Graz University of Technology

Institute for Computer Graphics and Vision

Master's Thesis

Realtime Incremental Image Stitching For Industrial Quality Inspection

---

Robert Lanner

Graz, Austria, February 2011

Thesis supervisors

Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof

Dipl.-Ing. Dr.techn. Matthias Rüther

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

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.

Graz, am ……. ……..
(Unterschrift)

Englische Fassung:

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.

……. ……..
date (signature)

Abstract

The task of image stitching is to create a high quality panorama from a set of partially overlapping input images. In order to align images, image registration is applied. A blending method is finally used to stitch the aligned images and to create smooth transitions between them without visible artifacts. Hence, the two main steps for the generation of a panorama are image registration and image blending. This thesis presents an image stitching system that creates a weld seam survey image for visual quality inspection of welding processes. The image registration is based on salient keypoint extraction and robust motion estimation. By incorporation of available tracking data, the system gurantees a successful mosaic generation. Furthermore, the system includes an incremental blending strategy to provide an 'online' generation of image mosaics and noise filtering methods to cope with e.g. smoke or sparks that typically occur during welding processes.

Keywords. Image stitching, image mosaic, panorama, image registration, blending

Kurzfassung

Image Stitching bezeichnet das Erzeugen eines Panoramabildes aus einer Menge von überlappenden Bildern einer Szene. Bildregistrierung wird verwendet, um die Bilder bestmöglich übereinander zu legen. Aufeinander ausgerichtete Bilder werden schließlich mittels einer Blendingmethode so zusammengefügt, dass sichtbare Übergänge oder störende Artefakte minimiert werden. Die zwei Hauptschritte zur Generierung eines Panoramabildes sind somit Bildregistrierung und Blending. Diese Masterarbeit stellt ein Image Stitching System vor, das ein Panoramabild einer kompletten Schweißnaht zur visuellen Qualitätskontrolle eines Schweißprozesses erstellt. Die Bildregistrierungsmethode beruht dabei auf markanten Bildpunkten und einer robusten Abschätzung der Bewegung zwischen aufeinander folgenden Bildern. Durch das Hinzufügen von verfügbaren Trackingdaten garantiert das System eine erfolgreiche Generierung eines Panoramabildes. Das System beinhaltet außerdem eine inkrementelle Blendingstrategie, durch die das fortlaufende Erstellen des Panoramabildes möglich ist. Weiters werden störende Effekte wie z.B. Rauch und helle Funken gefiltert, um ein sauberes Panorama als Endresultat zu liefern.

Schüsselwörter. Image Stitching, Panoramabild, Bildregistrierung, Blending

Acknowledgments

First and foremost I want to thank my supervisors Prof. Dr. Horst Bischof and Dr. Matthias Rüther for introducing me to the very interesting field of computer vision, for the smooth reviewing process and for assistance during my research.

I would also like to thank the robot vision community at the ICG for the pleasant collaboration, for profitable discussions and for invaluable support. Especially I would like to show my gratitude to Markus Heber, who was always available to answer my questions, advised me on any problems and carefully proofread my master thesis, as well as to Martin Lenz and Katrin Pirker, who were always available to give me advices and made helpful suggestions.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this thesis.

Contents

List of Figures

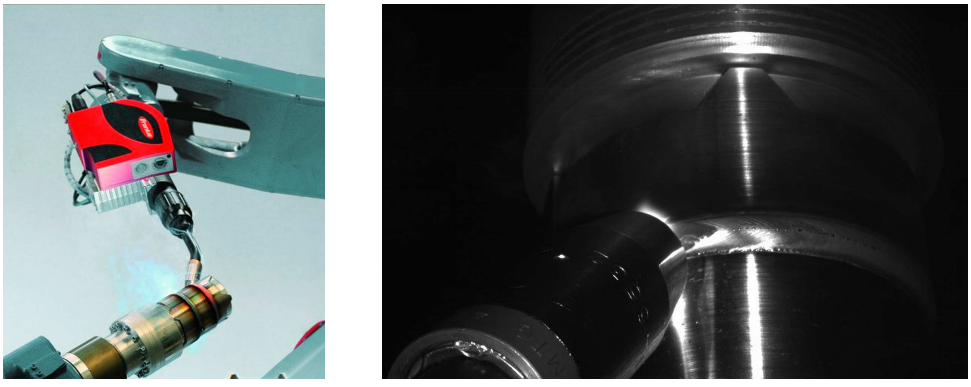
List of Tables

Chapter 1

Introduction

The task of image stitching is to create a high quality panorama from a set of (connected) partially overlapping input images. In order to align images, image registration is applied. A blending method is finally used to stitch the aligned images and to create smooth transitions between them without visible artifacts. Hence, the two main steps for the generation of a panorama are image registration and image blending.

The goal of this work is to implement an image stitching system for visual quality inspection of welding processes. The stitching system should be able to generate an incremental image mosaic, where a weld expert is able to inspect the quality of a weld seam at a glance. On the basis of the weld seam mosaic, it should be possible to evaluate when and where an error occurs. The welding process is carried out by a welding robot on which a camera is mounted. The high quality camera is able to acquire frames of the welding process despite the welding flare. Figure 1.1 shows the acquisition system. The camera unit is positioned next to the welding torch.



(a) (b)

Figure 1.1: Acquisition system to observe the welding process. (a) Camera unit positioned next to the welding torch, and (b) corresponding field of view. Image courtesy of Fronius Ltd.

High quality greyscale images are acquired containing the weld seam, which is tracked during the whole process. For a single weld seam, it may be that several hundred consecutive frames are acquired. In terms of computation time, the processing of that much data is not appropriate, because we want to focus on the weld seam. The other image information on the frame is not interesting. For example, the welding head and the melting bath which are visible on every frame are not required for the generation of the survey image. As tracking also allows to extract small patches around the weld seam from each frame, the input to the image stitching algorithm is not the original frame, but a patch which is extracted by a tracking algorithm. The obtained patches should be stitched into a single survey image. The size of these patches is variable (e.g. 140×140). The patches usually provide much overlap, so that one pixel in the final panorama can be related to numerous patches.

Commercial products (e.g. AutoStitch* or Autopano†) and open source applications (e.g. Hugin‡) for image stitching are basically intended for private use and can have some limitations, such as the projection plane of the panorama has to be cylindrical. For the acquired images of the weld seam, it is assumed that the scene is locally planar and that there is only little projective distortion. The distance between the camera and the weld contact area remains approximately constant. Thus, it is adequate to compute the relative rotation and translation between the extracted patches of the consecutive frames.

Challenges for the image stitching process are heavy smoke, bright sparks, evaporating water and further effects which appear during welding. It is a goal for our image stitching system to reduce such noisy effects in the final panorama. Requirements for the system are to deliver an accurate, robust result in adequate runtime. The survey image of the weld seam should be continous and not splitted, if registration fails. A further challenge is the creation of an incremental panorama. This means that whenever a new image is acquired, the panorama should be updated.

We present an image stitching system, that creates a weld seam survey image from extracted patches and that makes use of prior (tracking) data. Figure 1.2 illustrates the generation of a weld seam mosaic. The system is robust and a complete mosaic generation is guaranteed. If pairwise registration of patches fails, an incorporation of prior data ensures a successful registration. The mosaic is adjusted to a required end position in the mosaic coordinate frame by global refinement. This step further improves registration repeatability. Noisy elements like bright sparks or single wads of smoke are no problem. The weld seam mosaic is stiched by using appropritate blending methods. We also include strategies to overcome heavy smoke and over-exposure. Image stitching systems are often intended to create one panorama image not before all input images are processed. In our approach, the panorama is generated incrementally. Whenever a new image is acquired, the panorama is updated. As the image stitching system processes

*http://cvlab.epfl.ch/ brown/autostitch/autostitch.html
†http://www.autopano.net
‡http://hugin.sourceforge.net/

input images consecutively and incrementally, a 'growing' panorama can be observed. It is even possible to complete an image mosaic and to extend it afterwards. We use an incremental blending approach, which considers all overlapping patches that contribute to a single pixel in the final mosaic.

In Section 2 theory and background related to image stitching are presented. This includes image features, feature matching, image alignment and registration, as well as general image stitching and blending strategies. In Section 3 related work regarding registration and blending is discussed. Our methodology is presented in detail in Section 4. We introduce image registration relying on salient features and robust rigid motion estimation. Incremental image blending including exposure fusion is also introduced within the methodology section. In Section 5 we present different experiments and evaluations of the implemented image stitching system. The experimental results approve our choice of specific registration and blending methods.

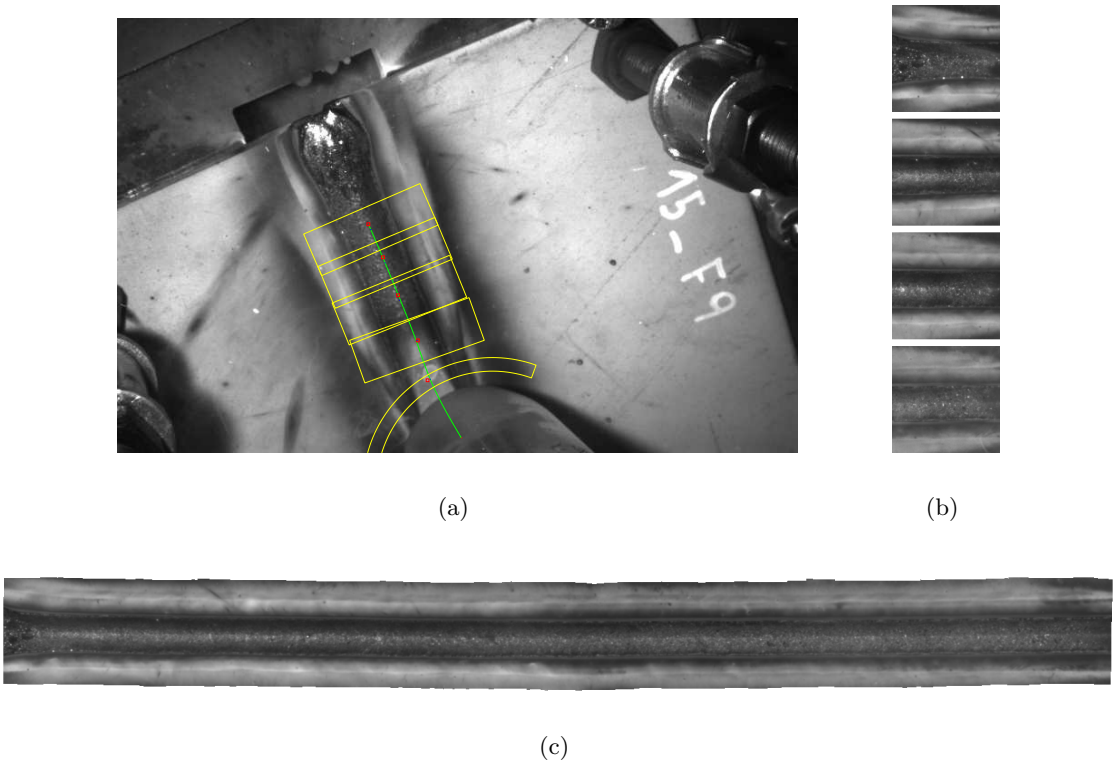


Figure 1.2: Generation of a weld seam mosaic. (a) Frame of a welding process and computed track, (b) extracted rectified patches and (c) generated weld seam mosaic.

Chapter 2

Theory and Background

Contents

The goal of image stitching is to create a panorama of high quality from a set of aligned input images. To align images an image registration method has to be applied. A motion model which relates pixel coordintes from one image to another has to be determined. An issue here is for example whether the camera which acquires images is positioned at a central point and is just rotating around its optical center or the camera position is changing as well. To represent the motion, a camera matrix or a homography (geometric transformtion) is estimated. The camera matrix describes the transformation which maps a point in world coordinates to the image plane. A homography is a planar projective transformation to project a pixel from one image to another, each image being considered as a projective plane [22]. There are two main approaches for image registration: direct and feature-based. Direct methods use the whole image information for a pixel-to-pixel mapping. Feature-based methods rely on distinctive image features. Section 2.1 gives an overview about two keypoint extractors and descriptors: SIFT and SURF. Furthermore two corner detection methods (Harris and FAST) are covered. Section 2.2 is concerned with methods and metrics to find matches between distinctive image features. In Section 2.3 a general overview about image registration and alignment is given.

If a set of images is aligned, the next task is to create a high quality panoramic image. A naive approach of just copying and pasting the images together would not yield in the desired result. The transition from one image to another should be smooth. There should be no visible seam between images. Challenges in this area could be different exposures in the acquired images, some small amount of misregistration between images or moving objects. Visible artifacts or blurring can appear on the final panorama. Hence,

an adequate image stitching and blending strategy has to be developed. Section 2.4 gives an general insight into this matter.

2.1 Image Features

Feature (interest point, keypoint, line, etc.) detection is crucial in computer vision tasks such as tracking, image matching or recognition. Many feature detection algorithms are based on corner response functions. The goal is to extract local maximum or minimum pixels, which exceed some threshold corneress value [41]. To allow successful matching of extracted image features, descriptors, which should be robust and distinctive, are computed from extracted features. The following sections give an overview about two corner detectors: Harris [21] and FAST [40, 41] corners. In further sections two well-known keypoint extractors and descriptors are discussed: SIFT [32] and SURF [5].

2.1.1 Harris Corners

Harris and Stephens proposed the well-known Harris corner detector [21] in 1988. It was an aim to enable explicit tracking with discrete image features. Corners are determined using a response function. For each image point (x,y) a matrix M (structure tensor) is calculated,

$$M = \begin{bmatrix} G_\sigma * {I_x}^2 & G_\sigma * I_{xy} \\ G_\sigma * I_{xy} & G_\sigma * {I_y}^2 \end{bmatrix} , \tag{2.1}$$

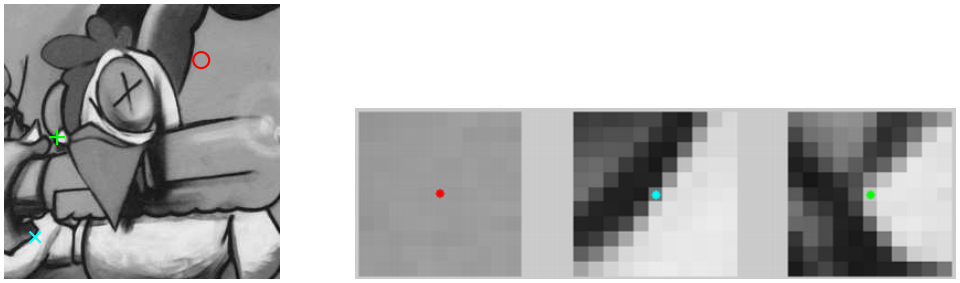
where G_σ is a Gaussian filter, $*$ is the convolution operator, and ${I_x}^2, {I_y}^2 and I_{xy}$ are second order derivatives of the input image I. By analyzing the eigenvalues of M, the following conclusions are made by Harris and Stephens:

- If both eigenvalues are small, a flat region around the point is detected.
- If one eigenvalue is high, and the other one is low, an edge is detected.
- If both eigenvalues are high, this indicates a corner is detected.

Figure 2.1 illustrates all three types. To avoid the explicit eigenvalue decomposition, the following corner response function R is used to extract interest points:

$$R = \det(M) - k(\mathrm{trace}(M))^2 . \tag{2.2}$$

A corner is indicated by a high response (R is positive). R is negative in edge regions and small in homogenous areas. A homogenous region is determined by $\mathrm{trace}(M) < \xi$, where ξ is a selected threshold. The constant parameter k can be set empirically (e.g. $k = 0.04$). A non-maximum supression is performed additionally on all corner responses, in order to get local maxima as interest points.



(a) (b)

Figure 2.1: Harris corner detection. In the left figure (a) a flat region (red), an edge (cyan) and a corner (green) are marked. The right figure (b) shows enlarged 11×11 patches of the region around the interest points. If both eigenvalues are small, a flat region is detected. The marked point of the flat region has eigenvalues $\lambda_1 =$ 1e-6, $\lambda_2 =$ 4e-6 and a corner response function $R = 0$. If one eigenvalue is high, and the other one is low, an edge is detected. The edge has $\lambda_1 =$ 3.4e-4, $\lambda_2 = 0.18$ and $R =$ -1.2e-3 at the marked point. If both eigenvalues are high, a corner is detected. The corner has the following values: $\lambda_1 = 0.01$, $\lambda_2 = 0.08$ and $R =$ 5.5e-4. The corner response function R is negative for edges and positive for corners.

2.1.2 FAST Corners

FAST [40, 41] was developed with an emphasis on computational cost reduction for corner detection. FAST is the acronym for *Features from Accelerated Segment Test*. The corner detector operates by considering a circle of sixteen pixel (a Bresenham circle of radius 3) around a corner candidate c. A corner is detected if the intensities of at least 12 contiguous pixels on the circle are brighter than the intensity of c plus a threshold t, or darker than the intesity of c minus t (see Figure 2.2).

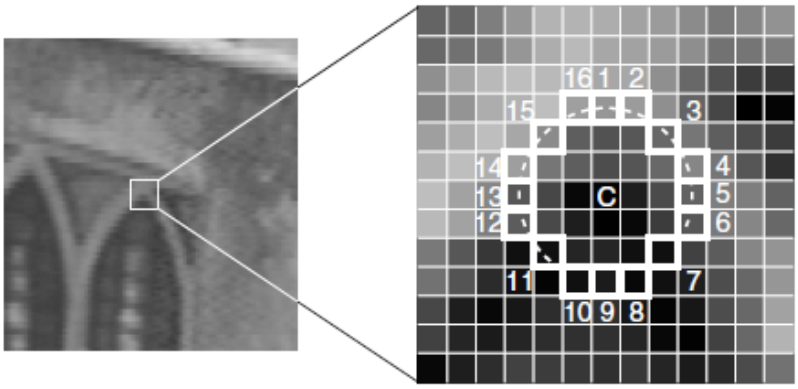


Figure 2.2: FAST corner detection (adopted from [40]). The highlighted squares are the 16 pixels of the Bresenham circle which are used for corner detection. A corner is detected at the center c. The dashed line represents 12 contigous pixels which are brighter than c plus a threshold.

An acceleration is achieved by examining only the four pixels at positions 1, 5, 9 and 13. This provides more rapid rejection of candidate pixels. A corner can only exist if at least three of these four pixels are brighter or darker than the intensity of c by a threshold t. The full segment test is only carried out for the remaining corner candidates. To yield a speed increase, a machine learning algorithm, which generates a decision tree, is employed. Additionally, a non-maximum suppression is performed to improve the results. The intensities from the 16 pixels on the circle can be used to compute a feature vector or descriptor. The feature vector elements are positive if pixels are brighter than the center c or negative vice-versa. This partitioning avoids comparison of positive with negative features.

2.1.3 SIFT

The Scale Invariant Feature Transform (SIFT) has been initially proposed by Lowe [32]. The extracted SIFT features are invariant to scale and rotation, partially invariant to change in illumination and partially invariant to change in 3D camera viewpoint. In the first step of the algorithm scale-space extrema are detected using Difference-of-Gaussians (DoG) images. Scale space theory has been developed by Lindeberg [31]. The scale space of an image is defined by

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \ , \tag{2.3}$$

where $G(x, y, \sigma)$ is a variable-scale Gaussian which is convolved with the input image $I(x, y)$. The DoG function $D(x, y, \sigma)$ is computed from the difference of two adjacent scales:

$$D(x, y\sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.4}$$

The two nearby scales are separated by a constant factor k. The generation of DoG images is shown in Figure 2.3. To detect local maxima and minima, a pixel is compared to its neighbors in 3×3 regions at the current and adjacent scales. Once a keypoint candidate location has been found, a detailed model is fit to determine location, scale, and ratio of principal curvatures. Keypoints with low contrast are rejected. Furthermore, keypoints with a ratio of principal curvatures below a threshold are also eliminated. The ratio r of principal curvature κ is estimated from a Hessian matrix H at the location and scale of the keypoint. The Hessian matrix for an image point (x, y) at scale σ is defined as

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \ , \tag{2.5}$$

where $L_{xx}(x, y, \sigma)$ is the convolution of the Gaussian second order partial derivative in x-direction at image point (x, y), analogously L_{yy} in y-direction, and L_{xy} in xy-direction.

Lowe estimates the derivatives by taking differences to neighboring points of the keypoint at the particular scale and gets accordingly the ratio r of principal curvature:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$
$$r = \frac{\text{trace}(H)^2}{\det(H)} \ . \tag{2.6}$$

In the next step, a consistent orientation is assigned to each keypoint location, based on local image gradient directions. As the keypoint descriptor can be represented relative to this orientation, invariance to image rotation is achieved. The keypoint descriptor is typically calculated for 16×16 patches (around the keypoint) which are further divided into 4×4 sample regions. For every sample region a weighted 8-bin orientation histogram is calculated. This leads to a 128 ($4 \times 4 \times 8$) element feature vector for each keypoint. The computation of a 2×2 keypoint descriptor is illustrated in Figure 2.4. Figure 2.5 shows an image with extracted SIFT features.

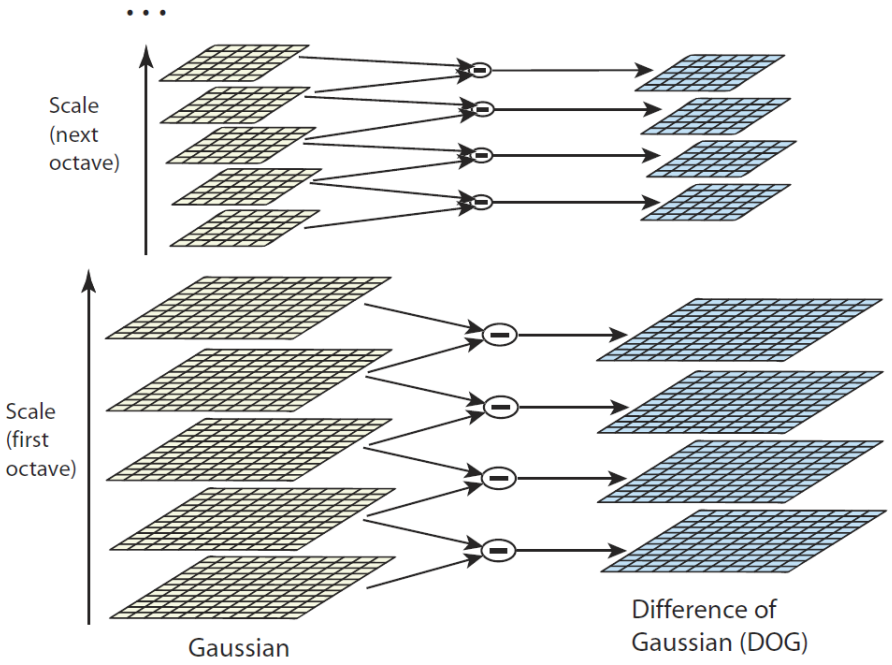


Figure 2.3: Efficient construction of Difference of Gaussian (DoG) images (taken from [32]). The initial image is incrementally convolved with Gaussians, to get a set of scale space images. The DoG images are computed from the difference of two nearby scales. The process is iterated after each octave. For a new octave the Gaussian image that has twice the initial value of σ is down-sampled by a factor 2.

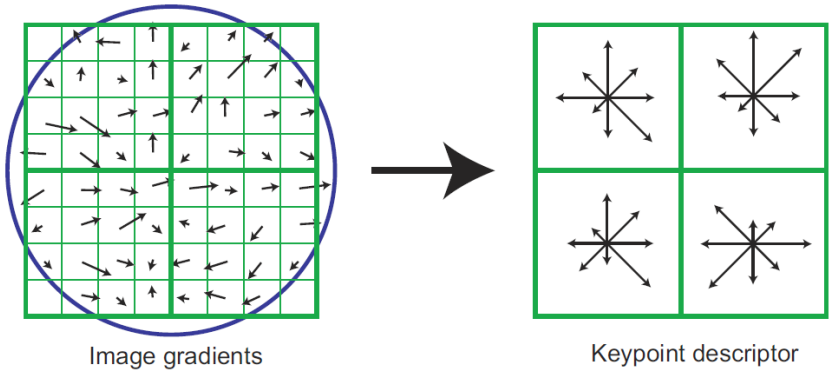


Figure 2.4: Computation of a SIFT keypoint descriptor (taken from [32]). Gradient magnitude and orientation are calculated at each point around the keypoint (left). A Gaussian weighting function (illustrated by the circle) is applied to the gradients. In this figure a 8×8 set of samples is divided into 2×2 sample regions. Weighted orientation histograms are computed for every sample region (right). An orientation histogram here contains 8 gradient directions with accumulated gradient magnitudes.

Figure 2.5: The 816×612 image shows about 300 extracted SIFT features. The circles represent the scales of the SIFT features. The radii represent their orientations. Note: There are SIFT keypoints where more than one orientation is assigned at the same scale and location.

2.1.4 SURF

SURF (Speeded Up Robust Features) [5] is a scale- and rotation-invariant feature detector and descriptor. The interest point detector is based on the Hessian matrix H (see Equation 2.5). To decrease the computational cost, the second order Gaussian derivatives are approximated by box filters (see Figure 2.6). The so-called Fast-Hessian detector relies on the determinant of an approximated Hessian,

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \ . \tag{2.7}$$

D_{xx}, D_{yy} and D_{xy} are the approximated filter responses. The box filters are evaluated on integral images. The scale space is analysed by applying such filters of various size on the same image. Instead of using image pyramids and down-scaling of image size, the filter size is increased iteratively to find keypoints at different scales. Further on, filter responses are normalized related to the mask size. The lowest scale is represented by 9×9 box filters (as shown in Figure 2.6) which are approximations for Gaussian second order derivatives with $\sigma = 1.2$.

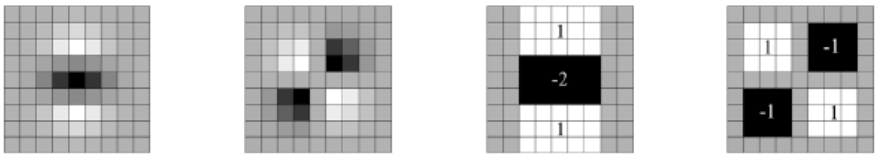


Figure 2.6: SURF box filters (adopted from [5]). The two left figures show Gaussian second order partial derivatives in y- and xy-direction. The two right figures are their approximations by using box filters. Grey regions are equal to zero.

In order to be rotation-invariant, an orientation is assigned to every keypoint. The dominant orientation is determined with the help of Gaussian weighted Haar-wavelet responses in x and y direction (see Figure 2.7 right) in a circular neighborhood around the interest points. The sampling step is chosen to be the scale s, at which an interest point is detected. The wavelet filters have a side length of $4s$. Next, the SURF descriptor is calculated for a square region (with window size $20s$), centered around the interest point. It is oriented along the assigned orientation. The region is divided into 4×4 sub-regions, where each sub-region contains 5×5 sample points. Gaussian weighted Haar-wavelet responses d_x and d_y are calculated for every sample point. Hence, a four-dimensional descriptor vector,

$$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_x|) \ , \tag{2.8}$$

is obtained for every sub-region, which results in a $4 \times 4 \times 4 = 64$ element descriptor vector. An extended version of the SURF descriptor uses eight features per sub-region, which leads to 128 elements (SURF-128). Figure 2.7 shows detected interest points at different scales with the corresponding descriptor windows and orientations, and the Haar-wavelet filters used for SURF.

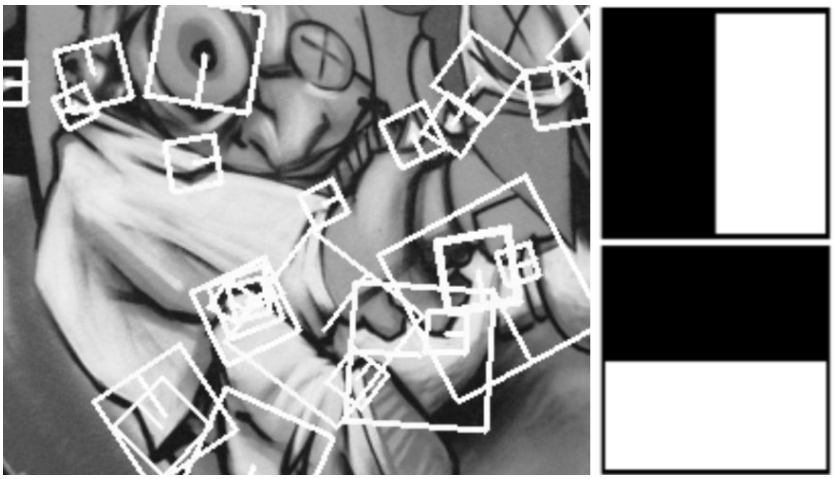


Figure 2.7: SURF features (taken from [5]). Left: Detected interest points with corresponding descriptor windows at different scales. Right: Haar-wavelet filter masks used for SURF descriptor. Black is negative (e.g. $-\sqrt{2}/2$), and white positive (e.g. $-\sqrt{2}/2$).

2.1.5 Comparison

The introduced keypoint and corner detectors respond to different keypoint locations. The differences are demonstrated in two figures. Figure 2.8 shows extracted Harris and FAST corners.

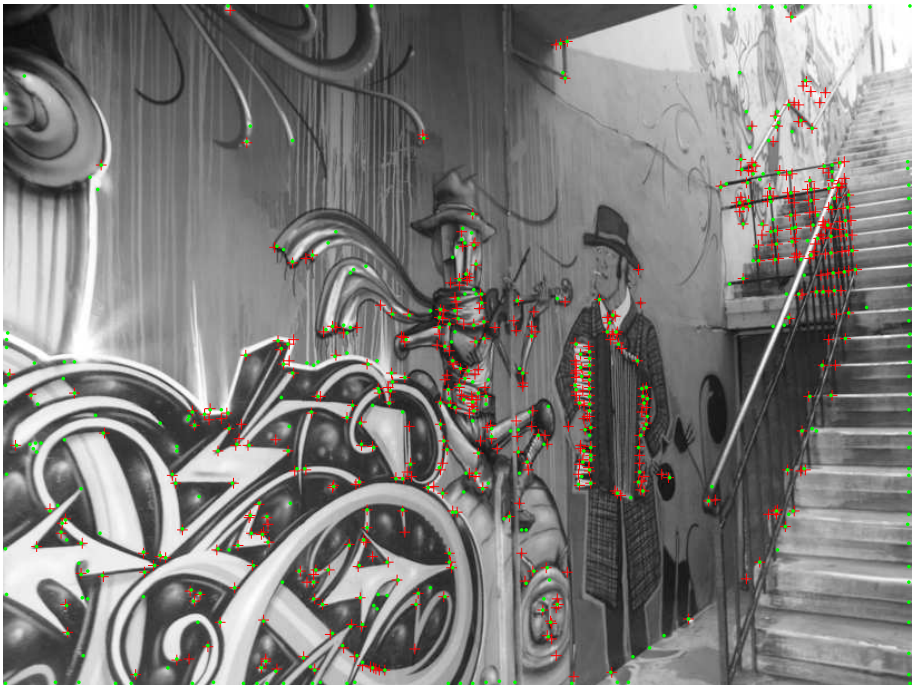


Figure 2.8: Extracted Harris and Fast corners. Harris corners are marked by a green dot. FAST corners are marked by a red plus.

In Figure 2.9 SIFT and SURF features are detected. The features are represented with respect to their scales and orientations.

Figure 2.9: The 816 × 612 image shows about 120 detected SIFT (yellow) and SURF (green) features each. Features are detected at different scales and with different orientations.

2.2 Matching Metrics

Matching metrics are required to find corresponding features (interest points) in two or more images. High-dimensional distinctive keypoint descriptors (as presented in Sections 2.1.3 and 2.1.4) are a requirement for reliable matching results. A suitable matching strategy has to be used to compare the descriptor vectors. A further approach is to track interest points from one image to another [33]. In this method features (e.g. Harris corners) are extracted in a first image and further monitored during tracking [43]. In the case of image registration, it is not only relevant to find feature matches, but also to verify image matches.

2.2.1 Feature Matching

In general, feature matching is a way to find the closest keypoint descriptors (feature vectors) between a set of images. The choice of an appropriate distance measure may depend on the type of keypoint descriptor. An often used distance measure is the Euclidean distance. Given two n-dimensional vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$

the Euclidean distance is

$$d = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \ . \tag{2.9}$$

Another distance measure to evaluate the similarity between keypoint descriptors is the Mahalanobis distance [35]. The Mahalanobis distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as [22]

$$d = \|\mathbf{x} - \mathbf{y}\|_\Sigma = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})} \ , \tag{2.10}$$

where Σ is the covariance matrix. Furthermore, an adequate matching strategy has to be chosen.

The best candidate match for a SIFT keypoint is determined by identifying its nearest neighbor [32]. The nearest neighbor is the keypoint with the minimum Euclidean distance for the 128 element descriptor vector. A modified variant of kd-tree search is used to identify the nearest neighbor. This algorithm is called Best-Bin-First (BBF) [6] and returns the closest neighbor with high probability. To discard false matches, the distance of the closest neighbor is compared to that of the second-closest one. If the ratio of the distance from the closest neighbor to the distance of the second closest is greater than a specified threshold (a typical value is 0.8), matches are rejected. Formally the matching criterion is defined as follows:

$$\frac{d_{N_1}}{d_{N_2}} < t \ , \tag{2.11}$$

where d_{N_1} and d_{N_2} are the Euclidean distances to the nearest and second nearest neighbor, and t is the threshold. This measure is based on the condition that correct matches need to have the closest neighbor considerably closer than the nearest false match. In contrast, there will likely be a number of closest incorrect matches within similar distances due to the high dimensionality of the descriptor vector.

The matching of SURF descriptors can be carried out in a similar manner [5]. Additionally, the sign of the filter response (see Equation 2.7) for a SURF keypont is included. As the sign of the Laplacian (i.e. the trace of the Hessian matrix) distinguishes dark blobs on bright background from bright ones on dark background, only features with the same contrast are compared in the matching stage. This minimal information enables faster matching.

In [35] two further matching strategies are used in addition to the nearest neighbor strategy which is used for SIFT matching. A naive nearest neighbor strategy is to match two descriptors if the distance of the nearest neighbor is below a threshold. The distance ratio between the first and second nearest neighbor is not considered. In the case of threshold-based matching, a descriptor can have several matches. A match is accepted, if the distance between two descriptors is below a threshold.

2.2.2 Image Matching

Considering images of arbitrary order and from different scenes, image matching is a necessary step, in order to find corresponding images. In contrast to feature matching, which determines one or more matches for a feature (keypoint), the task of image matching is to find overlapping images or image regions. Regions of interest (ROI), which appear in multiple images and form connected sets of image matches, should be found. Potential image matches are images which have a large number of feature matches in between [10]. An image match is verified by finding a compatible geometric mapping (see Section 2.3.3) between images. Noisy images can also be detected and rejected via image matching.

2.2.3 KLT Feature Tracker

The Kanade-Lucas-Tomasi (KLT) feature tracker is based on the work by Lucas and Kanade [33]. They state the translational image registration problem with two given image functions $I(\mathbf{x})$ and $J(\mathbf{x})$, where $\mathbf{x}$ is a vector with pixel locations of some region of interest. The goal is to find the displacement vector $\mathbf{d}$ that minimizes the difference between $I(\mathbf{x})$ and $J(\mathbf{x}+\mathbf{d})$ (see Figure 2.10). The problem statement is extended to arbitrary linear transformations, such as rotation, scaling, and shearing. The relationship between two images thus is represented by

$$I(\mathbf{x}) = J(\mathbf{x}A + \mathbf{d}) \ . \tag{2.12}$$

The matrix A expresses the linear spatial transformation between $I(\mathbf{x})$ and $J(\mathbf{x})$. The optimal solution is defined by the minimization of the error function

$$e = \sum_{\mathbf{x}} ((J(\mathbf{x}A + \mathbf{d}) - I(\mathbf{x}))^2 \ . \tag{2.13}$$

The feature tracking approach was further developed and is explained more in detail in [43]. The problem is defined by finding motion parameters A and $\mathbf{d}$ which minimize the dissimilarity

$$e = \int\int_W ((J(A\mathbf{x} + \mathbf{d}) - I(\mathbf{x}))^2 w(\mathbf{x}) d\mathbf{x} \ . \tag{2.14}$$

The error function is considered for a feature window W and is affected by a weighting function $w(\mathbf{x})$. A Gaussian function can be used for $w(\mathbf{x})$ to highlight the central area of the window. Smaller windows and a pure translational model is preferred for tracking. It is assumed that the affine transformation and motion of the feature window is rather small between adjacent frames. Finally, it is suggested to solve the following linear system with the goal to deteremine the displacement $\mathbf{d}$:

$$Z\mathbf{d} = \mathbf{e} \tag{2.15}$$

with

$$Z = \int \int_W \begin{bmatrix} {g_x}^2 & g_x g_y \\ g_x g_y & {g_x}^2 \end{bmatrix} w(\mathbf{x}) d\mathbf{x} \tag{2.16}$$

and

$$\mathbf{e} = \int \int_W (I(\mathbf{x}) - J(\mathbf{x})) \begin{bmatrix} g_x \\ g_y \end{bmatrix} w(\mathbf{x}) d\mathbf{x} \tag{2.17}$$

The formulation of Equations 2.16 and 2.17 includes the spatial gradient $\mathbf{g}$ of image intensity which can be calculated by

$$\mathbf{g} = (\frac{\partial J}{\partial x}, \frac{\partial J}{\partial y})^T \ . \tag{2.18}$$

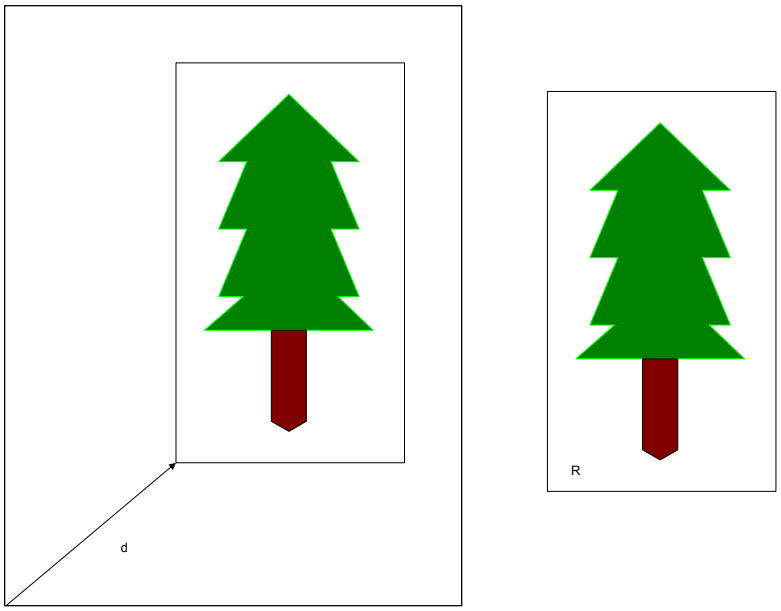


Figure 2.10: The translational image registration problem [33] for some region of interest R: The displacement vector d should be found.

