Seitz, S. M., Szeliski, R. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (TOG)*, pp. 835 – 846, 2006

[Stamos and Allen, 2001] Stamos, I., Allen, P. K. Automatic Registration of 2-D with 3-D Imagery in Urban Environments, *Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 731-736, 2001

[Stiny and Gips, 1972] Stiny, G., Gips, J. Shape grammars and the generative specification of painting and sculpture. *Information Processing 71, North-Holland Publishing Company*, pp. 1460–1465, 1972

[Stoter and Zlatanova, 2003] Stoter, J., Zlatanova, S. 3D GIS, where are we standing. *Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modeling and Analysis*, pp. 337-445, 2003

[Strat, 1993] Strat, T. M. Employing contextual information in computer vision. *In Proceedings of ARPA Image Understanding Workshop*, pp. 217–229, 1993

[Šochman, 2006] Šochman, J. Specification of AdaBoost IPM for use in SCENIC. *Technical Report TN-eTRIMS-CMP-01-2006*, 2006

[Taillandier, 2000] Taillandier, F. *Texture and Relief Estimation from Multiple Georeferenced Images.* Master of Science Thesis, DEA Algorithmique, Ecole Polytechnique, 2000

[Tell and Carlsson, 2000] Tell, D., Carlsson, S. Wide baseline point matching using affine invariants computed from intensity profiles. *European Conference on Computer Vision (ECCV)*, pp. 814-828, 2000

[Tell and Carlsson, 2002] Tell, D., Carlsson, S. Combining appearance and topology for wide baseline matching. *European Conference on Computer Vision (ECCV)*, pp. 68-81, 2002

[Tomlinson, 1967] Tomlinson, R.F. An Introduction to the Geo-Information System of the Canada Land Inventor. *ARDA, Canada Land Inventory, Department of Forestry and Rural Development*: Ottawa, ON, Canada, 1967

[Toshev et al., 2010] Toshev, A., Mordohai, P., Taskar, B. Detection and Parsing Architecture at City Scale from Range Data. *Computer Vision and Pattern Recognition (CVPR)*, pp. 398-405. 2010

[Van Gool et al., 2007] Van Gool, L., Zeng, G., Van den Borre, F., Müller, P. Towards Mass-produced Building Models. *Photogrammetric Image Analysis*, pp. 209-220, 2007

[Van Ruymbeke et al., 2008] Van Ruymbeke, M., Tigny, V., De Bats, E., Moreno, G.R., Billen, R. Development and use of a 4D GIS to support the conservation of the Calakmul site. *International Conference on Virtual Systems and Multimedia*, pp. 117 - 121, 2008

[Wallach, 2002] Wallach, H. *Efficient training of conditional random fields*. Master's thesis, University of Edinburgh, 2002

[Wainwright et al., 2002] Wainwright, M. J., Jaakkola, T. S., Alan, A.S. Tree-Based Reparameterization for Approximate Inference on Loopy Graphs. *Advances NISP*, pp. 1001-1008, 2002

[Wang et al., 2007] Wang, X., Yang, J., Liu, H., Ma, Y. Fast Stereo Matching Algorithm for Real-time Robot. *Application, IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 908 – 913, 2007

[Weis, 2007] Weis, S.A. RFID (Radio Frequency Identification): Principles and Applications. *MIT CSAIL*, 2007

[Wendel et al., 2010] Wendel, A., Donoser, M., Bischof, H. Unsupervised façade segmentation using repetitive patterns. *In Proceedings of the 32nd Annual Symposium of the German Association for Pattern Recognition (DAGM'10)*, LNCS 6376, pp. 51-60, Springer, 2010

[Wendel et al., 2011] Wendel, A., Irschara, A., Bischof, H. Natural Landmark-based Monocular Localization for MAVs. *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5792-5799, 2011

[Werner and Zisserman, 2002] Werner, T., Zisserman, A. New Techniques for Automated Architecture Reconstruction from Photographs. *European Conference on Computer Vision (ECCV)*, pp. 541—555, 2002

[Wikipedia, 2012] Zoning - Wikipedia, the free encyclopedia [online] http://en.wikipedia.org/wiki/Zoning (Accessed on July 4, 2012)

[Wolf and Bileschi, 2006] Wolf, L., Bileschi, S. A critical view of context. *International Journal of Computer Vision*, pp. 251-261, 2006

[Wonka et al., 2003] Wonka, P., Wimmer, M., Sillion, F. X., Ribarsky, W. Instant Architecture. *ACM Transaction on Graphics*, 22(3), pp. 669-677, 2003

[Xiao et al., 2009] Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L. Image-based street-side city modeling. *ACM Transactions on Graphics*, Article No. 114, 2009

[Yang et al., 2011] Yang, M., Ying, C. Y,, McDonald, J. Fusion of Camera Images and Laser Scans for Wide Baseline 3D Scene Alignment in Urban Environments. *In ISPRS Journal of Photogrammetry and Remote Sensing*, pp. 52-61, 2011

[Yoon and Kweon, 2006] Yoon, K. J., Kweon, I. S. Adaptive Support-Weight Approach for Correspondence Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), pp. 650–656, 2006

[Zara, 2004] Zara, J. Virtual Reality and Cultural Heritage on the Web. *International Conference on Computer Graphics and Artificial Intelligence*, pp. 101-112, 2004

[Zebedin, 2010] Zebedin, L. *Automatic Reconstruction of Urban Environments from Aerial Images*. PhD thesis, Graz University of Technology, 2010

[Zhilin et al., 2005] Zhilin, L., Qing, Z., Gold, C. Digital Terrain Modeling: Principles And Methodology. *CRC Press*, ISBN-13: 978-0415324625, 2005

[Zhu and Wu, 1998] Zhu, S. C., Wu, Y. N., Mumford, D. B. Filters, random field and maximum entropy: Towards a unified theory for texture modeling. *International Journal of Computer Vision*, pp. 1-20, 1998