2.3 Image Alignment and Registration

The goal of image alignment and registration is to transform two or more images of the same scene into a common coordinate system. Given the task to register a pair of images, a geometric mapping should be found which relates the second image to the first (reference) image. The geometric mapping is based on a chosen motion model to align images. Image registration is an important task in panorama and image mosaic generation as well as in image fusion. Challenges in image registration are for example low overlap between

images, moving objects in the scene and changing lighting conditions. Image alignment and registration can be divided into the following subtasks, which are covered in the next sections:

- Motion model definition
- Point matching and registration
- Motion estimation
- Image warping

The output of the registration are connected sets of images with their corresponding geometric transformations and sets of aligned images, respectively.

2.3.1 Motion Models

Before image registration, the mathematical relationship that maps pixel coordinates from one image to another has to be determined. There are different motion models possible [48]. One group is the set of 2D planar transformations (Section 2.3.1.1), from Euclidean (rotation and translation) transformation to projective transformation. Another motion model is the 3D camera rotatation, where it is assumed, that the camera is at a fixed point and undergoes rotation. In this case every camera position is parameterised by a rotation and focal length [10]. This approach is based on a camera model, which describes a mapping between a 3D world coordinate to a point on the image plane (Section 2.3.1.2). For multi-viewpoint panoramas [1], the camera is moving along the scene. The pixel-to-pixel mapping also depends on the surface of the panoramic image projection. The surface can be planar, cylindrical or spherical. Examples of the most common types of panoramas are shown in Figure 3.1 in Chapter 3.

2.3.1.1 2D Planar Transformations

The most simple coordinate transformations occur in the 2D plane and are called 2D planar transformations (see Figure 2.11). The hierarchy of transformations is defined by a preceding set of transformations, starting from the most specialized and generalizing up to projective transformations. Hartley and Zisserman [22] includes a detailed overview about the hierarchy of 2D planar transformations. Table 2.1 presents a summary of 2D transformation groups, their number of degrees of freedom (DOF), and their invariant properties.

The 2D Euclidean transformation models the motion of a rigid object and preserves Euclidean distance. It is a composition of rotation R and translation $\mathbf{t}$, and it is used to transform homogenous points $\mathbf{x}$ to $\mathbf{x}'$,

$$\mathbf{x}' = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \ . \tag{2.19}$$

R is a 2D rotation matrix, and $\mathbf{t}$ is a 2D translation vector. The Euclidean transformation has three degrees of freedom, one for the rotation angle and two for the translation in x and y direction.

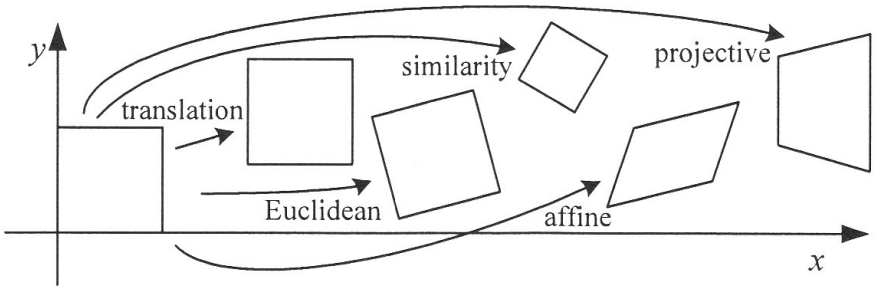


Figure 2.11: Basic set of 2D planar transformations [48]

A similarity transformation is an Euclidean transformation (rotation, translation) combined with an abritrary scale factor s:

$$\mathbf{x}' = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \; . \tag{2.20}$$

Hence, the similarity transformation has four degrees of freedom. Both transformation groups can be computed from two point correspondences.

An affine transformation is a non-singular linear transformation, followed by a translation. It has the following block form:

$$\mathbf{x}' = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} \; , \tag{2.21}$$

where A is an invertible 2×2 matrix. This transformation can be computed from three point correspondences. It has six degrees of freedom, one per matrix element of A, and two for the translation in x and y direction.

A planar projective transformation is a general non-singular linear transformation which is represented by a non-singular 3×3 matrix

$$\mathbf{x}' = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x} \; . \tag{2.22}$$

$H = [h_{ij}]$ is a 3×3 homogenous matrix. As only the ratio between the nine matrix elements is significant (excluding an arbitrary scaling factor), the transformation has eight degrees of freedom. Projective transformations are computed from at least four point correspondences, with no three points collinear on a plane.

Group	Matrix	DOF	Invariant properties
Euclidean	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	3	Length, area
Similarity	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	4	Ratio of lengths, angles
Affine	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	6	Parallelism, ratio of length of parallel line segments, ratio of areas
Projective	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$	8	Collinearity, cross ratio (ratio of ratio of lengths)

Table 2.1: Hierarchy of 2D planar transformations. Transformations higher in the table inherit all invariant properties of the ones below. Each simpler transformation is a subset of the more complex group below it.

2.3.1.2 Camera Models

Under the pinhole camera model, a 3D world point $\mathbf{X} = (X, Y, Z)$ is mapped to a 2D image point $\mathbf{x} = (x, y)$ through a pinhole at the camera origin C onto a 2D projection plane a distance f (focal length) along the Z axis [48],

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \ . \tag{2.23}$$

Figure 2.12 shows the so-called central perspective projection which can be expressed as a linear mapping between the homogenous world coordinates and homogenous image points [22]:

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.24}$$

The center of projection (camera center) is also known as the optical center. The principal axis of the camera is the ray from the camera center perpendicular to the image plane (see Figure 2.12). The intersection of the principal axis with the image plane is called principal point. In practice, the origin of the image plane may not be located at at the principal point. Hence, the coordinates of the principal point $\mathbf{p} = (p_x, p_y)^T$ are considered in the

mapping function:

$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} . \tag{2.25}$$

This can be rewritten in concise form as

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{X} , \tag{2.26}$$

where

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} , \tag{2.27}$$

K is called the camera calibration matrix. The Euclidean coordinate system with the camera located at its origin and with the principal axis pointing straight down the Z-axis is called the camera coordinate frame. In Equation 2.26 it is assumed that the world point $\mathbf{X} = (X, Y, Z, 1)^T$ is expressed within this coordinate system (compare Figure 2.12). Typically, world points are expressed in terms of a different coordinate system, namely the world coordinate frame. The camera coordinate frame and the world coordinate frame are related via rotation and translation (see Figure 2.13), also known as camera extrinsics.

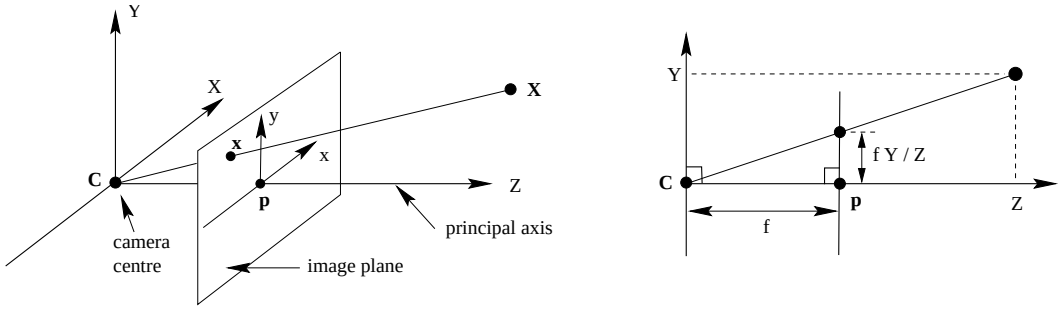


Figure 2.12: Pinhole camera geometry [22] with central projections of points in space onto the image plane. The camera center **C** placed at the coordinate origin. The principal axis intersects the image plane at the principal point **p**. The distance to the image plane is the focal length f.

If $\tilde{\mathbf{C}}$ represents the position of the camera center in world coordinates, and R defines a 3×3 rotation matrix, Equation 2.26 can be rewritten as

$$\mathbf{x} = KR[I| - \tilde{\mathbf{C}}]\mathbf{X} , \tag{2.28}$$

where $\mathbf{X}$ defines a point in the world coordinate frame. This general pinhole camera mapping has 9 degrees of freedom: 3 for K (f, p_x and p_y), 3 for R, and 3 for $\tilde{\mathbf{C}}$. The parameters of K are called the camera intrinsics, and those of R and $\tilde{\mathbf{C}}$ are the camera extrinsics, as already mentioned above. By introduction of a 3×4 camera projection

matrix $P = KR[I| - \tilde{\mathbf{C}}]$, Equation 2.28 can be written as

$$\mathbf{x} = P\mathbf{X} \ . \tag{2.29}$$

The camera projection matrix maps homogenous 3D world coordinates to homogenous 2D image coordinates in the image plane. The camera matrix can also be written as

$$P = K[R|\mathbf{t}] \tag{2.30}$$

by not making the camera center explicit. In this case the world-to-camera coordinate transformation (see Figure 2.13) is given by $\mathbf{X}_{cam} = R\mathbf{X} + \mathbf{t}$.

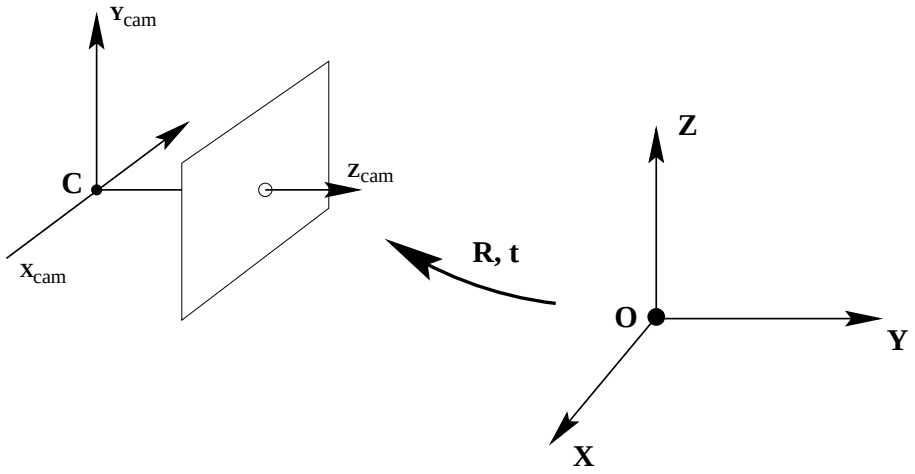


Figure 2.13: The transformation between world and camera coordinate frames [22]. They are related via rotation R and translation t.

Considering two images of a 3D scene from different cameras, a 3D point $\mathbf{X}$ is mapped to an image coordinate $\mathbf{x}_0$ in camera 0 through [48] the combination of a 3D rigid-body motion E_0,

$$E_0 = \begin{bmatrix} R_0 & t_0 \\ 0^T & 1 \end{bmatrix} \tag{2.31}$$

and a perspective projection P_0,

$$\mathbf{x}_0 \sim P_0 E_0 \mathbf{X} \ . \tag{2.32}$$

A pixel is mapped back to the 3D coordinate using

$$\mathbf{X} \sim E_0^{-1} P_0^{-1} \mathbf{x}_0 \ . \tag{2.33}$$

Hence, the mapping of an image coordinate in camera 0 to one in camera 1 can be formulated as

$$\mathbf{x}_1 \sim P_1 E_1 \mathbf{X} = P_1 E_1 E_0^{-1} P_0^{-1} \mathbf{x}_0 \ . \tag{2.34}$$

For a planar scene, the mapping equation can be reduced to a 8-parameter homography [48],

$$\mathbf{x}_1 \sim H_{10}\mathbf{x}_0 \ . \tag{2.35}$$

If a camera undergoes pure rotation (i.e. the camera is rotating around its center of projection), the homography for the mapping between two images I and J is given by [49]

$$H_{IJ} = K_I R_I R_J^{-1} K_J^{-1} \ . \tag{2.36}$$

Assuming that the focal length is given, it is sufficient to recover 3 parameters for the rotation matrix R, instead of estimating an 8-parameter homography, in order to obtain a relation between images. This motion model is also known as 3 parameter rotational model [44, 49] to create rotational panoramas.

2.3.2 Image Registration

Once a motion model has been chosen which represents the alignment between a pair of images, a method to estimate its parameters has to be applied [48]. In general there are two different categories for image registration: a) direct methods (Section 2.3.2.1) and b) feature-based methods (Section 2.3.2.2). Direct methods rely on the whole image data, while feature-based methods use distinctive image features only. As direct methods use all available image data, they can provide precise registration. But they are limited to a close initialisation. Feature-based registration does not require a close initialisation and has further advantages [10]. The scope for panoramic image generation is enlarged by using rotation- and scale-invariant features. Input images can be acquired under different rotation, zoom and illumination changes. Feature matching enables also the processing of unordered image sequences from different data sets, while in direct methods the images should be sequentially sorted. In addition to a local image-by-image registration, a global registration (Section 2.3.2.3) is an optimization step which tries to minimize a potential cumulative alignment error over a set of images.

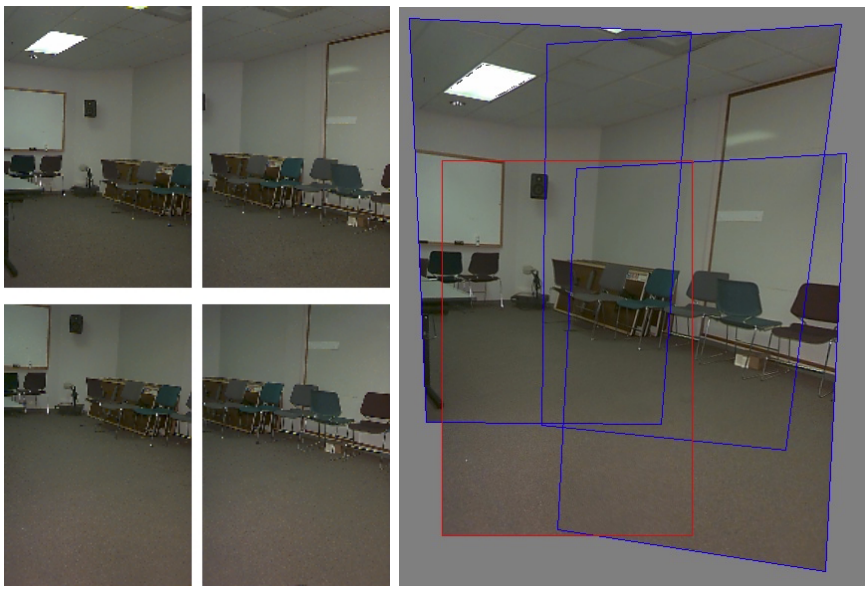
2.3.2.1 Direct Registration

Direct registration methods use a pixel-to-pixel matching over all available image data. A suitable error metric and search technique is used to compare images [48]. The simplest way to find an alignment is to shift one image over another one, and to minimize a least-squares function like the sum of squared differences (SSD)

$$E_{SSD}(\mathbf{u}) = \sum_i [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2 \ . \tag{2.37}$$

The goal is to find the location of image I_0 in image I_1, sampled at discrete pixel coordinates $\mathbf{x}_i = (x_i, y_i)$ and shifted by a displacement vector $\mathbf{u} = (u, v)$. In practice,

parameters for exposure differences are added to the error function. Moreover, the error function should be robust to reject outlier (noisy) pixels.



(a) (b)

Figure 2.14: 3D rotation registration (adopted from [44]). Four images (a) are aligned via 3D rotation registration. The 3-parameter rotational model is based on a pixel-to-pixel mapping over available image data, where a homography is estimated by minimizing an error function. Aligned images can be blended into a mosaic (b).

For example, in [52] a projective matrix H for two overlapping images is estimated by minimizing an error function, using gradient-based optimization:

$$E = \sum_{i \in A} (sI_1(\mathbf{x}_i) + b - I_0(H\mathbf{x}_i))^2 \ . \tag{2.38}$$

Here, $\mathbf{x}_i$ represents homogenous coordinates $(x_i, y_i, z_i)^T$, s and b are parameters for exposure differences, and A is the overlap area. The projective matrix H is initialized by internal and external camera parameters (see section 2.3.1.2). The camera motion is restricted to be rotational only (see Equation 2.36). The gradient-based optimization is combined with a correlation-based linear search, and is performed on subsequent finer levels of Gaussian pyramids. An example of a 3D rotation registration based on a pixel-to-pixel mapping is shown in Figure 2.14. An early direct image registration approach is the work of Lucas and Kanade [33] (see also Section 2.2.3), where a patch-based translational alignment (optical flow) technique is used.

2.3.2.2 Feature-based Registration

Feature-based registration extracts distinctive image features, and tries to find matchings between image pairs. In contrast to direct registration, close initialization is not required. Feature-based approaches show several advantages [10]. By using rotation- and scale-invariant features the scope for panoramic image generation is enlarged. Input images can be acquired under different rotation, zoom and illumination changes. Feature matching enables also the processing of unordered image sequences from different data sets.

An example for feature-based registration methods can be found in [10]. SIFT (see section 2.1.3) features are extracted and matched between all images. The images with the highest number of feature matches are potential candidates for an image match. By means of the feature correspondences a robust homography estimation is performed by using RANSAC (see section 2.3.3). It is supposed that each image undergoes affine transformation. Connected sets of image matches and their corresponding transformations are the output of the registration stage. Figure 2.15 depicts the stages of feature-based registration. In Figure 2.16 stages with unordered image data as input are shown.

2.3.2.3 Global Registration

Multiplying homographies (or any other kind of transformations) of pairwise registration can cause cumulative errors. Supposed that $H_{(i-1)i}$ is the homography of the i-th to the the $(i-1)$-th image, the global transformation H_{1i} from image i relative to the first image is calculated as

$$H_{1i} = H_{12}H_{23}...H_{(i-1)i} \ . \tag{2.39}$$

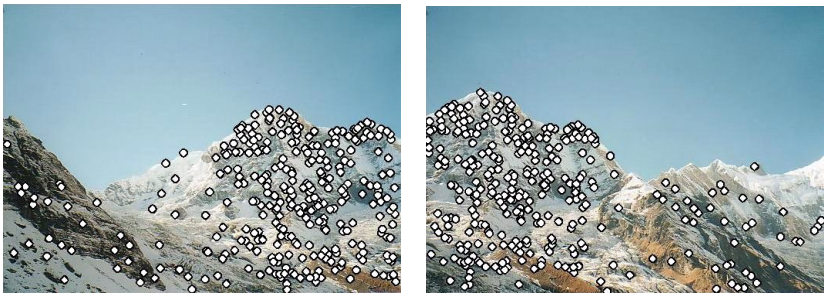
Global registration is a way to compensate for cumulative registration errors. The iterative global alignment [17] is based on the minimization of a cost function, which relates points of the mosaic with corresponding points in the source images. The cost function is defined as

$$e = \sum_{m=1}^{M} \sum_{\mathbf{x}_{i_k} \in \mathcal{N}_m} \| \mathbf{x}_{i_k} - H_{ij}\mathbf{x}_{j_m} \| \ . \tag{2.40}$$

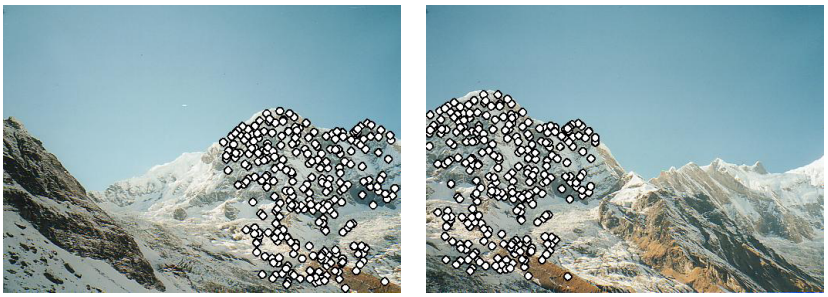
The k-th interest point $\mathbf{x}_{i_k}$ of image i is related to a point $\mathbf{x}_{j_m}$ in the mosaic by H_{ij}. All interest points that correspond to the same interest point in the mosaic are defined as the set $\mathcal{N}_m$. The minimization is solved by altering two linear steps iteratively. First, the real position of the feature points is estimated. In the second step, homographies are recalculated. An advantage of this approach is that it avoids non-linear optimization.

In [10] bundle adjustment is used to solve for accumulated errors. The following error function is a non-linear least squares function, which is solved with the Levenberg-Marquardt algorithm:

$$e = \sum_{i=1}^{n} \sum_{j \in \mathcal{I}(i)} \sum_{k \in \mathcal{F}(i,j)} h(r_{ij_k}) \ . \tag{2.41}$$



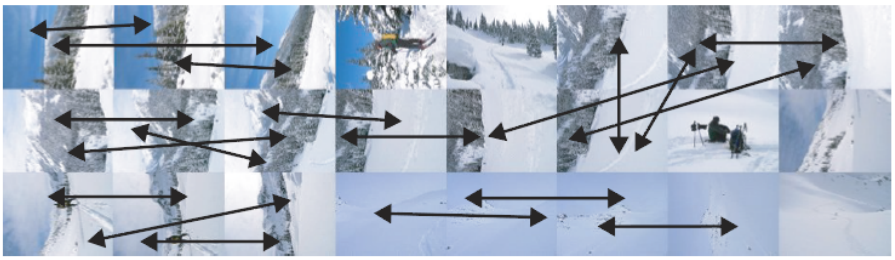
(a)



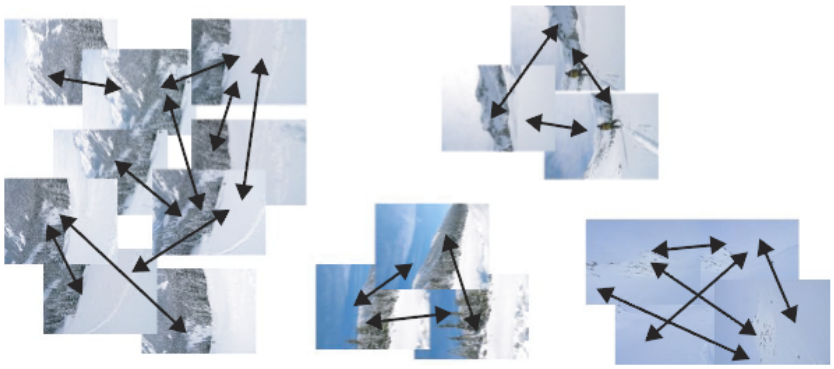
(b)

(c)

Figure 2.15: Image registration using SIFT features (adopted from [10]). SIFT features are extracted and matched (a). Using RANSAC, a homography is estimated and inliers are determined (b). The aligned image is shown in (c). Note: The visible seams in (c) can be removed with adequate blending strategies (see section 2.4).



(a)



(b)

(c)

Figure 2.16: Image matching (adopted from [10]). Consistent image matches are found in an unordered set of input images (a). Connected components of image matches (b) can be stitched into panoramas (c).

$\mathcal{I}(i)$ is the set of images matching to image i, and $\mathcal{F}(i,j)$ is the set of features between image i and j. A robust sum of squared errors function h is applied on the residual errors r over all n images. The residual r_{ij_k} is the difference between the position of feature k in image i and the projection of feature k from image j to i,

$$r_{ij_k} = \mathbf{x}_{i_k} - H_{ij}\mathbf{x}_{j_k} \tag{2.42}$$

Figure 2.17 shows an image mosaic before and after global alignment. It is obvious that (cumulative) registration errors are reduced after global alignment.

(a)

(b)

Figure 2.17: Image mosaic (a) before and (b) after global alignment (adopted from [17]). Before global alignment visible artifacts are visible (primarily around the tower).

2.3.3 Motion Estimation

Motion estimation is the task of establishing a geometric mapping between corresponding points of overlapping images. The calculation method depends on the underlying motion model. In particular, this section deals with the estimation of 2D planar transformations, which map points from one image to corresponding points in another image. For the computation of an 8-parameter homography or a 6-parameter affine transform, a direct linear transformation (Section 2.3.3.1) can be used. Euclidean (rigid) motion can be estimated by Procrustes alignment (Section 2.3.3.2). To be robust against outliers, a RANSAC procedure (Section 2.3.3.3) is applied. By means of a robust estimation of the geometric relation between a point set, an image match can be verified, because enough consistent point correspondences (inliers) are given.

2.3.3.1 Direct Linear Transformation

The direct linear transformation (DLT) [22] is an algorithm to estimate a 2D homography H from $n \geq 4$ 2D to 2D point correspondences that satisfy $\mathbf{x}'_i = H\mathbf{x}_i$. Typically, the normalized version of DLT is used, where point correspondencees are normalized before a homography is computed. The normalized version of DLT can be summarized as follows [22]:

Algorithm 2.1 Direct Linear Transformation (DLT)

1: Normalize $\mathbf{x}$ and $\mathbf{x}'$ by computing a similarity transformation T (and T') for both point sets independently:
(i) Translate the points so that the centroid of the points is at the origin $(0,0)^T$
(ii) Scale the points so that their average distance from the origin is $\sqrt{2}$

2: Compute the matrix

$$A_i = \begin{bmatrix} 0 & 0 & 0 & -x_i w'_i & -y_i w'_i & -w_i w'_i & x_i y'_i & y_i y'_i & w_i y'_i \\ x_i w'_i & y_i w'_i & w_i w'_i & 0 & 0 & 0 & -x_i x'_i & -y_i x'_i & -w_i x'_i \end{bmatrix}$$

for all normalized point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. A point $\mathbf{x}_i$ is a 3-vector of homogenous coordinates: $\mathbf{x}_i = (x_i, y_i, w_i)^T$.

3: Assemble n 2×9 matrices A_i into a single $2n \times 9$ matrix A.

4: Perform a SVD of A, $USV^T = A$, and obtain the 9-vector $\mathbf{h}$. The solution for $\mathbf{h}$ is the last column of V which is the unit singular vector corresponding to the smallest singular value.

5: Determine the matrix $H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$ from $\mathbf{h}$ with h_i the i-th element of $\mathbf{h}$.

6: Denormalize H: $H = T'^{-1}HT$

2.3.3.2 Procrustes Alignment

The generalized procrustes analysis was published by Gower [19] in 1975. Based on this approach two shapes, i.e. corresponding 2D point sets, can be aligned. It is a kind of least-squares solution to estimate a Euclidean transformation (rotation and translation). The two point sets are normalized with respect to their center of gravity (centroid). By means of the correlation matrix between the two point sets, a singular value decomposition is performed. The rotation between the point sets can be obtained from the singular value decomposition. The alignment procedure [16] is summarized in Algorithm 2.2.

Algorithm 2.2 Procrustes Alignment

1: Calculate the center of gravity for each point set:
$\overline{\mathbf{x}}_i = (\frac{1}{n}\sum_{j=1}^{n} x_j, \frac{1}{n}\sum_{j=1}^{n} y_j),\ i = 1, 2$

2: Align each point set to the center of gravity: $\overline{X_i} = X_i - \overline{\mathbf{x}},\ i = 1, 2$

3: $A = X_1 X_2^T$

4: Perform a SVD of A: $USV^T = A$

5: Get determinant d of (VU^T) and build matrix $D' = \begin{bmatrix} d & 0 \\ 0 & d \end{bmatrix}$

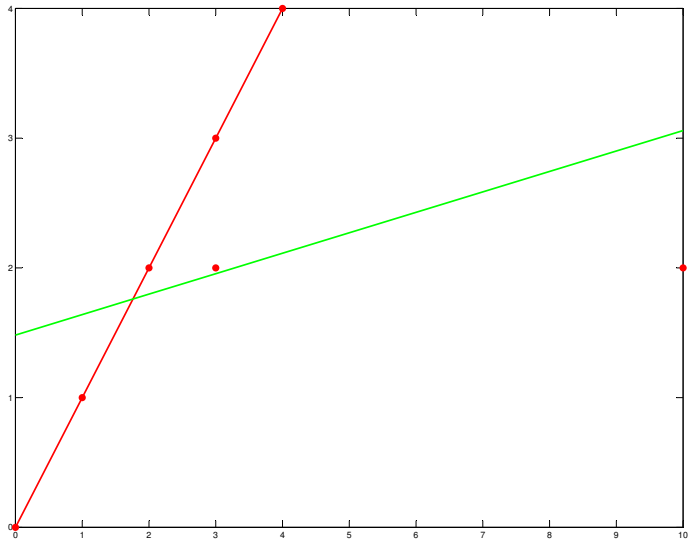
6: Calculate rotation $R = VD'U^T$

7: Get translation $\mathbf{t} = \overline{\mathbf{x}}_2 - R\overline{\mathbf{x}}_1$

2.3.3.3 RANSAC

The Random Sample Consensus (RANSAC) was introduced by Fischler and Bolles [18] for model fitting of experimental data. First, it is a classification problem to find the best match between one of the available models and the data. Second, it is a parameter estimation problem where the best values for the free parameters of the selected model should be computed. Instead of using an initial solution as large as possible, the RANSAC procedure uses an initial data set which is as small as acceptable. The data set is extended with consistent data when possible.

For example, if there is the task to fit a line through a set of points (see Figure 2.18), a subset of two points are selected by RANSAC to estimate the model. Two points are necessary to calculate a line. The line is generated based upon the two points, and points which are close enough to the line are determined. If this number of points is large enough, a set of consistent points has been found. The RANSAC paradigm is outlined in Algorithm 2.3.



Point	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

Figure 2.18: RANSAC illustration (adopted from [18]). Given the task of finding the best fit line, RANSAC terminates with the best possible consensus set of six data points. The six points (inliers) are fit by the red line. Point 7 is rejected as outlier. A least squares solution would provide an unsatisfactory solution (green line).

Algorithm 2.3 RANSAC procedure

1: **while** number of trials N is below a predefined maximum **do**
2: Select a subset $S1$ of n random data points from a set of data points P where n is the minimum number of required points to instantiate the free parameters of a given model and the number of points in P has to be greater than n
3: Instantiate the model $M1$ with $S1$
4: Calculate the subset $S1*$, the consensus set of $S1$, which are the points in P that are within a specified error tolerance ξ of the instantiated model $M1$
5: **if** the number of points in $S1*$ is greater than some predefined threshold κ **then**
6: break (suitable consensus set found)
7: **end if**
8: **end while**
9: Re-solve the model with the best consensus set

The RANSAC technique contains three parameters to control the model estimation process:

1. The error tolerance ξ, which is used to determine the inliers (compatible points) of a model

2. The number of inliers κ, which are necessary to implicate a correct model

3. The maximum number of iterations N

The maximum number of iterations can be determined adaptively [22]. The number of iterations N is chosen high enough, to ensure at probability p that at least one random sample of n points provides no outliers. The probability p is usually set to 0.99. Supposing that q is the probability that any selected data point is an outlier, then N is updated after every iteration by:

$$\begin{aligned} q &= \frac{1 - \text{number of inliers}}{\text{total number of points}} \\ N &= \frac{\log(1-p)}{\log(1-(1-q))^n} \end{aligned} \tag{2.43}$$

The number of inliers to implicate a correct model has not to be defined as the algorithm terminates if N iterations are reached.

2.3.4 Image Warping

Image warping defines a mapping of an image with given transformation, constraints or functions. If the transformation from one image to another is known, warping of the first image to the image space of the second one is possible. A geometric transformation [45] is applied, which is generally speaking a function T, that maps pixel coordinates (x, y) to a new position (x', y'). A geometric transformation includes on the one hand a pixel coordinate transformation, and on the other hand pixel brightness interpolation. The pixel coordinate transformation maps a point from the input image to a point in the output image. As the new point coordinates (x', y') do not generally fit the discrete raster of the output image a brigthness interpolation is required. Hence, the transformation T is done inverse, so that the point in the output raster is located on the discrete integer grid:

$$(x, y) = T^{-1}(x', y') \tag{2.44}$$

The brightness for the real coordinate (x, y), which corresponds now to a discrete point in the output image has to be interpolated. The interpolated brightness value is assigned to the discrete point coordinate (x', y'). There exist several interpolation methods that estimate the brightness value for non-discrete image points, e.g.:

- Nearest neighbor interpolation assigns the brightness value of the nearest neighboring point in the discrete raster.
- Bilinear interpolation uses a linear function on the four neighboring points to estimate a brightness value.
- Bicubic interpolation uses sixteen neighboring points and a bicubic interpolation kernel to approximate a brightness value.

Bicubic interpolation provides best results, nearest neighbor interpolation worst. Nearest neighbor interpolation can create step-like boundaries after transformation on objects with straight line boundaries. The position error can be up to half a pixel. The linear interpolation can produce blurring due to its averaging effect. Bicubic interpolation does not suffer from problems like step-like boundaries and blurring. In the case of creating high quality panoramas bicubic interpolation is recommended for image warping.

2.4 Image Stitching and Blending Strategies

After images are registered, they have to be stitched into a final panorama (mosaic). In literature the term *image stitching* is related either to the whole process of panoramic image generation, or just to the final composition of a mosaic image when source images are already aligned. In both cases, it is a merging of several images into a single mosaic. For the generation of the final composite it has to be chosen on which surface (e.g. flat, cylindrical or spherical) the pixels should be mapped. Image warping (see Section 2.3.4) can be a pre-processing step, in order to align images or to warp images to a certain surface. With a blending strategy overlapping images are fused. The images can have different exposures and changing lighting conditions. Images can have noisy elements as well as moving objects. It may also be that images are not well aligned. For these reasons a highly sophisticated blending strategy is applied. This involves the selection of pixels which contribute to the final panorama and how to blend these pixels as good as possible [48]. The final result should have no visible seams and no leaving artifacts. Figure 2.19 illustrates the impact of a blending strategies onto image stitching. If input images with different exposure are stitched into a panorama, a seam might be visible. Blending strategies address this issue and e.g. result in a smooth transition (as shown in Figure 2.19 (d)).

In the next sections different strategies to blend images are discussed. There are three main methods for image stitching [29]: a) optimal seam finding methods, b) smoothing the transition methods, and c) stitching in the gradient domain.

2.4.1 Optimal Seam Finding

Assuming that two images are aligned, an optimal seam finding algorithm searches for the optimal path in the overlap region, where the difference between the two images is optimal. The optimal seam divides the overlapping section into two regions. The final panorama is assembled from a set of regions where each pixel is sampled from a single source image. An example for an optimal seam algorithm can be found in [14]. The relative difference between two images is calculated as follows:

$$D = \frac{|I_i - I_j|}{\max(I_i - I_j)} \ . \tag{2.45}$$

(a) (b)

(c) (d)

Figure 2.19: Image blending. Two input images (a, c) with different exposure are stitched into a panorama image. A bad stitching result with visible seams (b), as well as a good blended image with smooth transition (d) are shown.

If a new source image is added to the mosaic, the difference image (equation 2.45) is created for the overlapping region. Dijkstra's algorithm is applied to find the shortest low intensity path on the low pass filtered difference image. On one side of the path the pixels of the mosaic are preserved. On the other side, previous information is discarded and pixels of the new image are used. The path usually does not go through moving objects. Hence, a moving object falls into one region provided that there exists at least one image with a complete view of the moving object. Effects like blurring from misalignments and ghosting from moving objects are avoided (Figure 2.20). The final panorama is divided into regions where every region is related to a single source image.

(a)

(b)

Figure 2.20: Mosaic of a scene with moving objects [14]. (a) Blending of a scene with moving objects can produce blurring and ghosting. (b) By using an optimal seam strategy the mosaic is divided into regions. Every region is related to a single source image, which results in a single focused mosaic image.

2.4.2 Transition Smoothing

These methods minimize visible seam artifacts by smoothing the transition between overlapping images. Two related approaches are pyramid blending and feathering. While feathering uses a weighted combination of the input images to create a smooth transition, in pyramid blending the images are blended on different frequency bands. Pyramid blending and feathering can also be combined as blending strategy.

2.4.2.1 Pyramid Blending

A very well known and often used blending algorithm is pyramid blending introduced by Burt and Adelson [12]. Pyramid blending is also known as multi-band blending [10]. The idea is to decompose the images into different frequency bands and to blend the images level by level with successive blurred weighting masks. The pyramid blending algorithm is based on the Laplacian pyramid.

Laplacian Pyramid. The Laplacian pyramid [11] represents different scales of an image. The first step is the generation of a Gaussian pyramid. A Gaussian pyramid is a set of images where an original image (level 0) is low-pass filtered and down-scaled (by a factor of 2) level-by-level (see Figure 2.21). The Laplacian pyramid is generated from a Gaussian pyramid. A level in the Laplacian pyramid is difference between two adjacent Gaussian pyramid levels: The expanded image at level $(i+1)$ is subtracted from the image at level i. Figure 2.22 shows an example for a Laplacian pyramid. The last level of the Laplacian pyramid is identical with the last one of Gaussian pyramid.

Figure 2.21: First six levels of a Gaussian pyramid. Each higher level is down-scaled by a factor of 2 the dimensions of its predecessor. The size of the original image is 512×512. The last level has a dimension of 16×16.

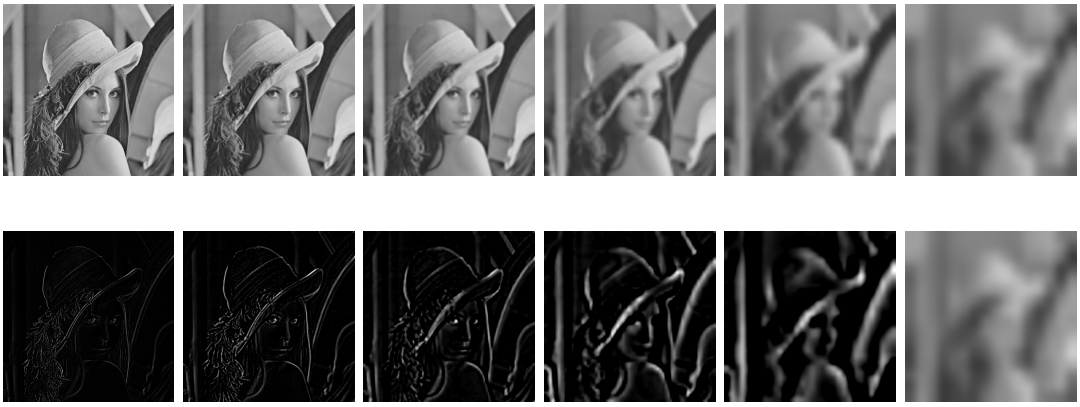


Figure 2.22: Laplacian pyramid with six levels. The upper row shows six expanded levels of the Gaussian pyramid which are used to generate the Laplacian pyramid. Two adjacent levels are subtracted to build a level of the Laplacian pyramid on the lower row. The last level of the Gaussian and Laplacian pyramid are identical.

Pyramid Blending Algorithm. With this technique an image I can be fused with a second image J under consideration of a weighting map W. In the following, W is assumed to be a binary image where all pixels inside the region of I are set to 1 and all remaining pixels are set to 0. This algorithm can also be adapted to more sophisticated weighting maps or weighting functions. The pyramid blending algorithm can be summarized as follows:

- Build Laplacian pyramids L_I and L_J for images I and J.
- Build a Gaussian pyramid G_W for image W.
- Form a combined pyramid L_S from L_I and L_J using G_W: $L_{S_l}(i,j) = G_{W_l}(i,j)L_{I_l}(i,j) + (1 - G_{W_l}(i,j))L_{J_l}(i,j)$
- Get the blended image S by expanding and summing the levels of L_S.

Figure 2.23 (d) shows how a smooth transition can be achieved with multi-band blending. For RGB images the blending process is repeated for each color channel.

2.4.2.2 Feathering

In feathering a mosaic is generated as a weighted combination of input images. In [49] a feathering algorithm is applied, where the pixels in each image are weighted proportionally to their distance to the nearest invisible pixel. Thereby, intensity and color discontinuities between the images being stitched are reduced. For the generation of a weighting map, a distance transform [3] with regard to the nearest invisible pixel can be adapted. The distance transform assigns to each pixel of a binary image the (Euclidean) distance to the nearest black pixel.

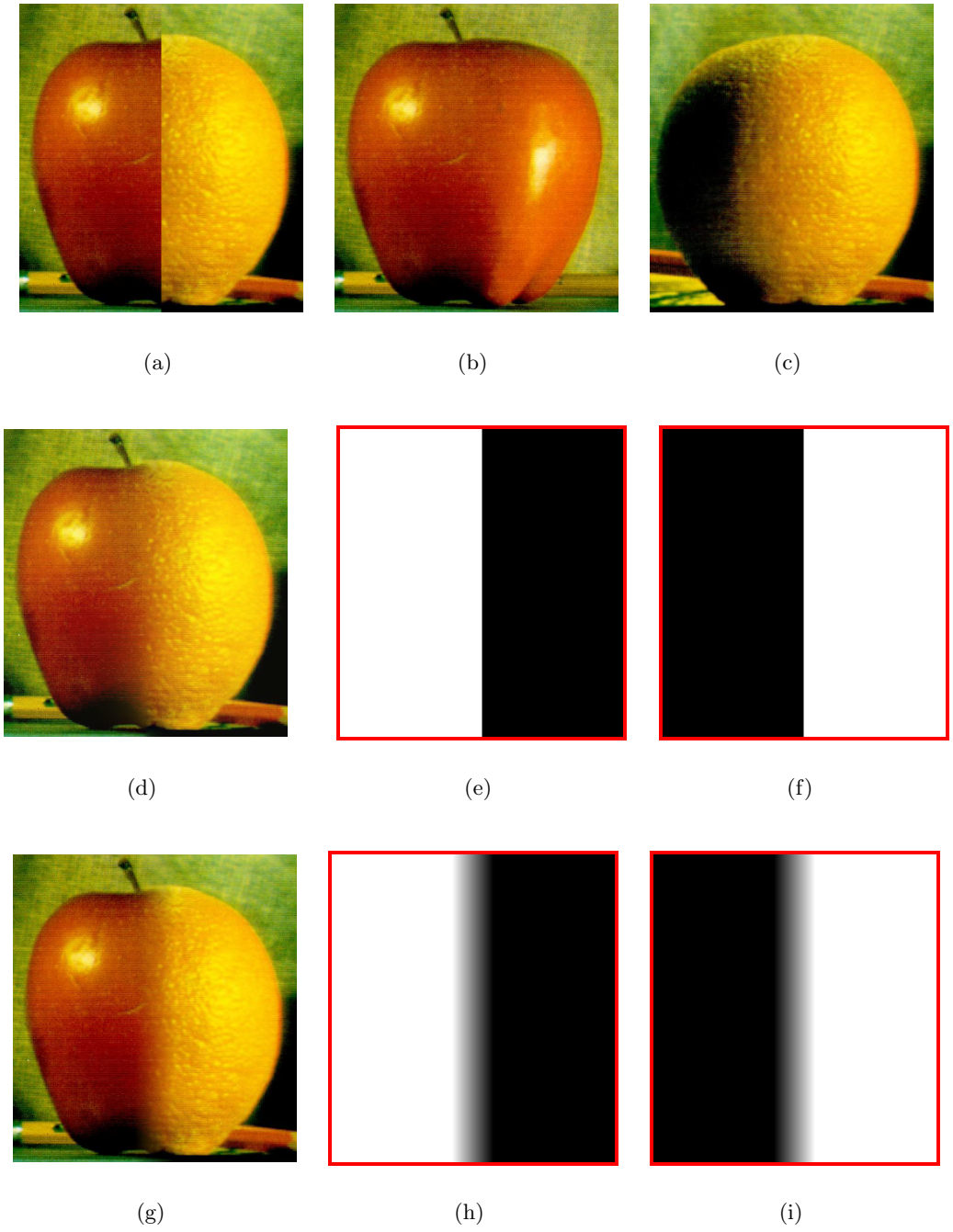


Figure 2.23: Feathering and pyramid blending. The left half of an apple (b) is fused with the right half of an orange (c): Without a blending strategy a seam is visible as in (a). Figure (d) shows an image stitched with pyramid blending by using the weights in (e) and (f). The image in (g) is created with feathering by using the weights in (h) and (i). The two input images are taken from [12].

Feathering can also be combined with pyramid blending, where spatially varying weighting masks are composited on different frequency bands. Figure 2.19 (d) shows a sample a panorama, generated by applying pyramid blending combined with feathering. Figure 2.23 presents images which are combined with feathering and their corresponding feathering weights.

2.4.3 Stitching in the Gradient Domain

Another approach for image stitching is to stitch images in the gradient domain [29]. Levin et. al present two different methods:

In the first one, a cost function over image derivatives is optimized. The goal is to minimize a dissimilarity measure E_p between the derivatives of two aligned input images ∇I_1 and ∇I_2, and the derivative of the stitched image $\nabla \widehat{I}$:

$$\begin{aligned} E_p &= d_p(\nabla \widehat{I}, \nabla I_1, \tau_1, W) \\ &\quad + d_p(\nabla \widehat{I}, \nabla I_2, \tau_2, U - W) \\ &\quad + d_p(\nabla \widehat{I}, \nabla I_1, \omega, W) \\ &\quad + d_p(\nabla \widehat{I}, \nabla I_2, \omega, U - W) \end{aligned} \tag{2.46}$$

where τ_i is the region viewed in I_i exclusively, ω is the overlap region, W is a weighting mask, U is a uniform image and $d_p(J_1, J_2, \upsilon, W)$ is the weighted distance between the two derivatives J_1 and J_2 on υ, defined by

$$d_p(J_1, J_2, \upsilon, W) = \sum_{\mathbf{q} \in \upsilon} W(\mathbf{q}) \left\| J_1(\mathbf{q}) - J_2(\mathbf{q}) \right\|_p \tag{2.47}$$

with $\|.\|_p$ defining the p-norm,

$$\left\| \mathbf{x} \right\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} . \tag{2.48}$$

The cost function is minimized with respect to $\widehat{I}$.

In the second approach the derivatives of the input images are stitched as follows:

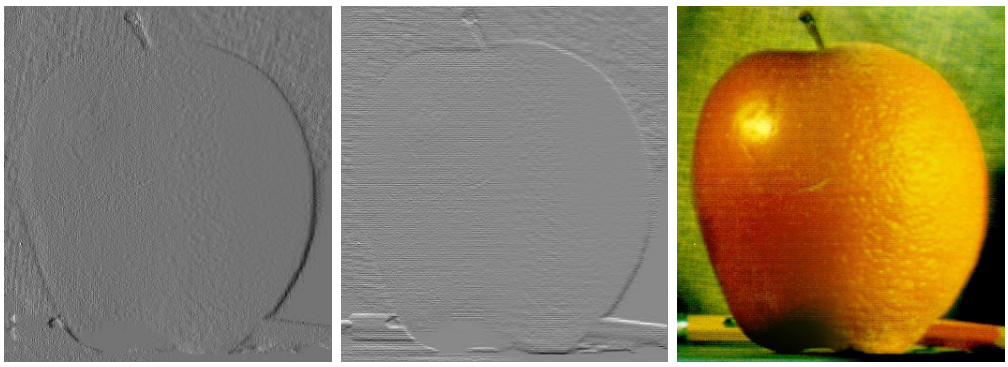
Algorithm 2.4 Stitching Derivative Images

1: Calculate the derivatives for I_1 and I_2: $\frac{\partial I_1}{\partial x}, \frac{\partial I_2}{\partial x}, \frac{\partial I_1}{\partial y}, \frac{\partial I_2}{\partial y}$

2: Stitch $\frac{\partial I_1}{\partial x}$ and $\frac{\partial I_2}{\partial x}$ to obtain F_x, and stitch $\frac{\partial I_1}{\partial y}$ and $\frac{\partial I_2}{\partial y}$ to obtain F_y. The stitched derivatives form a field $F = (F_x, F_y)$.

3: Minimize $d_p(\nabla \widehat{I}, F, \xi, U)$ (see Equation 2.47), where ξ is the total image area and U is a uniform image.

In step 3, the image $\widehat{I}$, whose gradients are closest to F, has to be found. This can be achieved by solving a Poisson equation for F: $\Delta\widehat{I} = \mathrm{div}(F)$. Figure 2.24 shows an example for stitching in the gradient domain. Calculated gradient images are composed with feathering and solved as Poisson equation.



(a) (b) (c)

Figure 2.24: Stitching in the gradient domain. The left half of an apple is fused with the right half of an orange. Calculated gradient images are composed with feathering, which results in a fused gradient image in x direction (a) and y direction (b). Both gradient images are normalized between 0 and 1 for the visualization. By solving a Poisson equation the stitched image (c) is obtained.

Chapter 3

Related Work

Contents

In this chapter an overview of approaches related to image stitching is presented. In literature, the term *image stitching* is related either to the whole process of panoramic image generation or just to the final composition of an image mosaic when source images are already aligned. There are several issues which should be considered for a system to create panoramic images. In [14] the following aspects are mentioned for the design of an image stichting system:

- A method for pairwise registration is needed.
- There can be difficulties if a large set of images is registered, e.g. small registration errors between pairs of images are accumulated.
- Moving objects are problematic.

Furthermore, the final panorama should be stitched without visible seams, which is done by an appropriate image blending strategy. In general, there are two main tasks concerning the generation of a panorama: a) registration and b) blending. In registration one image is aligned to another one, while blending is responsible for creation of a smooth transition between images, when they are stitched together.

3.1 Registration

Image registration algorithms can be either direct or feature-based. Direct methods use all available pixel to align images. Feature-based methods on the other hand solely rely on interest point correspondences between images. Furthermore, global registration methods

are often used to minimize cumulative registration errors due to local (pairwise) registration (see e.g. [10, 13, 14, 17, 44, 47, 49]). Many image stitching algorithms are based on the fact that the images are acquired from a stationary center of projection and that the camera motion is restricted to be rotational only [10, 14, 39, 44, 49, 52].

Mainly earlier approaches rely on direct methods such as an approach by Szeliski [47], where local image registration is performed to make statistically optimal use of all information. This technique minimizes the sum of squared intensity errors over all corresponding pairs of pixels i inside two overlapping images $I(x, y)$ and $I'(x', y')$ (by means of the Levenberg-Marquardt iterative nonlinear minimization algorithm). Furthermore, a global image registration is performed, where two techniques are applied: a) hierarchical matching, and b) phase correlation which can be used for larger misregistration.

There are approaches which use a 3 parameter rotational model [44, 49, 52] for image registration. To create full view panoramic image mosaics, 3D rotations are recovered instead of general projective 8-parameter transforms [49]. To apply this technique, first the focal length has to be estimated. Next, a 3D point $\mathbf{p}$ is related to its image coordinates $\mathbf{x}$: $\mathbf{p} = R^{-1}V^{-1}\mathbf{x}$, where V is the focal length scaling matrix and R is the 3D roation matrix. The problem of closing a gap due to cumulative registration errors is also adressed by using a refinement step, where one image is registered at the beginning and the end of the image sequence. The amount of misregistration can be obtained by the difference of computed rotation matrices. Finally, the input images with their associated transformations are mapped into spherical coordinates.

In the image mosaicing system proposed by Shum and Szeliski [44] a geometric transformation is associated with each input image. A parametric motion model is applied for the image alignment. The parametric motion estimation is performed between the current new input image and a warped version of the image mosaic. If it is known that the camera rotates around its optical axis, the 3-parameter rotational model is preferred. Nevertheless, the approach can be also used with different motion models (e.g. the 8-parameter model). The system utilizes patch-based alignment where the image is divided into subpatches, and a correlation-style search component is added to the registration algorithm, in order to be more robust. In a next step a global optimization technique is used to find the optimal overall registration and to avoid accumulated registration errors. In this way the misregistration between all overlapping pairs of images is minimized. The global alignment algorithm is feature based (as point correspondences between overlapping images should be found by dividing each image into patches) and the minimization is formulated as a constrained least squares problem. To overcome small amounts of motion parallax, local motion estimates (block-based optical flow) between pairs of overlapping images are computed.

Sawhney and Kumar presented an algorithm for true multi-image alignment that does not rely on a reference image being distortion free [42]. Each image is related to an ideal coordinate system through an interior camera transformation as well as an exterior view transformation. In addition to an 8-parameter plane projective transformation, a 1 or 2

parameter lens distortion transformation needs to be found. The optimization strategy uses progressively more complex models of motion. The optimization is divided into a sequence of steps, where the output from the previous step serves as an initial estimate for the next step:

- 2 unknown parameters (translation)
- 6 unknown parameters (affine transformtion)
- 8 unknown parameters (projective transformation) plus global lens distortion parameters

Another concept that is used for the optimization is coarse-to-fine minimization. Here the parameters are first estimated at the coarsest level of a Gaussian/Laplacian pyramid. The results are then used as an initial estimate for the next finer level. The coarse-to-fine technique is also used in approaches introduced in [15, 52].

Xiong and Turkowski [52] presented an approach with a calibration step between registration and blending. Camera motion is restricted to be rotational. In the pairwise registration step the projective 3×3 matrix given two overlapping images is estimated. The registration objective is solved by a gradient-based optimization on progressively finer levels of Gaussian pyramids. To be robust against exposure differences and large translations a combination of correlation-based linear search and a progressive damping of exposure parameters is used. In the calibration and global optimization step, camera internal and external parameters are extracted from the projection matrices. If the alignment results are unsatisfactory, registration and global optimization are iterated.

Dufaux and Moscheni [15] introduced a technique for background mosaicking. In their first step, background is segmented by using local motion estimation. In a second step, the camera motion is modelled by a parametric motion model. Here, the 8-parameter perspective model is used. To obtain the motion parameters, a disparity measure between a region in the current frame and the mapped region in the previous frame is minimized. The motion parameters are estimated progressively and with a coarse-to-fine technique using a Gaussian pyramid.

In [14] Davis assumes that images are acquired from a stationary center of projection. The pairwise registration is done with an extended version of the Mellin transform to use it for images which are related by a projective transform. In order to avoid cumulative registration errors, a global registration is carried out, where a linear system of equations, derived from the pairwise registration matrices, is solved.

Brown and Lowe [9, 10] presented a feature-based approach. The main advantage of direct methods is indeed that they provide very accurate registration as they use all available image data. A drawback though is that they require a close initialisation. In order to enable fully automatic panorama stitching, an invariant feature based approach (SIFT [32]) is used in [9]. It is not sensitive to the ordering, orientation, scale and illumination

of the input images. Furthermore, images which are not part of the panorama are even sorted out.

In [10], Brown and Lowe discuss and continue this approach. In the first stage of the automatic panorama generation algorithm, SIFT features are extracted and matched between all images (each feature to its k nearest neighbors by using a k-d tree with e.g. $k = 4$). Next, they find all matching images. Only a certain number of images that provide the largest number of feature matches are considered as potential image matches. In order to estimate a robust homography, RANSAC is applied. Each match is verified by applying a probabilistic model. The homography is computed between sets of $r = 4$ feature correspondences by using the direct linear transformation (DLT) method [22]. Connected sets of matching images build panoramic sequences. Bundle adjustment is used to overcome accumulated registration errors. A large benefit of the approach by Brown and Lowe is that input images can be in arbitrary order and orientation, and that panoramas are automatically recognised.

Agarwala et. al [1] present an approach to create multi-viewpoint panoramas. In a pre-processing stage radial distortion is removed, the camera projection matrices are recovered, and exposure variation is compensated for. The projection parameters are recovered by matching SIFT features [32] between pairs of input images. Obtained matches constrain the underlying optimization procedure.

Local misalignments, which appear as ghosting artifacts on stitched images, can be reduced by using local optical flow registration [39]. This approach uses a pixel by pixel registration. An update matrix is obtained for every pixel in the overlapping region of two images. A computational algebraic topology (CAT) approach of optical flow is applied to update the estimated transformation matrix of every pixel.

Elibol [17] et al. presented a new global alignment method for feature based image mosaicing. The registration is based on SIFT features. RANSAC is used to reject outliers. This new approach works on the mosaic frame and does not require non-linear minimization of an error term for global alignment. The iterative global alignment is based on the minimization of a cost function which relates points of the mosaic with corresponding points in the source images. It is solved by altering two linear steps iteratively.

A feature-based image registration is also introduced by Mills and Dudek [36]. SIFT features are matched and RANSAC is used to find the best consistent set of matches under a projective transformation. The projective transform is calculated on the consistent set of matches to map one image onto the other.

In [13] global registration is not just applied after the end of an image sequence, but every time a new image is submitted. In other words, the global registration is performed in an online manner. The adjustment is only made to the current image, and not to previously entered ones. The global registration relies on feature tracking. In every iteration, the transformation of the current image is revised due to tracked features in the overlapping area.

Zagrouba et al. [53] presented an image stitching method based on multifeature match-

ing. The proposed algorithm is divided into three levels: a) primitives extraction, b) primitives matching, and c) mosaic construction. In the beginning, each couple of successive images I_i and I_{i+1} is segmented into regions using a region growing algorithm. Interest points are extracted using a Harris detector. The primitives matching stage, which is partially invariant to translation, rotation and scale, starts with region matching. The matched regions are used to obtain a first estimation of the transformation (rotation angle and scale factor) between image I_i and I_{i+1}. The calculated transformation is applied on I_{i+1} to get RI_{i+1} which will be used for interest point matching. The Harris points are matched using zero-mean normalized cross correlation score and RANSAC. For the mosaic construction, the homography is calculated based on the matched points by solving an optimization problem using a QR-factorization technique followed by a relaxation algorithm to reject outliers.

A fast and robust feature-based method for planar image registration was introduced by Isgrò and Pilu [25]. For the interest point extraction SUSAN (smallest univalue segment assimilating nucleus) corners are used. Intensity-based normalized cross-correlation is adopted to find point correspondences between overlapping images. As the cross-correlation measure is only invariant to translation, a local deformation is included to take care about rotation. For the matching not the whole set of correspondences is determined. It is searched for three consistent matches defining an Euclidean transformation. Based on this early consensus, the estimated transformation is refined by cheaply searching for further matches. With enough matches given, a RANSAC procedure is applied to find and remove outliers.

The choice of an appropriate keypoint descriptor is also important for feature-based registration. Mikolajczyk and Schmid [35] present a performance evaluation of local descriptors. They compare descriptors like shape context, steerable filters, PCA-SIFT, differential invariants, spin images, SIFT, complex filters, moment invariants, and cross-correlation. They also propose a new descriptor, gradient location and orientation histogram (GLOH), which is an extension of the SIFT descriptor. The descriptors are compared for different challenges such as illumination changes, image rotation, scale changes, viewpoint changes, and image blur. The SIFT-based descriptors achieves the best results in most of their tests. GLOH performs best, closely followed by SIFT. In contrast, the results of cross-correlation are unstable.

3.2 Blending

Blending methods can be divided into a) transition smoothing methods [9, 10, 12, 47, 52, 53], b) optimal seam finding methods [14, 20, 36, 50], and c) gradient domain stitching methods [1, 2, 29].

A very well known and often referenced blending algorithm is pyramid blending developed by Burt and Adelson [12]. This blending algorithm is used and referenced in [9, 10, 52, 53]. Pyramid blending adresses the generation of a smooth transition.

There are different approaches to represent a panoramic scene [44, 47]. Panoramic scenes can be composited for instance

a) onto a planar viewing surface,

b) onto multiple planes where periodically a new base frame is chosen for alignment,

c) onto a cylindrical surface and

d) onto a spherical surface.

Figure 3.1 represents the most common kinds of panoramic images. Panoramas in spherical or cylindrical coordinates are limited to pure panning motion [44]. In [47] the world coordinates are mapped onto 2D cylindrical screen locations to represent a circular panorama. For blending, a weighted average is used where pixels near the center of each image contribute more to the final composite. This method is also known as 'feathering' in literature [48].

(a)

(b)

(c)

(d)

Figure 3.1: The most common types of panoramas: (a) multi-viewpoint panorama [1], (b) panorama in spherical coordinates [9], (c) planar panorama with images warped relative to a reference image [53], and (d) cylindrical panorama [44].

To avoid problems with overlaps of more than two images a labelling scheme can be used to obtain a blending mask where all overlaps are considered [52]. Every pixel in the panorama canvas is referenced to the source image which contributes most.

Ghosting is addressed in several approaches (e.g. [2, 14, 44, 50]). A characterization of ghosting is given in [50]. Ghosting is an effect which can appear, if objects in an image are not stationary. If there are moving objects, the overlap region contains different information from the contributing images. For example, if a person moves in an area of overlap of two images, the stitched image will contain a combination of the person and background from both images. Thus, the person will looks like a ghost and will appear on two locations. Figure 3.2 illustrates de-ghosting.

(a)

(b)

Figure 3.2: De-ghosting (adopted from [50]). The mosaic in (a) contains exposure and ghosting artifacts. (b) Applying exposure compensation and de-ghosting removes visible artifacts.

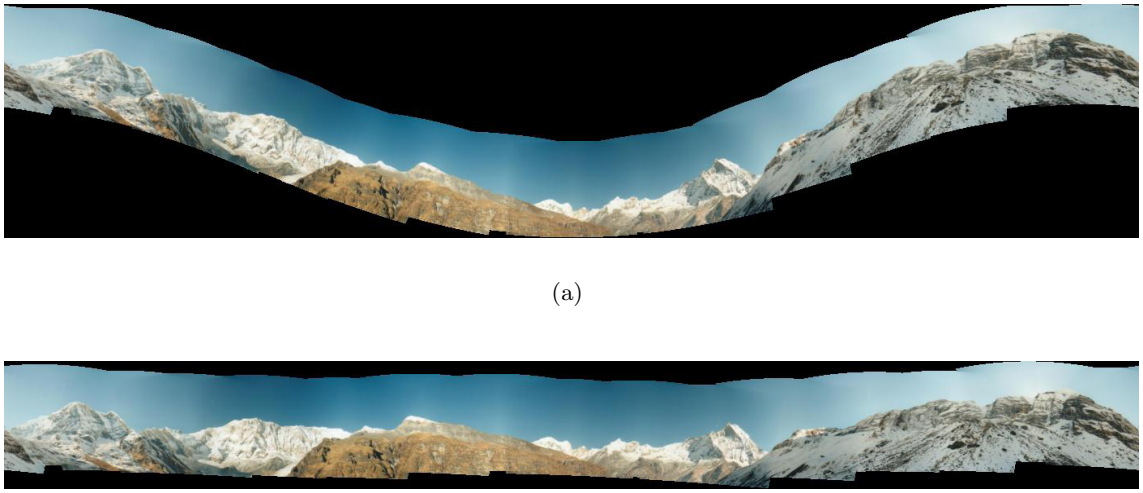
Davis [14] presents an approach to generate panoramic images of scenes with moving objects. The compositing algorithm divides the overlapping region between the existing mosaic and the source image that should be added to the mosaic. Dijkstra's algorithm is applied to find the shortest low intensity path on a low-pass filtered difference image of the overlapping section. The algorithm requires at least one image which contains a complete view of the moving object.

Uyttendaele et al. [50] introduce a work that addresses the elimination of ghosting and exposure artifacts in image mosaics. If exposure differences are not corrected, seams will appear in the panorama image. For exposure compensation, each image is divided into blocks and within each block a quadratic transfer function is applied to bring it more in line with its neighbors. To avoid ghosting, pixel values from only one image should be used for regions with moving objects. The first step of the de-ghosting approach is to determine where the movement occurs. A map is computed for each input image by identifying pixels which differ more than a certain threshold from pixels in overlapping images to get regions of differences (RODs). In a second step the image that is further used is determined. RODs are represented in a graph and corresponding RODs are linked by an edge. An exhaustive solver (for small graphs) and a randomized approximation algorithm (for large graphs) is used to find the minimum weight vertex cover of the graph which is removed. Then only a set of non-corresponding regions is left.

Brown and Lowe [10] include a preprocessing step called panorama straightening. The camera X vectors typically lie in a plane. A global rotation is applied after image registration so that the up-vector u is vertical in the rendering frame. This phase is called automatic panorama straightening. An example for straightening is shown in Figure 3.3. Next it is compensated for gain by using an error function which is the sum of gain normalised intensity errors for all overlapping pixels. In the last step, the panorama is composed using multi-band blending. A weighting function is assigned to each image, $W(x, y) = w(x)w(y)$. The weight $w(x)$ varies linearly from 1 at the center of the image to 0 at the boundary. For the blending mask i, the points for which image i has the highest weight are set to 1, and points where another image provides higher weights are set to 0. The output image is rendered in spherical coordinates (θ, ϕ). Limitations of this algorithm are that radial distortion is not modelled, and that moving objects are not detected.

In [53], also multi-band blending with a weighting function that varies linearly from 1 at the centre to 0 at the edge is applied to build the final mosaic. The mosaic warping also deals with more than two images. Therefore, the assumption is made that image I_i of a sequence only overlaps with images I_{i-1} and I_{i+1}. The reference image is the image in the middle of the sequence. Homographies are calculated accordingly.

Finding the optimal seam or smoothing the transition between images can also have disadvantages [29]. Optimal seam finding has the disadvantage that the seam is visible when there is a global intensity difference between the images. Smoothing the transition can leave artifacts in the mosaic image if there are misalignments between input images. Another approach is as mentioned above stitching in the gradient domain [2, 29].



(a)

(b)

Figure 3.3: Automatic panorama straightening (adopted from [10]). Panorama (a) without and (b) with automatic straightening. Positioning the up-vector u vertical in the rendering frame straightens the panorama.

Levin et. al [29] present two methods that address stitching in the gradient domain:

- GIST1 Optimizing a cost function over image derivatives: A dissimilarity measure between the derivatives of the stitched and the derivatives of the input images is minimized.

- GIST2 Stitching derivative images: e.g. pyramid blending or feathering on gradients.

According to experimental results presented in [29], it is recommended to use GIST1 under L1 norm.

A framework for interactive digital photomontage, which can be also applied for image stitching, is described in [2]. The user can interactively impact the result of the final composite. The framework is primarily based on two techniques. A graph-cut optimization is used to choose an optimal seam between aligned images. Furthermore, gradient domain fusion reduces any remaining visible artifacts. The framework supports also ghosting elimination.

Agarwala et. al [1] include a picture surface selection and a viewpoint selection. First, the coordinate system of the recovered 3D scene is defined. Then, a curve defining the image surface is drawn in the ground plane. The selection of the coordinate system can be done interactively or automatically via PCA. The system automatically computes the panorama using Markov Random Field optimization. A viewpoint is selected for each pixel in the output panorama. The final panorama is composited in the gradient domain

as presented in [2]. For interactive refinement, the system offers three different options: view selection, seam suppression and inpainting.

Another image stitching approach introduced in [26] uses structure deformation. Besides a smooth transition within the overlapping region of aligned images, the following requirements should be met when stitching images:

- The structure should be preserved, e.g. no misalignments.
- The intensity between images should be balanced.
- The context of objects in images should be considered.

Based on these characteristics, the proposed image stitching algorithm aligns not only intensities but also salient structures. It can be summarized as follows:

1. Optimal partitions are computed for the overlapping area of two images. A solution for an optimal seam is found by using the graph cuts method on a gradient alignment cost function which considers the gradient smoothness and penalizes pixel dissimilarity in the gradient domain.
2. Salient 1D features are detected along the optimal partitions and are matched by minimizing an energy function. Every matched feature pair is also associated with a deformation vector.
3. To align structure and intensity, the deformation vectors are propagated from the sparse 1D features to all other pixels.

An image blending algorithm using watersheds and graph cuts can be found in [20]. This approach is related to optimal seam finding. The intersection region of two images is segmented using the watershed transform which disjoints an input surface into a set of regions around local minima. The watershed algorithm is applied on the inverted, low-pass filtered difference image of the overlapping region to aggregate areas where the images are most dissimilar. The watershed segments are labelled in a binary mask according to the image they are assigned. The optimal labelling is found by minimizing a cost function which penalizes the dissimilarity along seams. Graph cuts are used to find a satisfactory solution. When dealing with more than two images, the mosaic space is divided into disjoint regions of image intersection (ROII), on which the presented pairwise image blending technique is performed independently.

Mills and Dudek [36] propose an image stitching algorithm that is also based on optimal seam finding. For the optimal seam selection an exposure correction is performed where gain and bias are adjusted between two images. For gain and bias computation only static pixels in the overlapping area are taken into account (no moving objects). The intensity and gradient difference is calculated for all pixels which are assembled in a combined difference image. The optimal seam is the minimum weight path in the combined difference

image, which is found using Dijkstras algorithm. For the blending step, a multi-resolution spline [12] is used, resulting in a Fourier decomposition of the source images and the mask with band-pass filters. The construction of the final image mosaic starts with the image pair that provides the best RANSAC inlier to outlier ratio. The next image, blended to the current mosaic, is the one with the best inlier to outlier ratio to one of the images which are already part of the panorama. Further images are selected according to the same criteria.

Flexible image blending [54] is an approach for incremental image blending of an image sequence. When a new image is added, the image mosaic is updated right away. To consider previous images that are already blended, the blending weights for the mosaic are accumulated in every iteration.

In contrast to image blending, in image fusion [38] multiple images of the same scene are combined. The goal of image fusion is to generate a new image of the same scene with considerably increased quality. Image fusion strategies also rely on multiple scale decomposition. A popular method for image fusion is the discrete wavelet transformation [38].

3.3 Summary

Image stitching is divided into two main processes: a) registration and b) blending. Image registration is the task of aligning images in a common coordinate system. Earlier approaches are often referenced as direct methods where the complete image information is used for registration. More recent registration methods are mostly feature-based. Feature-based methods are based on salient interest point correspondences to estimate robust transformations between images. Salient features like SIFT are extracted and matched. Consistent matches are obtained by using a robust outlier detection procedure like RANSAC. By means of global registration a cumulative registration error needs to be avoided. When images are aligned, they are stitched into a panorama with an adequate blending strategy. Blending methods are aimed to smooth the transition or to find an optimal seam between images. There exist also approaches, where images are stitched in the gradient domain. Further tasks are exposure correction, handling of moving objects or mapping of the final panorama on different surfaces. An early and widely used blending algorithm is pyramid blending which is applied in many image stitching systems. Dynamic image stitching approaches are quite interesting in order to create a 'growing' panorama. The panorama is updated as soon as a new image is added.

Chapter 4

Methodology

Contents

The goal of this diploma thesis is to design and to develop an image stitching system for visual quality inspection of a welding process. The stitching system should be able to generate an incremental image mosaic of a weld seam where a weld expert can evaluate the quality of the weld seam at a glance. In this chapter we give an overview about the methodoly of our image stitching system. We describe the individual parts of our system, which can be divided into two main processes: a) image registration and b) image blending.

The welding process is carried out by a welding robot. A camera is mounted on the robot and acquires high quality greyscale images during the welding procedure. The weld seam is tracked in each frame and small patches around the seam are extracted. The obtained patches should be stitched into a single survey image. Thus, the input to the image stitching algorithm is not the original frame, but a patch which is extracted by a tracker. The size of these patches is variable (e.g. 140×140). The patches usually provide much overlap, so that one pixel in the final panorama can be related to numerous patches. The generation of input images for the image stitching process is shown in Figure 4.1. On every frame an image patch of the welded seam is extracted. Figure 4.2 shows the position of an image patch on an acquired weld seam frame. This region of interest is determined, based on a tracked point on the weld seam.

There are different requirements for our system. It should be robust and real-time capable. So, it should deliver an accurate, robust result in adequate runtime. The survey

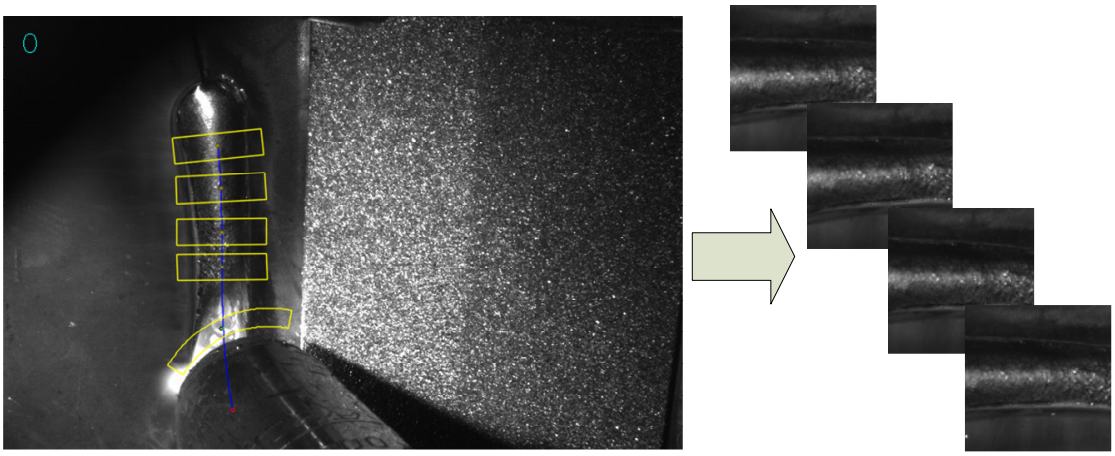


Figure 4.1: Weld seam, computed track and extracted patches. The weld seam is tracked in acquired frames. Based on the tracking data, rectified patches of the weld seam are extracted for each frame.

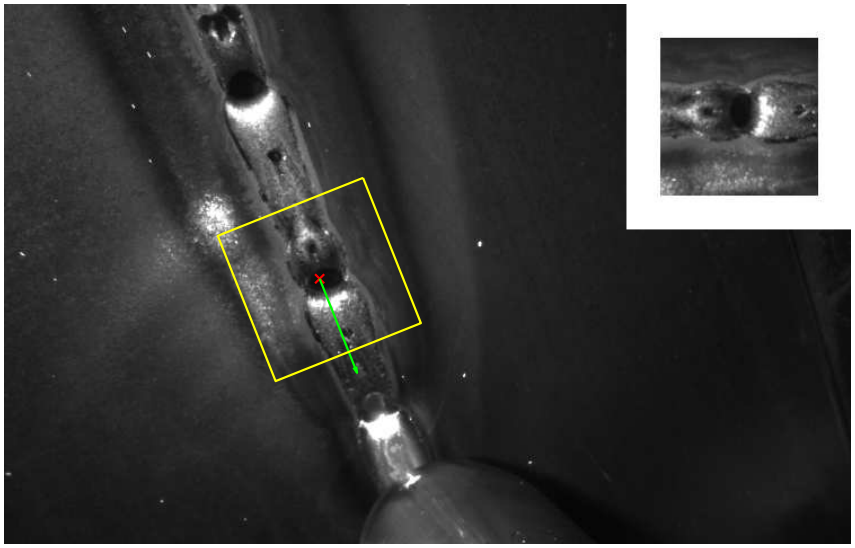


Figure 4.2: Image patches are obtained on each frame. The patch (top right) is extracted around a tracked point (marked by the red cross) and relative to the tracking direction (green).

image of the weld seam should be continous and not splitted if registration fails. Challenges for the image stitching process are heavy smoke, bright sparks, evaporating water and further effects which appear during welding. It is a goal for our image stitching system to reduce such noisy effects in the final panorama of the weld seam. There are two main questions. Which is an appropriate registration method to provide real-time capability and to guarantee a successful generation of a weld seam mosaic? How should input images be blended to reduce noisy artifacts and to provide an incremental ('online') generation of an image mosaic? We found out that a feature-based approach for image registration,

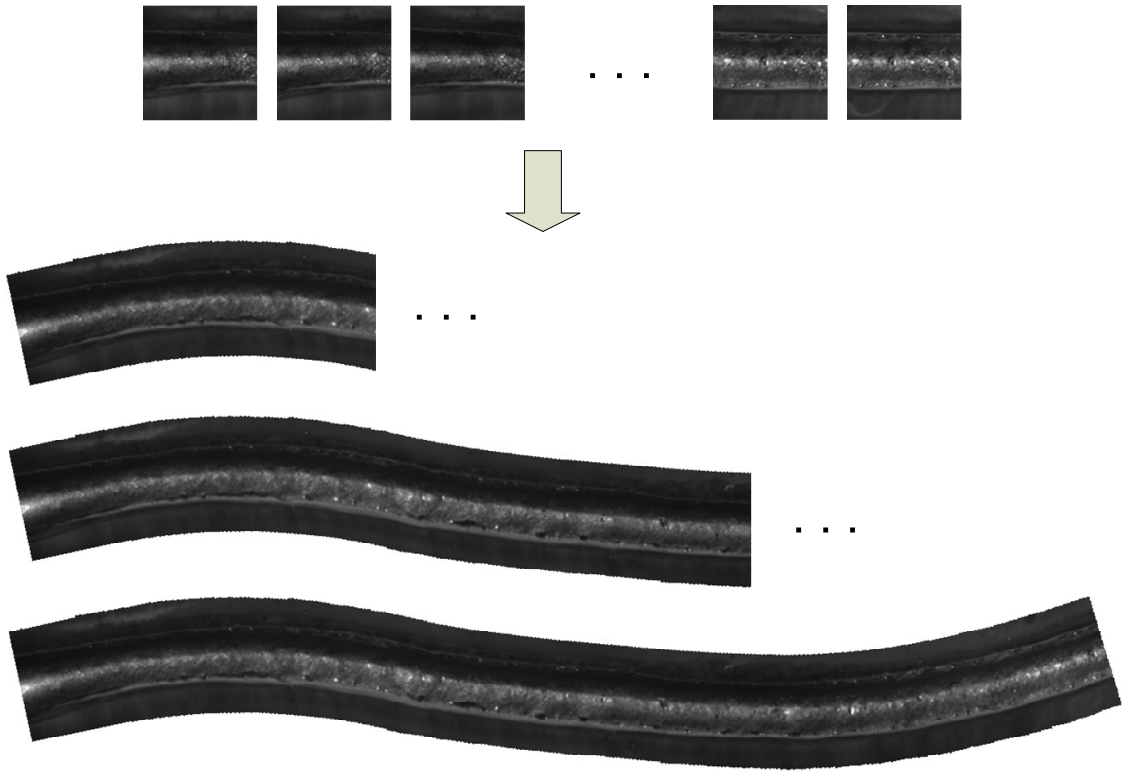


Figure 4.3: Incremental panorama generation. Input images are added sequentially. The panorama is generated incrementally. The image stitching process includes the registration of each new image, and the blending to the mosaic image. The final panorama above is generated from 186 140×140 patches. It has a size of 1351×214 pixels.

which is also primarily used in modern image stitching systems (e.g. [25], [10] and [53]), well meets our registration requirements. Such methods also include a robust motion estimation to reject outliers.

A challenge regarding image blending is the creation of an incremental panorama. Image stitching systems are often intended to create one panorama image not before all input images are processed. In our approach, the panorama is generated dynamically. Whenever a new image is acquired, the panorama is updated. Hence, a 'growing' panorama can be observed. For that purpose, a blending strategy is used, which considers an incremental processing of input images. It is even possible to complete an image mosaic and to extend it afterwards. For that purpose, we use the flexible blending approach presented in [54]. To reduce over-exposure in the final image mosaic, we adopt an exposure fusion approach [34].

Figure 4.3 illustrates the panorama generation process. Input images are added sequentially. The panorama is generated incrementally. Each new image is registered to the previous one, and is blended to the mosaic image. Hence, a growing panorama is

observable. An advantage of our incremental blending approach is that it can be used 'online' for a incremental panorama generation as well as 'offine' for a overall panorama generation if all input images are already registered. To improve the final registration result, we also provide a global step.

Section 4.1 gives a complete overview of our image stitching system. The individual parts of the system are discussed in further sections.

4.1 The Stitching Pipeline

In this section, we present our overall image stitching system. The aim is to generate an incremental mosaic of a weld seam where a weld seam expert can evaluate the quality of the weld seam at a glance. In order to review the survey image online during the welding process, the panorama is generated incrementally.

Figure 4.4 summarizes the main stages of our stitching system. The stages pre-processing, registration and incremental image blending are iterated for each input image. Hence, the current mosaic can be observed after each iteration. This approach enables real time applicability of our system. If all input images are processed, a global alignment step refines the final result.

The pre-processing step (see Section 4.2) includes a histogram equlization to improve the performance of registration as well as a dehazing procedure to enhance the blending result in the case of global haze. In Section 4.3, we discuss our robust feature-based approach for image registration. The incremental blending approach is discussed in Section 4.4. To refine the results of the completed image mosaics for archiving, we introduce a global alignment approach in Section 4.5. A more formal overview of our stitching system to process n input images is given in Algorithm 4.1.

Algorithm 4.1 Incremental Image Stitching

1: Initialize mosaic with first image I_1
2: **for** $k = 2$ to n **do**
3: Do pre-processing for image I_k
4: Register current image I_k to previous image $I_{(k-1)}$
5: Blend warped image I_k^W to corresponding ROI of mosaic M_{k-1} using multi-band blending with incremental blending weights
6: Update the mosaic with the blended image and get current mosaic M_k
7: **end for**
8: Do global refinement for completed mosaic M_n
9: Perform blending for the overall mosaic and get M'_n

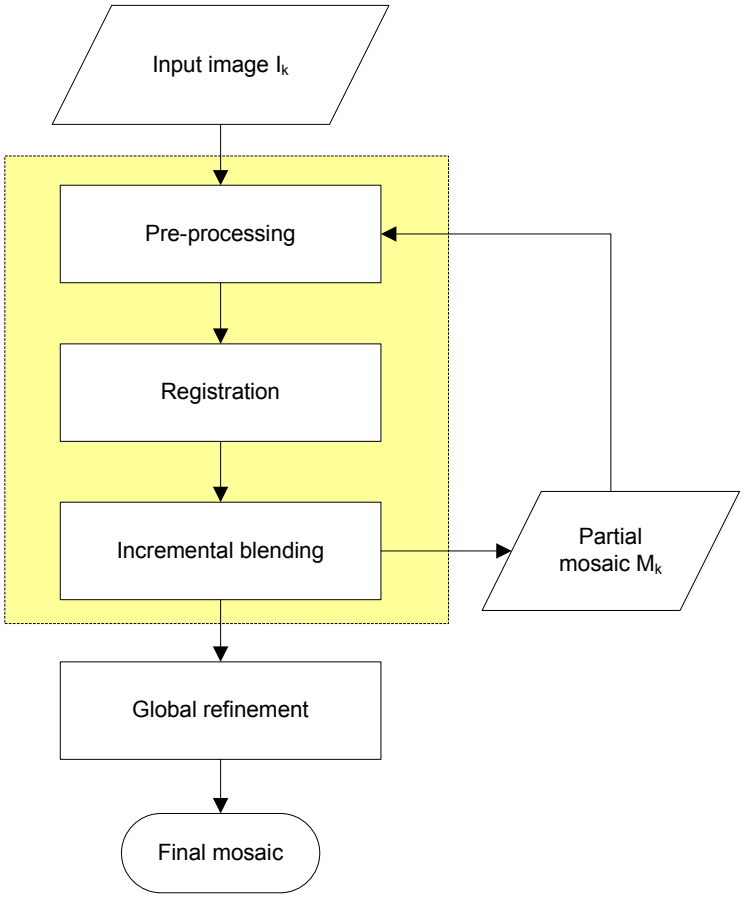


Figure 4.4: The Stitching Pipeline. Our image stitching system works dynamically. The stages pre-processing, registration and incremental image blending are iterated for each input image. Hence, the current mosaic can be observed in each iteration. If all input images are processed, a global alignment step refines the final result.

The image mosaic is initalized with the first image of the sequence. The registration includes prior data from tracking, so that the iterative generation of the mosaic is guaranteed. A precondition of the proposed stitching system is that input images has to be in sequential order. This has the advantage that one image do not have to be registered to several images but only to a predecessor, which is favorable in terms of runtime. As there is usually a large overlap between two subsequent images, the registration provides a satisfying result. The output of the registration is a local transformation H to map image I_k to image $I_{(k-1)}$. Local transformations of input images are acuumulated to a global transformation to warp the current image I_k to the mosaic coordinate frame.

A further advantage of the approach is that not the whole mosaic but only a relevant region of interest is used for blending. As blending weights are calculated for each new image and are accumulated for the existing mosaic, a pixel in the mosaic contains information of several patches. An incremental weighting function is used for the calculation of

blending weights but it is easy to substitute the blending weights calculation by a different approach (e.g. Feathering).

The approach above ensures a incremental generation of the panoramic weld seam image. It is possible to obtain an intermediate result of the mosaic at every time step. An existing mosaic can also be extended at a later date. After all n input images are processed, a final global alignment step refines the stitching result. If an online observation of the mosaic is not requested, the blending and mosaic generation can be dropped in each iteration (see line 5 and 6 in Algorithm 4.1) and just performed once after global alignment. An advantage of the incremental blending approach is that it can be used for a incremental ('online') panorama generation as well as for a overall panorama generation if all input images are already registered.

4.2 Pre-processing

We include two pre-processing steps to our stitching system: a) adaptive histogram equalization to improve the registration result and b) dehazing to enhance the blending result if there is global haze on the input images. The dehazing step is used in the case of heavy smoke on the input images. The two pre-processing steps are discussed in the following Sections 4.2.1 and 4.2.2.

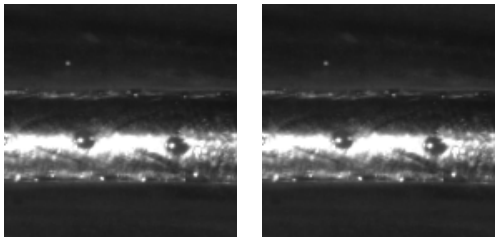
4.2.1 Adaptive Histogram Equalization

Before applying image registration (Section 4.3), we use a pre-processing step. Our image registration is based on local keypoints. We include a pre-processing step to improve the performance of feature extraction. This step is reasonable, as input images may have a low contrast. By using contrast-limited adaptive histogram equalization (CLAHE) [55], the contrast of a grayscale image is enhanced. This enables a more accurate and more comprehensive feature extraction, as the weld seam input images are generally rather small and do not contain much information. Figure 4.5 shows registration examples with and without CLAHE. The number of matched features is almost doubled in this case by using CLAHE.

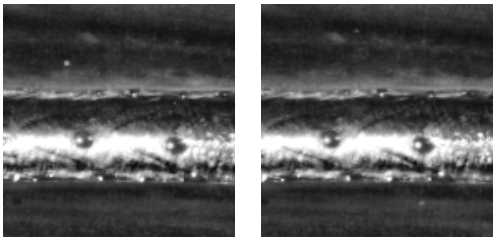
4.2.2 Dehazing

Frames of the welding process may contain heavy smoke. If there exist a global haze on these input images, a dehazing strategy can improve the stitching result. We present haze removal which can be applied on each input image before it is blended to the existing mosaic.

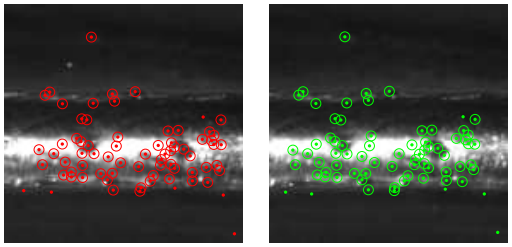
He et. al [23] presented an approach to remove haze from a single image using the dark



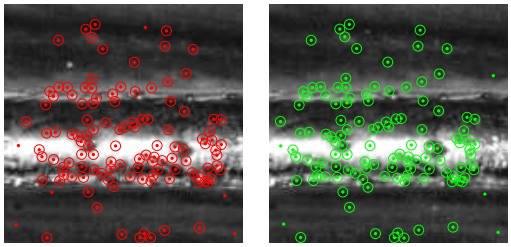
(a) Input images



(b) Images after CLAHE



(c) Inliers without CLAHE



(d) Inliers with CLAHE

Figure 4.5: Contrast-limited adaptive histogram equalization (CLAHE). Without adaptive histogram equalization 68 inliers are found out of 74 SIFT matches (91.89%) between the two input images. With CLAHE almost twice the number of SIFT features are extracted. This results in 124 inliers out of 133 matches (93.23%).

channel prior. The dark channel for a color image J is defined as

$$J^{dark}(\mathbf{x}) = \min_{c\in\{r,g,b\}} \left(\min_{\mathbf{y}\in\Omega(\mathbf{x})} \left(J^c(\mathbf{y}) \right) \right) , \tag{4.1}$$

where J^c is a color channel of J and $\Omega(\mathbf{x})$ is a local window centered at $\mathbf{x}$. The dehazing algorithm in [23] contains four steps: a) transmission estimation, b) soft matting, c) recovering the scene radiance, and d) atmospheric light estimation. Given the dark channel prior J^{dark}, the transmission $t(\mathbf{x})$ is estimated as

$$t(\mathbf{x}) = 1 - \omega J^{dark} , \tag{4.2}$$

where ω is an application-based haze constant ($0 < \omega \leq 1$). Soft matting is used to refine the transmission t. The refined transmission t_r is obtained by solving a sparse linear system:

$$(L + \lambda U)t_r = \lambda t . \tag{4.3}$$

L is a matting Laplacian matrix proposed in [28], U is an identity matrix of the same size than L, and λ is a regularization parameter. As soft matting and the calculation

of the matting Laplacian matrix are very expensive in terms of computation time, we use a guided filter approximation proposed in [24] instead. The transmission estimation t is filtered under the guidance of the hazy image to get the refined transmission t_r. Furthermore, t_r is restricted to a lower bound t_0,

$$t_r(\mathbf{x}) = max(t_r(\mathbf{x}), t_0). \tag{4.4}$$

He et. al [23] suggested a value of 0.1 for t_0. The final scene radiance for an image I is recovered by:

$$J(\mathbf{x}) = \frac{I(\mathbf{x}) - A}{t_r(\mathbf{x})} + A \ , \tag{4.5}$$

where A is the atomspheric light. For the estimation of the atmospheric light, the top 0.1% brightest pixels in the dark channel J^{dark} are chosen. The corresponding pixel with the highest intensity in the input image I is selected as atmospheric light. To use grayscale images for dehazing, the min_c operator in Equation 4.1 is omitted [23] and a grayscale version of the guided filter is used.

As the presented dehazing algorithm is intended for outdoor images and for global haze removal, it is rather difficult to adapt it for the weld seam images. But if smoke or haze arises globally on weld seam images, the final stitching result is enhanced. Dehazing can also improve the contrast in the image mosaic. The haze removal approach can be used as pre-processing step on all input images before image blending or as post-processing step on the final panoramic image. Figure 4.6 shows an example for haze removal in a color image. Figure 4.7 illustrates how dehazing of the weld seam patches results in a better panoramic image.

(a) (b)

Figure 4.6: Dehazing. The haze in the input image (a) is removed with a dehazing algorithm in (b).

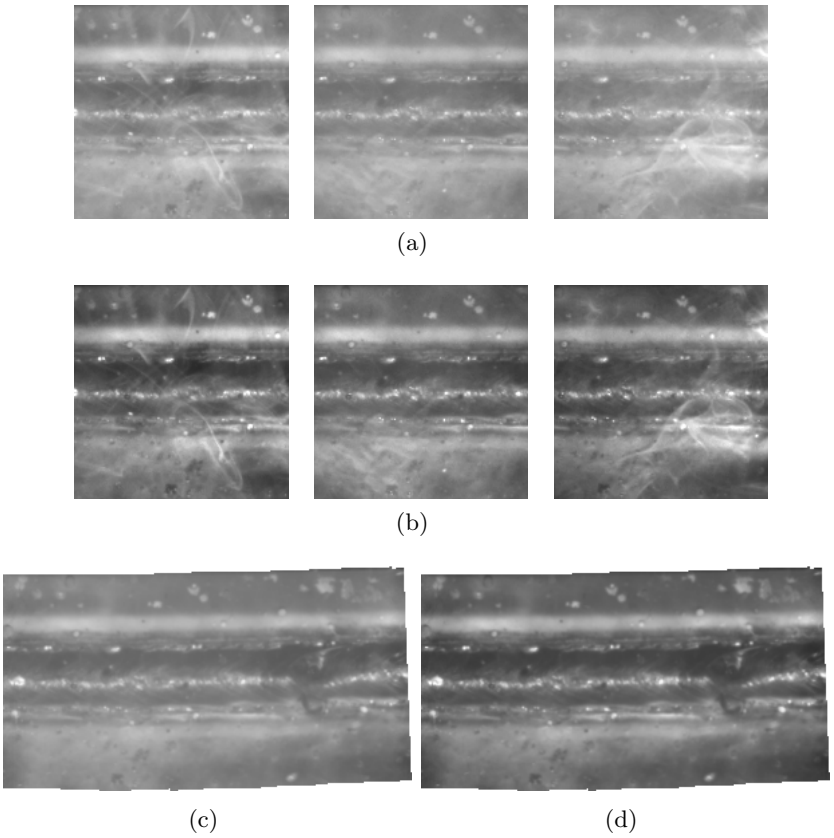


Figure 4.7: Dehazing of weld seam images. If there is particular global haze on the input images, dehazed input images provide a better result. The contrast is enhanced compared to the panorama without haze removal. Figures (a) and (b) show a sample of input images and its dehazed versions, respectively. The global haze is reduced. Single wads of smoke are still visible. Below are the panoramas (c) without and (d) with dehazed input images.

4.3 Image Registration

Image registration is the task of aligning images due to an underlying motion model (see Section 2.3). In the case of our image stitching sytem, extracted weld seam images should be aligned. As the scene is locally planar and the distance between the camera and the weld contact area remains approximately constant, it is adequate to compute the relative rotation and translation between the extracted patches of the consecutive frames. Hence, we have a 2D Euclidean transformation as motion model (see Section 2.3.1.1). Image registration can be direct or feature-based, as discussed in Section 2.3.2. As computation time and robustness is an issue, we decided to apply feature-based registration. Feature-based registration relies on local feature extraction and matching as well as robust motion estimation. Such an approach seems appropriate in our case.

Thus, we apply a robust feature-based image registration which is based on local

keypoints extraction and a RANSAC [18] routine for rigid motion estimation. The stages of the image registration are outlined in Figure 4.8.

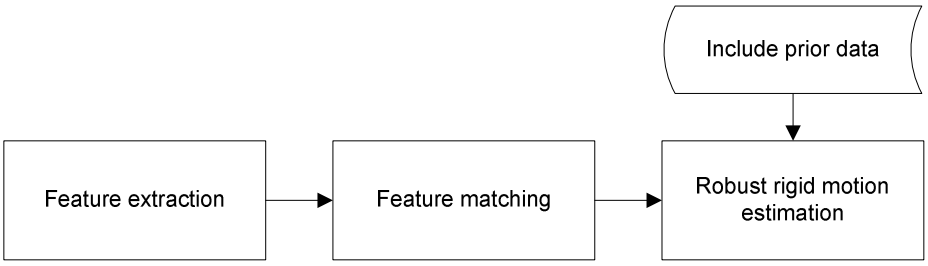


Figure 4.8: The stages of the proposed image registration based on salient feature extraction and robust motion estimation.

Salient keypoints are accordingly extracted and matched between the input images. There are different keypoint extractors (see Section 2.1) for choice. We recommend to use SIFT [32] features due to their good performance in our experiments as presented in Section 5.3. Furthermore, SIFT features are invariant to rotation and partially invariant to illumination change. The rigid motion is estimated in a robust manner by using RANSAC and using a tracking data in the case of registration failure. Hence, a successful generation of the image mosaic is guaranteed. As already mentioned, the motion model to describe the mapping between images is assumed to be an Euclidean transformation (rotation and translation). The rigid motion is computed by using Procrustes alignment [19]. The input images are in sequential order and the registration is performed pairwisely. Hence, a new image I_k is registered to the previous one I_{k-1}.

The registration results in a local transformation $H_{(k-1)k}$ from the k-th to the $(k-1)$-th image. The first image of the sequence is the reference image for the global mosaic coordinate frame. Hence, the global transformation H_{1k} from image k relative to the first image is calculated as

$$H_{1k} = H_{12}H_{23}...H_{(k-1)k} \ . \tag{4.6}$$

The pairwise registration of two successive images I_{k-1} and I_k can be summarized as follows:

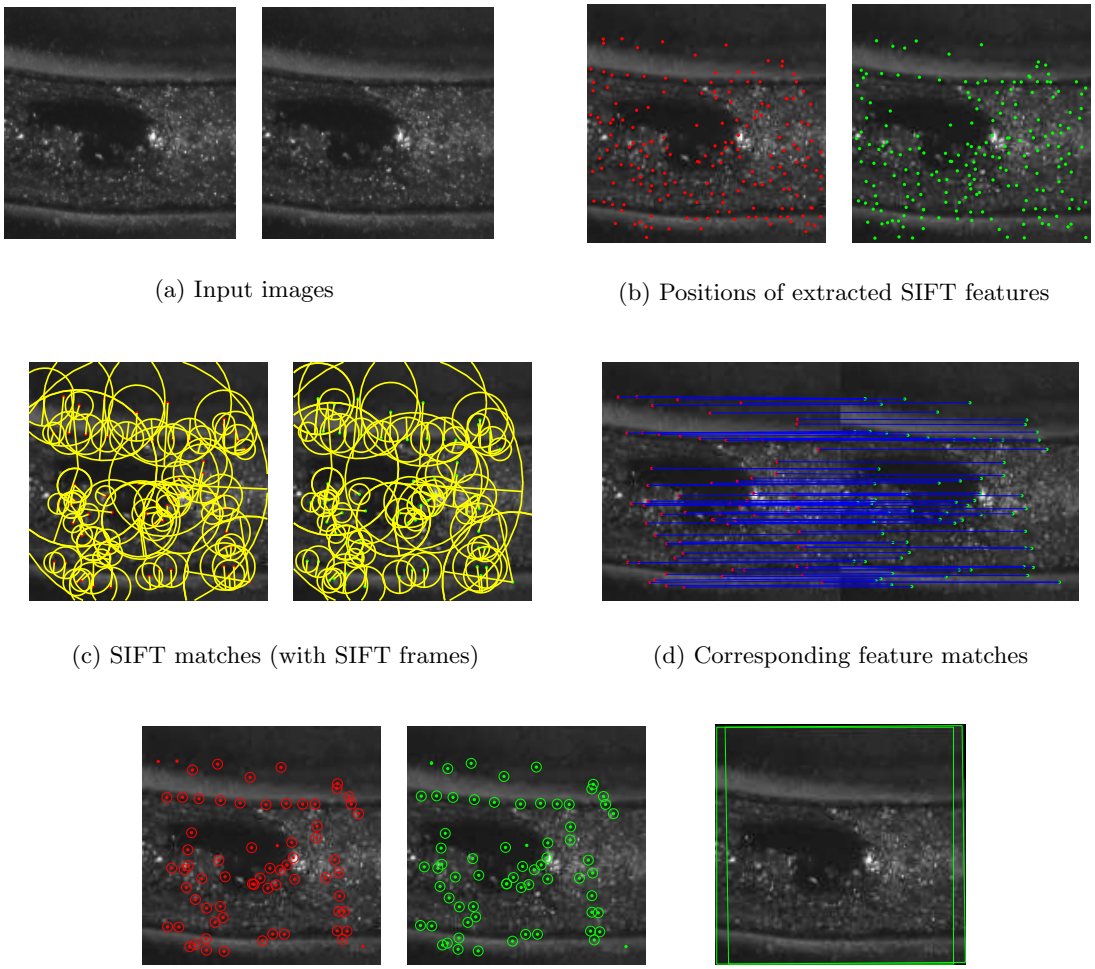
1. Extract salient features from I_k and I_{k-1}, respectively

2. Match extracted features and get point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i{}'\}$ between the two images

3. Estimate the rigid motion $H_{(k-1)k}$, i.e. rotation R and translation t, in a robust manner to map $\mathbf{x}_i$ to $\mathbf{x}'_i$, and I_k to I_{k-1}

4. If the rigid motion estimation fails, obtain $H_{(k-1)k}$ due to prior (tracking) data

Feature extraction and matching is discussed in Section 4.3.1. Section 4.3.2 addresses the robust rigid motion estimation. Furtheron, the consideration of tracking data is introduced in Section 4.3.3. Figure 4.9 illustrates the stages of the image registration based on SIFT features and robust rigid motion estimation. The output of a successfully completed registration is a local transformation $H_{(k-1)k}$ to align image I_k with image I_{k-1}.



(a) Input images

(b) Positions of extracted SIFT features

(c) SIFT matches (with SIFT frames)

(d) Corresponding feature matches

(e) RANSAC inliers (encircled)

(f) Aligned images

Figure 4.9: Image registration: SIFT features are extracted from the images. Then SIFT matches between the images are determined. From the point correspondences a rigid motion transformation is computed. RANSAC is performed to get a robust estimation and to reject outliers. The images have a size of 140×140. There are 69 verified feature matches (inliers).

4.3.1 Feature Extraction and Matching

In the first stage of image registration, distinctive image features are extracted. We decided to use SIFT features [32] (see Section 2.1.3) for this purpose, as they are invariant to rotation and partially invariant to illumination change. As computation time is an issue, we use the *Fast SIFT Image Features Library*[*] (libsiftfast). It is more than three times as fast as the standard SIFT implementation and almost reaches the runtime of SURF. In Section 5 we compare registration strategies using different image features. Due to the overall performance and to have the ability for a real-time application, we recommend to use SIFTfast features. The matching between two SIFT descriptors relies on a nearest neighbor strategy (see Section 2.2). Two SIFT descriptors $\mathbf{d}_1$ and $\mathbf{d}_2$ are matched if the distance $d(\mathbf{d}_1, \mathbf{d}_2)$ between them multiplied by a threshold t is smaller than the distances to all other N descriptors:

$$d(\mathbf{d}_1, \mathbf{d}_2)t < d(\mathbf{d}_1, \mathbf{d}_i),\ i = 3 \dots N\ . \tag{4.7}$$

We choose a threshold $t = 2.2$, which has proved to be a suitable value in our experiments. To be conform with Equation 2.11, this equation is rewritten to

$$\frac{d(\mathbf{d}_1, \mathbf{d}_2)}{d(\mathbf{d}_1, \mathbf{d}_i)} < \frac{1}{t},\ i = 3 \dots N\ . \tag{4.8}$$

As the set of SIFT matches may contain outliers, we use a RANSAC procedure to reject outliers. The inliers of SIFT matches are used to estimate the rigid motion to map I_k to I_{k-1}. The rigid motion estimation based on RANSAC is discussed in the following Section 4.3.2.

4.3.2 Robust Rigid Motion Estimation

The rigid motion estimation is based on RANSAC (see Section 2.3.3.3) and Procrustes alignment (see Section 2.3.3.2). The corresponding point matches $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i{}'\}$ which are found between two images may contain outliers. To get a robust transformation for image alignment and to reject outliers, a RANSAC [18] routine is used. It is implemented in the following way:

1. Randomly select a subset S of $s = 3$ point correspondences from the set of point correspondences $P : \{\mathbf{x}_i \leftrightarrow \mathbf{x}_i{}'\}$
2. Perform Procrustes alignment on the subset S to get a transformation matrix H (rotation and translation)
3. Determine the set of inliers $S_{\mathcal{I}} : \{\mathbf{x}_j \leftrightarrow \mathbf{x}_j{}'\}, j \in \mathcal{I}$, which are within a distance threshold: $\|\mathbf{x}_i - H\mathbf{x}_i{}'\|_2 < t$

[*] http://sourceforge.net/projects/libsift/

4. Get the number of inliers: $n_{\mathcal{I}} \leftarrow \text{length}(S_{\mathcal{I}})$.

5. If the number of inliers $n_{\mathcal{I}}$ is the best so far, update the best inliers set $P_{\mathcal{I}} \leftarrow S_{\mathcal{I}}$

6. If best inliers set so far: Update the estimate of number of trials N to ensure we pick with probability p (we set p to 0.99) one sample free from outliers:

$$f_{\mathcal{I}} = \frac{n_{\mathcal{I}}}{\text{length}(P)},\ N = \frac{log(1-p)}{log(1-{f_{\mathcal{I}}}^s)}$$

7. Repeat from 1 until break condition for the loop is fulfilled: The total trial count reaches the estimate of number of trials N or the maximum number of iterations.

8. Re-calculate H on the best inlier set of point correspondences $P_{\mathcal{I}}$ (by using Procrustes alignment)

Step 8 equals a final least squares fit on the set of inliers. The output of the RANSAC procedure is a transformation matrix H that maps the corresponding point matches between the two images. As additional condition to accept an image match and the alignment between two images I_{k-1} and I_k respectively, the fraction of inliers $f_{\mathcal{I}}$ is considered. If the fraction of inliers with respect to the total number of point correspondences is above a specified threshold t, the image alignment is accepted. If not, the image alignment is rejected. We include this step to eliminate rigid motion estimations which are based on an unsufficient number of inliers. We accept a rigid motion estimation, if

$$f_{\mathcal{I}} > t + \frac{2}{length(P)}\ . \tag{4.9}$$

The last term is included to penalize a low number of feature matches. We have found out in experiments that 0.25 is an appropriate value for t.

The output of a successful (pairwise) registration is a local transformation to map I_k to I_{k-1}. The registration fails, if an insufficient number of inliers is found to fulfill the specified requirement (e.g. fraction of inliers). In the case of several successive registration failures, the generation of a single survey image of the weld seam is no longer ensured. For that purpose, a fallback strategy is introduced. For example, there are the following possibilities if no successful registration is obtained:

a) The previous rigid motion estimation is used for the current registration.

b) The current mosaic is finished and a new mosaic starts with the new image.

c) Alternative information provided by the tracker is used.

As we can obtain appropriate prior data from the tracker, we have implemented a fallback strategy that utilizes tracking data (see Section 4.3.3).

4.3.3 Incorporating Prior Data

The worst case would be given if several consecutive registrations fail. In this case the image mosaicing process cannot be continued as new input images no longer have overlapping regions with the existing mosaic. To avoid this, a fallback strategy using prior data is included for the registration process. We calculate the local tranformation between two consecutive patches due to tracking data, if pairwise registration fails.

The position of the weld seam in the image is tracked during the welding process. Input patches to the image stitching system are extracted from each frame by the tracker. As there exist tracking related data that can be used for registration too, it is possible to use this data for a fallback strategy if the image registration fails.

For each frame, the following tracking data is available: central position $\mathbf{x}$ of the extracted patch (in frame coordinates), tracking angle θ relative to the frame, and an estimated rigid motion H_T of the tracker between the current and the previous frame. But it should be noted that H_T is only a raw estimate based on 3 or 4 point correspondences.

The center position of the patch is a tracked point on the weld seam. Figure 4.10 shows the position and orientation of an extracted patch in a frame. The corresponding information is obtained from the tracking procedure. Accordingly, the transformation of the k-th patch from patch to frame coordinates can be expressed as concatenation of 3 transformations based on tracking data:

$$H_{P_k} = T_k R_k T_c \ , \tag{4.10}$$

where T_c is the translation to the patch center $\mathbf{c} = (c_x, c_y)$, so that the origin of the patch coordinate frame is at the patch center,

$$T_c = \begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & 1 \end{bmatrix} \ , \tag{4.11}$$

R_k is the rotation of the centered patch k with the tracking angle θ,

$$R_k = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \ , \tag{4.12}$$

and T_k is the translation of the centered and rotated patch k to the position $\mathbf{x} = (x, y)$ in frame coordinates,

$$T_k = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \ . \tag{4.13}$$

The transformation from patch k to patch $(k-1)$ is calculated by:

$$H_{(k-1)k} = H_{P_{(k-1)}}^{-1} H_{T_k} H_{P_k} \ . \tag{4.14}$$

Thus, the rigid motion between two successive patches is approximated by

1. Transformation H_{P_k} of the current patch to the frame position
2. Frame to Frame transformation H_{T_k} monitored by the tracker
3. Transformation $H_{P_{(k-1)}}^{-1}$ from the position in the last frame to patch coordinates

The major advantage of this fallback strategy is that there is achieved a rigid motion estimation in any case and thus the panorama always can be generated. The successful generation of a weld seam survey image has indeed the condition that the tracker does not fail. In the case of a tracker failure the adequate patch extraction could not be guaranteed and the image mosaicing process is condemned to failure.

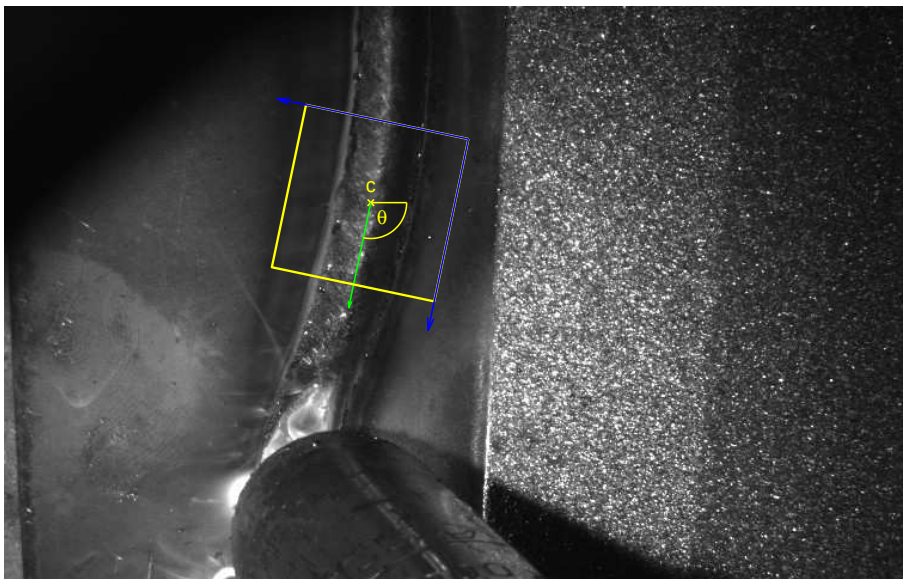


Figure 4.10: Tracking data. The patch is extracted around the tracked point **x**, where the central position **c** of the patch is located. The tracking angle θ is spanned between the tracking direction (green) and the x axis of the frame. The extracted patch is rotated relative to the tracking angle. The coordinate frame of the patch is represented by the blue vectors.

4.4 Incremental Image Blending

A requirement for our system is to be able to generate a incremental image mosaic. Thus, the input images are processed incrementally. As soon as a new image is passed to the image stitching system, it is blended to the already existing mosaic. In other words, the mosaic grows over time. This provides following advantages: The mosaic can be observed

at every time step. Hence, it is possible to follow the 'growing panorama'. A completed mosaic can be extended at any time. The system is also capable for a real-time application.

There exist several blending strategies (see Section 2.4). In the case of an optimal seam finding strategy, a region in the final mosaic corresponds exactly to one region of an input image. Overlapping regions of multiple images are not considered. Noisy artifacts like bright sparks or wads of smoke may completely survive with such an approach, while they are eliminated in a large part with a transition smoothing method. As we want to take into account all images which contribute to a region in the final mosaic, a transition smoothing approach with adequate blending weights seems to be appropriate. The blending weights should consider an incremental generation of the panoramic image. Another issue for the blending strategy is hat there may be over-exposed regions in input images which should be suppressed for the final result.

To enable a incremental blending and to adjust the blending weights at each time step respectively, we adopted the flexible blending approach by Zhao [54]. The blending weights are additionally modified by exposure fusion weights [34] to suppress over-exposed regions. A transition smoothing between overlapping images is achieved by using a scale space based blending approach [12]. If heavy smoke affects the input images, a dehazing [23] step can further improve the stitching result (see Section 4.2.2). An overview about our approach is given in Figure 4.11.

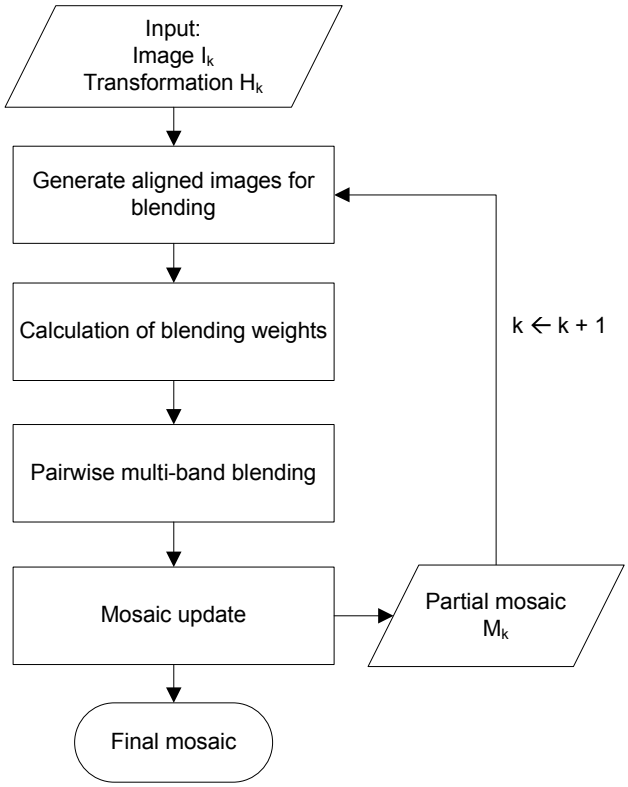


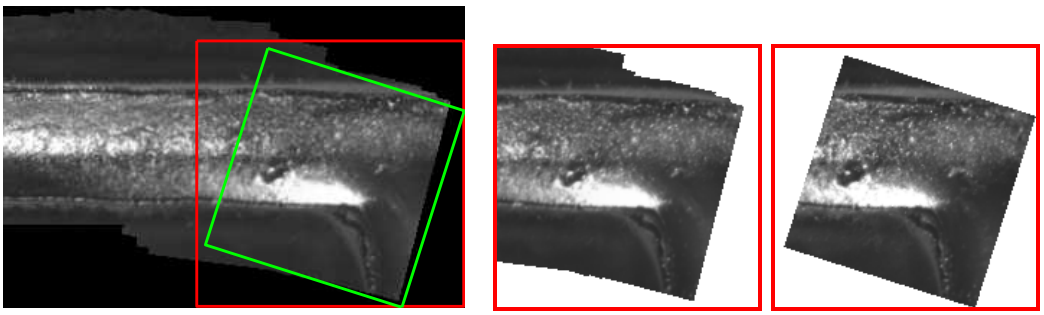
Figure 4.11: Overview of the incremental image blending approach.

The image mosaic is initialized with the first image I_1, which is also the reference image for the mosaic coordinate frame. Furthermore, each new image I_k is added to the current mosaic. Our incremental image blending strategy includes the following steps, which are discussed in the following sections:

- Generation of aligned images for blending (Section 4.4.1)
- Calculation of blending weights considering incremental panorama generation (Section 4.4.2)
- Pairwise multi-band blending to add the new image to the existing panorama (Section 4.4.3)
- Update the panorama with the new blended image (Section 4.4.4)

4.4.1 Generation of Aligned Images

In the image registration stage, the mapping of a new image to its previous image is calculated. Before the new image is blended to the existing mosaic, it has to be warped relative to the mosaic coordinate frame. Image warping is discussed in Section 2.3.4. For blending only a small part of the mosaic is considered, namely the regions which are affected by the new image. Two patches of equal size are generated: a) the region of interest (ROI) of the mosaic and b) the warped version of the new image. Figure 4.12 demonstrates the generation of the two aligned patches.



(a) (b) (c)

Figure 4.12: Generation of aligned images. To warp and blend a new image to the mosaic, two aligned images are generated. (a) A region of interest (red) around the area of the new warped image (green) is extracted. The output is given by two aligned patches of equal size: (b) mosaic patch and (c) warped new image.

4.4.2 Calculation of Blending Weights

The calculation of blending weights is divided into two sections: a) incremental weighting and b) exposure fusion. The blending weights are updated whenever a new image is processed. On the one hand, weights for the current mosaic are accumulated. On the other hand, weights are calculated for each new image. In Section 4.4.2.1 we discuss the calculation and application of incremental blending weights. How to generate exposure fusion weights and to include them for incremental blending is presented in 4.4.2.2.

4.4.2.1 Incremental Weighting

The implementation of incremental image blending is mainly adopted from the flexible blending method introduced by Zhao in [54]. We want to use an image blending function, that considers previous images similarily to the current image. If we consider a ray to the mosaic, one pixel in the final mosaic contains information of several images. Flexible blending is generally defined by:

$$I_m(x,y) = \sum_{k=1}^{t} w_k(x,y) I_k^W(x,y) \ , \tag{4.15}$$

where $I_m(x,y)$ is the mosaic image, and $w_k(x,y)$ is the weighting function for the warped version of the k-th input image $I_k^W(x,y)$. Pixel coordinates (x,y) are given in the mosaic coordinate frame. The weighting function for an input image $I_t(x,y)$ at step t can be applied in a geometry-based manner. This kind of weighting function considers the constraint that the center of an image has sharper focus and less distortions. The geometry-based weighting function $g_t(u,v)$ for a single image is defined as:

$$g_t(u,v) = (1 - d(u,v))^r \ , \tag{4.16}$$

where d is the normalized distance (< 1) from the pixel location (u,v) to the center of the image, and r is a power term. In contrast to geometry-based weighting, averaging uses similar weights for each pixel. These two approaches can be comined, e.g. 50% averaging, 50% weighting based on geometry. For that purpose, a control parameter p in the range of $[0,1]$ is introduced. It indicates the percentage of averaging. The percentage of geometric-based weighting is $(1-p)$. If p is 0, 100% geometric-based weighted is used. The combined weighting is expressed as

$$w_a = (1-p) * g_t^W + p \ , \tag{4.17}$$

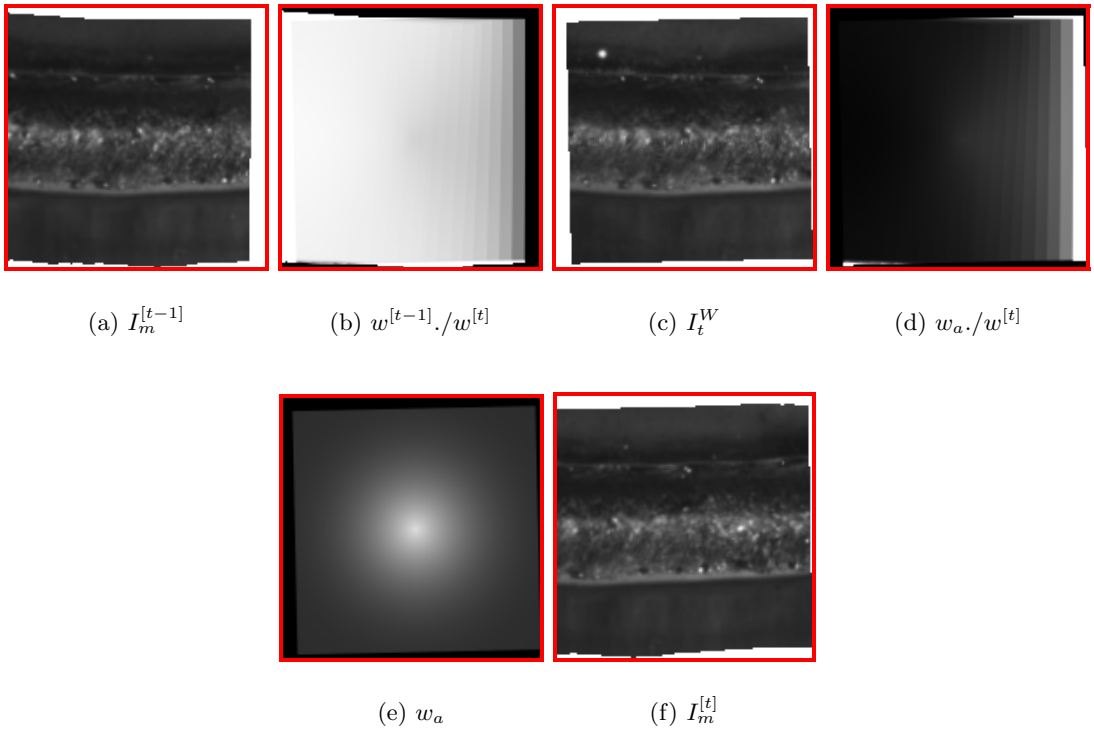
where g_t^W is a warped version of the geometry-based weights. By using this weighting function, the mosaic $I_m^{[t]}$ at time step t can be obtained by

$$I_m^{[t]} = \begin{cases} I_m^{[t-1]}, & I_t^W \text{has no pixel} \\ \frac{w^{[t-1]} * I_m^{[t-1]} + w_a * I_t^W}{w^{[t-1]} + w_a}, & \text{else} \end{cases} \tag{4.18}$$

The weights $w^{[t]}$ are updated according to

$$w^{[t]} = w^{[t-1]} + w_a \ . \tag{4.19}$$

Figure 4.13 illustrates the general characteristics of incremental blending weights. The more patches overlap each other, the less a single patch contributes to the final mosaic.



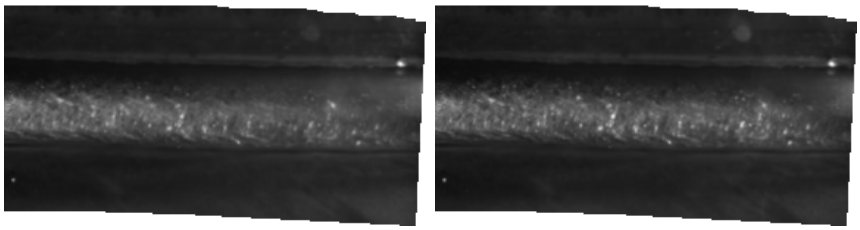
(a) $I_m^{[t-1]}$ (b) $w^{[t-1]}./w^{[t]}$ (c) I_t^W (d) $w_a./w^{[t]}$

(e) w_a (f) $I_m^{[t]}$

Figure 4.13: Incremental blending: (a) mosaic ROI, (b) relative weights for mosaic ROI, (c) warped image, (d) relative weights warped image, (e) absolute weights warped image, (f) blended image. The mosaic ROI is blended to the warped image due to the corresponding blending weights. The gradation of the particular patches is clearly visible in the relative weights. The more patches overlap each other, the higher are the weights of the mosaic ROI. The absolute weights of the warped image shows the characteristics of geometry-based weights. The following parameters are used for the calculation of the incremental blending weights: averaging ratio $p = 0.2$, power term $r = 3$.

Geometry-based weighting is not applicable in its classical meaning in our case. We do not use a complete image, i.e. one frame, but only a small region of interest, i.e. an extracted image patch. The optimal parameterisation depends on the input images. However, it can make sense to use weights based on geometry. Thereby, pixels around the center of a patch contribute more to the final mosaic. This approach can even sharpen the focus of the mosaic. It can happen that extracted patches are too close to the melt bath. In this case, a geometry-based approach is advantageous by all means. By using

geometry-based weights some kind of feathering (see Section 2.4.2.2) is also included into the blending weights. Figure 4.14 shows a part of a mosaic with pure averaging and one with pure geometry-based weighting. The geometry-based approach increases the contrast and reduces the negative effects of the melt bath visible on some patches.



(a) Average weighting (b) Geometry-based weighting

Figure 4.14: Average and geometry based weighting. In (a) every pixel of a patch contributes to the final mosaic similarly. Geometry-based weighting in (b) assigns higher weights to pixels around the center of a patch. The latter can sharpen the focus. In the example above patches are extracted too close to the melt bath, whereby the geometry-based approach provides favorable results.

4.4.2.2 Exposure Fusion

An approach to fuse images with multiple exposures into a high quality image is presented in [34] and is called exposure fusion. The goal is to keep only the best parts of a multi-exposure image sequence. There are three quality measures which are provided for exposure fusion: a) contrast, b) saturation, and c) well-exposedness. For the contrast a Laplacian filter is applied to the greyscale version of an image. The indicator C for the contrast is obtained from the absolute value of the Laplacian filter response. The saturation measure S is calculated as the standard deviation within the R, G and B channel at each pixel. The well-exposedness E should decrease intensities near zero (underexposed) as well as near one (overexposed). For that purpose, a Gaussian curve is used to weight each intensity i: $\exp(-\frac{(i-0.5)^2}{2\sigma^2})$ with σ equals 0.2. This well-exposedness measure evaluates how close an intensity i is to 0.5. Combining the quality measures C, S and E, an exposure weight can be assigned to each pixel of an image:

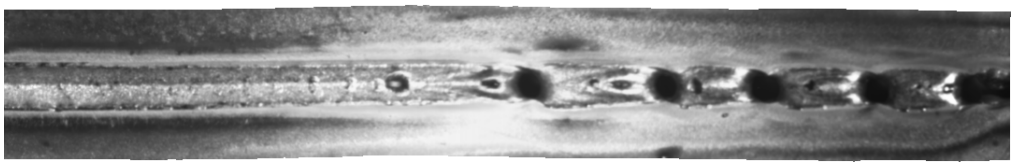
$$W_k^{\mathcal{E}}(x,y) = C_k(x,y)^{wc} \times S_k(x,y)^{ws} \times E_k(x,y)^{we} \ , \tag{4.20}$$

where wc, ws and we are weighting exponents, and k indicates the k-th image. Applying exposure fusion to the weld seam images of our image stitching system, the saturation measure S can be rejected as only greyscale images are used. A possibility to implement exposure fusion is to multiply the exposure weights to existing (incremental) blending

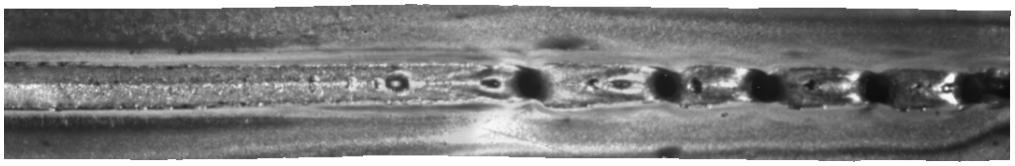
weights. Thus, an exposure weight map is combined with another weight map:

$$W_k(x,y) = W_k^{\mathcal{E}}(x,y) \times W_k^{\mathcal{F}}(x,y) \ , \tag{4.21}$$

where $W_k^{\mathcal{E}}$ and $W_k^{\mathcal{F}}$ indicates the exposure weights and blending weights as in Equation 4.17 respectively for the k-th image. Figure 4.15 illustrates the advantages of exposure fusion. It shows that overexposure is reduced, and that the structure of the weld seam is emphasized.



(a)



(b)

Figure 4.15: Exposure fusion. The image mosaics are created (a) without, and (b) with exposure fusion. Overexposed regions are reduced and the structure of the weld seam is focused.

To consider some effects on the weld seam images, we enhanced the exposure fusion approach with two functions. Weld seam images can contain bright sparks which should be reduced on the final mosaic. By the contrast enhancement effect of exposure fusion and with occurrence of bright sparks on very dark background, it may be that sparks are highlighted with exposure fusion. Additionally, it may be that the acquired weld seam images are generally quite dark. For that reason, a contrast adjustment step is introduced before calculating the exposure weights. The contrast is adjusted with an sigmoid function which is defined as follows [27]:

$$I'(x,y) = \frac{1}{(1 + \exp(g * (c - I(x,y))))} \ , \tag{4.22}$$

where g represents the gain and c is the cutoff pixel value about which the contrast is increased or decreased. The control parameter g is in the range from 1 to 10. In our experiments we set g to 5. A low value for g reduces the contrast, and a high one increases the contrast. The image I is normalized between 0 and 1 before applying the

sigmoid function. Pixel differences are emphasized by contrast adjustment. The weighting distribution is improved for too dark exposures. The calculation of the well-exposedness weights E is also modified. The weight map undergoes a block processing. For each pixel p_i in the weight map E, a local neighborhood (e.g. 5×5) around this pixel is analyzed. The lowest weight in the window is chosen as the new weight for pixel p_i. By using such a filtering function, outliers like bright sparks are suppressed.

A challenge of using the exposure fusion approach for weld seam images is that on the one hand high frequencies and high contrast should be highlighted to get a final panorama of higher quality. On the other hand, high frequncies and high contrast should be decreased to reduce bright sparks and other noisy artifacts. The challenging task is to find the optimal medium to get the best solution. To completely omit the contrast enhancement step, may hide bright sparks, but at the expense of the contrast quality of the final mosaic. One way to meet this challenge is the block processing and contrast adjustment step for the calculation of exposure weights. We suggest to use block processing in any way and to set the parameters for contrast adjustment just if for example it is known that the input sequences are acquired under dark exposure. Oversaturation can be well reduced with the exposure fusion approach provided that there exists images with well-adjusted saturation in the overlapping area.

4.4.3 Multi-band Blending

In the previous section we have defined a method to calculate successive blending weights for an image sequence. A simple approach to blend a new warped image $I^n(x, y)$ to the current mosaic $I^m(x, y)$ is to calculate a weighted sum of image intensities along each ray:

$$I^m(x,y) = \frac{W^m(x,y)I^m(x,y) + W^n(x,y)I^n(x,y)}{W^m(x,y) + W^n(x,y)} \ , \tag{4.23}$$

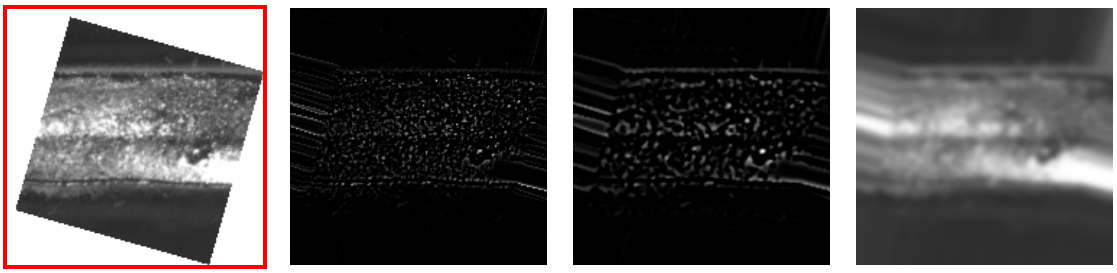
where $W^m(x, y)$ are the blending weights of the current mosaic and $W^n(x, y)$ are the blending weights for the new image. The pixel locations (x, y) are expressed within the mosaic coordinate frame. To improve the blending result, we adopt multi-band blending as presented in [10]. Brown and Lowe use a multi-band blending approach similar to pyramid blending introduced by Burt and Adelson [12].

A high-pass version $B^i_\sigma(x, y)$ of an image is defined as

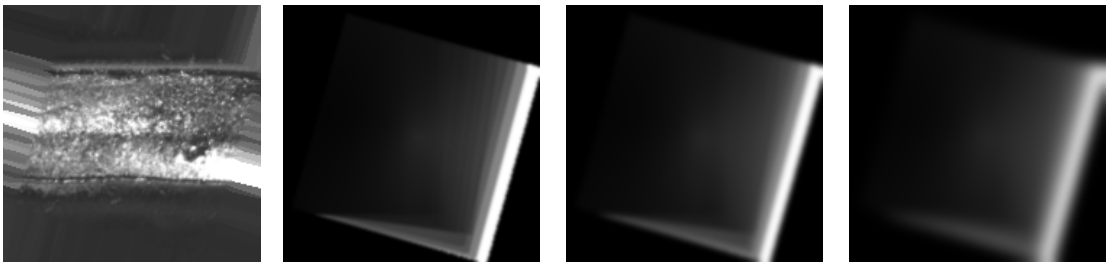
$$B^i_\sigma(x,y) = I^i(x,y) - I^i_\sigma(x,y) \tag{4.24}$$

$$I^i_\sigma(x,y) = I^i(x,y) * g_\sigma \tag{4.25}$$

where g_σ is a Gaussian kernel with standard deviation σ, and $*$ indicates the convolution operator. B_σ depicts spatial frequencies of wavelengths between 0 and σ. Further band-



(a) Warped image (b) Band-pass images



(c) filled (d) Blending weights

Figure 4.16: Multi-band blending. The original (warped) image (a) is decomposed into band-pass images (b) which are applied for multi-band blending. To avoid abrupt transitions in the band-pass images, pixels outside the image boundaries of the original image after warping are filled with the nearest pixel value (c). For each band-pass image corresponding (continously blurred) weights (d) are used to blend the image to another one.

pass images are generated by:

$$B^i_{(k+1)\sigma} = I^i_{k\sigma} - I^i_{(k+1)\sigma} \tag{4.26}$$

$$I^i_{(k+1)\sigma} = I^i_{k\sigma} * g_{\sigma'} \tag{4.27}$$

The standard deviation of the Gaussian kernel is adjusted to the range of wavelengths for subsequent bands:

$$\sigma' = \sqrt{(2k+1)}\sigma \ . \tag{4.28}$$

For the fusion of the band-pass images at each level k, the blending weights are blurred continously with the same Gaussian kernels:

$$W^i_{(k+1)\sigma} = W^i_{k\sigma} * g_{\sigma'} \ . \tag{4.29}$$

Having N band-pass images, the new image I^n and the current mosaic I^m are combined

at each band by using the corresponding blending weights:

$$I^m(x,y) = \sum_{k=1}^{N} \frac{W^m_{k\sigma}(x,y)B^m_{k\sigma}(x,y) + W^n_{k\sigma}(x,y)B^n_{k\sigma}(x,y)}{W^m_{k\sigma}(x,y) + W^n_{k\sigma}(x,y)} \ . \tag{4.30}$$

Thus, high frequency bands are blended over short ranges, and low frequencies are blended over larger ranges. Before applying multi-band blending for two images (I^m, I^n), we normalize the corresponding weight maps, so that the sum of weights equals 1.

We do not consider the entire mosaic for blending. Just a region of interest (ROI) of the mosaic is blended to the new image. The ROI is selected, based on the overlapping region between the mosaic and the warped new image (see Section 4.4.1). Figure 4.16 depicts the band-pass images and the blending weights for a warped image.

This multi-band blending approach can be used, independent of the type of weight maps. It works for incremental blending weights (see Section 4.4.2.1) as well as for binary weight maps or feathering. For this reason, our blending approach is divided into two stages: a) the calculation of the blending weights for the ongoing panorama generation process, and b) the pairwise blending of a mosaic ROI to a new image by using e.g. multi-band blending.

4.4.4 Mosaic Update

The weld seam survey mosaic is updated in each iteration a new image is processed by the system. Two aligned images, the warped new image and the corresponding ROI of the mosaic, are blended by using multi-band blending and incremental blending weights. The current mosaic is updated with the blending result. As there is the possibility that the region of a new image can be outside the existing mosaic coordinate frame, the coordinate frame of the mosaic has to be adjusted if necessary.

The first image of a sequence is per definition the reference image for the mosaic coordinate frame. In the beginning, the coordinate plane of the first image and the image mosaic are the same. The upper left corner defines the origin $(0, 0)$. A new image is warped to the mosaic coordinate frame by a transformation H. A sample point $\mathbf{x}_i$ of the image is mapped to a point $\mathbf{x}_m$ in the mosaic:

$$\mathbf{x}_m = H\mathbf{x}_i \ , \tag{4.31}$$

where $\mathbf{x}_m = (x_m, y_m, 1)$ are homogenous mosaic coordinates, and $\mathbf{x}_i = (x_i, y_i, 1)$ are homogenous image coordinates. If mapped mosaic coordinates $\mathbf{x}_m$ are negative (i.e. points are located outside the existing mosaic), the mosaic coordinate plane needs to be updated. This is done by shifting the mosaic coordinate frame. The necessary shifts (s_x, s_y) are

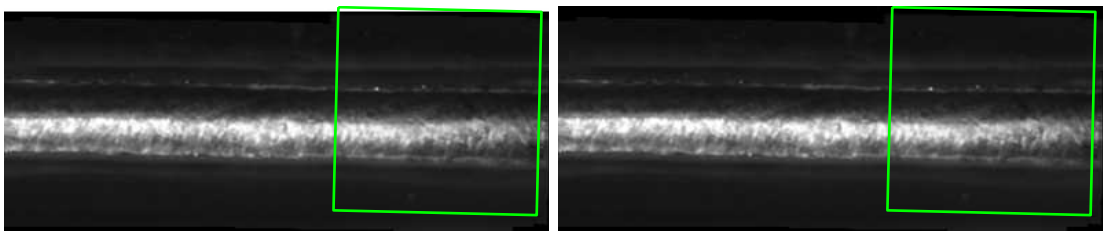
calculated in the following way:

$$\begin{aligned} s_x &= -\min(0, x_{min}) \\ s_y &= -\min(0, (y_{min}) \ , \end{aligned} \tag{4.32}$$

where x_{min} is the minimum x coordinate and y_{min} is the minimum y coordinate of the warped image in mosaic coordinates. The current mosaic is shifted by (s_x, s_y), resulting in an update of transformation H of the current image:

$$\begin{aligned} H(1,3) &= H(1,3) + s_x \\ H(2,3) &= H(2,3) + s_y \ . \end{aligned} \tag{4.33}$$

By updating the mosaic and the transformation H, it is guaranteed that the warped image always has positive image coordinates. Figure 4.17 gives an example for the here introduced mosaic coordinate frame update process.



(a) (b)

Figure 4.17: Update of the mosaic coordinate frame. The figure shows the mosaic coordinate frame (a) before and (b) after the update. The update is necessary because the new image would exceed the visible region of the existing mosaic. The green border represents the position of the new image. After the update the complete new image is visible in the mosaic.

As the image blending process only concerns regions of the total panorama, a panorama update is performed in each iteration. After a new image is warped and blended with its corresponding (overlapping) region in the panorama, the panorama is updated with the blending result. Figure 4.18 illustrates the steps of the panorama update process. These steps can be summarized as follows:

1. Update mosaic coordinate frame
2. Determine overlapping region between existing mosaic and new warped image
3. Extract mosaic ROI I_m and warp new image relative I_k to it
4. Blend I_m and I_k^W to obtain I'_m
5. Update mosaic with blending result I'_m

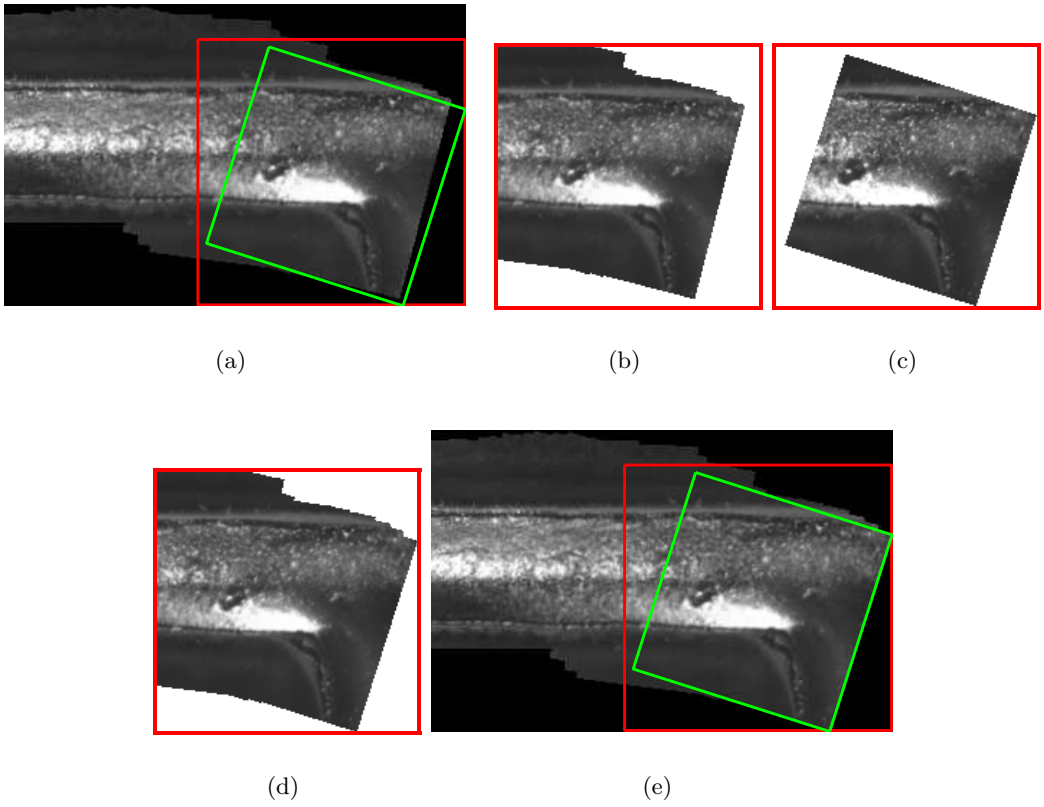


Figure 4.18: Mosaic update. (a) A region of interest (red) around the area of the new warped image (green) is determined. (b) The mosaic ROI is extracted and (c) the new image is warped relative to it. (d) The two patches are blended and (e) the mosaic is updated.

4.5 Global Refinement

We present a global refinement step which is processed if all input images are registered. Our global refinement is a combination of a graph based optimization [46] and an iterative global alignment [17]. First, the global transformations are adjusted to a required end position of the weld seam in the mosaic coordinate frame. Second, the registration result is refined by using an iterative global alignment strategy. Thus, sequences of the same welding task do not only start from the same point, they also end at the same point in the mosaic coordinate frame. For the global alignment step, feature tracking is necessary during the pairwise registration of all patches.

In the first step, we perform a graph based optimization of camera poses. We assume all patch positions and orientations in the mosaic coordinate frame as poses which lie all on a plane. We fix the first and the last pose. The last pose equals the required end position and orientation. It is defined via a global mosaic transformation for the last patch. Therefore, the rotation and translation has to be defined. We apply the loop

closure correction as presented in [46] to optimize poses and to close the gap between the current and the required end pose. Figure 4.19 shows a set of mosaic transformations before and after graph based optimization.

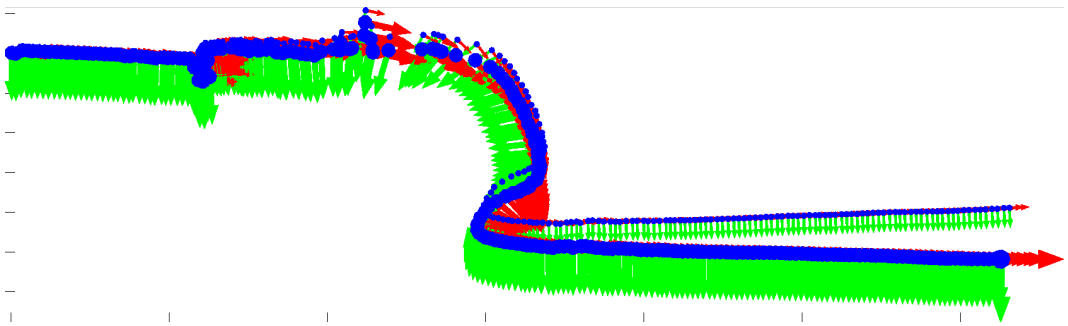


Figure 4.19: Graph based pose optimization. A set of transformations is adjusted to a given end position (bottom right). The figure shows the coordinate frames of each patch in the mosaic coordinate system. The smaller ones are before the graph based optimization, the larger ones after it.

In the second step, the iterative global alignment alternately refines furtheron the feature positions in the mosaic coordinate frame and the global transformations from patch to mosaic coordinates. It is applied according to [17] where two linear steps are altered iteratively. First, feature positions ${}^{m}\widehat{\mathbf{x}}_j$ in the mosaic coordinate frame are estimated:

$$ {}^{m}\widehat{\mathbf{x}}_j = \frac{1}{N_j} \sum_{{}^{k}\mathbf{x}_i \in \eta_j} ({}^{m}\mathbf{x}_i^k) \ , \tag{4.34} $$

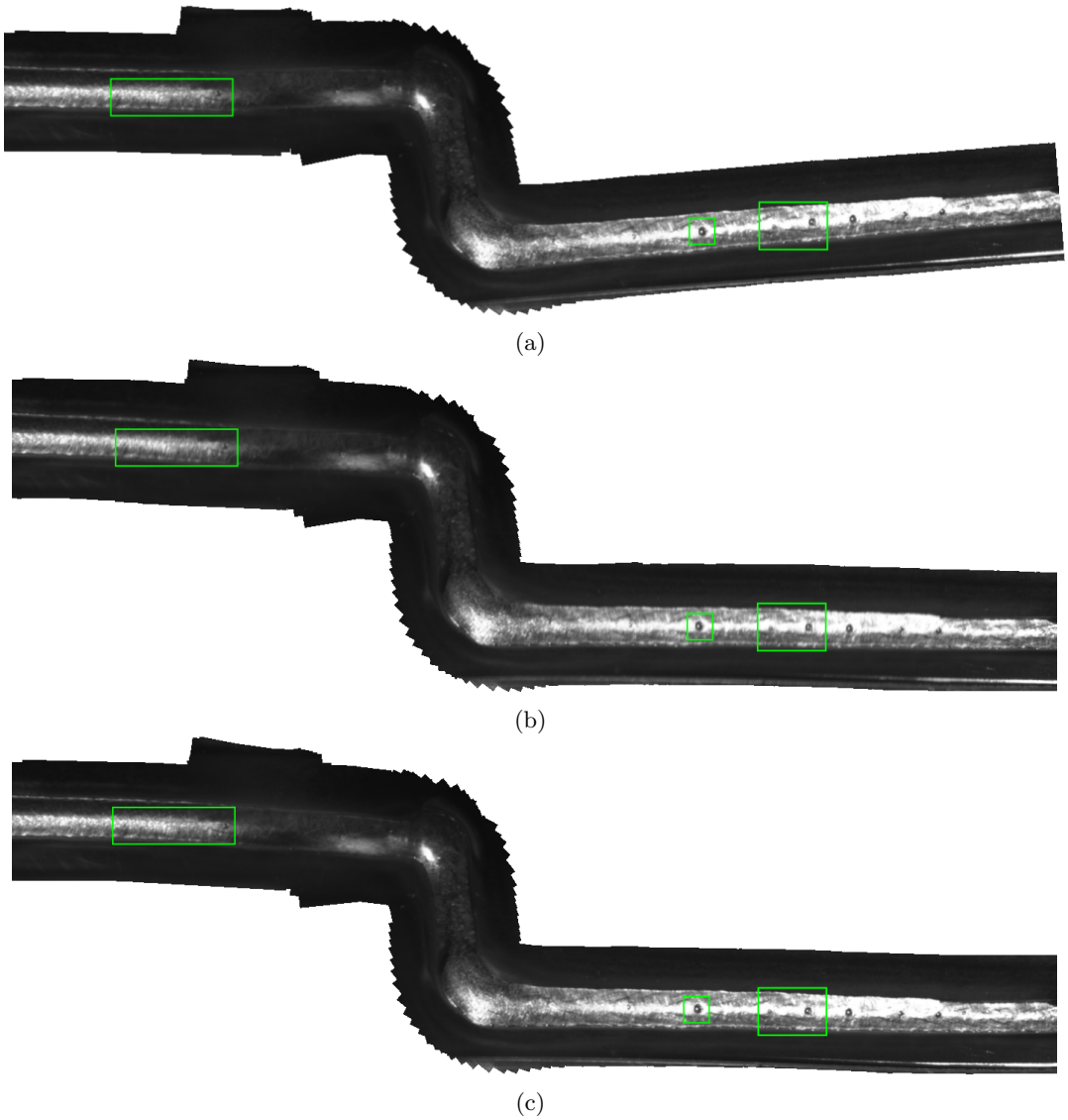
where N_j is the number of images in which feature point ${}^{m}\widehat{\mathbf{x}}_j$ appears, η_j is the set of image points matching to the same feature point in the mosaic, and ${}^{m}\mathbf{x}_i^k$ is a projection of the i-th feature point in the k-th image onto the mosaic point ${}^{m}\mathbf{x}_j$:

$$ {}^{m}\mathbf{x}_i^k = {}^{m}H_k \cdot {}^{k}\mathbf{x}_i \ . \tag{4.35} $$

Then, the list of transformations ${}^{m}H_k$ is recalcualted by using the new point set $({}^{k}\mathbf{x}_i, {}^{m}\widehat{\mathbf{x}}_j)$. The following cost function should be minimized by iterating the two linear steps until a stopping criterion is reached:

$$ E = \sum_{j=1}^{M} \sum_{{}^{k}\mathbf{x}_i \in \eta_j} \left\| {}^{m}\mathbf{x}_j - {}^{m}H_k \cdot {}^{k}\mathbf{x}_i \right\|_2 \tag{4.36} $$

A threshold for the decrease rate of error term E can be used as stopping criterion. An advantage of the iterative global alignment is that it avoids non-linear optimization and accordingly has low computational costs. Figure 4.20 illustrates the steps of global refinement for an exemplary image mosaic.



(a)

(b)

(c)

Figure 4.20: Global refinement. Image mosaic (a) before global refinement, (b) after graph based optimization, and (c) after iterative global alignment on (b). In the graph based optimization the mosaic is adjusted to a given end position. The visible blurring effect in (b) is eliminated with iterative global alignment.

4.6 Summary

We have presented an image stitching system which provides incremental generation of a weld seam survey image. A camera, which is mounted on a welding robot, acquires images during the automatic welding process. The weld seam is tracked on each acquired image. The input to the stitching system is a sequence of weld seam patches which are extracted by the tracker. There are two main stages of the stitching system: a) image registration and b) image blending.

The registration is based on salient features (e.g. SIFT) extraction and matching. A RANSAC routine is used to get a consistent set of matches and to provide a robust rigid motion estimation. To guarantee a successful panoramic image generation we incorporate prior data from tracking into the rigid motion estimation. Thus, the rigid motion information of the tracker is used in the case of registration failure.

The incremental blending method includes a mosaic coordinate frame update if necessary. The blending of a new image and the existing mosaic is only performed for the affected region of the mosaic. By using an incremental blending strategy, the blending weights for the mosaic and the new image are calculated dynamically. Multi-band blending is used to reduce visible seams. If there is global haze on the weld seam patches, haze removal is applied to improve the stitching result. In order to suppress over-exposed regions in the final mosaic, exposure fusion is included in the blending strategy. Exposure fusion provides favorable results if only parts of the input images are over-exposed.

An advantage of our incremental blending approach is that it can be used 'online' for an incremental panorama generation as well as 'offline' for an overall panorama generation if all input images are already registered. In order to improve the result of the final mosaic for archiving, a global refinement step is applied if all images of a welding sequences are registered. If global refinement is used, sequences of the same welding task do not only start from the same point, they also end at the same point in the mosaic coordinate frame.

Chapter 5

Evaluation and Experiments

Contents

We have performed different experiments to evaluate our image stitching system for generation of weld seam image mosaics. The goal of the stitching system is to deliver a robust and accurate weld seam mosaic in adequate runtime. This means that the registration procedure has to output an accurate result to generate a continous panorama. The panorama should be not splitted up. The blending strategy has to provide a good quality of the depicted weld seam. Furthermore, the system should be real-time capable. Experiments are performed in order to check how well the proposed system meets the requirements and if the used approaches are appropriate.

The evaluations are based on visual quality inspections as well as quantitative measurements. For the quantitative evaluation similarity measures like Structural SIMilarity (SSIM) [51] are used. In order to generate ground truth data and to have data for similarity evaluation, we have created synthetic test data. Approaches for generation of ground truth image sequences and evaluation methodologies for image mosaics are discussed in [4, 8, 37].

There are many challenges in the case of stitching weld seam images. Input images can contain bright sparks, heavy smoke and over-exposed regions. The reduction of such noisy elements is also considered in our experiments. We show that our approach can eliminate bright sparks on the final image mosaic. We also applied a method to improve the final result if the input images are over-exposed. In general, the incremental blending approach combined with exposure fusion weights works well for weld seam images compared to other

blending methods. The proposed incremental blending method is also compared to other blending approaches. It achieves the best performance in most of the experiments.

Another topic in our experiments is the performance evaluation of registration results. We found out that registration with SIFT (or SIFTfast) features performs best for our purposes. We also show that global refinement is appropriate to improve the registration repeatability. The experiments also show that our introduced robust rigid motion estimation including prior data work well to guarantee a successful overall panorama generation.

Section 5.1 introduces our applied evaluation metrics. The generation of synthetic test data is explained in Section 5.2. The experiments and evaluations are divided into two main sections. On the one hand, we perform experiments to test the registration (Section 5.3), and on the other hand experiments are also used to analyze challenges regarding image blending and to compare different blending methods against each other (Section 5.4). Section 5.5 summarizes the overall stitching system. It shows that the incremental panorama generation works well to meet the given requirements.

5.1 Evaluation Metrics

We use evaluation metrics in our experiments to measure the stitching results in a quantitative way. In this section these evaluation metrics are defined. Similarity measures provide the possibility to analyze the equality of two images. For the quantitative evaluation of the image mosaics, we use the following similarity measures:

- Structural similarity
- Normalized sum of squared differences
- Normalized cross correlation

5.1.1 Structural Similarity

The Structural SIMilarity (SSIM) [51] is a measure to quantify the visibility of errors between two images. It is a composition of three values: a) luminance, b) contrast and c) structure. First, the luminance for a signal $\mathbf{x}$ is estimated as the mean intensity μ_x. Second, the mean intensity is substracted from the signal $\mathbf{x}-\mu_x$. The contrast is estimated by the standard deviation σ_x. Third, the signal is divided by its standard deviation: $(\mathbf{x}-\mu_x)/\sigma_x$. The Structural SIMilarity (SSIM) Index of two signals $\mathbf{x}$ and $\mathbf{y}$ is a combination of the luminance comparison $l(\mathbf{x},\mathbf{y})$, contrast comparison $c(\mathbf{x},\mathbf{y})$ and structure comparison $s(\mathbf{x},\mathbf{y})$:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha} \cdot [c(\mathbf{x},\mathbf{y})]^{\beta} \cdot [s(\mathbf{x},\mathbf{y})]^{\gamma} \ , \tag{5.1}$$

where α, β and γ are power terms (> 0) that influence the importance of each comparison measure. In the following, the comparison measures are defined. The luminance is

compared due to the mean intensity:

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \ . \tag{5.2}$$

The constant $C1$ is included to guarantee stability when $\mu_x^2 + \mu_y^2$ is close to zero. The contrast comparison is composed in a similar manner with respect to the standard deviation:

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \ . \tag{5.3}$$

Once again a constant $C2$ is used to ensure stability. The structure comparison is based on the correlation between the two unit vectors $(\mathbf{x}-\mu_x)/\sigma_x$ and $(\mathbf{y}-\mu_y)/\sigma_y$ and is defined as

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \ . \tag{5.4}$$

In the work of Wang. et. al [51], SSIMs are calculated for a local window, weighted by a Gaussian function, which moves over the whole image. To get one similarity measure in order to evaluate the overall image quality, a mean SSIM (MSSIM) is used:

$$\mathrm{MSSIM}(X, Y) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{SSIM}(\mathbf{x}_i, \mathbf{y}_i) \ , \tag{5.5}$$

where X and Y are two images, $\mathbf{x}_i$ and $\mathbf{y}_i$ are the image contents at the i-th local window, and N is the total number of local windows in the image.

5.1.2 Normalized Sum of Squared Differences

The normalized sum of squared differences (NSSD) is robust to intensity differences between two images. Given two images I and J of equal size $M \times N$, it is defined as follows:

$$\mathrm{NSSD}(I, J) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y) - J(x,y))^2}{\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y)^2 \cdot \sum_{x=1}^{M}\sum_{y=1}^{N} J(x,y)^2}} \ . \tag{5.6}$$

5.1.3 Normalized Cross Correlation

The correlation between two signals is also a measure of similarity. The normalized cross-correlation (NCC) [30] between two images is calculated by

$$\mathrm{NCC}(I, J) = \frac{\sum_{x,y}(I(x,y) - \mu_I)(J(x,y) - \mu_J)}{\sqrt{\sum_{x,y}(I(x,y) - \mu_I)^2 \cdot \sum_{x,y}(J(x,y) - \mu_J)^2}} \ , \tag{5.7}$$

where μ_I is the mean of image I and μ_J is the mean of image J.

5.2 Synthetic Test Data Generation

For some experiments we generate synthetic test data to perform a quantitative performance evaluation. The goal is to create weld seam input images from existing panoramas. Similar methods for the generation of ground truth data for mosaic evaluation are presented in [4, 8, 37]. We performed this inverse strategy on straight and curvy panoramas.

If a straight panorama is used, weld seam patches are extracted along the x-axis of the mosaic coordinate frame. A patch is extracted every k pixels, where k is randomly chosen between 5 and 12. To simulate disturbing factors during a welding process, three different types of noise are added to the input images: a) salt & pepper, b) Poisson, and c) Gaussian noise.

A second method to generate synthetic test data is to extract warped patches from a curvy panoramic image. The transformation of each weld seam patch into global mosaic coordinates is known. Based on this information, we can generate weld seam input images from the mosaic. Patches are re-warped from mosaic to patch coordinates. Again, three different types of noise are added to the images: a) salt & pepper, b) Poisson, and c) Gaussian noise. Figure 5.1 illustrates both methods for synthetic test data generation.

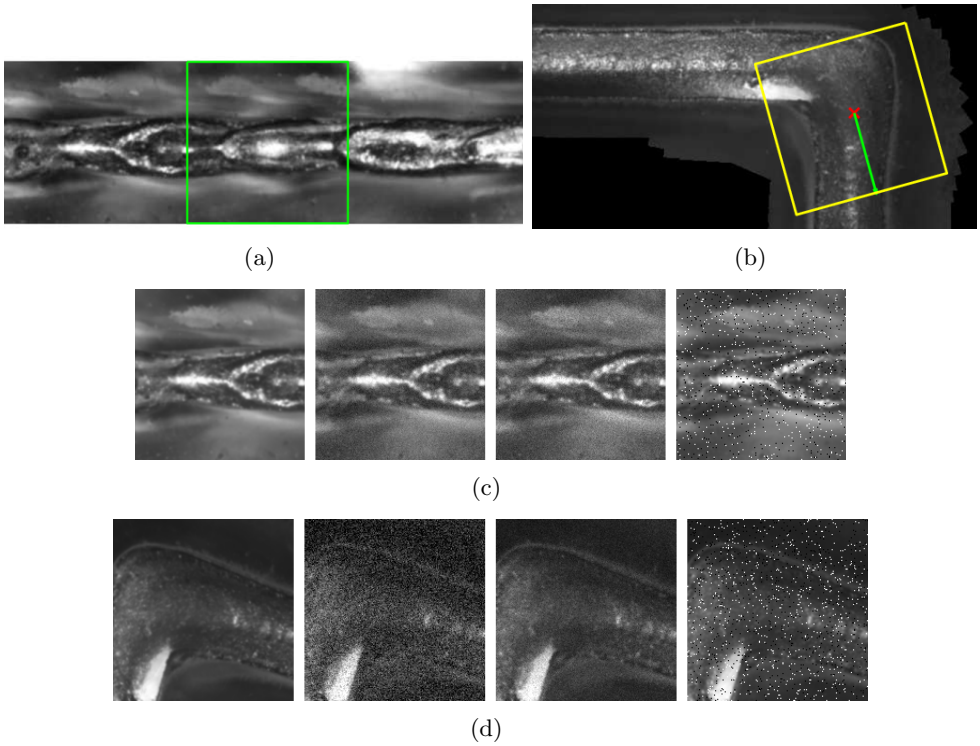


Figure 5.1: Generation of synthetic test data. (a) Patches are extracted on a straight panorama along the x axis. (b) Patches are re-warped due to their mosaic transformation. In (c) and (d) the extracted patches are shown with and without noise. Three different types of noise are used: Gaussian, Poisson and salt & pepper noise.

5.3 Evaluation of Registration Quality

A goal of our stitching system is to deliver a robust and accurate registration result. The panorama should be continous and not splitted up. To evaluate the performance of our robust rigid motion estimation, we have tested the system on 14 data sets which include curved and straight weld seams. The data sets include also different types of challenges like bright sparks, heavy smoke, evaporating water, fast robot motions and illumination changes. The panoramic image generation was successful for each data set. The results are presented in Section 5.3.1.

In order to find out which keypoint descriptor is most suitable for our registration requirements, we compare the matching performance of different descriptors in our data. We use the registration method described in Section 4.3 and compare the matching performance of the following descriptors:

- SIFT
- SURF
- KLT
- OpenCV pyramid optical flow (OFPyrLK)
- SIFTfast

The OpenCV pyramid optical flow matching [7] is used with FAST corners. For SIFT features we use a faster implementation, called SIFTfast (*Fast SIFT Image Features Library*[*] (libsiftfast)), too. It is more than three times as fast as the standard SIFT. A description of these descriptors is given in Section 2.1. Furthermore, an overview about matching approaches is given in 2.2.

An issue is how accurate the run of the weld seam is represented via the registration result. For that purpose, we test the repeatability and accuracy of registration results. The repeatability testing is performed on five different (three straight and two curvy) data sets each containing 19 sequences of a repeated welding task. The 19 sequences are acquired under different conditions. The results of registration repeatability are shown in Section 5.3.2.

For the registration accuracy evaluation, synthetic test data is generated and the results are compared to the ground truth. The synthetic test data consists of three curvy and two straight welding sequences. In order to have also a visual similarity measure of registration results, we blend the registered images and perform a radiometric evaluation on the same synthetic test data too. If synthetic test data is used, no prior data is incorporated into the registration for robust rigid motion estimation. The performance comparisons on synthetic test data are presented in Section 5.3.3 and 5.3.4.

[*] http://sourceforge.net/projects/libsift/

In Section 5.3.5 we test global refinement of registration results as introduced in Section 4.5. We show that this refinement step provides favorable results and strongly improves the repeatability of registration.

5.3.1 Complete Panorama Generation

The successful complete panorama generation is tested on 14 data sets, where each data set contain 19 sequences of a repeated welding task. We use the registration method as described in Section 4.3. The size of the weld seam patches is 140×140. The threshold for the ratio of RANSAC inliers is set to 0.25. We have performed the registration on all 266 sequences. Each registration has been completed successfully. Table 5.1 summarize the registration results. It shows that a successful panorama generation can be guaranteed for all welding sequences. The incorporation of prior tracking data to complete the registration is important especially for welding sequences with heavy smoke.

Description	Runs	Registrations completed	Images	Fallback ratio
Set01	19	19	10408	0.0270
Set02	19	19	5662	0.0019
Set06	19	19	4119	0.0007
Set07	19	19	4285	0.0042
Set08	19	19	4107	0.0467
Set09	19	19	4462	0.2298
Set12	19	19	4179	0.0000
Set13	19	19	4073	0.0000
Set14	19	19	3980	0.0000
Set15	19	19	4407	0.0077
Set21	19	19	4254	0.0002
Set22	19	19	2012	0.0396
Set23	19	19	1976	0.0588
Set24	19	19	1984	0.0010
Total	**266**	**266**	**59908**	**0.0294**

Table 5.1: Registration summary for 266 welding sequences of 14 data sets. The registration has been completed successfully for all welding sequences. The fallback ratio indicates how often prior tracking information is used, in relation to the number of images, to continue the registration.

5.3.2 Repeatability

A characteristic which is evaluated for different registration strategies is the repeatability. This means if we have the same welding task which is repeated under different conditions, in the ideal case the geometry of the computed panorama should always be the same. In this section, we evaluate the repeatability of the registration results. We have used

five different data sets, where each data set contains 19 welding sequences of a repeated welding task. Figure 5.2 shows exemplary panoramas of one welding sequence per data set. Three data sets contain sequences of approximately straight weldings. The other two data sets contain welding tasks with a curved weld seam. The threshold for the fraction of RANSAC inliers is set to 0.25. With other words at least 25% of the matches have to be inliers to accept a rigid motion estimation.

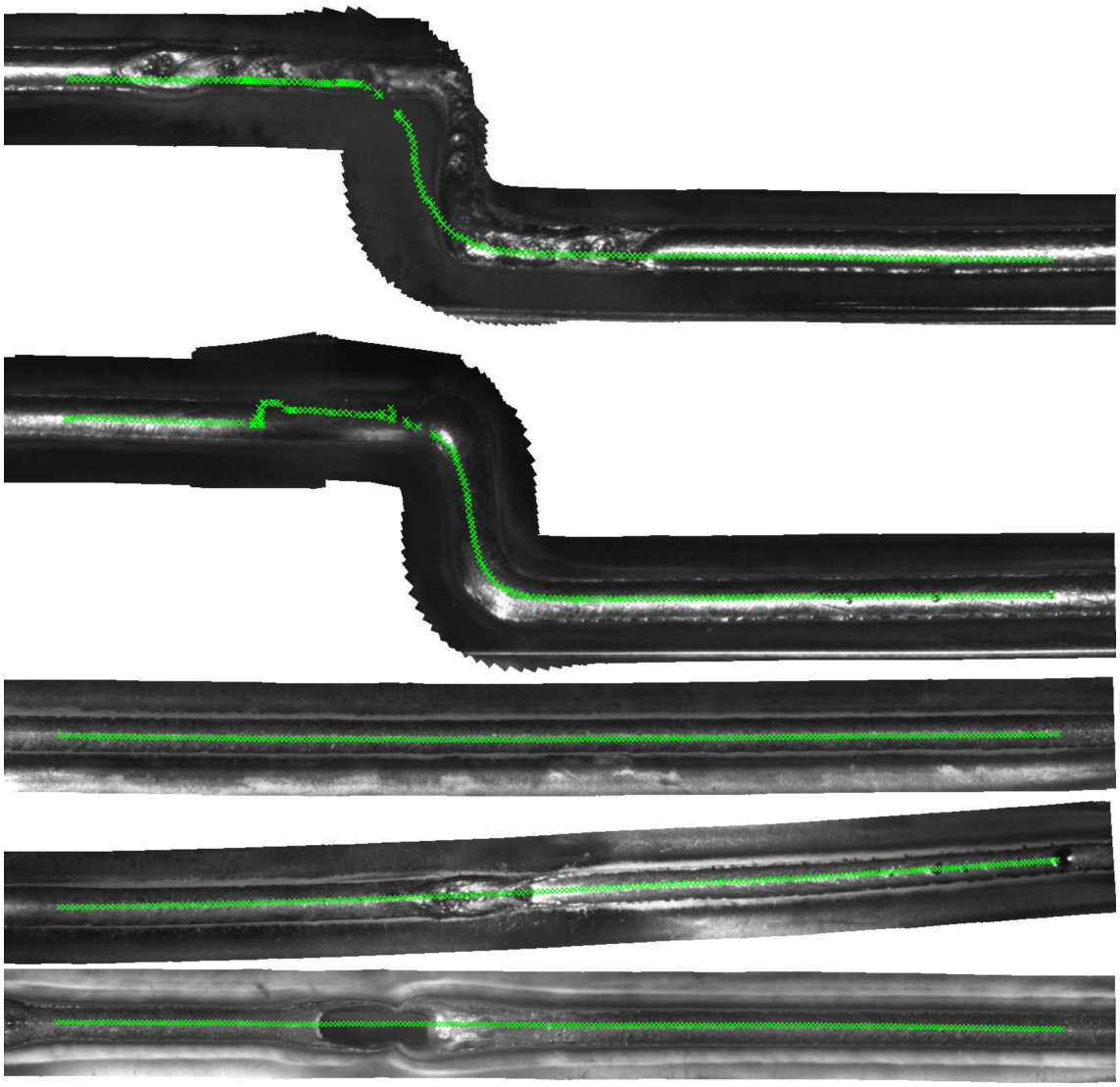


Figure 5.2: Sample panoramas of the five data sets used for registration repeatability evaluations. The center positions of all patches are marked in the panoramas.

If a registration for a welding sequence is finished, the position of each patch in a common mosaic coordinate frame can be calculated. More precisely, we transform the center position of each patch to the mosaic coordinate frame to generate a registration trajectory

of each welding sequence. All sequences start at the same point in the mosaic coordinate frame. Figure 5.3 visualizes a registration result. The figure shows the position of each patch in mosaic coordinates. Connected center positions of patches form a registration trajectory.

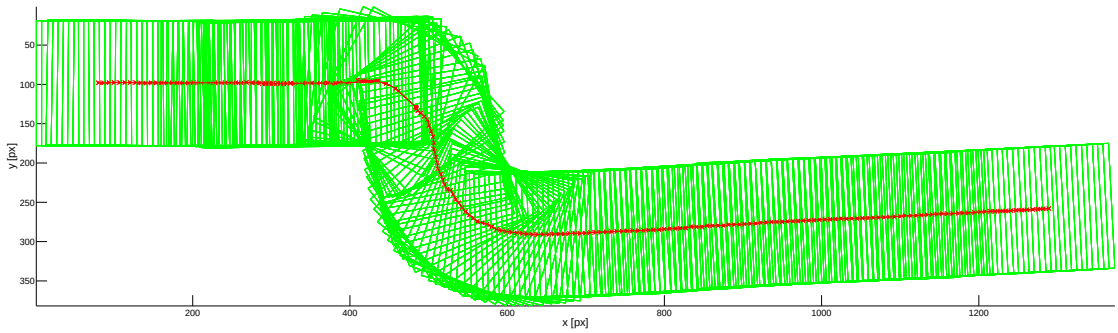


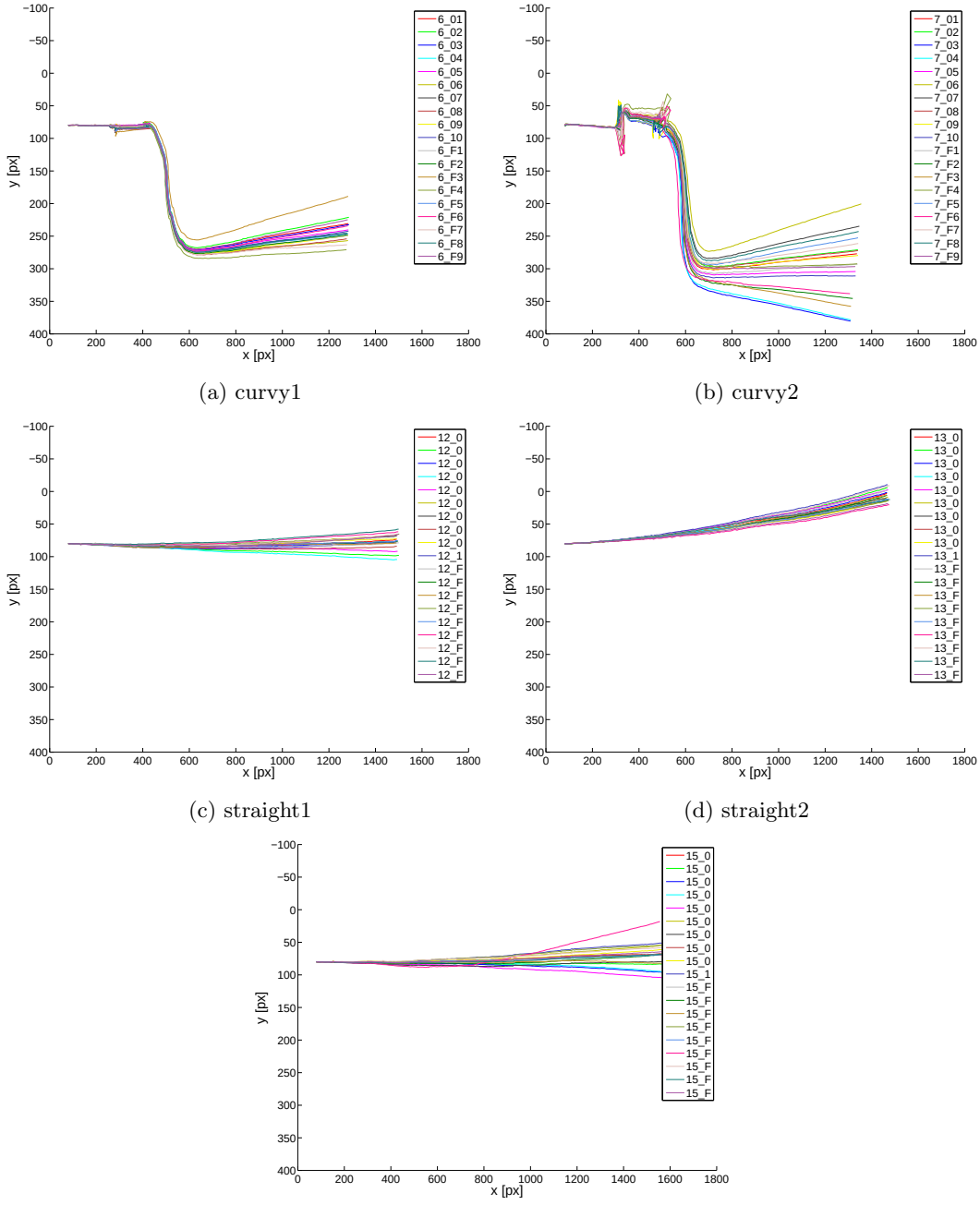
Figure 5.3: Visualization of a registration result. The registration result of a welding sequence is visualized by showing the patch positions in mosaic coordinates. Connecting the center positions of each patch results in a registration trajectory (red).

In this way, 19 trajectories are generated for one registration method. This trajectories can be compared amongst each other. For each complete data set we evaluated the standard deviation for three nodes along all trajectories for one data set. A node is located after one third, after two thirds and after the end of the registrations. Table 5.2 summarizes the results. The tests demonstrate that the registration methods SIFT and SIFTfast achieved best results. They have the lowest standard deviations. The large deviations of the OpenCV method (OFPyrLK) in x direction are remarkable. The number of fallbacks indicates how often prior tracking information is used to continue the registration. It is also significantly lower for SIFT and SIFTfast. This result suggests that SIFT and SIFTfast are most robust for pairwise image matching. Prior tracking information is for example necessary if there is too heavy smoke on weld seam patches and thereupon the standard pairwise registration fails. The repeatability performance of our proposed method with SIFTfast features is visualized in Figure 5.4.

Another way of visualization is the coverage-cumulative error score graph. We adapted the method of Paalanen et. al [37] to visualize the cumulative standard deviation of the 19 trajectories at 20 nodes. For each trajectory, we computed the difference (in x and y direction) between consecutive nodes. For these differences we computed the standard deviation along all trajectories and sorted all values in ascending order. The graph denotes the smallest possible cumulative standard deviation when selecting n nodes. As the graph is monotonically increasing, the repeatability performance of the registration methods is straighforward comparable (see Figure 5.5). It is obvious that SIFT and SIFTfast performed best.

	Standard deviations [px]						Number of fallbacks
	Node1		Node2		Node3		
	x	y	x	y	x	y	
curvy1							
SIFT	2.69	2.01	4.13	8.87	4.21	18.66	1
SURF	2.50	3.31	4.33	10.90	4.22	24.02	30
KLT	2.67	2.86	4.46	9.87	4.66	19.28	37
OFPyrLK	10.71	1.54	10.93	9.49	10.55	26.81	44
SIFTfast	2.52	1.63	2.76	8.63	2.89	17.73	2
curvy2							
SIFT	8.10	7.01	14.86	21.17	15.71	54.21	6
SURF	8.84	17.25	26.11	41.14	28.09	91.11	60
KLT	4.69	14.03	25.07	25.68	24.91	69.70	64
OFPyrLK	79.99	6.77	83.36	25.80	83.13	65.12	92
SIFTfast	7.68	6.56	14.67	21.13	14.38	48.79	9
straight1							
SIFT	1.46	2.17	2.36	6.76	3.52	12.69	0
SURF	1.74	2.86	3.20	8.73	4.70	19.95	7
KLT	1.98	5.30	4.39	20.45	8.44	49.77	1
OFPyrLK	1.73	5.26	3.62	15.10	5.02	34.02	4
SIFTfast	1.51	2.25	2.51	6.17	3.89	12.17	0
straight2							
SIFT	1.00	1.72	2.20	4.35	4.30	7.84	0
SURF	1.17	2.35	3.22	6.64	5.19	13.77	4
KLT	0.85	4.21	2.17	11.31	4.44	20.20	3
OFPyrLK	1.08	1.82	2.86	6.05	23.55	12.51	3
SIFTfast	1.00	2.09	2.08	5.10	4.14	9.02	0
straight3							
SIFT	2.23	2.09	6.54	6.94	6.85	15.55	30
SURF	8.29	5.10	21.58	20.53	24.71	47.15	83
KLT	1.88	2.48	5.56	8.78	6.06	20.13	23
OFPyrLK	4.07	4.58	17.55	13.93	17.36	27.13	21
SIFTfast	1.15	3.01	3.75	8.17	4.51	19.28	21
average							
SIFT	3.10	3.00	6.02	9.62	6.92	21.79	7.4
SURF	4.51	6.17	11.69	17.59	13.38	39.20	36.8
KLT	2.41	5.78	8.33	15.22	9.70	35.82	25.6
OFPyrLK	19.52	3.99	23.66	14.07	27.92	33.12	32.8
SIFTfast	2.77	3.11	5.15	9.84	5.96	21.40	6.4

Table 5.2: Registration repeatability results. Standard deviations at three nodes along registered welding sequences for each data set, and number of fallbacks. A node is located at the central position of the relevant patch in the mosaic coordinate frame. Each data set contains 19 sequences of a repeated welding task, yielding in about 4200 pairwise registrations in total per data set. The number of fallbacks indicates how often prior tracking data is used to continue the registration.



(a) curvy1

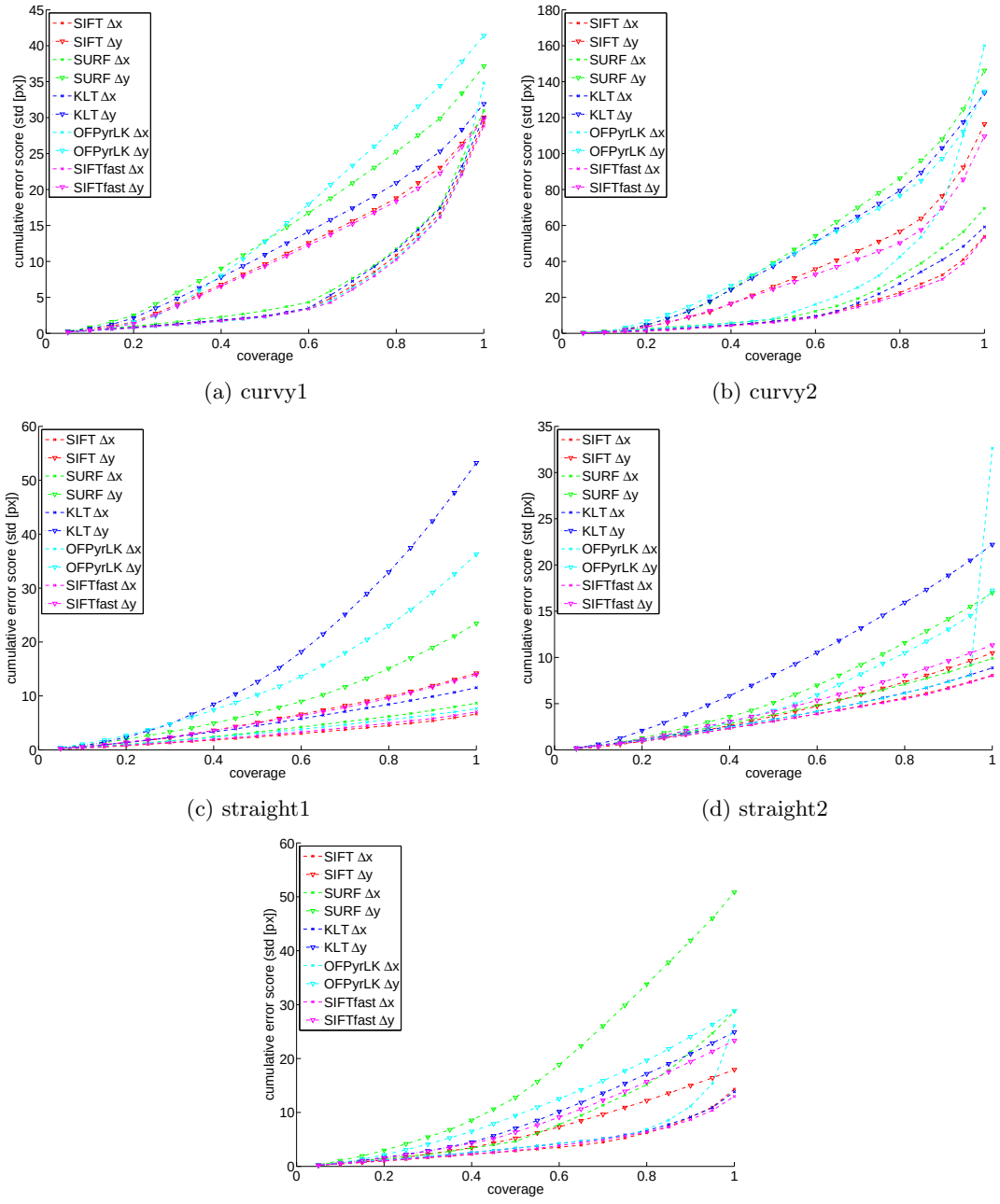
(b) curvy2

(c) straight1

(d) straight2

(e) straight3

Figure 5.4: Registration repeatability. The figure shows the registration results for five different data sets with 19 sequences of a repeated welding task each. The visualized trajectories are connected center points of the registered patches in the mosaic coordinate frame. The registration is based on SIFTfast features. The trajectories on the top represent curved weld seams. The other three trajectories correspond to approximate straight weldings. All sequences start left. It is obvious that the distribution of registration results is increasing towards the end of the welding sequences.



(a) curvy1

(b) curvy2

(c) straight1

(d) straight2

(e) straight3

Figure 5.5: Registration repeatability CCES. The cumulative error score (CCES) graphs show the minimum possible cumulative standard error for $n\%$ of the registration (coverage) for 19 sequences each. The error value is given by the standard deviation for the difference in x and y direction between two nodes along all 19 sequences. There are 20 nodes for each registered welding sequence.

5.3.3 Evaluation of Accuracy

The evaluation of accuracy is done with synthetic test data (see Section 5.2). In order to have ground truth image sequences, we generated five different synthetic test data sets: three curvy and two straight weld seam panoramas. Figure 5.6 shows the image mosaics which the synthetic test data is generated from. Two different noisy data sets are generated from each input sequence. In each case, 70% of the input images contain Gaussian and salt & pepper noise, respectively. The registration is performed for all five test data sets. The center positions of all registered patches are compared to the ground truth. Table 5.3 gives an overview of the mean distance to ground truth in pixel for all tested registration methods.

	Number of	Mean distance to ground truth [px]				
	images	SIFT	SURF	KLT	OFPyrLK	SIFTfast
Salt & pepper						
Set1	341	6.93	22.70	12.37	54.04	2.9114
Set2	186	16.57	34.39	16.10	25.43	14.1846
Set3	226	8.15	45.30	6.7901	17.56	9.55
Set4	163	2.6726	17.33	5.14	3.21	9.77
Set5	169	3.37	16.89	1.5539	6.43	6.61
Gaussian						
Set1	341	12.74	14.79	9.12	236.08	7.1046
Set2	186	7.76	29.43	15.20	63.55	5.1582
Set3	226	3.70	30.81	43.62	33.82	2.8367
Set4	163	9.54	6.05	14.77	2.8986	4.43
Set5	169	5.8822	11.88	9.96	96.04	11.28
No noise						
Set1	341	1.27	6.96	2.68	37.34	0.9224
Set2	186	0.7922	1.80	10.46	15.07	1.12
Set3	226	1.5987	7.70	7.74	10.45	1.97
Set4	163	0.9404	4.05	3.78	1.96	1.72
Set5	169	1.00	2.62	0.77	0.99	0.3405

Table 5.3: Evaluation of registration accuracy with synthetic test data. The registration accuracy is evaluated on five different synthetic test data sets: three curvy and two straight weld seam panoramas. Each set contains one welding sequence for different noise types. We calculate the Euclidean distance to the ground truth of the center position of each patch in the mosaic coordinate frame.

Our tests show that SIFT and SIFTfast achieve best results in most cases. Hence, SIFTfast is a good alternative to standard SIFT with almost similar results. KLT has the best performance on two data sets with salt & pepper noise. SURF and OFPyrLK perform rather poor in most test cases. In the tests without noise the mean distance of SIFT and SIFTfast is around or below one pixel. The registration results are visualized

via trajectories in Figure 5.7 (70% of input images with Gaussian noise) and Figure 5.8 (70% of input images with salt & pepper noise). The following parameters has been used for the generation of the noisy input images: density of salt & pepper noise 4%, Gaussian noise $\mu = 0, \sigma = 0.001$.

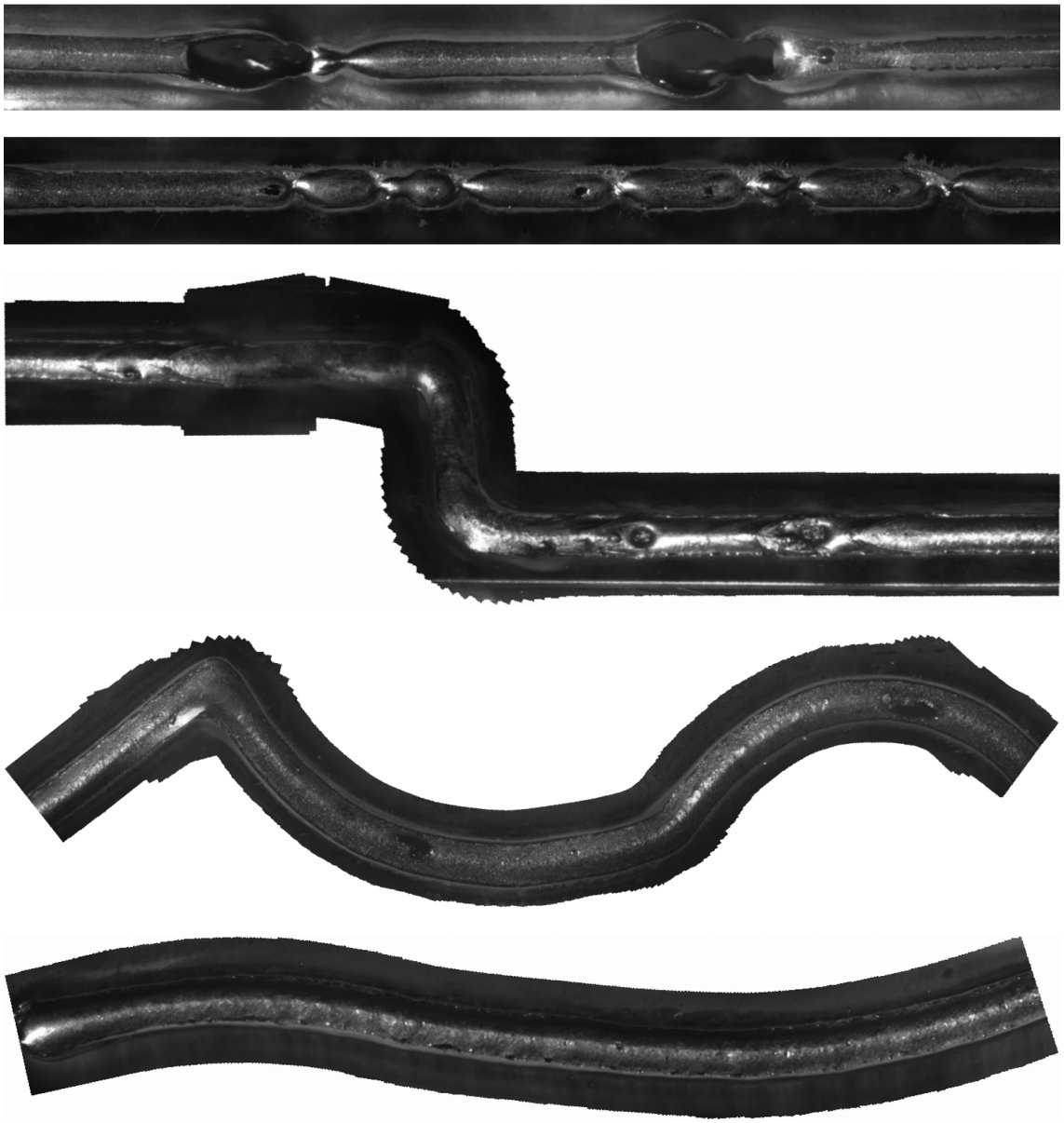
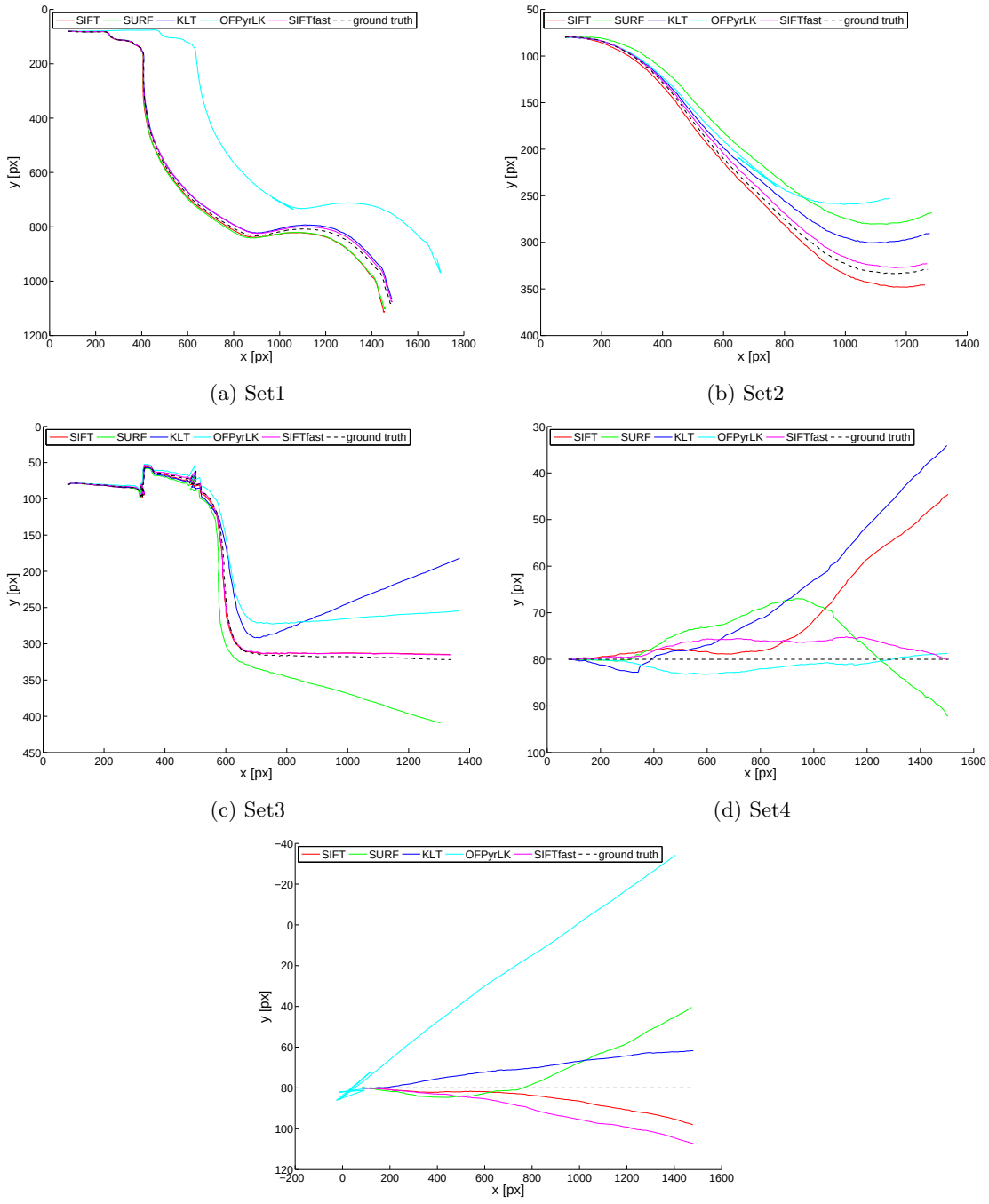


Figure 5.6: Weld seam mosaics from which synthetic test data for registration testing is generated.



(a) Set1

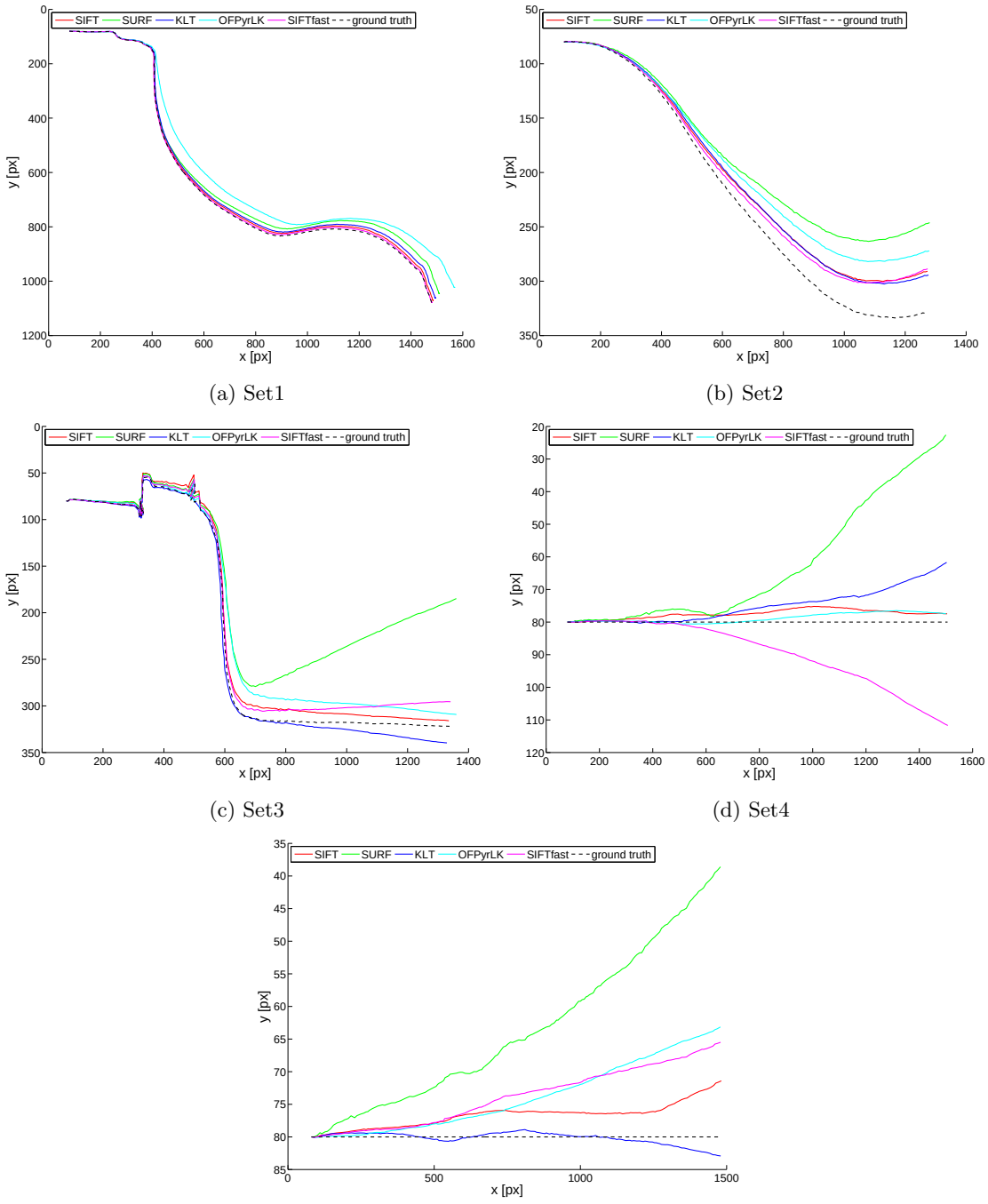
(b) Set2

(c) Set3

(d) Set4

(e) Set5

Figure 5.7: Evaluation of registration accuracy with input images with Gaussian noise. The figure shows the connected center positions of all registered patches in the mosaic coordinate frame. All five synthetic test data sets are registered with five different methods. The results are also compared to ground truth data (dashed line).



(a) Set1

(b) Set2

(c) Set3

(d) Set4

(e) Set5

Figure 5.8: Evaluation of registration accuracy with input images with salt & pepper noise. The figure shows the connected center positions of all registered patches in the mosaic coordinate frame. All five synthetic test data sets are registered with five different methods. The results are also compared to ground truth data (dashed line).

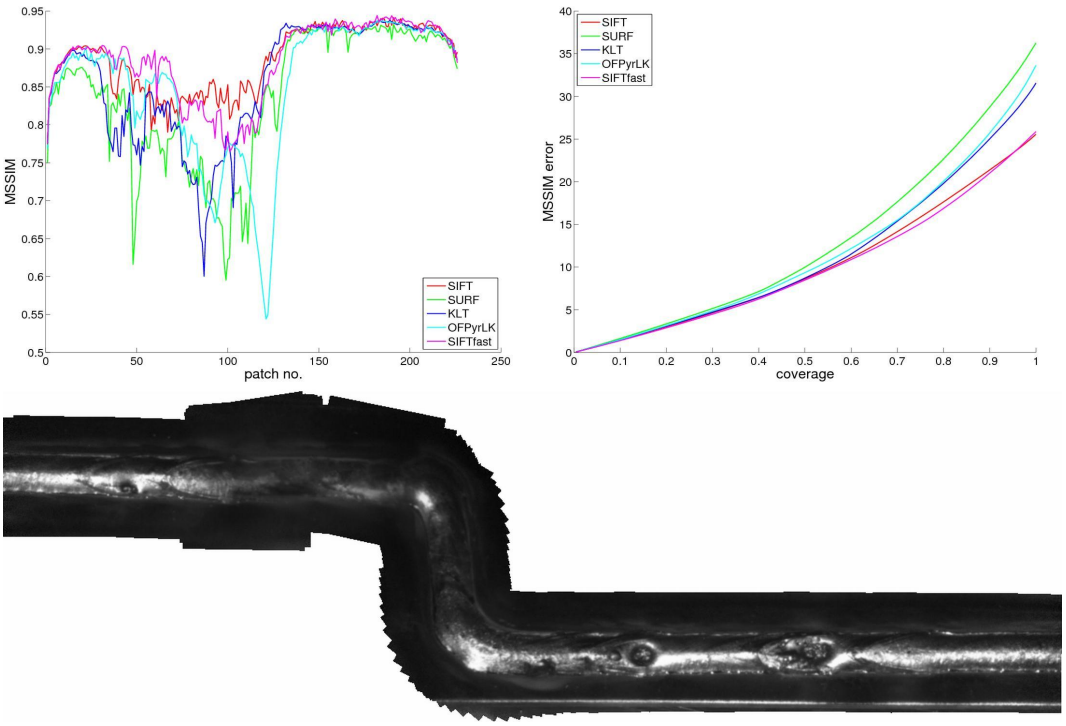
	Number of images	Mean MSSIM				
		SIFT	SURF	KLT	OFPyrLK	SIFTfast
Salt & pepper						
Set1	341	0.7438	0.7154	0.7536	0.7133	0.7411
Set2	186	0.6545	0.6323	0.6589	0.6547	0.6556
Set3	226	0.6378	0.6080	0.6322	0.6161	0.6395
Set4	163	0.7350	0.7097	0.7223	0.7329	0.7195
Set5	169	0.5993	0.5787	0.5891	0.5814	0.5926
Gaussian						
Set1	341	0.9086	0.8865	0.9263	0.8173	0.9115
Set2	186	0.8999	0.8970	0.9100	0.8413	0.9054
Set3	226	0.8869	0.8396	0.8604	0.8512	0.8855
Set4	163	0.9175	0.9085	0.9082	0.9236	0.9186
Set5	169	0.8643	0.8519	0.8666	0.8364	0.8665
No noise						
Set1	341	0.9622	0.9575	0.9695	0.9334	0.9630
Set2	186	0.9803	0.9776	0.9764	0.9731	0.9804
Set3	226	0.9709	0.9683	0.9687	0.9281	0.9709
Set4	163	0.9845	0.9819	0.9793	0.9766	0.9833
Set5	169	0.9816	0.9825	0.9796	0.9830	0.9832

Table 5.4: Registration radiometric evaluation of synthetic test data. The radiometric evaluation is performed on five different synthetic test data sets: three curvy and two straight weld seam panoramas. Each set contains one welding sequence for different noise types. For the similarity evaluation, noise-free patches are compared to their corresponding regions in the mosaic.

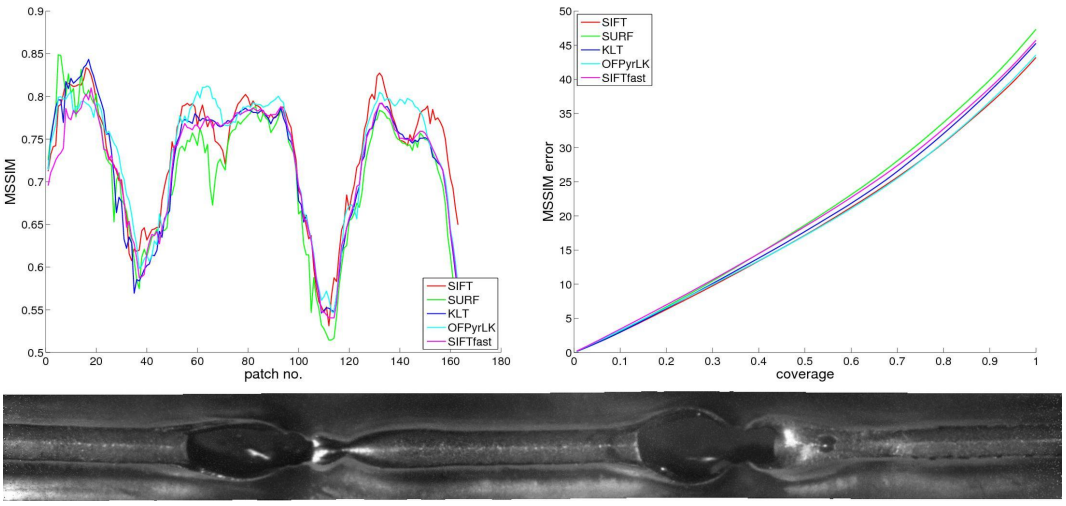
5.3.4 Radiometric Evaluation

For the radiometric evaluation of our registration results, we use the same test data as for the geometric tests described in Section 5.3.3. Each test data set is registered with the five different registration methods and stitched into a mosaic by using incremental image blending (see Section 4.4). To obtain a quantitative comparison between the registration results, we use the similarity measure MSSIM (see Section 5.1). Each noise-free patch of an image sequence is compared to its corresponding part in the mosaic. It must be mentioned here, that the region in the mosaic contains information from all overlapping patches. Thereby noise from the input images should be reduced. The evaluations have been done for three different types of noise. The first run uses input images, where 70% contain salt & pepper noise. For the second one Gaussian noise is added to 70% of the input images. In the third one the registration is performed with noise-free images.

The results for the five test data sets are summarized in Table 5.4. The table presents the mean score of all similarity evaluations per image sequence. Two exemplary visualizations of the evaluation results are depicted in Figure 5.9. SIFT, SIFTfast and KLT achieve best results for all five data sets. It has to be mentioned that for KLT and Gaussian noise input images a successful registration is not always accomplished with the first attempt. With respect to robustness SIFT and SIFTfast provide the best performances, where in



(a)



(b)

Figure 5.9: Radiometric registration evaluation. Two exemplary visualizations for the radiometric evaluation with synthetic test data with (a) 70% Gaussian noise patches and (b) 70% salt & pepper noise patches: MSSIM of each patch, cumulative MSSIM error relating to patch coverage and blended image mosaic. For the similarity evaluation, noise-free patches are compared to their corresponding regions in the mosaic.

each case the first registration was already successful.

The comparison to the geometric evaluation in Section 5.3.3 indicates that the method with the best geometric result does not necessarily lead to the best radiometric performance. However, the differences between the similarity results of a test data set are partially small.

5.3.5 Global Refinement

In this section we show that global refinement improves the repeatability of registration results. We have implemented the global refinement strategy as described in Section 4.5. We use a complete data set with 19 sequences and perform registration with and without global refinement at the end. If global refinement is used, the position and the orientation of the last image is given and the registration is arranged to it. Thus, all welding sequences do not only start from the same point, they also end at the same point. It is obvious, that the repeatability is improved with global refinement (see Figure 5.10 and Figure 5.11). The trajectories are less distributed, and start and end at the same positions. It has to be considered that the global refinement step does not implicitly improve the radiometric quality of the final image mosaic, but the geometry of the image mosaics of all welding sequences is considerably improved.

Table 5.5 shows that the positions of the nodes are quite more similar by using global refinement. The standard deviation is calculated at three nodes along 19 registered welding sequences for each data set. A node is located at the central position of the i-th patch in the mosaic coordinate frame. The difference of the similarity measure MSSIM is marginal. For the similarity evaluation the mosaics shown in Figure 5.10 and Figure 5.11 are compared with their corresponding frames.

	Standard deviations [px]						Mean
	Node1		Node2		Node3		MSSIM
	x	y	x	y	x	y	
curvy							
No BA	2.6666	1.4522	3.1383	5.7687	3.5550	12.1349	0.9301
With BA	1.7255	1.5368	0.7752	0.7631	0.0000	0.0000	0.9302
straight							
No BA	0.7674	1.8794	2.2290	4.4341	4.2283	8.7297	0.9416
With BA	1.2495	0.9586	0.9335	1.4201	0.0000	0.0000	0.9415

Table 5.5: Registration with global refinement. Global refinement considerably improves the registration repeatability. The standard deviation is calculated at three nodes along 19 registered welding sequences for each data set. A node is located at the central position of the i-th patch in the mosaic coordinate frame. Additionally, the resulting mosaics are compared with their corresponding frames. The mean of the calculated similarity measure MSSIM is nearly the same.

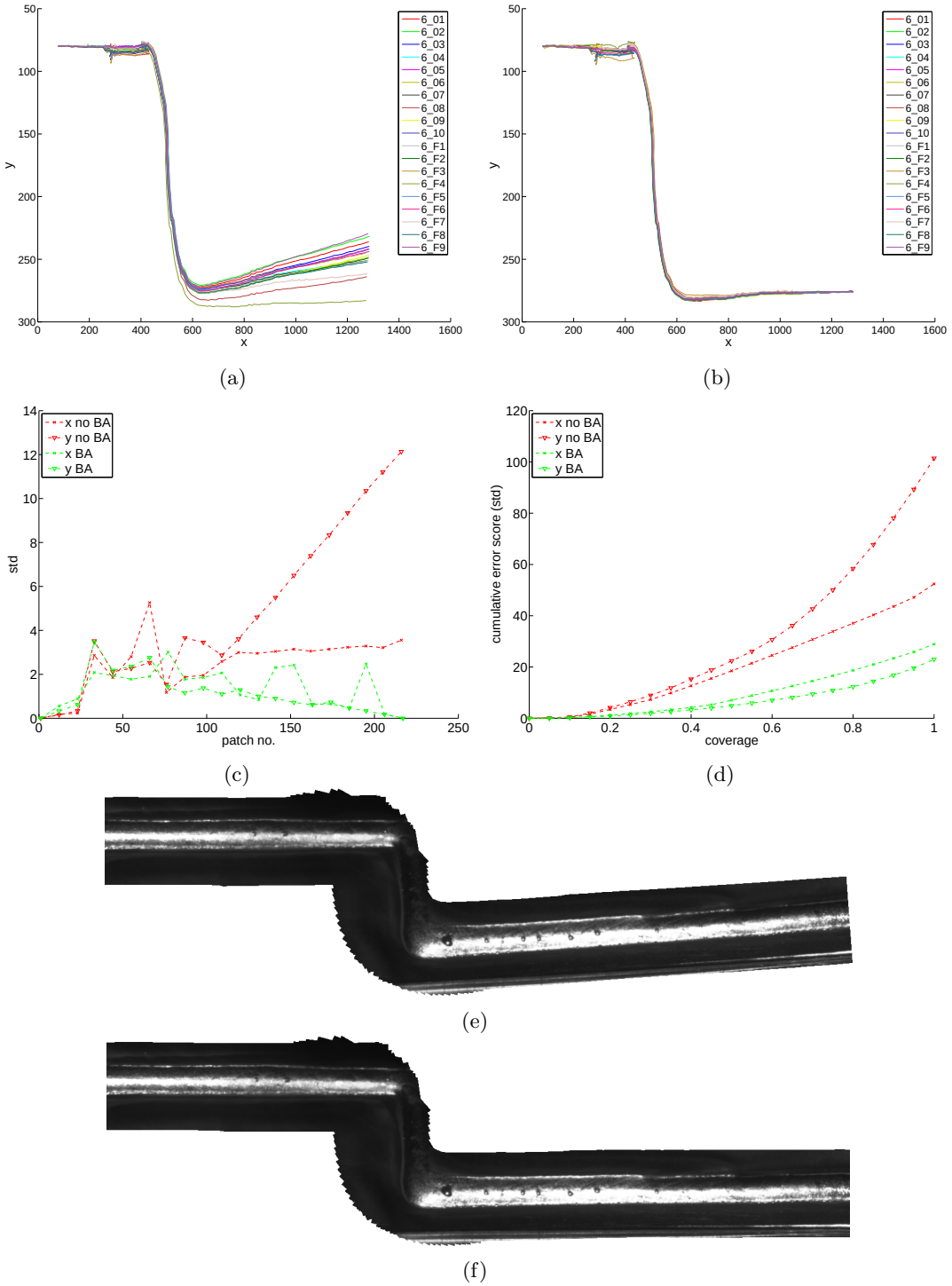


Figure 5.10: Registration with global refinement, for a curvy weld seam. The repeatability of the registration result for 19 welding sequences is improved with global refinement: (a) registration trajectories without global refinement, (b) registration trajectories with global refinement, (c) standard deviations along 20 nodes of the trajectories, (d) cumulative error score graph showing the minimum cumulative standard deviation of n nodes, and exemplary image mosaics (e) before and (f) after global refinement.

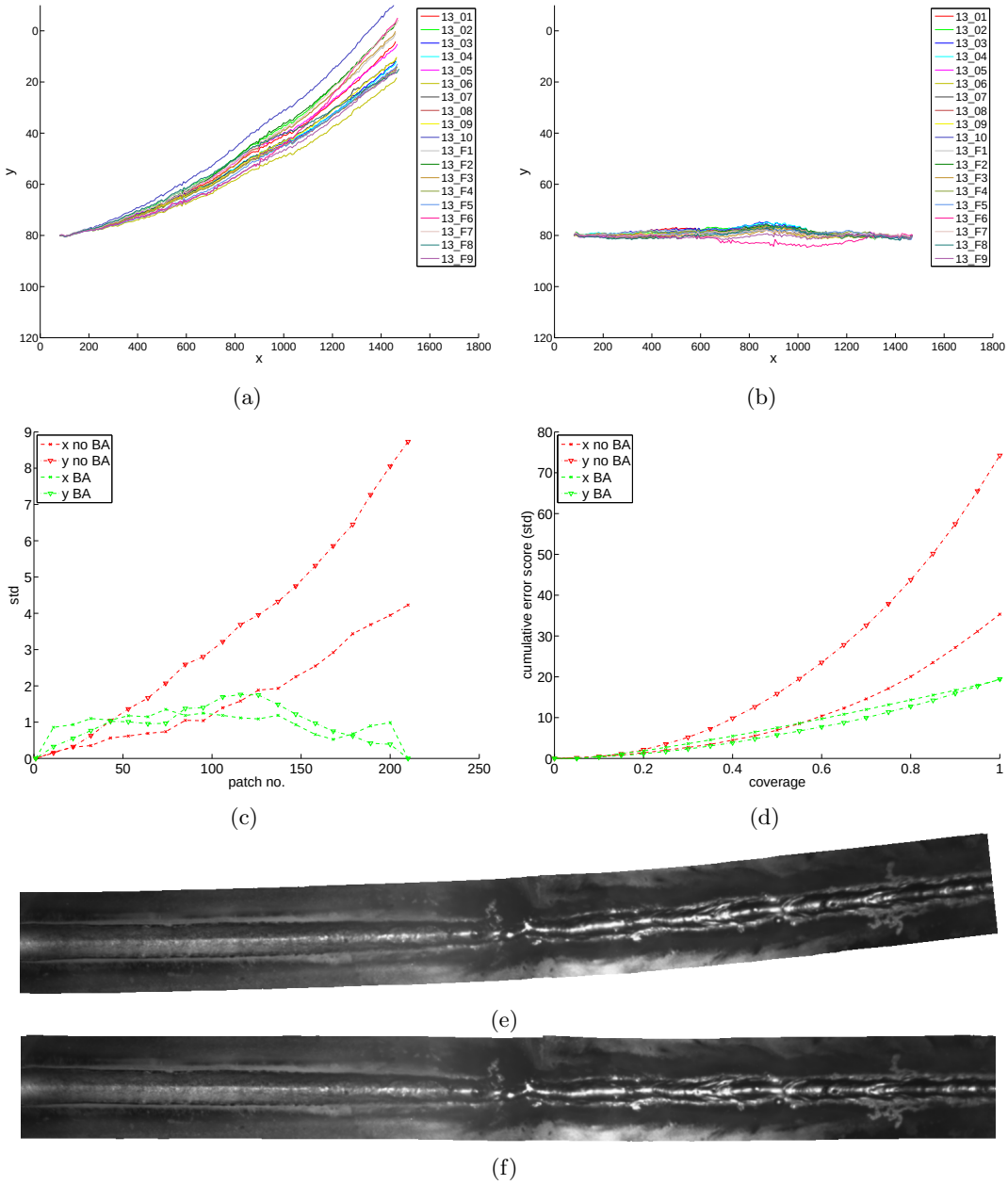


Figure 5.11: Registration with global refinement, for a straight weld seam. The repeatability of the registration result for 19 welding sequences is improved with global refinement: (a) registration trajectories without global refinement, (b) registration trajectories with global refinement, (c) standard deviations along 20 nodes of the trajectories, (d) cumulative error score graph showing the minimum cumulative standard deviation of n nodes, and exemplary image mosaics (e) before and (f) after global refinement.

5.4 Evaluation of Blending Results

With respect to image blending, the final stitching result should be enhanced. Challenges are to eliminate noisy effects like bright sparks and wads of smoke as well as to reduce over-exposure. In general, the structure of the weld seam should be well depicted. We use incremental image blending as described in Section 4.4. The following blending challenges are tested in the next sections:

- Bright sparks (Section 5.4.1)
- Smoke (Section 5.4.2)
- Over-exposure (Section 5.4.3)
- Synthetic noise (Section 5.4.4)

In order to have comparisons to other blending strategies, we evaluate also different approaches. A radiometric evaluation is performed for different synthetic test data sets to determine the most appropriate blending method (see Section 5.4.5). The following general blending approaches are used in our test cases:

- Incremental image blending
- Feathering
- Blending with binary weights
- Stitching in the gradient domain
- Median blending

Based on the evaluation results, we recommend to use incremental image blending with adaptions depending on the welding task. For example, if there are over-exposed images, exposure fusion weights should be adjusted accordingly.

5.4.1 Bright Sparks

Bright sparks occur quite often during a welding process. If the overlap between weld seam patches is rather large, sparks can eliminated very well with multi-band blending and incremental blending weights. A spark is eliminated in a large part with the information of the overlapping images. Figure 5.12 shows some patches with bright sparks and corresponding stitching results. Larger sparks on dark background are still visible using traditional blending methods. One way to completely eleminate sparks is to use median blending. This means that for each pixel in the final mosaic the median pixel from a stack of overlapping images is chosen. The median method has the disadvantage that it is computationally expensive. Furthermore, the information of all warped images of a sequence has to be kept for the median calculation. For each input image all corresponding

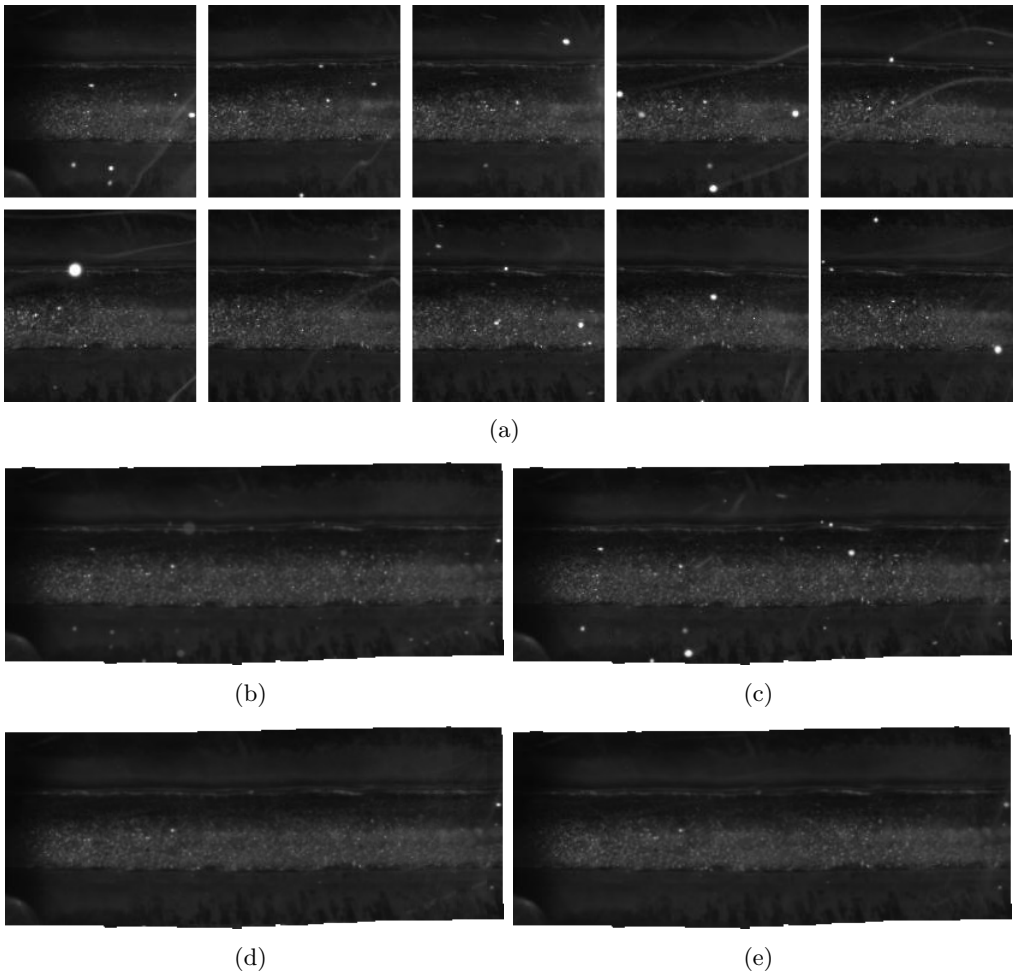
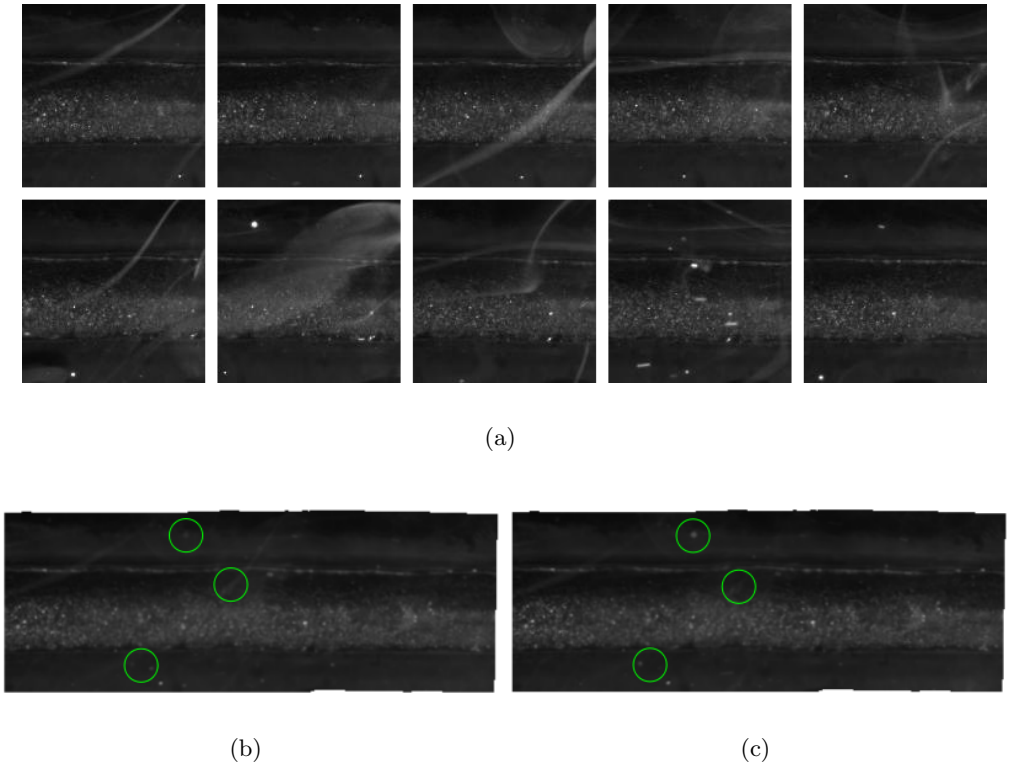


Figure 5.12: Blending of noisy sequences with bright sparks. 20 weld seam patches of size 160×160 are stitched together with different strategies. In (a), 10 input images are shown. (b) Geometry-based incremental blending partially eliminates bright sparks. (c) If using a binary weighting mask for blending, sparks might survive completely. (d) Median blending is an optimal strategy to eliminate sparks. (e) Similar results are achieved if exposure fusion is added to the incremental blending weights of (b).

overlapping images have to be determined. Our implementation of incremental blending is faster by approximately a factor of 6 compared to median blending. Incremental blending combined with exposure fusion achieves similar results like median blending. The parameters for exposure fusion have to be chosen appropriately to eliminate the sparks. The following parametrization is used for the panorama shown in Figure 5.12 (e): incremental blending: $p = 0.0$, $r = 3$; exposure fusion: $\sigma = 0.1$, $wc = 0.5$, $we = 1.0$, $g = 5$, $c = 0.1$ and window size is 19.

5.4.2 Smoke

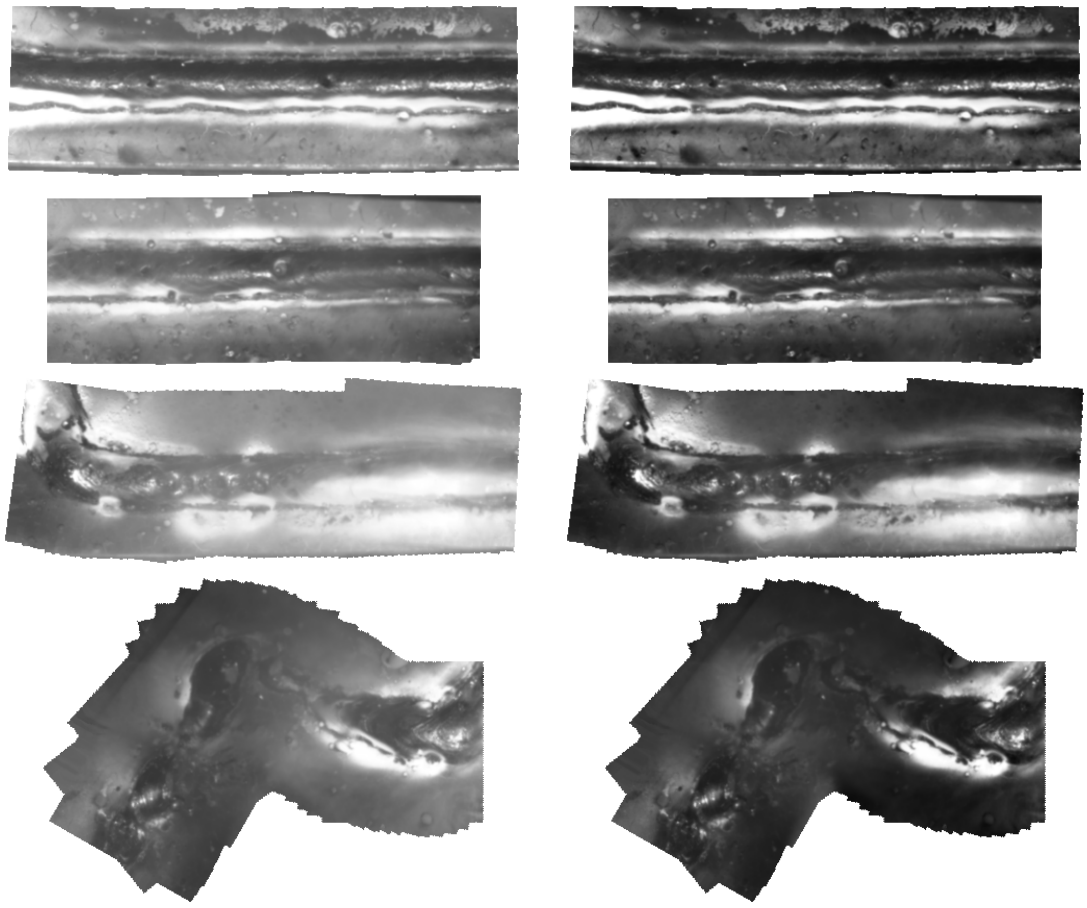
The task of reducing heavy smoke on the stitched panorama is not a trivial one. If only some isolated wads of smoke appear on patches, the smoke is well reduced via blending (see Figure 5.13). If there is heavy smoke on almost each patch, the quality of the final result suffers. In order to enhance the final result, we include a dehazing strategy (see Section 4.2.2). This method yields partially improvements, but it might have the disadvantages that regions in the image mosaic sometimes become too dark. Figure 5.14 illustrates the effects of image dehazing. All image mosaics in the figure are generated with incremental blending weights and multi-band blending. The results with dehazing have a better contrast and the structure is improved in some regions.



(a)

(b) (c)

Figure 5.13: Smoke reduction. By using an appropriate blending method, smoke on input images is reduced on the stitched panorama. In (a), 10 input images are shown. In addition to smoke, there are also bright sparks visible. Both image mosaics are obtained with multi-band blending, (b) with incremental blending weights and (c) with feathering. Minor differences between the stitching results are visible.

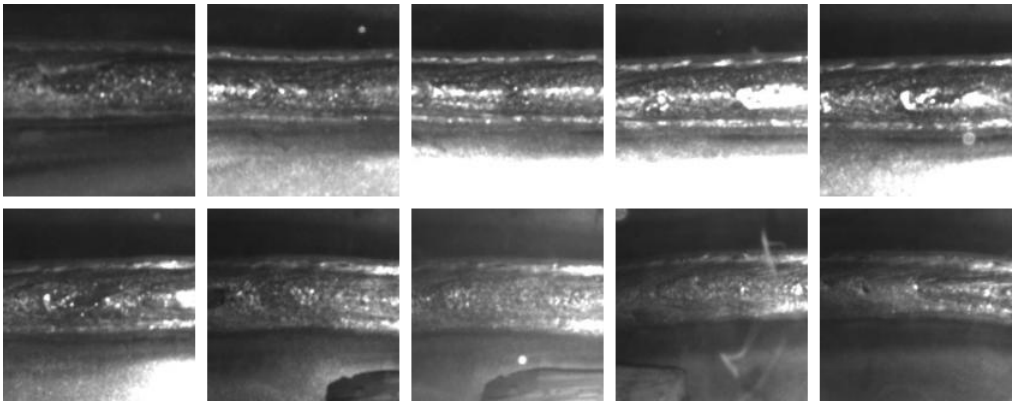


(a) (b)

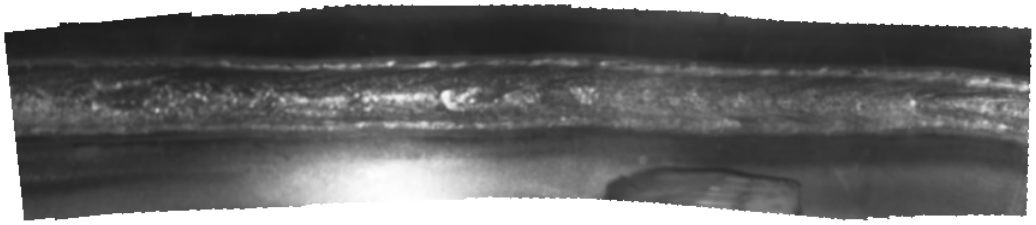
Figure 5.14: Dehazing results. The figure shows image mosaics of weld seam images (a) without and (b) with dehazing. By applying a dehazing strategy, some regions are enhanced. The contrast and the structure of some regions are improved. Each image mosaic is created from 50 input patches of size 160×160. The difficulty is that the used patches are also over-exposed in large parts.

5.4.3 Over-exposure

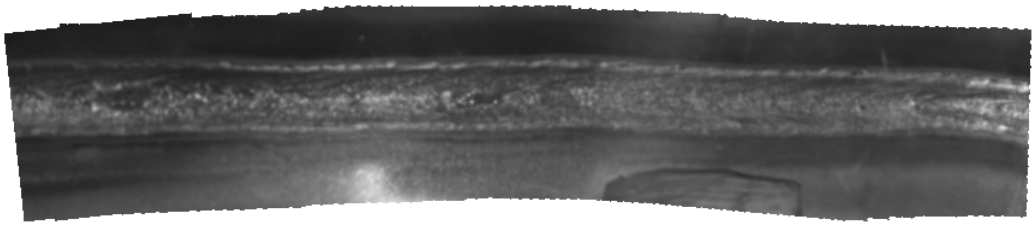
During the welding process it may happen that the illumination changes. In some cases the weld seam is partly or fully over-exposed. If the over-exposure does not concern the whole scene, an adaption of the blending weights due to exposure provides better results. Figure 5.15 compares stitching results with and without exposure compensation. The following parameters have been used for the calculation of the exposure fusion weights: $\sigma = 0.1$, $wc = 0.2$, $we = 1.0$, $g = 5$, $c = 0.3$ and window size is 5.



(a)

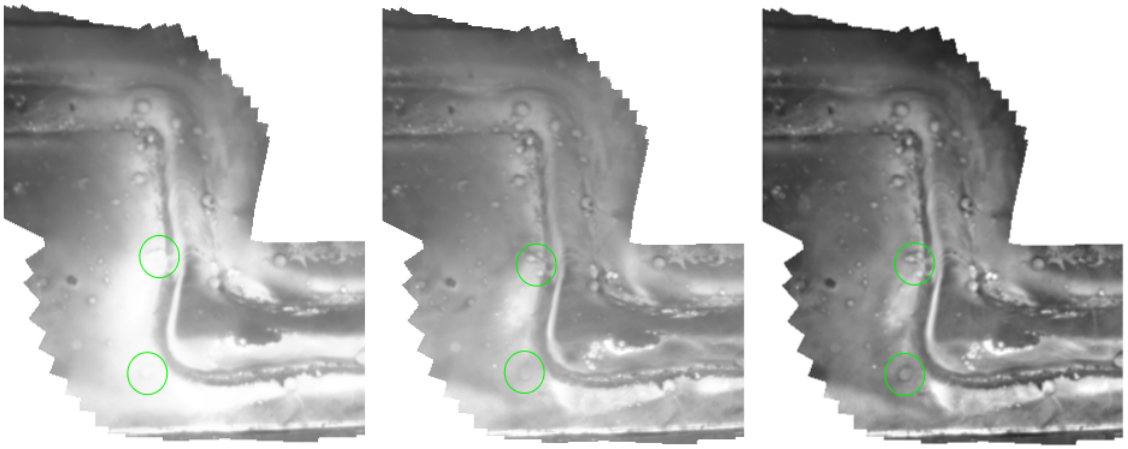


(b)



(c)

Figure 5.15: Exposure compensation. (a) Some weld seam images contain over-exposed regions. The figure shows a sample of the 40 input images. (b) With traditional blending, over-exposed regions are visible in the final mosaic. (c) By adapting the blending weights, it is compensated for over-exposure. The illumination in the final result is more consistent.



(a) (b) (c)

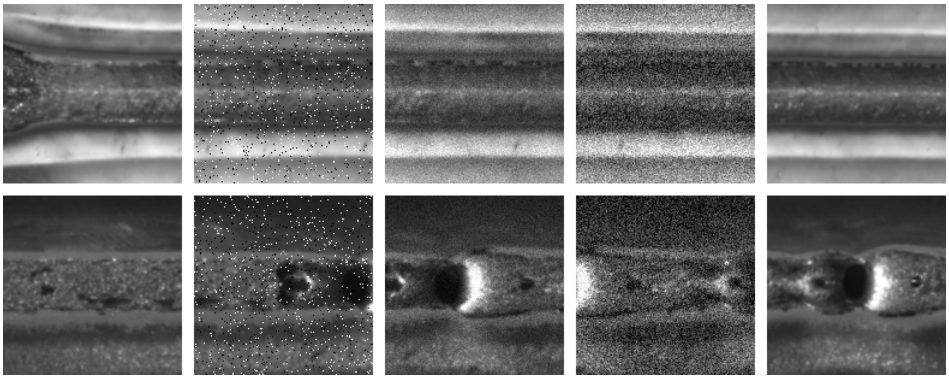
Figure 5.16: Exposure compensation and dehazing. Image mosaics generated with three different methods: (a) without exposure fusion, (b) with exposure fusion, and (c) with exposure fusion and dehazing. The dehazing preprocessing step and the addition of exposure fusion blending weights improve the stitching result. All three panoramas are generated with multi-band blending and incremental blending weights.

Figure 5.16 represents a further example for exposure fusion. Here, the result is further enhanced by adding a dehazing preprocessing step. The exposure weights are here calculated without contrast adjustment and block filter processing. The mosaic, where dehazing and exposure fusions is used, provides the best contrast and makes more structure.

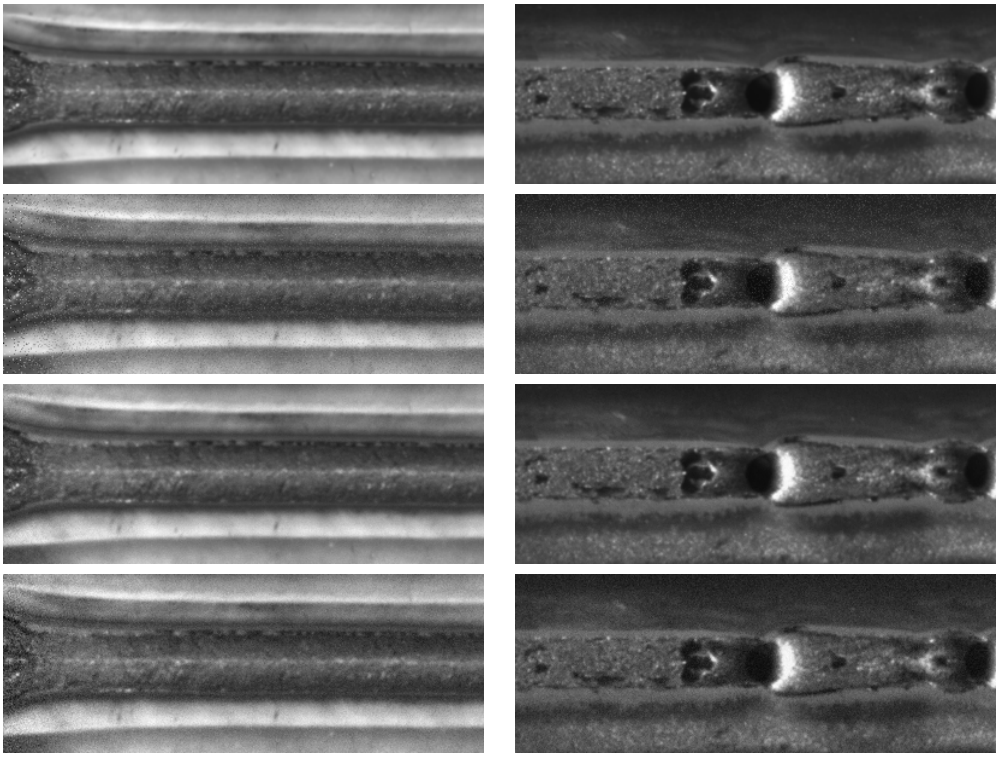
5.4.4 Noise

For noise evaluation, we have generated synthetic test data (see Section 5.2). For that purpose, two existing straight weld seam panoramas are chosen. Along the x-axis of a panorama, patches of size 149×149 are extracted. As we only have a translation into x direction, we can easily stitch a new panorama out of it. Three different kinds of noise are used in this case: a) salt & pepper, b) Poisson, and c) Gaussian noise.

Figure 5.17 shows the resulting weld seam mosaics generated from noisy patches, as well as the noise-free results. As one can see, the stitching results are quite well in spite of the noise. Here again incremental blending weights are used, but with 100 % averaging. Averaging weights reduce noise more efficient, if the noise remains constant on all patches.



(a) Input images



(b) Stitching results

Figure 5.17: Blending with noisy images. (a) Sample of the input images generated from two different panoramas: The first and the last column show noise-free patches. The patches from the second to the fourth column are provided with salt & pepper, poisson and gaussian noise, respectively. (b) Stitching results: no noise, salt & pepper, poisson and gaussian noise (from top to bottom).

5.4.5 Radiometric Evaluation

In this section we perform a radiometric evaluation of our blending results. Our applied evaluation metrics are introduced in Section 5.1. The test is performed for synthetic noisy test data as described in Section 5.2. We use four different test data sets, two straight and two curvy welding sequences. The region of a warped weld seam patch in the panorama is compared to the corresponding original patch without noise. Five different blending methods are compared: a) incremental blending, b) feathering, c) blending with binary weights, d) stitching in the gradient domain (with feathering weights), and e) median blending. In the methods a, b and c just different weight maps are used and the images are blended with multi-band blending.

The results for two straight panoramas are summarized in Figure 5.18. MSSIM is used as similarity measure, where each noise-free input patch is compared to its corresponding region in the final mosaic. As this region contains information of multiple overlapping patches, the noise from the input images should be reduced in the stitching result. Incremental blending seems to be the best method for Gaussian and Poisson noise. In the presence of salt & pepper noise it is obvious that median blending is the best alternative. The corresponding panoramas are shown in Figure 5.20 (a) and (b).

We have tested the blending methods also for curvy panoramas. 70 % of all patches (randomly chosen 7 out of every 10) are provided with noise. Again, five different blending strategies are compared. We have performed the incremental blending with and without exposure fusion. We have found out that the inclusion of exposure fusion weights here improve the results for incremental blending considerably. Figure 5.19 shows the results for two different synthetic test data sets. The corresponding panoramas are shown in Figure 5.20 (c) and (d). The evaluation results prove that the dynamic blending approach is an appropriate method. The following parameters has been used for the generation of the noisy input images: density of salt & pepper noise 5%, Gaussian noise $\mu = 0, \sigma = 0.01$.

A similarity evaluation for the whole mosaic compared to the ground truth mosaic is performed in addition to the patch-wise evaluation for all four test data sets. Table 5.6 summarizes the results. The similarity is evaluated by using three different measures: a) mean structural similarity (MSSIM), b) normalized cross-correlation (NCC), normalized sum of squared differences (NSSD). The straight panoramas are just compared with the ground truth panoramas. The curvy panoramas are registered to the ground truth panorama before evaluating our approaches. The best scores are achieved by the incremental blending method. Exposure fusion weights are a good strategy to further reduce noise from input images in the panorama.

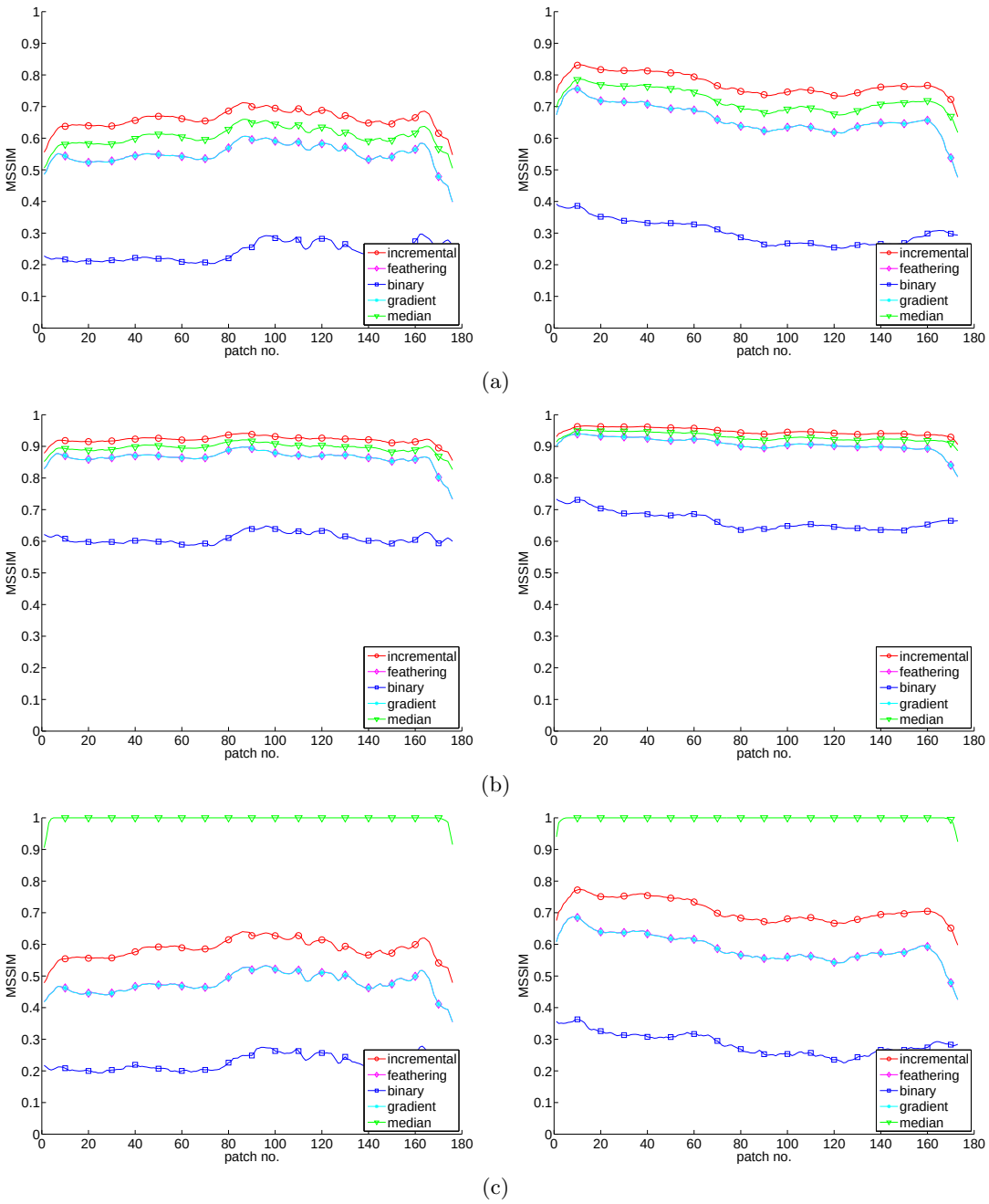


Figure 5.18: Comparison of blending methods with noisy input images of two straight welding sequences: (a) Gaussian, (b) Poisson, and (c) salt & pepper noise. In each case two different data sets are evaluated. The noise-free patch is compared with its corresponding region in the mosaic for evaluation. MSSIM is used as similarity measure. Incremental blending is the best method in two cases. By occurence of salt & pepper noise it is obvious that median blending is the best alternative.

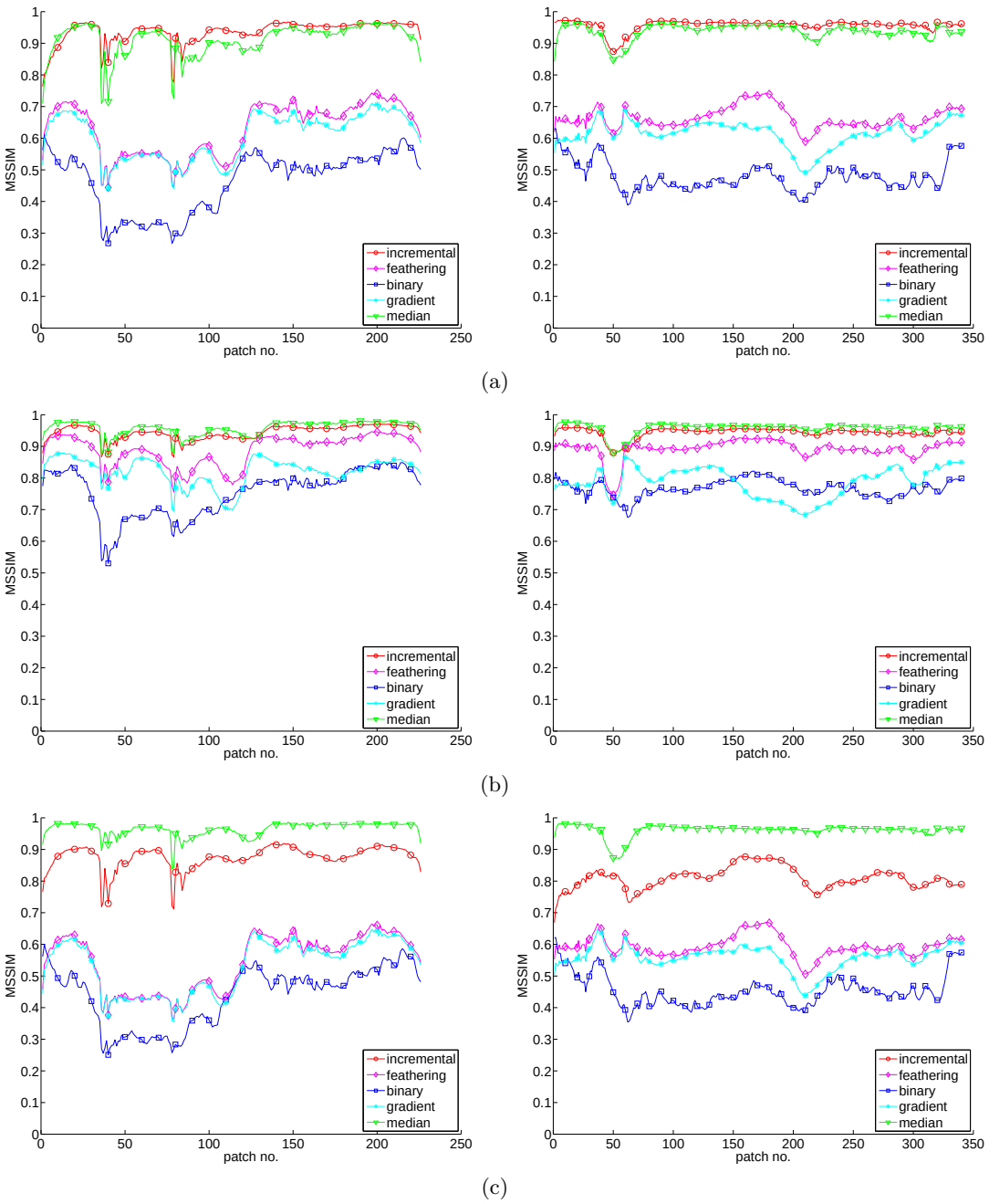
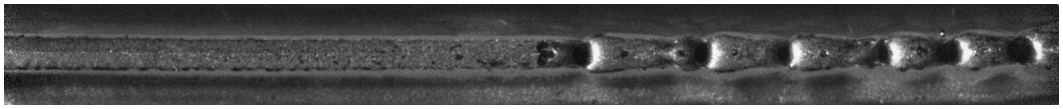
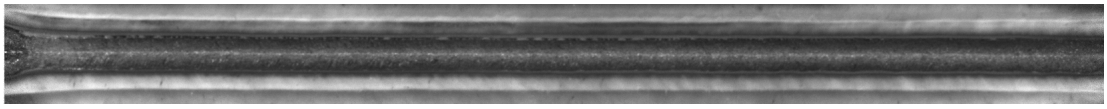


Figure 5.19: Comparison of blending methods with noisy input images of two curvy welding sequences: (a) Gaussian, (b) Poisson, and (c) salt & pepper noise. In each case two different data sets are evaluated. The noise-free patch is compared with its corresponding region in the mosaic for evaluation. MSSIM is used as similarity measure. To obtain better results, exposure fusion weights are added to incremental blending. This approach provides suitable results for all types of noise.



(a) straight1



(b) straight2

(c) curvy1

(d) curvy2

Figure 5.20: Panoramas from noisy input images: (a) all patches with Gaussian noise, (b) all patches with Poisson noise, (c) 70 % of patches with Gaussian noise, (d) 70 % of patches with Poisson noise. For all four panoramas incremental blending is used.

	Gaussian			Poisson			Salt & pepper		
	MSSIM	NCC	NSSD	MSSIM	NCC	NSSD	MSSIM	NCC	NSSD
straight1									
Incremental	0.6546	0.9717	0.0086	0.9156	0.9968	0.0010	0.5821	0.9527	0.0153
Feathering	0.5353	0.9433	0.0180	0.8535	0.9934	0.0020	0.4649	0.9080	0.0299
Binary	0.2340	0.7746	0.0902	0.5996	0.9644	0.0110	0.2194	0.6823	0.1404
Gradient	0.5350	0.9433	0.0180	0.8534	0.9933	0.0020	0.4644	0.9080	0.0300
Median	0.5926	0.9615	0.0123	0.8882	0.9956	0.0013	0.9841	0.9906	0.0029
straight2									
Incremental	0.7716	0.9861	0.0039	0.9459	0.9976	0.0007	0.7078	0.9782	0.0060
Feathering	0.6483	0.9722	0.0079	0.9002	0.9952	0.0013	0.5781	0.9565	0.0119
Binary	0.3025	0.8786	0.0397	0.6659	0.9759	0.0069	0.2837	0.8249	0.0569
Gradient	0.6481	0.9721	0.0079	0.9002	0.9952	0.0013	0.5776	0.9563	0.0120
Median	0.7095	0.9807	0.0055	0.9260	0.9967	0.0009	0.9885	0.9960	0.0011
curvy1									
Incremental	0.8727	0.9958	0.0024	0.8855	0.9963	0.0021	0.8471	0.9956	0.0026
Feathering	0.7795	0.9956	0.0025	0.8630	0.9959	0.0023	0.7515	0.9953	0.0028
Binary	0.7228	0.9942	0.0033	0.8123	0.9952	0.0028	0.7145	0.9930	0.0041
Gradient	0.7693	0.9948	0.0034	0.8363	0.9950	0.0034	0.7435	0.9942	0.0037
Median	0.8585	0.9956	0.0026	0.8809	0.9959	0.0023	0.8750	0.9955	0.0026
curvy2									
Incremental	0.9874	0.9997	0.0001	0.9872	0.9997	0.0001	0.9672	0.9996	0.0002
Feathering	0.9471	0.9995	0.0002	0.9828	0.9997	0.0001	0.9285	0.9993	0.0004
Binary	0.8907	0.9984	0.0007	0.9662	0.9996	0.0002	0.8679	0.9973	0.0012
Gradient	0.9162	0.9975	0.0024	0.9416	0.9978	0.0023	0.9042	0.9972	0.0023
Median	0.9851	0.9997	0.0001	0.9882	0.9997	0.0001	0.9874	0.9997	0.0001

Table 5.6: Blending evaluation with noisy input images. Image mosaics are compared to the noise-free ground truth mosaics.

5.5 Summary (Overall System)

In the previous sections, we have performed different experiments to evaluate registration and blending results of weld seam images. The registration repeatability evaluations demonstrated that the registration with SIFT and SIFTfast is most suitable. This registration methods achieved the best repeatability performance as well as the best matching performance. Hence, with an appropriate quality of the input images a successful registration can also be guaranteed without using tracking data. Nevertheless, a successful registration is guaranteed with a robust rigid motion estimation incorporating prior tracking data.

In terms of computation time, it is recommended to use the SIFTfast implementation which outperforms the registration with SURF features. By using global refinement the repeatability of registration results is strongly improved. This does not really have an impact on the radiometric quality of the final mosaic. The evaluation of registration accuracy for synthetic noisy test data also indicates that the usage of SIFT or SIFTfast

features for registration is recommandable. A radiometric evaluation on the same test data confirms this results.

The blending results show that incremental blending combined with exposure fusion is a good method to create dynamic weld seam panoramas. Median blending is no alternative in terms of run time, because incremental blending is multiple times faster. Challenges like bright sparks or single wads of smoke are well accomplished with our proposed blending strategy. The biggest problems occur if over- or under-exposed areas on weld seam patches are too large and if heavy smoke covers nearly the whole patch.

An advantage of our approach is that it can be adapted according to the application. Over- or under-exposed regions in input images can be reduced by using exposure fusion weights. A further advantage of the incremental blending method is that a pixel in the final mosaic is generated based on the information of multiple input images. By contrast, binary blending weights only assign the information from one input image. We recommend to use incremental blending with adaptions depending on the welding task and characteristics of the weld seam images, respectively. If there are over-exposed input images, exposure fusion weights should be included accordingly. In general, incremental blending guarantees a smooth panoramic image with no visible seams. It benefits most from the large overlap of consecutive input images.

Finally, we want to give a concluding overview of the overall performance of our stitching system. For that purpose, we have used 14 data sets, where each data set contains 19 sequences of a repeated welding task. The data sets are acquired under different conditions and challenges like illumination changes, camera rotation, bright sparks, heavy smoke, fast robot motion. In total, 266 panoramas are generated by using our approach described in Chapter 4. This includes the following workflow:

1. Image registration based on SIFTfast features and robust rigid motion estimation including prior data

2. Global refinement of registration results

3. Incremental image blending including exposure fusion

All panoramas are successfully generated. The registration results for these 14 data sets are summarized in Table 5.1 in Section 5.3.1. Based on the successful registrations the global refinement step also works well. The results of incremental blending allows a good visual quality inspection of the depicted weld seam. The major challenge is heavy smoke. By incorporating prior data, a successful generation of the weld seam mosaic is guaranteed as well, but the final stitching result is unsufficient in some cases. If weld seam patches contain heavy smoke, the setting of the image stitching system has to be adapted due to the challenge. A dehazing pre-processing and an adjustment of the exposure fusion contrast parameters might improve the stitching result. A further issue in the case of bad conditions is that the region in the frame where the structure of the weld is well depicted

is very small. The position of the patch extraction as well as the patch size is in such cases important to provide favorable results.

Another issue is the real-time capabability of our system. We have performed all experiments on a Intel Core i5 540M with 4GB RAM. The image stitching prototype has been implemented in Matlab. We tested the time performance on four different data sets, where a sequence has about 225 input patches. We set the patch size to 120×120. The run time for 'online' registration and blending was about 6 frames per second for each data set. Due to a typical speed gain of $\sim 1:10$ between Matlab and C applications, and as further speed-up might be gained by bringing our implementation onto the GPU, our system is real-time capable, meaning that we can process X frames per second.

Chapter 6

Conclusion

We have presented an image stitching system for incremental weld seam mosaic generation. A welding process is performed by an autonomous robot with a rigidly mounted camera. The input to our stitching system is a sequence of rectified weld seam patches that are extracted by a tracking algorithm on each frame. There are two main stages of the stitching system: a) image registration and b) image blending.

The image registration is based on salient feature extraction and matching. A RANSAC routine is used to get a consistent set of matches and to provide a robust rigid motion estimation. In order to guarantee a successful panoramic image generation, we incorporate prior tracking data into the rigid motion estimation if necessary. A global refinement step is used to adjust the image mosaic to a required end position in the mosaic coordinate frame.

The incremental blending method creates a 'growing' panorama where multi-band blending is used to reduce visible seams and artifacts. In order to reduce over- or under-exposure, the blending weights are adapted with an exposure fusion approach. If weld seam patches contain heavy smoke, the final result is further improved by image dehazing. Our incremental blending approach emerged to be robust against bright sparks or wads of smoke. An advantage of our incremental blending approach is also that it can be used 'online' for an incremental panorama generation as well as 'offline' for an overall panorama generation if all input images are already registered.

In our experiments we have evaluated registration and blending results of weld seam images. We recommend to use SIFT or SIFTfast features for the image registration step. In terms of computation time, SIFTfast is highly recommended as similar results to SIFT are obtained. By using global refinement the repeatability of registration results is drastically improved. The evaluation of blending results showed that our incremental blending approach is well suited for creation of dynamic weld seam mosaics. An advantage of the incremental blending method is that a pixel in the final mosaic is generated by means of the information of multiple overlapping input images. Challenges like bright sparks or single wads of smoke are well accomplished.

Adding exposure fusion weights further enhances the blending results, especially if weld seam patches contain partial over- or under-exposure. If input images have different exposures, e.g. several are over-exposed and several are well-exposed, exposure fusion is a good posibility to improve the final blending result. In order to overcome heavy smoke, a dehazing method is introduced. The usage for weld seam images is limited. Dehazing makes only sense in the case of global haze on the input images. In a bad case, the final panorama can get too dark with dehazing.

By incorporating prior tracking data, our system was able to generate mosaics from 100% of all evaluated sequences. The lowest performance on pairwise registration is achieved in the case of heavy smoke, where the usage of prior data is essential. For a good blending result of the weld seam, heavy smoke was also the biggest challenge.

Future work will address the implementation of faster registration and blending methods to improve real-time capability. The system should work in real by now, depending on the size of the weld seam patches. As there is usually a large overlap between patches, it could be also adequate to register and blend only every second or third patch of a sequence in order to decrease the computation time. Another idea for weld seam quality inspection based on our system would be to include an automatic error detection and localization. The system could then mark possible errors in the computed mosaic.

A contrary approach to blending is image fusion [38]. Image fusion uses different images of the same scene and tries to get the best information out of it. This would be an optional approach to generate a weld seam survey image. Image fusion can be applied for overlapping regions of the weld seam images. But this method has also a disadvantage. As structures are refined by image fusion, this strategy is only applicable for weld seam images without noisy elements.

Another enhancement would be to use a more adaptive weighting function for image blending, in order to eliminate noisy artifacts more efficiently and to enhance the structure of the weld seam at the same time. A fast blending strategy that combines all advantages of multi-band blending, incremental blending, dehazing, exposure fusion and image fusion would be ideal. Another possibility is to adapt our system to more general applications where sequences of images or even unordered images should be stitched together.

Bibliography

[1] Agarwala, A., Agrawala, M., Cohen, M., Salesin, D., and Szeliski, R. (2006). Photographing long scenes with multi-viewpoint panoramas. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 853–861, New York, NY, USA. ACM.

[2] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. (2004). Interactive digital photomontage. *ACM Transactions on Graphics*, 23(3):292–300.

[3] Alsuwaiyel, M. H. and Gavrilova, M. (2000). On the distance transform of binary images. In *CISST2000: Proceedings of the International Conference on Imaging Science, Systems and Technology*, pages 83–86.

[4] Azzari, P., Di Stefano, L., and Mattoccia, S. (2008). An evaluation methodology for image mosaicing algorithms. In Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., and Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems*, volume 5259 of *Lecture Notes in Computer Science*, pages 89–100. Springer Berlin / Heidelberg.

[5] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision - ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin / Heidelberg.

[6] Beis, J. and Lowe, D. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1000 –1006.

[7] Bouguet, J.-Y. (2000). Pyramidal implementation of the lucas-kanade feature tracker: description of the algorithm. OpenCV Document.

[8] Boutellier, J., Silvén, O., Tico, M., and Korhonen, L. (2009). Objective evaluation of image mosaics. In Braz, J., Ranchordas, A., Araújo, H. J., and Pereira, J. M., editors, *Computer Vision and Computer Graphics. Theory and Applications*, volume 21 of *Communications in Computer and Information Science*, pages 107–117. Springer Berlin Heidelberg.

[9] Brown, M. and Lowe, D. G. (2003). Recognising panoramas. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1218–1225 vol.2.

[10] Brown, M. and Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73.

[11] Burt, P. and Adelson, E. (1983a). The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540.

[12] Burt, P. J. and Adelson, E. H. (1983b). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236.

[13] Cervantes, A. and Kang, E. Y. (2006). Progressive multi-image registration based on feature tracking. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas*, pages 633–639.

[14] Davis, J. (1998). Mosaics of scenes with moving objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 354–360.

[15] Dufaux, F. and Moscheni, F. (1996). Background mosaicking for low bit rate video coding. In *Proceedings of the International Conference on Image Processing*.

[16] Eggert, D. W., Lorusso, A., and Fisher, R. B. (1997). Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5):272–290.

[17] Elibol, A., Garcia, R., Delaunoy, O., and Gracias, N. (2008). A new global alignment method for feature based image mosaicing. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y., Rhyne, T.-M., and Monroe, L., editors, *Advances in Visual Computing*, volume 5359 of *Lecture Notes in Computer Science*, pages 257–266. Springer Berlin / Heidelberg.

[18] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[19] Gower, J. (1975). Generalized procrustes analysis. *Psychometrika*, 40(1).

[20] Gracias, N., Mahoor, M., Negahdaripour, S., and Gleason, A. (2009). Fast image blending using watersheds and graph cuts. *Image and Vision Computing*, 27(5):597 – 607.

[21] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[22] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA.

[23] He, K., Sun, J., and Tang, X. (2009). Single image haze removal using dark channel prior. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1956 –1963.

[24] He, K., Sun, J., and Tang, X. (2010). Guided image filtering. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Computer Vision - ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin / Heidelberg.

[25] Isgrò, F. and Pilu, M. (2004). A fast and robust image registration method based on an early consensus paradigm. *Pattern Recognition Letters*, 25(8):943–954.

[26] Jia, J. and Tang, C.-K. (2008). Image stitching using structure deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):617–631.

[27] Kovesi, P. (2010). Matlab and octave functions for computer vision and image processing. `http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/`. Last checked on November 25, 2010.

[28] Levin, A., Lischinski, D., and Weiss, Y. (2006). A closed form solution to natural image matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 61 – 68.

[29] Levin, A., Zomet, A., Peleg, S., and Weiss, Y. (2004). Seamless image stitching in the gradient domain. In *Eighth European Conference on Computer Vision (ECCV 2004)*, volume IV, pages 377–389.

[30] Lewis, J. P. (1995). Fast normalized cross-correlation. `http://scribblethink.org/Work/nvisionInterface/nip.pdf`. Last checked on February 23, 2011.

[31] Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):225–270.

[32] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[33] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 674–679.

[34] Mertens, T., Kautz, J., and Van Reeth, F. (2007). Exposure fusion. In *15th Pacific Conference on Computer Graphics and Applications*, pages 382 –390.

[35] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615 –1630.

[36] Mills, A. and Dudek, G. (2009). Image stitching with dynamic elements. *Image and Vision Computing*, 27(10):1593 – 1602.

[37] Paalanen, P., Kämäräinen, J.-K., and Kälviäinen, H. (2009). Image based quantitative mosaic evaluation with artificial video. In Salberg, A.-B., Hardeberg, J., and Jenssen, R., editors, *Image Analysis*, volume 5575 of *Lecture Notes in Computer Science*, pages 470–479. Springer Berlin / Heidelberg.

[38] Pajares, G. and de la Cruz, J. M. (2004). A wavelet-based image fusion tutorial. *Pattern Recognition*, 37(9):1855 – 1872.

[39] Rebiere, N., Auclair-Fortier, M.-F., and Deschenes, F. (2008). Image mosaicing using local optical flow registration. In *19th International Conference on Pattern Recognition*, pages 1–5.

[40] Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1508 –1515 Vol. 2.

[41] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision - ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. Springer Berlin / Heidelberg.

[42] Sawhney, H. and Kumar, R. (1999). True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):235–243.

[43] Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593 –600.

[44] Shum, H.-Y. and Szeliski, R. (2000). Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130.

[45] Sonka, M., Hlavac, V., and Boyle, R. (2007). *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering.

[46] Strasdat, H., Montiel, J. M. M., and Davison, A. J. (2010). Scale drift-aware large scale monocular slam. In *Proceedings of Robotics: Science and Systems Conference*.

[47] Szeliski, R. (1996). Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30.

[48] Szeliski, R. (2006). Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104.

[49] Szeliski, R. and Shum, H.-Y. (1997). Creating full view panoramic image mosaics and environment maps. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258.

[50] Uyttendaele, M., Eden, A., and Skeliski, R. (2001). Eliminating ghosting and exposure artifacts in image mosaics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–509–II–516 vol.2.

[51] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 –612.

[52] Xiong, Y. and Turkowski, K. (1998). Registration, calibration and blending in creating high quality panoramas. In *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision*, pages 69–74.

[53] Zagrouba, E., Barhoumi, W., and Amri, S. (2009). An efficient image-mosaicing method based on multifeature matching. *Machine Vision and Applications*, 20(3):139–162.

[54] Zhao, W. (2006). Flexible image blending for image mosaicing with reduced artifacts. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 20(4):609–628.

[55] Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In *Graphics gems IV*, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA.