THE UNIVERSITY *of* ADELAIDE

Doctoral Thesis

Moving Least Squares Registration in Computer Vision: New Applications and Algorithms

Author:
William Liu

Supervisors:
Dr. Tat-Jun CHIN
Assoc. Prof. Gustavo CARNEIRO
Prof. David SUTER

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Faculty of Engineering, Computer and Mathematical Sciences
School of Computer Science

July 2017

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed: ____

Date: ____

THE UNIVERSITY OF ADELAIDE

Abstract

School of Computer Science

Doctor of Philosophy

Moving Least Squares Registration in Computer Vision:
New Applications and Algorithms

by William Liu

Registration is a fundamental task in computer vision, and is often used as a preliminary step in diverse applications. In the process of registration, the transformation model needs to be estimated to establish the correspondence relationships between input images. Most transformation models are built upon certain assumptions. However, in practice, when given uncharacteristic data, applying such a model may result in critical deviations/artifacts in the registration output. The research conducted in this thesis focuses on the step of transformation model estimation in registration problems, where the underlying model assumptions do not hold. A central theme of this thesis is the usage of moving least squares (MLS) technique to handle violations to model assumptions. This thesis contributes in three specific applications: radial distortion estimation, image stitching and video stabilization.

First, real cameras approximate ideal pinhole cameras using lenses and apertures. This leads to radial distortion effects that are not characterizable by the standard epipolar geometry model and impacts the efficacy of point correspondence validation based on the epipolar constraint. Many previous works deal with radial distortion by augmenting the epipolar geometry model with additional parameters such as distortion coefficients and centre of distortion. In this thesis, radial distortion is treated as a violation to the basic epipolar geometry. To account for the distortion effects, the epipolar geometry is adjusted via the MLS approximation combined with M-estimators to allow robust matching of interest points under severe radial distortion. Compared to previous works, the proposed method is much simpler and exhibits a higher tolerance in cases where the exact model of radial distortion is unknown.

Secondly, spatially varying warps are increasingly popular for image alignment as alternatives to homographic warps, since the basic homography model carries the assumptions that images were taken under pure rotational motions, or that the scene is sufficiently far away such that it is effectively planar – conditions unlikely to be satisfied in casual photography. However, estimating spatially varying warps requires a sufficient number of feature matches. In image regions where feature detection or matching fail, the warp loses guidance and is unable to accurately model the true underlying warp, thus resulting in poor registration. This thesis proposes a correspondence insertion method for As-Projective-As-Possible (APAP) warps, which are extensions of MLS to the projective setting. The proposed method automatically identifies misaligned regions, and inserts appropriate point correspondences to increase the flexibility of the warp and improve alignment. Unlike other warp varieties, the underlying projective regularization of APAP warps reduces overfitting and geometric distortion, despite increases to the warp complexity.

Lastly, video stabilization is achieved by estimating the camera trajectory throughout the video and then smoothing the trajectory. In practice, most approaches directly model and filter the camera motion using 2D image transforms (e.g., affine or projective). From the smoothed motions, update transforms are obtained to adjust each frame of the video such that the overall sequence appears to be stabilized. However, the update transform is also customarily defined by the basic 2D transforms, which cannot preserve the image contents well. As a result the stabilized videos often appear distorted and "wobbly". Therefore, estimating good update transforms is more critical to success than accurately modeling and characterizing the motion of the camera. Based on this observation, this thesis proposes homography fields for video stabilization. A homography field is a spatially varying warp that is regularized to be as projective as possible, so as to enable accurate warping while adhering closely to the underlying geometric constraints. It has been shown that homography fields are powerful enough to meet the various warping needs of video stabilization, not just in the core step of stabilization, but also in video inpainting. This enables relatively simple algorithms to be used for motion modeling and smoothing. Results on various publicly available testing videos demonstrate the merits of the proposed video stabilization pipeline.

Acknowledgements

I thank my supervisor, Dr. Tat-Jun Chin, for all his advices, comments, and critical revisions throughout this thesis and articles I published during my PhD. I really appreciate his interest in this research. Working with him has certainly been an enriching life experience. I would also like to thank my co-supervisors Assoc. Prof. Gustavo Carneiro and Prof. David Suter for their advices, revisions and comments during research meetings. I extend my thanks to Dr. Anders Eriksson for his valuable comments on works that are part of this thesis.

I would like to express my sincere appreciation to my family for their supports and encouragements.

During my PhD candidature, I have shared incredible time with incredible people. I would like to extend my thanks to Dr. Quoc Huy Tran, Dr. Trung Thanh Pham, and Dr. Julio Zaragoza for their general advices and discussions.

Lastly I would like to thank the University of Adelaide for funding my PhD.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

APAP	**A**s-**P**rojective-**A**s-**P**ossible
DoF	**D**egrees **o**f **F**reedom
LS	**L**east **S**quares
MDLT	**M**oving **D**irect **L**inear **T**ransformation
MLS	**M**oving **L**east **S**quares
RANSAC	**RAN**dom **SA**mple **C**onsensus
SVD	**S**ingular **V**alue **D**ecomposition
TLS	**T**otal **L**east **S**quares
WLS	**W**eighted **L**east **S**quares

Publications

This thesis is in part the result of the work presented in the following papers:

- William X. Liu, Tat-Jun Chin, Gustavo Carneiro and David Suter,
  "Point Correspondence Validation under Unknown Radial Distortion",
  International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2013.
  (DOI:10.1109/DICTA.2013.6691513)

- William X. Liu, Tat-Jun Chin, Anders Eriksson and Michael S. Brown
  "Correspondence Insertion for As-Projective-As-Possible Image Stitching",
  Submitted to arXiv as arXiv:1608.07997

- William X. Liu and Tat-Jun Chin,
  "Smooth Globally Warp Locally: Video Stabilization Using Homography Fields",
  International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2015.
  Awarded DSTO Best Fundamental Contribution to Image Processing Paper Prize.
  (DOI: 10.1109/DICTA.2015.7371309)

Chapter 1

Introduction

1.1 Introduction

Registration is a fundamental task in computer vision, and is often used as a preliminary step in diverse applications that require (a) integrating information taken from different sensors, (b) observing changes in images taken at different times or under different conditions, (c) extracting 3D information from images, and (d) identifying objects in images to estimate their poses (location and orientation) [111].

In the most basic form of registration, the task is to align two or more images of the same scene taken from different viewpoints, and/or by different devices. Although many methods [24, 16, 21, 113, 6, 29, 15, 143, 114, 135] have been proposed on this topic, the majority of image alignment methods consist of the following steps as shown in Figure 1.1. First, in the step of feature detection, as shown in the top row in Figure 1.1, salient and distinctive objects are detected. Many computer vision applications employ feature detection as the initial step, thus, a very large number of feature detectors have been developed. Feature detectors examine every pixel to determine whether there is an image feature (e.g., corner, interest points, edges), and at last output feature point representatives, i.e., feature point localizations and feature descriptors. Then, the correspondence relationships between features from input images can be established with various feature descriptors and similarity metrics. Given established correspondences, the mapping is estimated. There are several transformation models that can be used to characterize the mapping, but they can be roughly categorized as global mapping

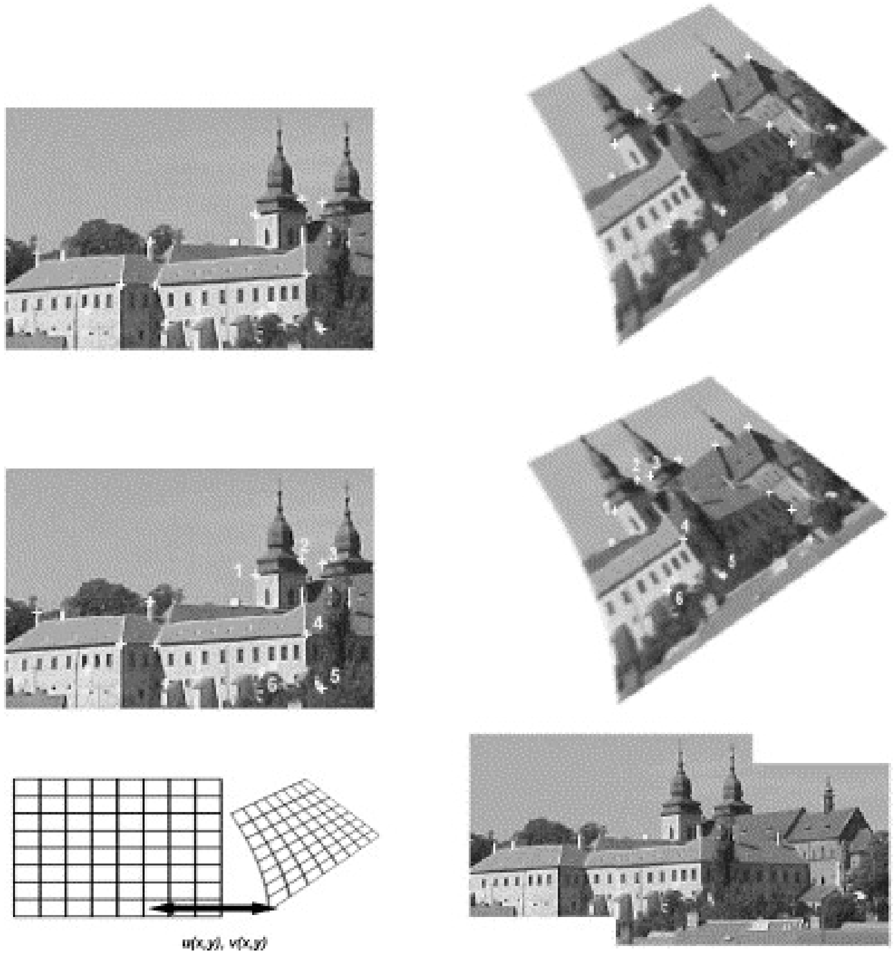


Figure 1.1: Image alignment consists of four steps: feature detection (as shown in the top row), feature matching (as shown in the middle row), transformation model estimation and image resampling and transformation. Figure is taken from [143].

transformations and local mapping transformations (more below). Finally, images are optionally warped onto the same canvas with estimated transformation. The warping output is shown in bottom row in Figure 1.1.

Apart from image alignment, there are other registration problems, such as estimating the epipolar geometry. Similar to image alignment, the epipolar geometry estimation begins with the search for corresponding feature points in stereo images. Then the transformation model – the fundamental matrix, which is the algebraic representation of epipolar geometry – is obtained. With the help of fundamental matrix, the poses of a calibrated camera from stereo images can be determined, which are crucial to the reconstruction of 3D structure. In all registration problems, inevitably the transformation model requires to be estimated to establish the correspondence relationships between input images, which is where the focus of this thesis lies.

When applying the global mapping transformation, after choosing a certain global model, e.g., rigid transformation or affine transformation, all feature correspondences are used for estimating the set of model parameters valid for the entire image. In general, the number of correspondences is usually higher than the minimum number required for the estimation of the mapping model. The mapping model is then computed by least square methods, so that the sum of squared residuals at correspondences is minimized. However, such global model mapping does not map all correspondences onto their correspondences exactly, and cannot properly handle images deformed locally. More fundamentally, the usage of a particular model is based on certain assumptions about the underlying data which may not hold in practice. However, when the underlying model assumptions do not hold, applying such a model may result in critical deviations/artifacts in the registration output. In this thesis, a central theme is to deal with violations to model assumptions with the usage of moving least squares (MLS) technique.

The MLS method was proposed by Shepard [118] for smoothing and interpolating data. Shepard applied MLS to generate two-dimensional interpolants from irregularly-spaced data to produce a continuous surface, which was later extended by Lancaster and Salkaukas in [72, 71]. More recently, the MLS approximation was used in the reconstruction of two and three-dimensional curves from unorganized and noisy data [73] and in the reconstruction of surfaces [3, 74]. In [3], data in the form of raw points, acquired with a range scanner and therefore contains errors, were replaced by surfaces derived from MLS technique. This was achieved by down-sampling (i.e., iteratively removing points which have little contribution to the shape of the surface) or up-sampling (i.e., adding points and projecting them to the MLS surface where point-density is low). The projection procedure was later augmented and further analyzed in the work of Amenta and Kil [5]. The MLS method has also been successfully applied to simulating and animating elastoplastic materials [14, 104], partition of unity implicits [108], and image deformation [115]. In [115], an image deformation method was proposed based on linear MLS. To construct deformations that minimize the amount of local scaling and shear, Schaefer *et al.* [115] restricted the classes of transformations used in MLS to similarity and affine transformations. Following this, Zaragoza *et al.* [140] proposed moving direct linear transformation (MDLT), which is conceptually the homogeneous version of MLS and based on projective warps, to produce as-projective-as-possible (APAP) image warps.

1.2 Overview of Thesis

This thesis provides mathematical adjustments, inspired by the work of APAP image stitching [140], to solve three challenging problems: (a) correspondence validation under unknown radial distortion, (b) correspondence insertion for spatially varying warps, and (c) video stabilization.

First, radial distortion is viewed as a violation to the basic epipolar geometry equation. Instead of augmenting the standard epipolar constraint with a radial distortion model, the proposed method adjusts the epipolar geometry as warranted by the data to account for the distortion effects. The model adjustment is achieved through the framework of MLS. Specifically, MLS is extended to allow for epipolar geometry estimation, and is combined with M-estimators [112] to enable robust point matching under severe radial distortion. Compared to previous methods, the proposed technique is much simpler and involves just solving linear subproblems. It also exhibits a higher degree of flexibility and generality, especially when the true model of radial distortion is unknown.

Second, this thesis attempts to insert correspondences for APAP warps, with a focus on panoramic stitching. In correspondence-poor regions, the proposed method automatically identifies misaligned regions, and inserts appropriate point correspondences to increase the flexibility of the warp and improve alignment. It has been shown how correspondence search can be accomplished for MDLT. On panoramic mosaicing problems that are challenging, the proposed approach has also exhibited the ability to achieve accurate alignment without being handicapped by insufficient feature matches.

Finally, homography fields are proposed for video stabilization. Conceptually, video stabilization is achieved by estimating the camera trajectory throughout the video and then smoothing the trajectory. In practice, the pipeline invariably leads to estimating update transforms that adjust each frame of the video such that the overall sequence appears to be stabilized. Contrary to the recent works, this thesis argues that the key to effective video stabilization is in designing better update transforms rather than accurately modeling and characterizing the motion of the camera. To capture the camera motion, it is sufficient to estimate the frame global image motion – following [98, 51], simple 2D homographies are used in the novel approach. To construct the all-important update transform, the homography fields, which are spatially varying warps that are regularized to be as projective as possible [140], has been used. This enables flexible and accurate warping that adheres closely to the underlying scene geometry. The obtained update transform is powerful enough to eliminate unwanted jerky motions, while at the same time prevents the warped sequence of frames from appearing wobbly or distorted. Apart from removing shakiness, homography fields are powerful enough to fill in blank regions arising from video re-rendering. The work conducted in this thesis is one of the first to treat trajectory smoothing and video inpainting in a single unified pipeline.

1.3 Background on Moving Least Squares

The simple problem of line fitting in 1D will be used to explain the concept of MLS method. The idea of least squares method will be given first to inspire the theory of MLS technique.

1.3.1 Least squares

Assume there exists a linear relationship between two variables a and b such that

$$b = xa. \tag{1.1}$$

Given a set of observations $\{a_i, b_i\}$ as shown by green circles in Figure 1.2, the goal is to find the linear model that matches the observations as best as possible. It is reasonable to require the discrepancy, or residual, between the predictions by the fitted model and the data to be as small as possible. The least squares criterion sums the squared residual at each observation, i.e.,

$$e = \sum_{i=1}^{N} (b_i - xa_i)^2, \tag{1.2}$$

where N is the number of observations and $(b_i - xa_i)$ is the residual for the i-th observation. Thus the problem becomes finding the x that minimizes the sum of squared residuals,

$$x = \operatorname*{argmin}_{x} \sum_{i=1}^{N} (b_i - xa_i)^2. \tag{1.3}$$

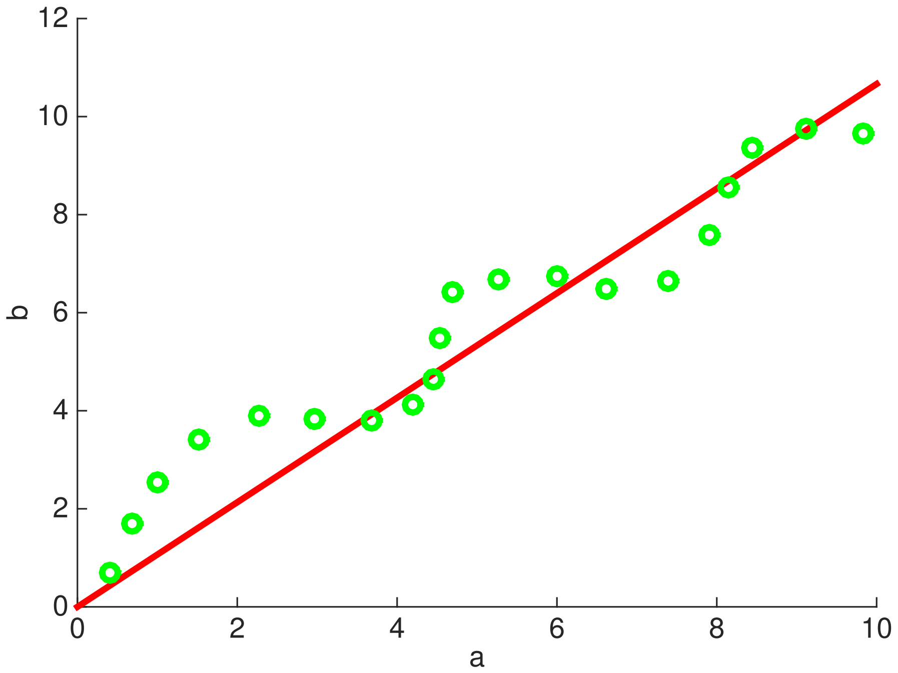


FIGURE 1.2: The concept of line fitting. Data points are plotted in green on the $a - b$ plane. The fitted line with least squares is marked in red.

Using the notations

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \textit{ and } \mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}, \tag{1.4}$$

the error function can be rewritten as:

$$\mathbf{E} = (\mathbf{b} - \mathbf{A}x)^T(\mathbf{b} - \mathbf{A}x) = \mathbf{b}^T\mathbf{b} - 2(\mathbf{A}x)^T b + (\mathbf{A}x)^T(\mathbf{A}x). \tag{1.5}$$

To solve for x, the following can be obtained

$$\frac{\partial \mathbf{E}}{\partial x} = 2\mathbf{A}^T\mathbf{A}x - 2\mathbf{A}^T\mathbf{b} = 0. \tag{1.6}$$

The least squares solution is simply

$$x_{LS} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}, \tag{1.7}$$

and the fitted line is shown in red in Figure 1.2.

1.3.2 Moving least squares

The standard least squares technique solves a large linear system and produces a global solution. Different from this global fitting strategy, MLS allow the fit to change locally depending on where the function is evaluated. To achieve this property, the solution for each point in the problem domain is solved for using weighted least squares approximation.

In the weighted least squares estimation, as in regular least squares, the unknown x is estimated by minimizing the sum of the squared residuals. Unlike least squares, however, each term in the weighted least squares approximation contains an additional weight, w_*, that determines how much each observation in the data set influences the final parameter estimation. The problem of weighted least squares for arbitrary point a_* can be formulated as following

$$x_* = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{N} w_*(b_i - xa_i)^2, \tag{1.8}$$

where $w_*(\cdot)$ is a distance weighting function. By selecting an appropriate spatially varying weighting function, a variety of interpolating or approximating behaviors can be achieved. Here, the Gaussian function is used, not only for its broad usage in practice, but also because it has been used in MDLT [140] and in this thesis. Thus, the scalar weights $\{w_*^i\}_{i=1}^N$ change according to a_* and are calculated as

$$w_* = e^{-(a_* - a_i)^2/\sigma^2}, \tag{1.9}$$

where σ is a scale parameter. Let the weight matrix $\mathbf{W}_* \in \mathbb{R}^{N\times N}$ be a diagonal matrix containing all weights:

$$\mathbf{W}_* = \begin{bmatrix} w_*^1 & 0 & \dots & c \\ 0 & w_*^2 & \dots & c \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_*^n \end{bmatrix}. \tag{1.10}$$

Thus, Equation (1.8) can be written in the matrix form

$$x_* = \underset{x}{\operatorname{argmin}} \, (\mathbf{W}_*(\mathbf{b} - \mathbf{A}x))^T(\mathbf{W}_*(\mathbf{b} - \mathbf{A}x)), \tag{1.11}$$

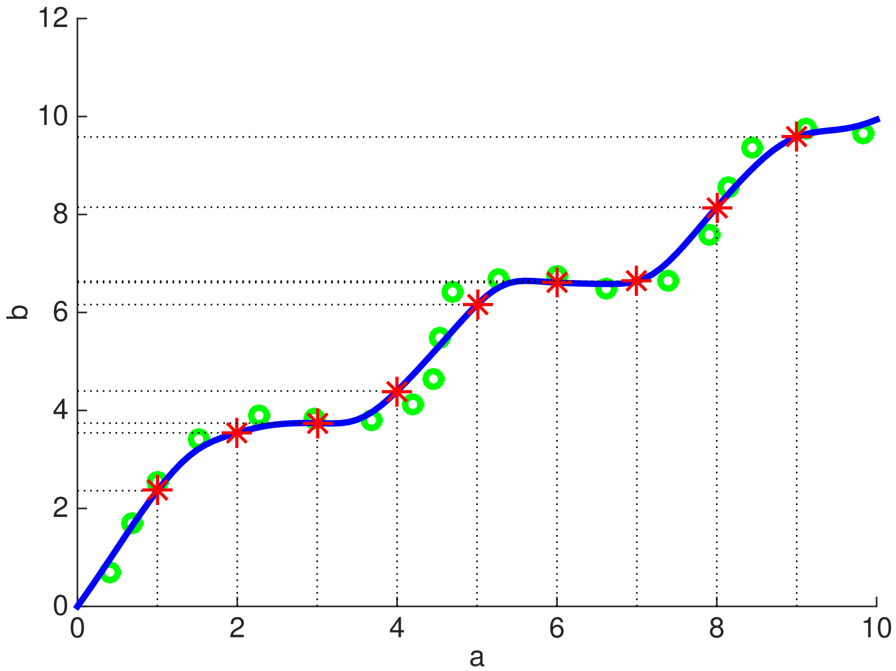


Figure 1.3: For each point a_* on a-axis, the corresponding x_* is estimated using weighted least squares technique. Estimated points (in red) can be calculated by $b_* = x_* a_*$. By using all estimated points, the MLS curve is obtained as shown in blue.

from which the following equation can be derived

$$\mathbf{A}^T \mathbf{W}_*^T \mathbf{W}_* \mathbf{A} x = \mathbf{A}^T \mathbf{W}_*^T \mathbf{W}_* \mathbf{b}. \tag{1.12}$$

As a result, for each point, the solution is

$$x_* = (\mathbf{A}^T \mathbf{W}_*^T \mathbf{W}_* \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}_*^T \mathbf{W}_* \mathbf{b}. \tag{1.13}$$

The MLS solution is obtained by solving the weight least squares problem for each point in the problem domain, i.e., all points on a-axis; see Figure 1.3. For each point a_*, the corresponding x_* is estimated with (1.13). By calculating $b_* = x_* a_*$, the estimated points $\{a_*, b_*\}$ are obtained, as shown in red in Figure 1.3. Notice that due to the unique position of a_* with respect to all observations $\{a_i, b_i\}$, a unique weight matrix $\mathbf{W}_*$ is composed, which results in a corresponding x_* that is specific to a_*. Thus, the MLS solution appears a curve instead of a straight line, as shown in Figure 1.4.

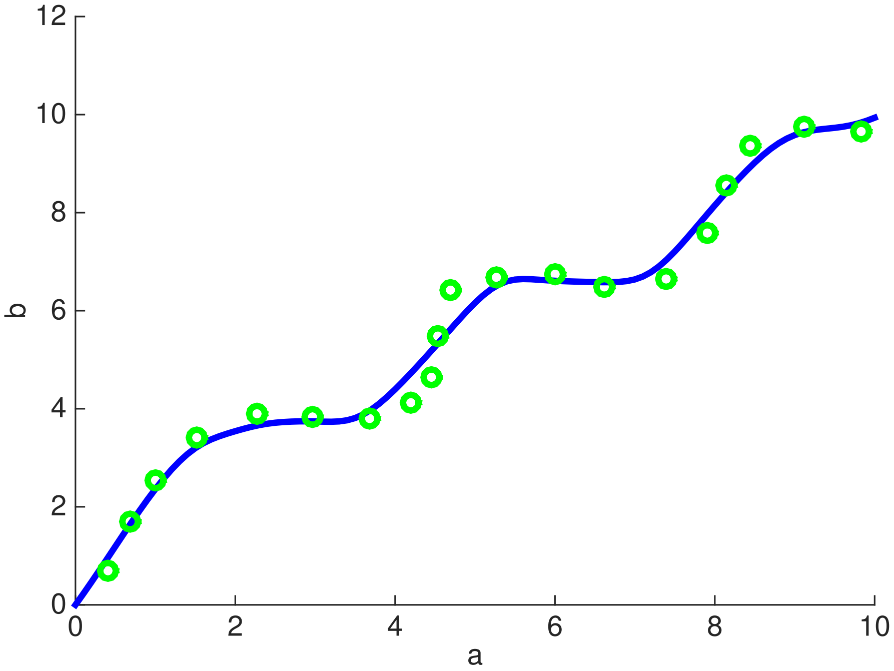


FIGURE 1.4: The MLS technique allows the fit to change locally. The MLS curve is marked in blue.

1.4 Thesis Outline

Chapter 2 is a review of image registration and several areas in computer vision which are related to the target problems in this thesis, namely feature correspondences extraction, radial distortion estimation, image stitching and video stabilization.

In Chapter 3, a practical algorithm is developed for correspondence validation under radial distortion. As opposed to recent methods that augment the standard epipolar constraint, the thesis adopts a different approach. The radial distortion is treated as a violation to the basic epipolar geometry. To account for the distortion effects, the epipolar geometry is adjusted using MDLT. The proposed technique is much simpler compared to previous methods and exhibits a high degree of flexibility.

Chapter 4 introduces a method that inserts correspondence for APAP warps. APAP warps produce accurate panoramic stitching, especially in cases with significant depth parallax. However, in image regions where feature detection or matching fail, the warp loses guidance and is unable to accurately model the true underlying warp. While, the proposed method automatically identifies misaligned regions, and inserts appropriate point correspondences to increase the flexibility of the warp and improve alignment.

Chapter 5 proposes the usage of homography fields for video stabilization. It has been shown that homography fields are powerful enough to meet the various warping needs of video stabilization, not just in the core step of stabilization, but also in video inpainting.

The last chapter draws the thesis to a close. After summarizing the main contributions of the preceding chapters, possible directions for future work are offered.

Chapter 2

Literature Review

2.1 Feature Detection and Matching

Many automatic methods have been developed to extract distinctive features from stereo images, and to then match these features to estalish correspondences. In this section, methods for detecting and matching distinctive features are reviewed.

2.1.1 Feature detection

Feature detectors extract salient features from images. Salient features include corners [17, 132, 62], line intersections [121], and lines/edges [78]. Shi and Tomasi [119] proposed to use the minimum eigenvalue of the subimage Hessian matrix to find good features. Förstner [46] introduced a method to find keypoints using a Gaussian weighting function to replace the equal weighting used in [119]. Instead of using the minimum eigenvalue, Harris and Stephens [57] proposed a simpler quantity, which was later extended by Triggs [127]. Brown *et al.* [26], on the other hand, used the harmonic mean. More recently, feature detectors become invariant to scale [93, 101] and geometric transformations [12, 65, 116, 100]. Lindeberg [81] and Lowe [93] achieved scale invariance by searching for scale-space maxima of Difference of Gaussian (DoG), whereas Mikolajcayk and Schmid [101] and Triggs [127] used Harris corner detector [57] computed over a sub-octave pyramid.

Besides keypoints, there are other features that can be used to register images. Line features [62, 133, 103, 77, 34, 53, 78] can be extracted from images. Alhichri and Kamel [4] proposed a method that employs virtual circles to produce region features. Matas *et al.* [97] introduced a feature extraction method based on maximally stable extremal regions. Tuytelaars and Van GooL [130] extracted "variant regions" based on detected corners and their nearby edges.

2.1.2 Feature matching

Once features are detected from input images, they will be encoded into descriptors to describe the characteristics of their surrounding neighborhoods. Feature descriptors are suitable for discriminative matching, and are used to establish feature correspondences. Mikolajczyk and Schmid [101] compared several local descriptors [82, 92, 116], among which the Scale Invariant Feature Transform (SIFT) [93] outperforms others. When extracting a SIFT feature, an orientation histogram is estimated from local image gradients. SIFT feature descriptor typically extractes 4×4 sub-blocks from a 16×16 neighborhood image gradients. Each sub-block contains 8 orientation bins. This results in a total of $4 \times 4 \times 8 = 128$ bin values. Thus, a SIFT feature descriptor consists of a 128-dimensional vector.

To establish the correspondences, the easiest way is to compare all features in one image against all features in the other, using one local descriptor (in this thesis SIFT is used). However, this approach leads to quadratic time complexity in the expected number of features. Nene and Nayar [106] proposed to use indexing scheme to cull down the number of candidates for each feature points. Beis and Lowe [13] developed a Best-Bin-First (BBF) algorithm to identify the nearest neighbors for each feature using only a limited amount of computation. Shakhnarovich *et al.* [117] extended the locality sensitive hashing technique to be more sensitive to the distribution of points in parameter space. Brown and Szeliski [26] greatly sped up the search for correspondences based on low-frequency Haar wavelet coefficients. Nister and Stewenius [107] proposed the vocabulary tree to compare feature descriptors more efficiently. Although many feature detection and matching methods have been published, it is not the focus of this thesis. In this thesis, SIFT features are detected and matched with functions released in the VLFeat open source library [131].

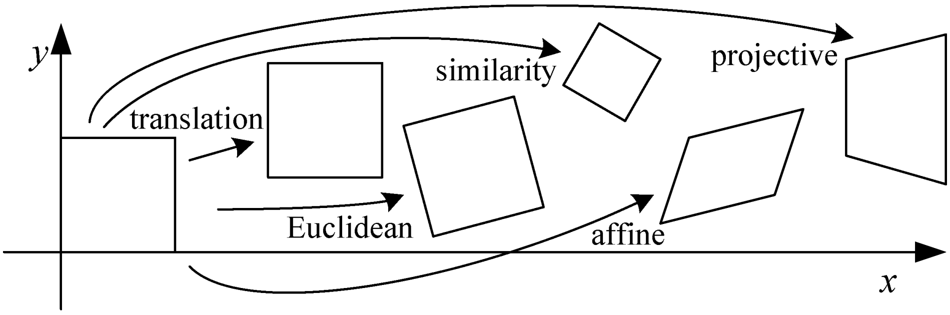


FIGURE 2.1: Basic set of global transformation models. Figure is taken from [126]

2.2 Transformation Functions

With the feature correspondences established, the transformation function that maps one input image to another needs to be estimated. A variety of transformation functions are possible. Generally, transformations can be categorized into two kinds. Global transformation functions have a fixed number of parameters and are defined for the entire image. On the other hand, local mapping functions have a varying number of parameters and are more flexible to local deformation.

2.2.1 Global transformation functions

A variety of global transformation models can be used, like translation, rigid, similarity, affine, and projective transformations. The transformations are illustrated in Figure 2.1.

Translation. 2D translations can be written as $\mathbf{x}' = \mathbf{x} + \mathbf{t}$ or

$$\mathbf{x}' = [\ \mathbf{I}\ \mathbf{t}\ ]\ \tilde{\mathbf{x}}, \tag{2.1}$$

where $\mathbf{I}$ is a 2×2 identity matrix and $\tilde{\mathbf{x}} = (x, y, 1)$ is $\mathbf{x}$ in homogeneous coordinate. Given a correspondence in the images, vector $\mathbf{t}$ can be estimated.

Rigid. This transformation is also known as the Euclidean transformation since Euclidean distances are preserved. It can be expressed as $\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}$, which can also be written as

$$\mathbf{x}' = [\ \mathbf{R}\ \mathbf{t}\ ]\ \tilde{\mathbf{x}}, \tag{2.2}$$

where

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.3}$$

At least two correspondences are required to determine the rigid transformation.

Similarity. The similarity transformation can be expressed as $\mathbf{x}' = s\mathbf{R}\mathbf{x} + \mathbf{t}$, which can also be written as

$$\mathbf{x}' = [\ s\mathbf{R}\ \mathbf{t}\ ]\ \tilde{\mathbf{x}} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \end{bmatrix} \tilde{\mathbf{x}}, \tag{2.4}$$

where s is an arbitrary scale factor. The similarity transformation preserves angles. At least two correspondences are required to estimate the parameters of similarity transformation.

Affine. The affine transformation can be expressed as

$$\mathbf{x}' = \mathbf{A}\tilde{\mathbf{x}} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \tilde{\mathbf{x}}, \tag{2.5}$$

where $\mathbf{A}$ is an arbitrary 2×3 matrix. The affine transformation preserves parallelism. A minimum of three correspondences are required to calculate the affine transformation.

Projective. The projective transformation is also known as the planer perspective transformation or more commonly homography. The homography can be expressed as

$$\tilde{\mathbf{x}}' \sim \mathbf{H}\tilde{\mathbf{x}}, \tag{2.6}$$

where $\sim$ denotes equality up to scale and $\mathbf{H}$ is an arbitrary 3×3 scale invariant matrix. Straight lines remain straight under perspective transformation. Four corresponding points are required to determine the projective transformation parameters.

2.2.2 Local transformation functions

Since global transformation functions have fixed numbers of degree of freedom and are defined for the entire image, they cannot accommodate local non-rigid deformations. Local transformation functions can be used to deform local patches of the image.

Radial basis functions. A radial basis function (RBF) is a real-valued function whose value decreases (or increases) monotonically on the distance (usually Euclidean distance) from a centre point. There are many kinds of radial basis functions. A commonly used RBF is the Gaussian function:

$$\o(\|\mathbf{x}-\mathbf{c}\|) = \exp\left(-\epsilon^2\|\mathbf{x}-\mathbf{c}\|^2\right), \tag{2.7}$$

where $\o$ is the RBF associated with the distance $\|\mathbf{x}-\mathbf{c}\|$ from point $\mathbf{x}$ to center $\mathbf{c}$, $\mathbf{x}$ is the point at which the function is evaluated, and scalar parameter $\epsilon \neq 0$ may be adjusted to adapt the approximation. A Gaussian RBF monotonically decreases with the distance from a centre point.

RBFs are usually used in interpolation of scattered data. Given N points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ at which values of the function to be estimated are known $f_1, f_2, \ldots, f_N$, the RBF interpolation is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \o(\|\mathbf{x}-\mathbf{x}_i\|), \tag{2.8}$$

where $\|\mathbf{x}-\mathbf{x}_i\|$ is the distance between $\mathbf{x}$ and $\mathbf{x}_i$ so that each RBF $\o(r_i)$ is associated with a different centre $\mathbf{x}_i$ meanwhile weighted by scalar parameter w_i. To estimate $f(\mathbf{x})$ at an arbitrary point $\mathbf{x}$, one just needs to estimate the coefficients w_i. According to the interpolation conditions $f(\mathbf{x}_i) = f_i, i = 1, \ldots, N$, the following linear system can be obtained:

$$\mathbf{A}\mathbf{w} = \mathbf{f}, \tag{2.9}$$

where

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{bmatrix}, \tag{2.10}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}, \text{ and } \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}. \tag{2.11}$$

In $\mathbf{A}$, each entry $a_{i,j}$ is calculated as $a_{i,j} = \o(\|\mathbf{x}_i - \mathbf{x}_j\|)$. With w_i determined, following (2.8), $f(\mathbf{x})$ can be easily solved for at any arbitrary point $\mathbf{x}$.

Thin plate spline. The thin plate spline (TPS) method is an augmented RBF interpolation method with additional linear terms. Given N points $\mathbf{x}_1 = \{x_1, y_1\}, \mathbf{x}_2 = \{x_2, y_2\}, \ldots, \mathbf{x}_N = \{x_N, y_N\}$ at which values of the function to be estimated are known $f_1, f_2, \ldots, f_N$, the TPS interpolation is defined as

$$f(\mathbf{x}) = c_1 + c_x x + c_y y + \sum_{i=1}^{N} w_i \o(\|\mathbf{x} - \mathbf{x}_i\|), \tag{2.12}$$

where $\o$ is defined as $\o(r) = r^2 \ln r$, under the following conditions

$$\sum_{i=1}^{N} w_i = 0, \ \sum_{i=1}^{N} w_i x_i = 0 \text{ and } \sum_{i=1}^{N} w_i y_i = 0. \tag{2.13}$$

Similar to the basic RBF interpolation method, the following linear system can be obtained to solve $c_1, c_2, c_3, w_1, w_2, \ldots, w_N$:

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}, \tag{2.14}$$

where

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N, \end{bmatrix}, \text{ and } \mathbf{c} = \begin{bmatrix} c1 \\ c_2 \\ c_3. \end{bmatrix}. \tag{2.15}$$

$\mathbf{A}$ is the matrix in (2.10), $\mathbf{w}$ and $\mathbf{f}$ are the vectors in (2.11). Once $\mathbf{w}$ and $\mathbf{c}$ are computed, (2.12) is used to solve for $f(\mathbf{x})$.

Moving least squares. Given control points $\mathbf{p}_i$ with associated data values f_i, moving least squares (MLS) methods strive to find function $f(\mathbf{x})$ at each point $\mathbf{x} = (x, y)$ that minimizes the following problem:

$$\sum_{i=1}^{N} w_i(\mathbf{x}) \| f(\mathbf{x}) - f_i \|^2, \tag{2.16}$$

where $w_i(\mathbf{x})$ is a non-negative monotonically decreasing radial function centered at control point $\mathbf{p}_i$, such as the one used in [140]

$$w_i(\mathbf{x}) = e^{-(\mathbf{x}-\mathbf{p}_i)^2/\sigma^2}. \tag{2.17}$$

The derivation in this section is the multivariate extension of the 1D version in Section 1.3.2.

MLS technique has been widely used in many research areas in computer graphics and vision, such as surface reconstruction [3, 74, 5], elastoplastic material simulation and animation [14, 104], partition of unity implicits [108], and image deformation [115]. More recently, Zaragoza *et al.* [140] have further developed the work of [115] and proposed moving direct linear transformation (MDLT) to produce as projective as possible image warps. MDLT is conceptually the homogeneous version of MLS and based on projective warps. Section 4.1.2 will present MDLT in detail.

2.3 Camera Calibration and Radial Distortion

In this section, the concepts of camera calibration, radial distortion and the state-of-the-art estimation methods will be reviewed.

2.3.1 Camera calibration

Camera calibration has been studied extensively in computer vision [50, 48, 41, 129, 136, 42, 99, 134] and photogrammetry [23, 39]. It is the process of estimating extrinsic and intrinsic parameters of a camera from 2D images. The camera model considered here is the pinhole camera model. A pinhole camera is a simple camera without a lens

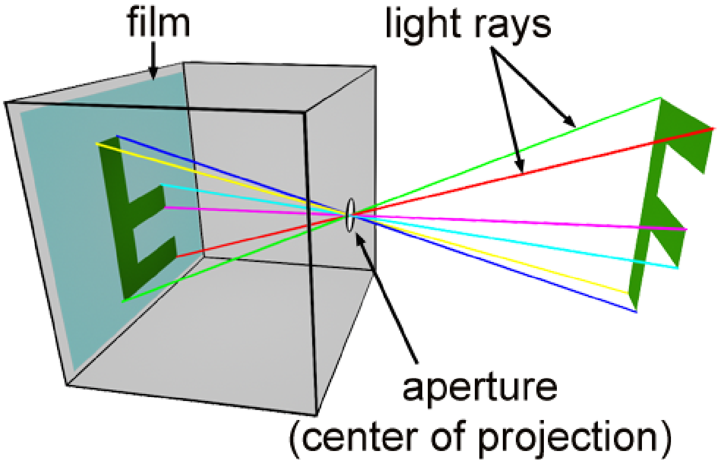


FIGURE 2.2: Principle of a pinhole camera[1]. A pinhole camera is a simple camera without a lens and with a hole as the aperture. Light rays pass through the aperture and project an inverted image on the opposite side of the camera.

and with a single small aperture. Light rays pass through the aperture and project an inverted image onto the film, as shown in Figure 2.2.

Let $\mathbf{m} = [u, v]^T$ denote a 2D point coordinate, and $\mathbf{M} = [x, y, z]^T$ be a 3D point coordinate. Use $\tilde{\mathbf{x}}$ to denote vector in homogeneous coordinate by adding 1 as the last element: $\tilde{\mathbf{m}} = [u, v, 1]^T$ and $\tilde{\mathbf{M}} = [x, y, z, 1]^T$. The equation of the projection between a 3D point $\mathbf{M}$ and its image projection $\mathbf{m}$ is given by

$$\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\tilde{\mathbf{M}}, \tag{2.18}$$

where $\mathbf{P} = \mathbf{K}[\ \mathbf{R} \mid \mathbf{t}\ ]$ is known as the camera matrix, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is called the camera intrinsic matrix, and $\mathbf{R} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^{3\times1}$ define a 3D Euclidean transformation on the basis of position and orientation of the camera.

$\mathbf{R}$ is obtained from multiplying three basic rotation matrices that rotate vectors by angles θ_x, θ_y and θ_z about the x, y, and z axes,

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}, \tag{2.19}$$

[1] http://scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera.

$$\mathbf{R}_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \tag{2.20}$$

$$\mathbf{R}_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.21}$$

i.e.,

$$\mathbf{R} = \mathbf{R}_z \cdot \mathbf{R}_y \cdot \mathbf{R}_x. \tag{2.22}$$

The translation vector $\mathbf{t}$ is notated as

$$\mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \tag{2.23}$$

where t_x, t_y and t_z specify respectively the translation amount in the x, y and z direction. $\mathbf{R}$ and $\mathbf{t}$ together contain six parameters, which are called extrinsic: θ_x, θ_y, θ_z, t_x, t_y and t_z.

The intrinsic matrix, $\mathbf{K}$, has the following form

$$\mathbf{K} = \begin{bmatrix} \alpha & s & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.24}$$

where (c_x, c_y) are the coordinates of the principal point, α and β represent focal length in pixel units, and s is the skew coefficient between two image axes. Primarily, the goal of camera calibration is to estimate six extrinsic parameters and five intrinsic parameters.

Once extrinsic and intrinsic are extracted, they can be used to correct for lens distortion, measure the size of an object, infer 3D information from 2D information, and vice versa. In short, camera calibration is a prerequisite for any application where the relation between 2D images and the 3D world is needed.

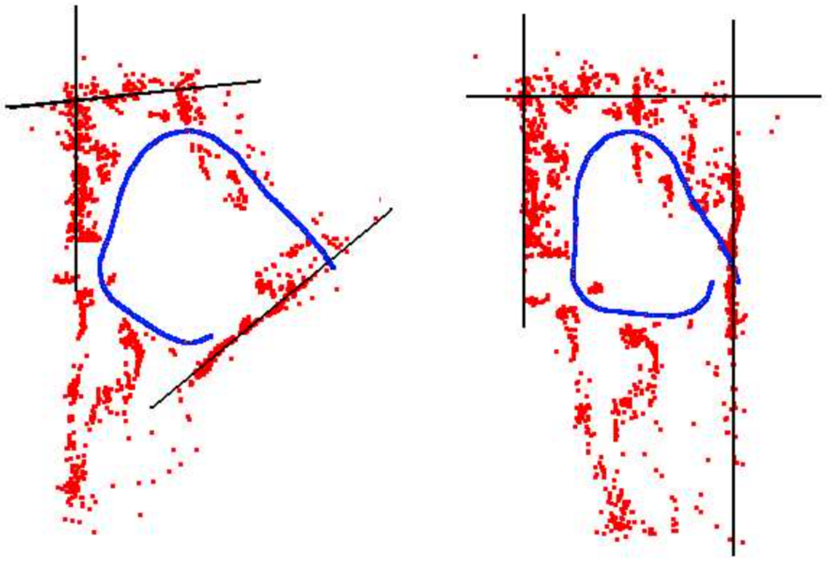


FIGURE 2.3: A 3D reconstruction of an 800-frame video of an office scene computed by a commercial camera tracker[2], without distortion correction (left). The lens distortions appeared in images lead to incorrectly estimated focal length and rotation. Therefore, the reconstruction result is far from the correct geometry. While, the correct reconstruction is shown on the right. Figure is taken from [44].

2.3.2 Radial distortion models

A pinhole camera performs a perfect perspective transformation, and is free from lens distortion [61], since no lens is involved. However, a pinhole camera has a few severe limitations. The size of the pinhole must be very small, otherwise the image will be blurry. Using pinhole with smaller size improves the image resolution, which however reduces the amount of captured light. As a result, a tripod and long exposures are necessary to photograph a pinhole image, which is impractical for daily use. Thus, real cameras approximate ideal pinhole cameras using optical lenses.

Due to the imperfections in lenses, to accurately characterize a real camera, camera lens distortion needs to be considered. Input images effected by lens distortions will lead to incorrectly estimated focal length and camera poses, which are fatal to the 3D reconstruction, as shown in Figure 2.3. Although distortion can be irregular or follow many patterns, the most commonly encountered distortion is radially symmetric.

Denote the perspective, pinhole point using $\tilde{\mathbf{p}} = (p, q, 1)$, and denote the distorted image point as $\tilde{\mathbf{x}} = (x, y, 1)$, both in homogeneous coordinates. Denote a set of two-view

[2] http://www.2d3.com

correspondences as $\tilde{\mathbf{p}} \leftrightarrow \tilde{\mathbf{p}}'$, and denote fundamental matrix and essential matrix as $\mathbf{F}$ and $\mathbf{E}$ respectively. Since only radial distortion is considered here, the relationship between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{p}}$ is purely dependent on their distances from the center of distortion (COD).

Earlier works on radial distortion calibration are the so-called plumb line methods, e.g., [35, 68, 123]. These methods function by identifying (either manually or automatically) distorted straight lines in the input images and then attempting to straighten them. However, since straight lines are not always available in the real world, such methods may easily confuse real curves with distorted lines [44].

More recent methods introduce radial distortion directly into the mathematical model of the imaging process, with the aim of deriving epipolar geometry with radial distortion. Zhang [142] introduced one of the earliest epipolar geometry model with radial distortion. Zhang's model was later extended to solve for the system parameters and camera motions [67]. Under the assumption of mild radial distortion, Fitzgibbon [44] proposed the division model with a single distortion parameter. In the division model, all points are expressed in a 2D coordinate system with origin at the distortion center, which is assumed known or to be located at the center of the image. The radial distortion model is a function about the magnitudes $||\tilde{\mathbf{x}}||$ and $||\tilde{\mathbf{p}}||$, and can be written as

$$\tilde{\mathbf{x}} = \mathbf{L}(||\tilde{\mathbf{p}}||)\tilde{\mathbf{p}}. \tag{2.25}$$

Fitzgibbon expanded the distortion function $\mathbf{L}$ as a Taylor series, and kept only the first nonlinear even term

$$\tilde{\mathbf{x}} = (1 + \lambda||\tilde{\mathbf{p}}||^2)\tilde{\mathbf{p}}. \tag{2.26}$$

The inverse function is the division model, also known as the undistortion model, since it theoretically describes the mapping from distorted coordinates to undistorted coordinates:

$$\tilde{\mathbf{p}} = \frac{1}{1 + \lambda||\tilde{\mathbf{x}}||^2}\tilde{\mathbf{x}}. \tag{2.27}$$

Given image point correspondences $\tilde{\mathbf{x}} \leftrightarrow \tilde{\mathbf{x}}'$, the fundamental matrix $\mathbf{F}$ which only works for undistorted points $\tilde{\mathbf{p}} \leftrightarrow \tilde{\mathbf{p}}'$ can be estimated,

$$\tilde{\mathbf{p}}'^T\mathbf{F}\tilde{\mathbf{p}} = \mathbf{0}. \tag{2.28}$$

Claus and Fitzgibbon [33] extended the undistortion model to the rational function model and showed how to estimate epipolar curves and recover highly distorted images. Barreto and Daniilidis [9] introduced a parameter into the augmented fundamental matrix to quantify the amount of distortion. Kukelova *et al.* [70] proposed minimal solvers for such augmented projective models. All these methods assume that the COD is either known or simply the image centre. However, it has been shown that one cannot assume that the COD is simply at the image centre [76]. Brito *et al.* [19] included the COD as a parameter in the second-order radial distortion model, and recovered the location of the COD in [18].

2.3.3 Robust radial distortion estimation

Registration methods rely on robust feature point correspondences. The most commonly used techniques for establishing feature point correspondences are based on geometric relationships (e.g, fundamental matrix, homgraphy) derived from multiple-view geometry [58]. Such geometric transformations can be easily solved using fast linear solvers, which allows their computations to be integrated into the RANdom SAmple Consensus (RANSAC) framework to identify correct correspondences (inliers) from incorrect ones (outliers). However, when images have radial distortion, these transformations cannot be applied, especially in the image periphery. Thus, Fitzgibbon [44] applied the division model (2.27) in the kernel of RANSAC to validate feature correspondences under radial distortion.

The RANSAC algorithm [43] is an iterative resampling technique that estimates parameters of a mathematical model from input data which contains outliers. In [44], the basic fundamental matrix is combined with division model, which is then used in the RANSAC framework to eliminate outliers and recover fundamental matrix and distortion parameter. The algorithm is summarized in Algorithm 2.1. The algorithm has proven its ability of finding more correspondences and covering more of input images. However, as tested in [90], the accuracy of [44] severely drops as the distortion level is increased.

Other augmented fundamental matrices [33, 9, 19, 18] reviewed in Section 2.3.2 can also be embedded in RANSAC framework to enable robust estimation of epipolar geometry under radial distortion, just by replacing the number in step 1 in Algorithm 2.1 with required number of correspondences and then estimating the augmented fundamental matrix in step 2. However, the fact remains that these methods are based on assumed models of radial distortion, and thus may yield low accuracy if the assumed models are wrong (see the experiments in [90]).

Algorithm 2.1 RANSAC with fundamental matrix combined with division model.

```
1: Select randomly 9 point matches, which are the minimum for a solution in [44].
2: Solve for distortion parameter $\lambda$ and fundamental matrix $\mathbf{F}$.
3: Determine the number of matches that fit $\mathbf{F}$ within a predefined tolerance $\epsilon$.
4: Repeat steps 1 through 3 for $N = 1000$ times.
5: Return the $\mathbf{F}$ that has the highest number of agreement (consensus) among all point
   matches.
```

2.4 Image Stitching

The construction of mosaic images have been an active area of research for many years. There have been a variety of commercial tools based on or incorporating image stitching, such as photography editor Adobe Photoshop, web-based photo organizer Microsoft Photosynth, smartphone application Autostitch. However, many tools give unconvincing results when the input photos violate fairly restrictive imaging assumptions. More recently, more effective algorithms have been proposed to build image mosaics, including the moving least squares method. In this section, previous works on image stitching and panorama construction will be reviewed.

2.4.1 Image stitching pipeline

In general, image stitching algorithms have two steps: estimate the warping functions that align input images, and then composite the aligned images onto a common canvas. Most of the current techniques model the alignment functions as 2D projective transforms or homographies. Homographies are justified only if the images were taken under pure rotational motions, or the scene is sufficiently far away such that it is effectively planar [124]. On top of the usage of basic homography, Shum and Szeliski [120] used methods of photogrammetric bundle adjustment [128] to simultaneously optimize the relative rotations of the input images, and then align all images to a common frame of reference. Such bundle adjustment technique is also adopted in Autostitch [2]. However, the usage of homographic warps for image stitching has been questioned [49, 140] due to the mentioned assumptions homography carries.

Instead of producing perfect alignment throughout the overlap region, pixel blending techniques can be applied to remove misalignments. Agarwala *et al.* [2] and Eden *et al.* [38] applied seam cutting methods to select pixels among the overlapping images to minimize visible seams. Apart from seam cutting, there are other pixel blending techniques, such as Laplacian pyramid blending [28, 25] and Poisson image blending [109], have been proposed to minimize misalignments. However, such post-processing methods strive to reduce errors in the compositing step, which may not work all the time (see [64] for examples).

2.4.2 3D reconstruction and plane-plus-parallax

Given a set of overlapping images of a scene, Agarwala *et al.* [1] recovered the 3D structure and camera parameters (via structure-from-motion (SfM) [58] and freely available software PtLens [3]), and then reprojected each scene point to produce panoramas from images taken along streets. However, a 3D reconstruction can be "overkill" and only works for scene points in the overlapping regions [140].

Instead of computing the 3D structure, Dornaika and Chung [37] combined a parallax component with the classical planar transformations associated with three different images (two of them are the images to be stitched). However, their method can only approximate the parallax at each pixel, which still results in significant parallax errors.

2.4.3 Panorama creation

State-of-the-art methods [120, 25] optimize the focal lengths and camera poses (relative rotations) of all views by performing bundle adjustment [128], and then estimate inter-image homographies to construct panoramas. Shum and Szeliski [120] defined the error terms based on pixel values at regularly sampled patch positions and conducted a second refinement stage. For each patch position, rays from each view were first back-projected, whose average was then projected again onto each view to yield the desired patch position in 2D. Shum and Szeliski utilized the differences between the original and desired patch

[3] http://epaperpress.com/ptlens/

positions to form a correction field to compensate for parallax. In [25], a panorama recognition step based on SIFT keypoint correspondences [93] is introduced, which is able to classify images belong to the same panorama. Instead of estimating relative camera poses, Kang *et al.* [66] and Marzotto *et al.* [96] proposed to chain the inter-image homographies to stitch all images onto a common canvas. Controlling the chaining error becomes the crux of the method.

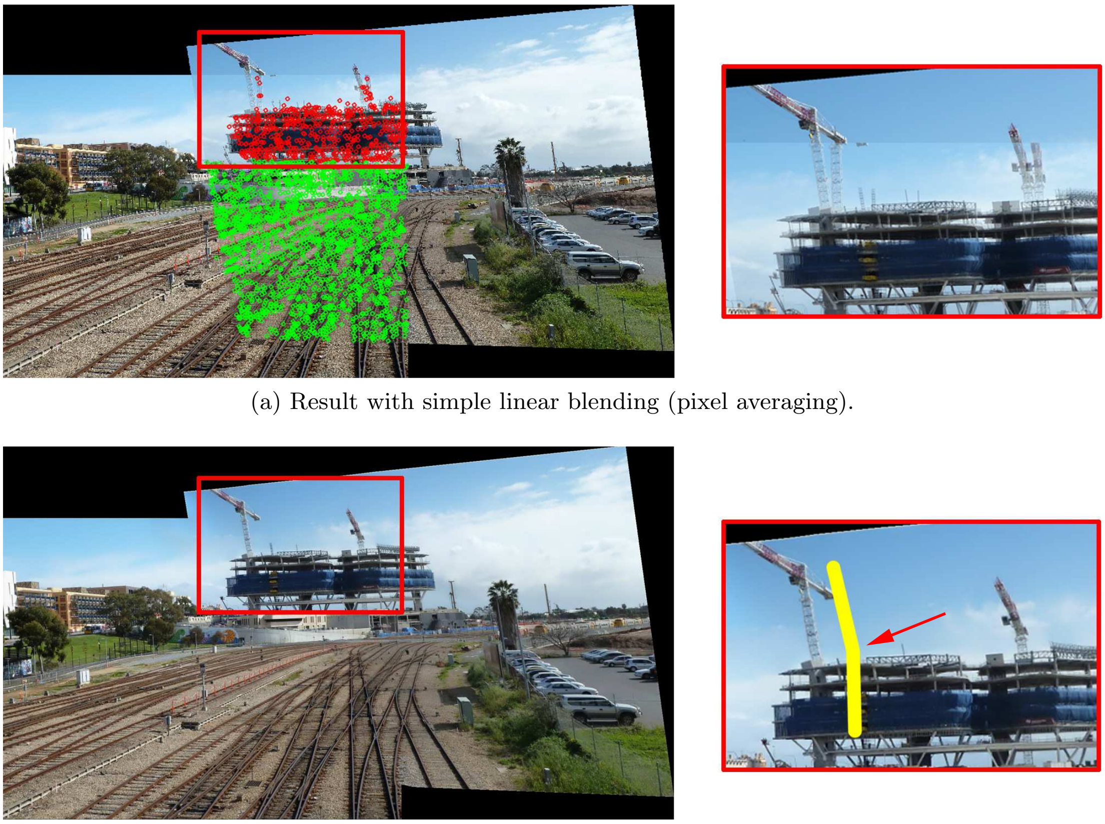
2.4.4 Spatially varying warps

Spatially varying warps have been proposed as alternatives to homographic warps [86, 49, 140, 31]. Lin *et al.* [80] proposed the smoothly varying affine warp for image stitching, which is conceptually similar to the as-affine-as-possible warp [115]. However, using affine regularization is inadequate to achieve a perspective extrapolation [124]. In the context of video stabilization, Liu *et al.* [86] proposed content preserving warps to render stabilized video frames. Gao *et al.* [49] proposed the usage of dual-homography to construct image mosaics of a panoramic scene containing a distant back plane and a ground plane. More recently, Zaragoza *et al.* [140] proposed the as-projective-as-possible (APAP) warps which is able to interpolate the data flexibly, while maintaining a global projective trend so as to extrapolate correctly. Half-projective warps [31] improve upon APAP by preventing excessive stretching when extrapolating.

Ultimately, spatially varying warps are only as flexible as warranted by available feature matches. Without a sufficiently dense sampling of the underlying interpolant, the warp reduces to the baseline warp (similarity [86], projective [140]), thus defeating its spatially varying ability. A large number of feature matches are thus required to obtain good alignment, especially in areas where the true alignment function deviates from a simple homography.

2.4.5 Parallax-tolerant image stitching

Except spatially varying warps, an alternative idea is that perfect alignment throughout the overlap region is unnecessary [141]. Rather, images need only to be aligned well in a local area, and a randomized algorithm was proposed to find a local homography. Seam cut [2] is then used to remove misalignments elsewhere. Such an approach is heavily



(a) Result with simple linear blending (pixel averaging).

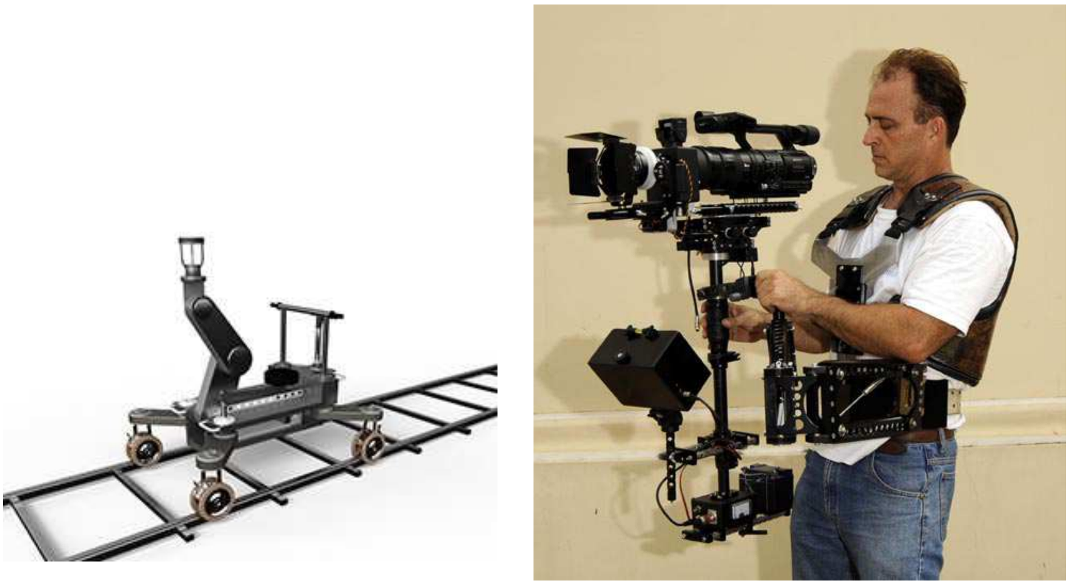
(b) Result after seam cut pixel selection to remove ghosting.

Figure 2.4: (a) Parallax-tolerant image stitching finds a homography that aligns a local region as well as possible. Here, green points are correspondences that are fitted by the local homography. Expectedly, regions that do not lie on the same plane cannot be aligned well; (b) Seam cut removes ghosting, but produces perceptually awkward results; note that the left crane appears to be bent. This result was taken directly from the project website of [141].

dependent on post-processing by seam cut. However, if misalignments are too severe, seam cut may not produce geometrically correct results [64]. This will occur when the true alignment function deviates significantly from a homography, e.g., when there are two apparent planes; see Figure 2.4. More crucially, this method is reliant on the existing set of keypoint matches and cannot introduce new correspondences.

2.5 Video Stabilization

The proliferation of hand-held video recording devices has resulted in large quantities of videos taken by amateurs or casual users. Many amateur videos are taken in an undirected and spontaneous manner, which yields low quality videos with significant amounts of shakiness. Professional tools such as dollies (see Figure 2.5a) or steadicams



(a) Camera dolly.

(b) Steadicam.

FIGURE 2.5: Professional tools such as camera dollies or steadicams mechanically isolate the operator's movement. They allow for smooth shot, even when moving quickly over an uneven surface.

(see Figure 2.5b) used in movie making are too impractical for casual recording purposes. Consequently, there is a need for post-processing software that can remove shakiness in recorded videos, especially amateur videos that exist in large quantities on video sharing websites. In this section, the state-of-the-art methods that automatically stabilize videos, remove rolling shutter effects and lastly inpaint rendered sequence will be surveyed.

2.5.1 Motion compensation methods

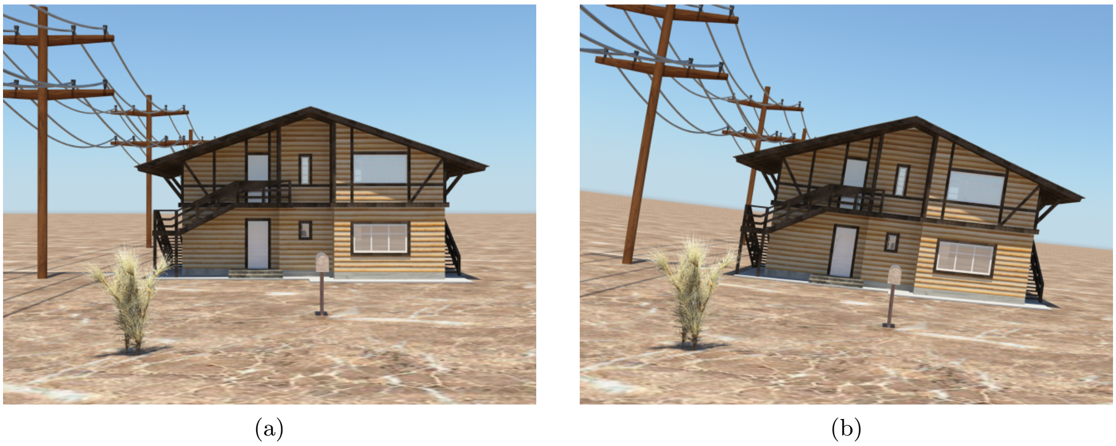
Most video stabilization methods conduct the following steps: estimate the camera trajectory throughout the input shaky video, then smooth the trajectory to remove high-frequency motions. From the smoothed trajectory, construct update transforms that can be applied on each frame of the video to "undo" the jerky motions. Theoretically, the idea can be realized via SfM, where the 3D camera pose trajectory is estimated (along with a set of features in 3D) and smoothed. The projection of the original frames onto the smoothed camera poses amounts to performing the update transforms. However, conducting SfM on a long shaky video can be error-prone.

Unsurprisingly many approaches avoid explicit 3D reconstruction. The class of 2D approaches estimate 2D image transforms from feature matches across two successive frames (e.g., via SIFT matching) to capture the camera motion. The estimated sequence of transforms are then filtered to perform stabilization. Earlier methods relied on simple 2D image transforms (e.g., affine or projective), which are very efficient to estimate [102, 98, 51]. Fundamentally, however, such simple transforms cannot adequately model the 3D camera motion. Further, the update transform is also defined using simple 2D image transforms, which cannot preserve the image contents well. As a result the stabilized videos may appear distorted and "wobbly" [86]. Nevertheless, techniques based on projective transforms (homographies) can work well in practice, especially if good motion planning is conducted during stabilization [51]. In particular, Gleicher and Liu [51] proposed a motion planning framework based on cinematographic principles. Grundmann *et al.* [54] divided the original camera path to segments with constant, linear and parabolic motions, which are stabilized with L1-norm optimization.

Recent works have proposed more sophisticated motion models and smoothing algorithms. Liu *et al.* [87] tracked local features in the video, then smoothed the set of trajectories based on low-rank matrix factorization. Goldstein and Fattal [52] estimated fundamental matrices that encapsulate the epipolar constraint between successive frames, then generated virtual point trajectories from the epipolar relations which were then filtered. More recently, Liu *et al.* [89] spatially partitioned the video frame into a grid of subwindows, then estimated a chain of homographies across the video for each subwindow. The separate chains of homographies are then smoothed in a bundled manner to avoid drift. In all three approaches, an update transform must be performed on each frame based on the original and stabilized feature locations; content preserving warps (CPW) [86] have been used or adapted for this purpose.

2.5.2 Rolling shutter removal

Most current imaging sensors are based on CMOS technology. Such sensors are subject to the phenomenon known as electronic rolling shutter, where image rows are exposed and readout at different times [105]. This causes distortions in the video known as rolling shutter effects. For example, as shown in Figure 2.6, the house appears as slanted since



(a)

(b)

FIGURE 2.6: Synthetic images (a) and (b) are two input images taken from [45], with rolling shutter effect in the second one.

the camera is panned while recording. Methods have been proposed for the removal of rolling shutter effects in recorded videos [32, 79, 8, 45, 110, 55].

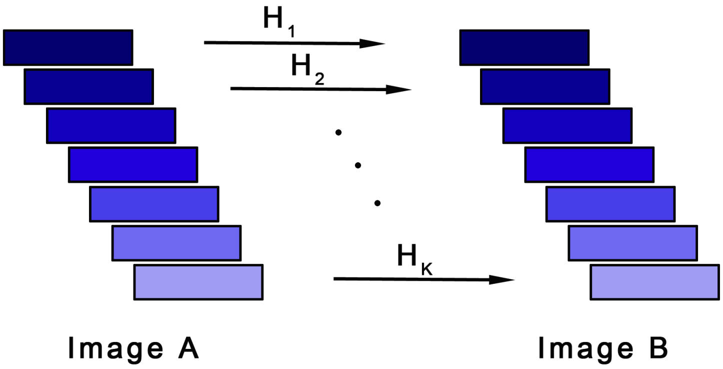
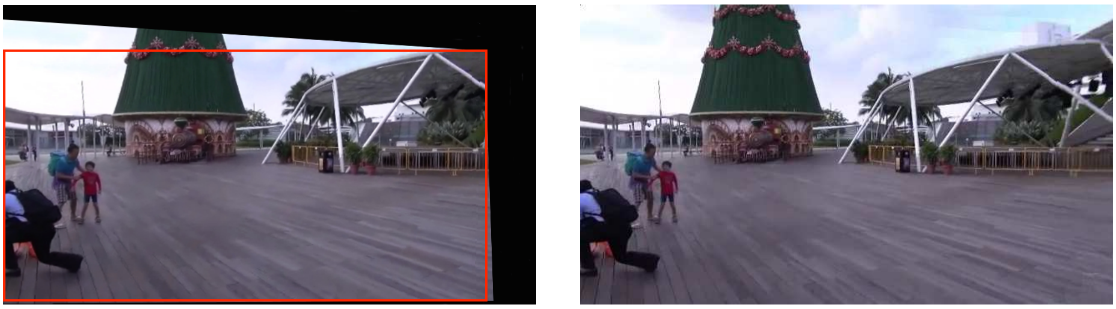


FIGURE 2.7: Grundmann *et al.* [55] proposed the usage of homography mixtures to remove rolling shutter effects. A frame is partitioned vertically into K equal-sized strips, and for each strip a homography that maps to the corresponding strip in the next frame is estimated.

Grundmann *et al.*. [55] proposed the usage of homography mixtures to remove rolling shutter effects in the context of video stabilization. Basically, a frame is partitioned vertically into K equal-sized strips, as shown in Figure 2.7, and for each strip a homography that maps to the corresponding strip in the next frame is estimated. Conceptually this is a kind of spatially varying warp. However, homography mixtures are allowed to vary only vertically, and this constrains the flexibility of the warp. Note that Liu *et al.* [87], Goldstein and Fattal [52] and Liu *et al.* [89] do not explicitly model and compensate for rolling shutter effects, beyond treating it as part of high-frequency jitter.



(a) Blank pixels lead to cropping.

(b) Inpainting result.

Figure 2.8: Due to the deviation of the smoothed camera path from the original trajectory, the existence of blank pixels in the stabilized video is unavoidable. Each frame has to be cropped to render the output video following a fixed-sized rectangular as marked in red in (a). As a result, video inpainting methods have been developed to avoid significant blank regions in the rendered video; see (b).

2.5.3 Video inpainting

In essence, video stabilization is re-rendering the video frames as viewed from smoothed camera positions. The existence of blank pixels in the new views is thus unavoidable; see Figure 2.8a. Further, the size of the blank regions is large if the new viewing positions significantly differ from the original views. Eventually each image will have to be cropped to a fixed-sized rectangular frame to allow saving the video. This creates unsightly blank regions in the video, unless the blank pixels are inpainted using information from other frames.

Differing from inpainting for censored or damaged videos (e.g., [137]), inpainting for stabilized videos is an extrapolation problem, since the missing regions almost always occur at the sides of the video frames. The state-of-the-art method of Matsushita *et al.* [98] eschewed the usage of standard mosaicing techniques for video inpainting [83], due to significant misalignment errors. Instead, Matsushita *et al.* propagated 2D motion fields (derived from optic flow) to guide the inpainting. However, Matsushita *et al.*'s method produces visible artifacts if the blank region is large. This stems from the difficulty of extrapolating motion fields to large empty regions without an underlying function to guide the extrapolation.

Note that approaches that avoid inpainting must either limit the magnitude of smoothing (as shown in some of Liu *et al.*'s [89] results) or aggressively crop the output video (such as Grundmann *et al.* [54]) to avoid significant blank regions in the output video.

2.6 Summary

In this chapter, the state-of-the-art methods on feature detection and matching are first reviewed in Section 2.1. Many computer vision applications are developed based on feature correspondences from input images. In this thesis, all works employ feature detection and matching as the initial step.

After feature correspondences are established, the geometric transformation that registers input images needs to be estimated. There are several transformation models can be used to characterize the mapping, like the global mapping models reviewed in Section 2.2.1 and the local mapping models mentioned in Section 2.2.2. Due to the ability of registering images locally and providing smooth results, the MLS method can be used to solve many challenging problems. Among many, three specific applications have been surveyed: radial distortion estimation in Section 2.3, image stitching in Section 2.4 and video stabilization in Section 2.5.

Chapter 3

Point Correspondence Validation Under Unknown Radial Distortion

3.1 Introduction

3.1.1 Point correspondence validation

Most registration problems rely on robust feature point correspondences. The most commonly used techniques for establishing feature point correspondences are based on geometric relationships, such as epipolar geometry. The epipolar geometry is the intrinsic projective geometry of stereo vision, when a 3D scene is captured by cameras from multiple distinct positions. A 3D point $\mathbf{p}$ that is imaged by two cameras will yield a pair of corresponding 2D image points $\mathbf{x} = [x\ y]^T$ and $\mathbf{x}' = [x'\ y']^T$, denoted in homogenous coordinates as $\tilde{\mathbf{x}} = [x\ y\ 1]^T$ and $\tilde{\mathbf{x}}' = [x'\ y'\ 1]^T$. A pair of corresponding points will satisfy the epipolar constraint, given by

$$\tilde{\mathbf{x}}'^T \mathbf{F} \tilde{\mathbf{x}} = \begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0, \tag{3.1}$$

where the fundamental matrix $\mathbf{F}$ is a 3×3 matrix of rank 2 that algebraically represents the epipolar geometry [58]. The epipolar constraint is derived based on the assumption that $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ were observed using pinhole cameras. See [40] for details.

For each correspondence, the epipolar constraint can be written as follows

$$\mathbf{af} = 0, \tag{3.2}$$

where

$$\mathbf{a} = [\; xx' \quad yx' \quad x' \quad xy' \quad yy' \quad y' \quad x \quad y \quad 1 \;], \tag{3.3}$$

$$\mathbf{f} = [\; F_{11} \quad F_{12} \quad F_{13} \quad F_{21} \quad F_{22} \quad F_{23} \quad F_{31} \quad F_{32} \quad F_{33}]^T \;. \tag{3.4}$$

By stacking n correspondences, the following system of homogeneous equations can be constructed

$$\mathbf{Af} = \mathbf{0}, \tag{3.5}$$

where $\mathbf{A} \in \mathbb{R}^{n \times 9}$ contains the n point coordinates in the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} . \tag{3.6}$$

Imposing the constraint $\|\mathbf{f}\| = 1$ and formulating the problem in the least square sense leads to the following equation

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmin}} \; \|\mathbf{Af}\|^2 \tag{3.7}$$

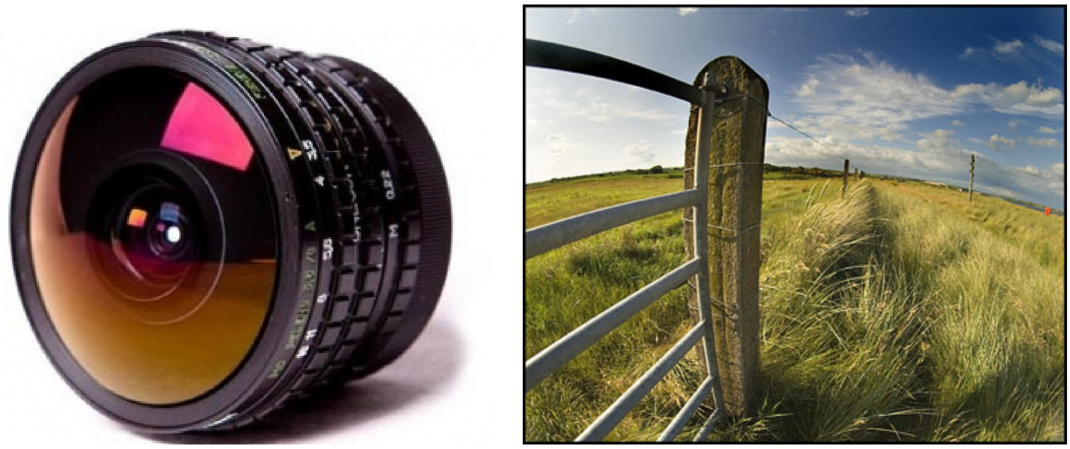
Then $\hat{\mathbf{f}}$ can be estimated by taking the least significant right singular vector of $\mathbf{A}$. The process of rewriting linear equations as a matrix system and then solving the problem is referred to as the direct linear transformation (DLT) algorithm. The resulting fundamental matrix is then truncated to rank 2. In general, at least 7 correspondences are required to compute the fundamental matrix [58]. In this chapter, the 8-point algorithm [91] is used. For numerical stability, the point coordinates may also be normalized following [59].

The fundamental matrix can be easily embedded into the RANSAC framework to identify inliers from outliers, and thus to validate feature point correspondences. Please refer to Section 2.3.3 for more details on RANSAC.

3.1.2 Radial distortion calibration

The epipolar geometry of a scene describes how corresponding geometric entities are mathematically related across stereo images [40]. It serves an important role in recovering the 3D shape of the scene and the relative camera poses. The epipolar geometry is derived based on the assumption that ideal pinhole cameras (as shown in Figure 2.2) are used to capture the images. However, due to the strict demand for the pinhole size and the associated problem of low brightness, pinhole cameras are not practical. Thus, real cameras conduct the projection operation using lenses and apertures. Unfortunately imperfections in lenses lead to artifacts in images known as radial distortion. In reality, radial distortion occurs in every image, and it is especially pronounced in mobile phone cameras with cheap lenses and catadioptric (fisheye lens) cameras; see Figure 3.1.

Radial distortion in lenses must be calibrated for, or the distorted images will lead to biased 3D reconstructions; see Figure 2.3. Recent approaches on radial distortion calibration augment the standard epipolar constraint with a radial distortion model. Starting from the seminal undistortion model of Fitzgibbon [44], successively more complex distortion models have been incorporated into epipolar geometry; see [18] for a survey and comparison. For example, Claus and Fitzgibbon [33] extended the undistortion model to allow for a broader class of distortions. Brito *et al.* [20, 18] incorporated the center of distortion (COD) into the constraint to deal with images with unknown COD. Embedded in a RANSAC framework, these augmented epipolar constraints are more effective than the standard epipolar constraint for outlier removal. However, the fact remains that these methods are based on assumed models of radial distortion, and thus may yield low accuracy if the assumed models are wrong. Concrete evidences will be given in the experiments in this chapter (also see the experiments in [18]). It is worth noting that, though dominant, radial distortion is but one type of lens distortion [18, 129].



(a) Peleng Fisheye 3.5/8 M42 Lens.

(b) Sample picture using a fisheye lens.

FIGURE 3.1: A fisheye lens and a sample picture[1].

3.1.3 Chapter overview

The emphasis of this chapter falls under the online camera calibration, i.e., the method only has a set of overlapping images which exhibit radial distortion, and has no access to the original camera(s). Online radial calibration is usually initiated with procurement of a set of reliable point correspondences across the images. Such correspondences are often obtained using automatic algorithms, which invariably yield wrong correspondences or outliers. The standard epipolar constraint, embedded in a RANSAC framework, cannot be effectively used to identify the outliers due to the presence of radial distortion. In this chapter, a novel approach to validate point correspondences under severe radial distortion is proposed.

Note that the goal of the proposed method does not extend to the recovery of the radial distortion parameters and the correction of radial distortion; to achieve these, *"there is no recourse but to bundle adjustment, initialized with (1) reasonable estimates of camera geometry and (2) good correspondences"* [44]. The contribution of this chapter, therefore, is to provide good correspondences in sufficient quantities to support online radial distortion calibration. A significant advantage of the proposed method is not relying on a radial distortion model. Eventually, however, undoing radial distortion will require knowing (or assuming) the underlying distortion model and how to reverse its effects. The proposed method allows to decouple this model selection task from the point correspondence validation stage.

[1] http://www.rugift.com/photocameras/peleng_fisheye_lens.htm

In this chapter, the radial distortion effect is considered as a violation to the basic epipolar geometry. Instead of attempting to characterize radial distortion, the proposed method adjusts the epipolar geometry as warranted by the data to account for the distortion effects. The model adjustment is achieved through the framework of moving least squares (MLS). The proposed method extends MLS to allow for epipolar geometry estimation, and combines it with M-estimators [112] to enable robust point matching under severe radial distortion. Compared to previous works, the proposed method is much simpler and exhibits a higher tolerance in cases where the exact model of radial distortion is unknown.

The rest of the chapter is organized as follows: Section 3.2 gives an overview of the proposed approach. Section 3.3 captures a rough model of the epipolar geometry. Then a robust version of moving direct linear transformation (MDLT) is used to validate point correspondence under unknown radial distortion in Section 3.4. The overall algorithm is summarized in Section 3.5. Section 3.6 compares the proposed method with state-of-the-art algorithms on synthetic and real data. A summary is then given in Section 3.7.

3.2 Proposed Method Overview

To validate point correspondences obtained from automatic matching algorithms, the proposed method has four main steps; see Figure 3.2. First, SIFT feature points are extracted from input stereo images. Then, RANSAC embedded with the standard epipolar constraint (3.1) is applied to initialize the fundamental matrix. A loose inlier threshold is used in RANSAC to accommodate the deviation from the epipolar geometry caused by radial distortion. The initialized fundamental matrix is then refined by M-estimator. With the help of the converged weights, the proposed method will adjust or “tweak” the epipolar constraint with the robust version of MDLT [140] to account for radial distortion, thereby removing the false positives from the final result.

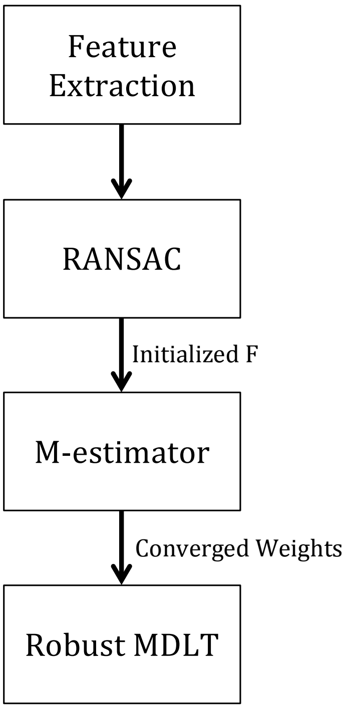


FIGURE 3.2: Point correspondence validation method overview.

3.3 Robust Estimation for Epipolar Geometry

RANSAC embedded with standard epipolar constraint may output feature point correspondences containing outliers, which can severely bias the estimation of fundamental matrix. However, the fundamental matrix $\mathbf{F}$ estimated using matched interest points $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^{N}$ can still serve as the initialization of M-estimator [112].

Let $r_i := \mathbf{a}_i\mathbf{f}$ be the residual of the i-th correspondence with respect to $\mathbf{f}$. To reduce the impact of outliers, the M-estimator replaces the squared residual in (3.7) by a robust loss function $\rho(u)$

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_i \rho(r_i). \tag{3.8}$$

Differentiating the objective function and setting to 0 yields

$$\sum_i \psi(r_i) = 0, \tag{3.9}$$

where $\psi(u)$ is the first derivative of $\rho(u)$. In general, the robust loss function has the form $\psi(u) = u \cdot w(u)$, where $w(u)$ is the weight function. This leads to the equation

$$\sum_i w(r_i) r_i = \sum_i w(r_i) \cdot (\mathbf{a}_i^T \mathbf{f}) = 0, \tag{3.10}$$

which has the equivalent matrix form

$$\mathbf{WAf} = 0, \tag{3.11}$$

where $\mathbf{W} = diag([w_1, w_2, \ldots w_N])$ is the weight matrix, and $w_i = w(r_i)$. Several kinds of robust loss functions are possible [112]. In this work, the Tukey's biweight function is used, whose weight function is

$$w(u) = \begin{cases} [1 - (\frac{u}{\epsilon})^2]^2 & \text{if } u \leq \epsilon \\ 0 & \text{if } u > \epsilon \end{cases}, \tag{3.12}$$

and ϵ is the error scale or inlier threshold.

Note that in (3.11) $\mathbf{W}$ also depends on $\mathbf{f}$. Given $\mathbf{W}$, however, $\mathbf{f}$ can be obtained as the solution of a weighted DLT problem, which is achieved by performing a singular value decomposition (SVD) on $\mathbf{WA}$. Given $\mathbf{f}$, $\mathbf{W}$ can be calculated using the chosen weight function. Therefore, the computations of $\mathbf{f}$ and $\mathbf{W}$ alternate until convergence. This is the well-known iteratively reweighted least squares (IRLS) method, which guarantees convergence to a local minima. Typically, a small number of iterations are required (less than 20 iterations in conducted experiments in this chapter). At convergence, correspondences with higher weights are more likely to be inliers, and vice versa.

The reader may question the efficacy of RANSAC and M-estimator on data affected by radial distortion. Indeed, Fitzgibbon [44] demonstrated that RANSAC (embedded with the standard epipolar constraint) fails under radial distortion since the inlier threshold ϵ needs to be set to a high value to accommodate the deviation of the inliers from the expected trend, thus producing also many false positives. It is conceivable that M-estimator will also suffer from the same weakness. However, in the proposed method, RANSAC with a loose inlier threshold is used to exclude obvious outliers, and the aim of using M-estimator is simply to capture a rough model of the inliers.

3.4 Epipolar Constraint Adjustment for Radial Distortion

First, assume there are no outliers in the data $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^{N}$. However, let there be radial distortion in the images such that the epipolar constraint is violated. Let $\mathbf{p}_*$ be an arbitrary position in the first image. Inspired by MLS surface approximation, adjust the epipolar constraint for $\mathbf{p}_*$ by calculating the local fundamental matrix

$$\mathbf{f}_* = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{i=1}^{N} (v_i^* \mathbf{a}_i \mathbf{f})^2, \tag{3.13}$$

subject to the usual norm constraint $\|\mathbf{f}\| = 1$. The non-stationary weights v_i^* are obtained as

$$v_i^* = \exp\left(-\|\mathbf{x}_i - \mathbf{p}_*\|^2/\sigma^2\right), \tag{3.14}$$

where σ is the neighbourhood scale. Intuitively, since the function (3.14) assigns higher weights to correspondences that are closer to $\mathbf{p}_*$, $\mathbf{f}_*$ adapts better to the local deviation (due to radial distortion) around $\mathbf{p}_*$. Further, if point $\mathbf{p}_*$ is moved continuously within the first image, a set of $\mathbf{f}_*$ that collectively define the epipolar geometry that is globally adjusted for radial distortion will be obtained.

For a given $\mathbf{p}_*$, (3.13) is again a weighted DLT problem, which can be rewritten into the matrix form

$$\mathbf{f}_* = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{V}^* \mathbf{A} \mathbf{f}\|^2 \tag{3.15}$$

where $\mathbf{V}^* = diag([v_1^*, v_2^*, \dots v_N^*])$. Again, the solution is the least significant right singular vector of $\mathbf{V}^*\mathbf{A}$. This estimation procedure is called MDLT. Zaragoza *et al.* [140] applied MDLT for estimating projective transforms. The work in this thesis is the first to apply MDLT for epipolar geometry. Moreover, in the following, a novel outlier identification scheme for images with radial distortion will be proposed; such an algorithm was not available in [140].

To deal with data contaminated with outliers, the proposed method incorporates the converged weights $\{w_i\}_{i=1}^{N}$ from M-estimator into MDLT. Specifically, (3.13) is modified to become

$$\mathbf{f}_* = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{i=1}^{N} (w_i v_i^* \mathbf{a}_i \mathbf{f})^2, \tag{3.16}$$

subject to the usual norm constraint $\|\mathbf{f}\| = 1$. Intuitively, the converged weights $\{w_i\}_{i=1}^N$ of M-estimator will globally reduce the influence of outliers, whereas the non-stationary weights $\{v_i^*\}_{i=1}^N$ will locally adapt the epipolar constraint to the deviation around $\mathbf{p}_*$. This simple combination creates a robust version of MDLT. The solution to (3.16) is simply the least significant right singular vector of $\mathbf{W}\mathbf{V}^*\mathbf{A}$.

3.5 Point Correspondence Validation Under Unknown Radial Distortion

By incorporating the converged weights from M-estimator into MDLT, the epipolar constraint is adjusted for radial distortion. The overall correspondence validation method is summarized in the Algorithm 3.1. MDLT gives more flexibility in the radial distortion model. However, there is a potentially higher risk of false positives by increasing the flexibility. To combat false positives, the key is to set the initial rough threshold ϵ correctly, so that obvious outliers will not be identified as inliers in the flexible fitting step. To this point, one may argue instead of following the algorithm pipeline in Figure 3.2, why do not put MDLT inside the RANSAC loop. However, if MDLT is applied on the minimal subset sampled in each iteration of RANSAC, no flexibility can be achieved since the model can be fitted on the minimal subset exactly.

Algorithm 3.1 Point correspondence validation method based on MDLT.

```
Require: Two overlapping input images, rough inlier threshold $\epsilon$, neighborhood scale
    $\sigma$, fine inlier threshold $\beta$.
Obtain a set of matching points $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ across the input images. Normalize the
    point coordinates following [59].
Invoke RANSAC to initialize $\mathbf{f}$. Refine $\mathbf{f}$ using IRLS until convergence. Save the
    converged weights $\{w_i\}_{i=1}^N$.
for $i = 1, \ldots, N$ do
    Set $\mathbf{p}_* := \mathbf{x}_i$ and compute weights $\{v_i^*\}_{i=1}^N$ from (3.14).
    Solve for $\mathbf{f}_*$ based on (3.16) using SVD.
    Calculate the Sampson distance

        $s_i = \frac{\tilde{\mathbf{x}}_i'^T \mathbf{F}_* \tilde{\mathbf{x}}_i}{(\mathbf{F}_*\tilde{\mathbf{x}}_i)_1^2 + (\mathbf{F}_*\tilde{\mathbf{x}}_i)_2^2 + (\mathbf{F}_*^T\tilde{\mathbf{x}}'_i)_1^2 + (\mathbf{F}_*^T\tilde{\mathbf{x}}'_i)_2^2}$    (3.17)

    where $\mathbf{F}_*$ is the $3 \times 3$ version of $\mathbf{f}_*$ and symbol $(\mathbf{a})_j$ indicates the $j$-th entry of
    vector $\mathbf{a}$.
    If $s_i \leq \beta$, then set $(\mathbf{x}_i, \mathbf{x}'_i)$ as inlier, else set it as outlier.
end for
```

3.6 Results

In this section, the proposed method (henceforth, MDLT) will be tested and benchmarked. Both synthetic and real image data will be used. The main aim of conducted experiments is to demonstrate the effectiveness of MDLT in dealing with radial distortion without requiring or assuming a particular distortion model. MDLT is compared with the following methods:

1. Fitzgibbon's [44] undistortion model;
2. Brito *et al.*'s [19, 20, 18] one-sided and two-sided radial fundamental matrices;
3. Kukelova *et al.*'s [70] method.

The methods of Fitzgibbon and Brito *et al.* are implemented in this thesis, while an implementation of Kukelova's method is available on the project website[2]. Another recent method by Barreto and Daniilidis [9] has also been studied. However, since their method uses the same lifting approach as Brito *et al.*, only the latest methods by Brito *et al.* are compared.

3.6.1 Synthetic data experiments

First, synthetic point correspondence data without outliers are generated to investigate the accuracy of MDLT in characterizing radial distortion. The process is summarized in Figure 3.3. 128 scene points were created and captured by two cameras. The 1st camera had fixed pose and intrinsics, while 1000 random poses and focal lengths were produced for the 2nd camera. This yielded 1000 sets of point correspondence data.

One-sided and two-sided radial distortion were then introduced to image points. The points were distorted following the traditional second-order distortion model [23]

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} x_u \\ y_u \end{pmatrix} + \lambda r^2 \left(\begin{pmatrix} x_u \\ y_u \end{pmatrix} - \begin{pmatrix} d_x \\ d_y \end{pmatrix} \right), \tag{3.18}$$

where $[x_d, y_d]^T$ is the distorted point, $[x_u, y_u]^T$ is the undistorted point, $[d_x, d_y]^T$ is the COD, $\lambda \in \mathbb{R}$ is the distortion coefficient, and $r^2 = \|(x_u, y_u)^T - (d_x, d_y)^T\|^2$. Figure 3.4

[2] http://www2.maths.lth.se/vision/downloads/

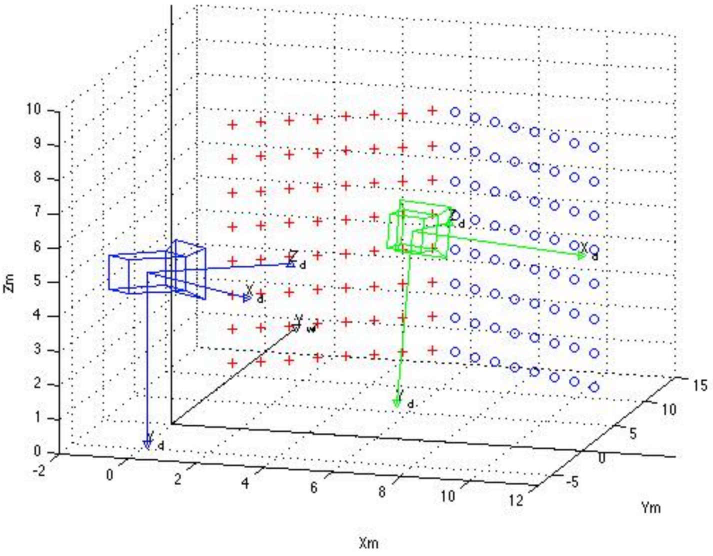
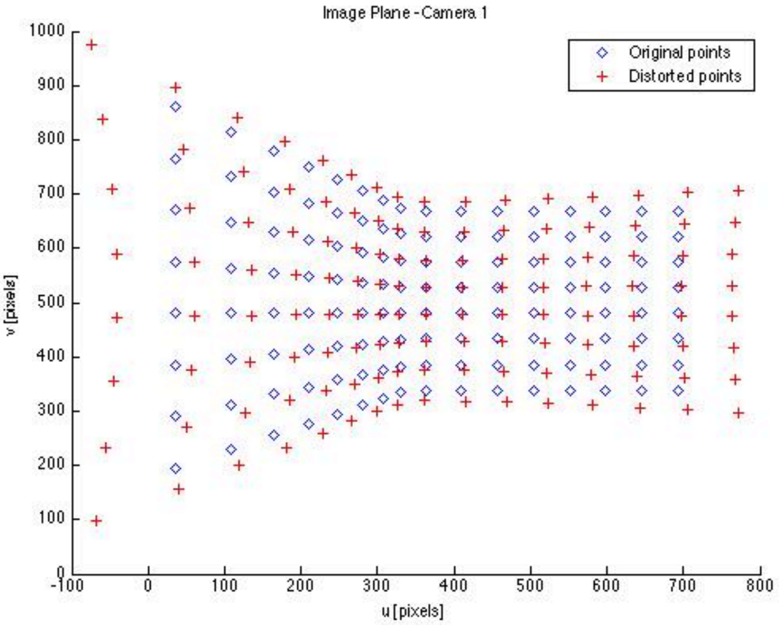


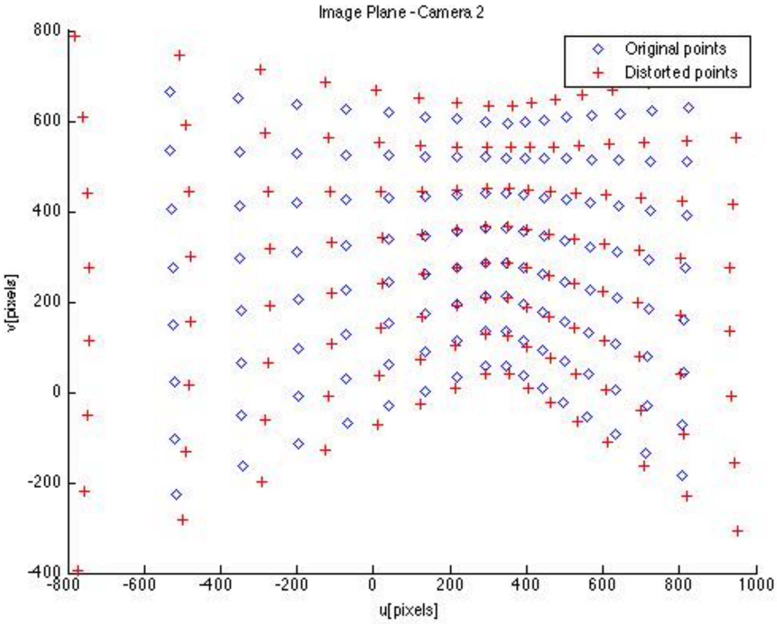
Figure 3.3: Scene and imaging setup for the synthetic data experiment. In total, 128 scene points are generated. The pose and intrinsics of the 1st camera (blue) is fixed, while the pose and focal length are randomly generated for the 2nd camera (green). The images captured using the cameras are then corrupted with varying degrees of radial distortion.

shows an example of two-sided radial distortion. A variant of distortion model used in Brito *et al.* [19, 20, 18] was also tested, where the only distinction is that $r^2 = \|(x_d, y_d)^T - (d_x, d_y)^T\|^2$ is now computed between the distorted point and the COD.

Two levels of distortion, specified by $\lambda = 0.01$ and 0.1, were tested. To further vary the difficulty of the data, (1) Gaussian noise with standard deviation in the range of $[0.0, 5.0]$ pixels was added, and (2) the COD was displaced to the right of the image centre by $d(width/2)$ pixels, where $width$ is the width of the image and d controls the actual displacement amount.



(a) Image from camera 1.



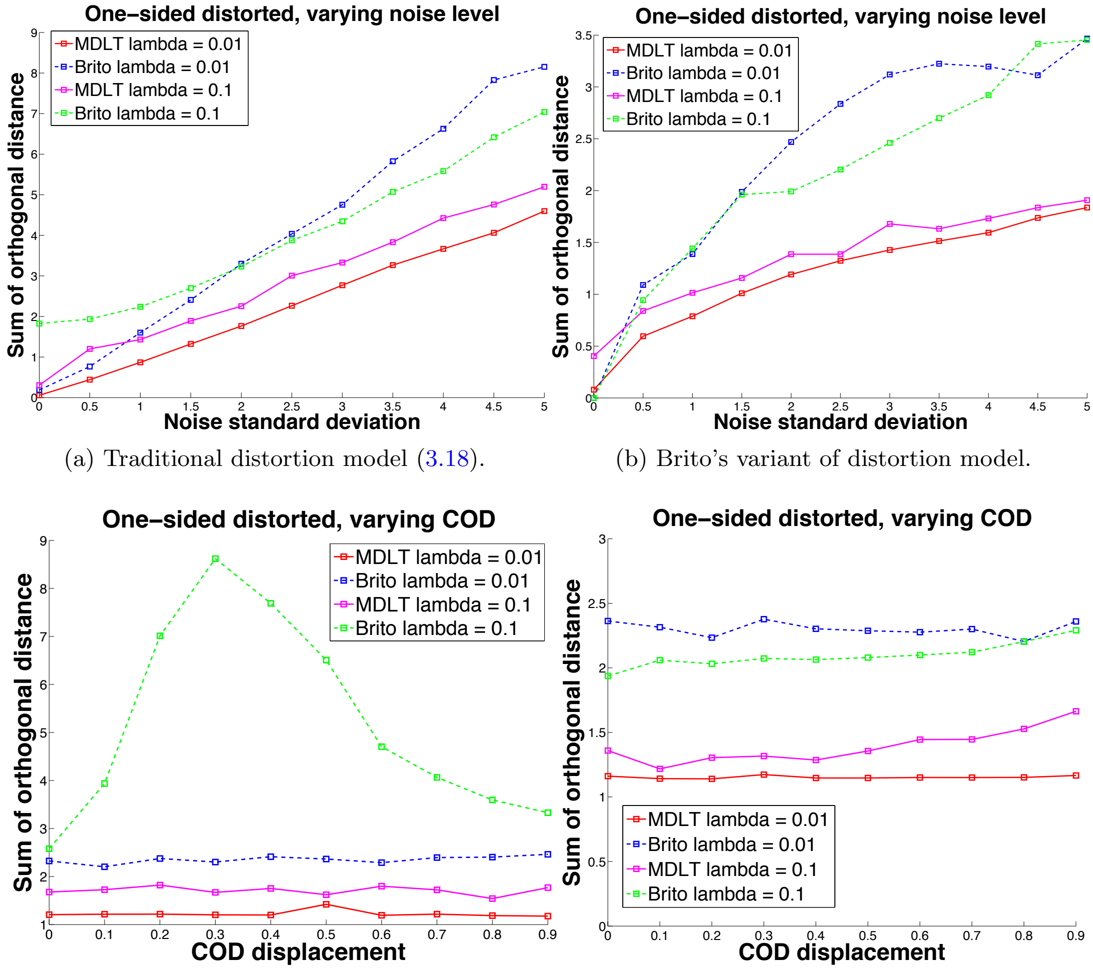
(b) Image from camera 2.

FIGURE 3.4: A sample data with two-sided radial distortion.

One-sided radial distortion. Here the proposed method is compared against Brito *et al.*'s [19, 20] one-sided radial fundamental matrix. The convention was to take the distorted image as the 1st image, and the undistorted image as the 2nd image. For MDLT, each point $\mathbf{x}_i$ in the distorted image was iteratively taken as $\mathbf{p}_*$, and the local fundamental matrix $\mathbf{f}_*$ was obtained as in (3.13). Using $\mathbf{f}_*$, $\mathbf{p}_*$ was then mapped to an epipolar line l_* in the 2nd image. The point-to-line distance (or orthogonal distance) between $\mathbf{x}'_i$ and l_* was obtained as the error measure. For Brito *et al.*, the one-sided radial fundamental matrix F_1 was estimated using all available correspondences simultaneously (since there are no outliers), then F_1 was used to compute the epipolar line (which is straight, since F_1 will map from the distorted to the undistorted image) for each $\mathbf{x}_i$. Again, the orthogonal distance between the epipolar line and the corresponding point was recorded. The results are shown in Figure 3.5.

From Figures 3.5a and 3.5b, it is obvious that as the Gaussian noise on the point coordinates increases, the error becomes larger for both methods. Given the same radial distortion magnitude λ, MDLT consistently has equal or lower error than the one-sided fundamental matrix. Interestingly, when Brito's variant of the distortion model is used to generate the radial distortion, and there is no noise on the point coordinates, it is possible for the one-sided fundamental matrix to achieve zero error. However, this good performance disappears when the traditional distortion model is used or when the Gaussian noise magnitude increases. This dependence on the underlying distortion model becomes more apparent in Figures 3.5c and 3.5d. Whilst the error of MDLT is largely stably regardless of the underlying distortion model, Brito's method appears to break down at certain ranges of the COD displacement when the traditional distortion model was used to distort the data.

Two-sided radial distortion The previous experiment was repeated with two-sided radial distortions. MDLT was compared with Brito *et al.*'s [19, 18] two-sided fundamental matrix, as well as Kulelova *et al.*'s [70] method. Since the epipolar line becomes an epipolar curve [142, 33] on an image with radial distortion, the orthogonal distance cannot be calculated as the error measure. Therefore, the Sampson distance for comparisons was used instead. The results are presented in Figure 3.6.



(a) Traditional distortion model (3.18).

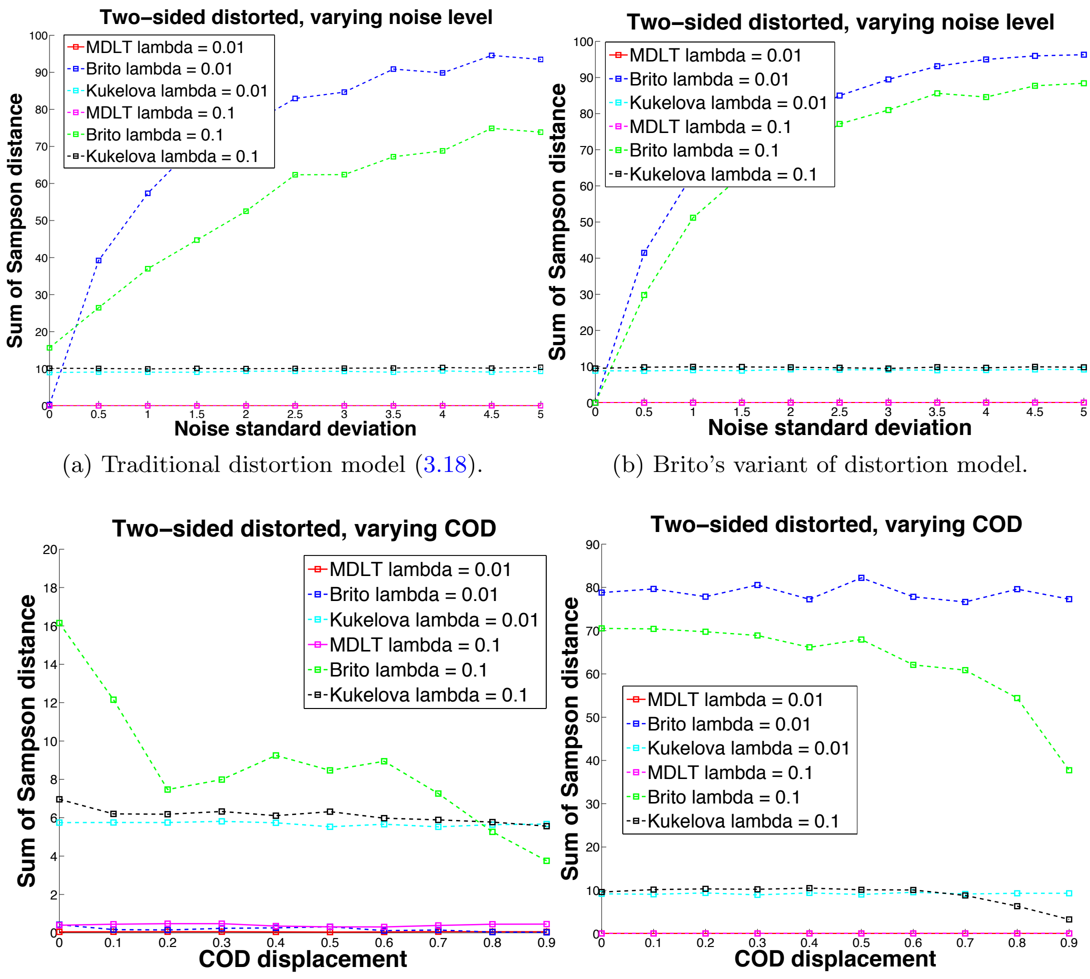
(b) Brito's variant of distortion model.

(c) Traditional distortion model (3.18).

(d) Brito's variant of distortion model.

FIGURE 3.5: Synthetic data experiment with one-sided radial distortion. (a)(b) The standard deviation of Gaussian noise is increased from 0.0 to 5.0, while the COD is placed at the image centre. (c)(d) The standard deviation of Gaussian noise is fixed at 2.0, while the COD is displaced by the amount controlled by d following the equation $d(width/2)$. Note that in (a)(c) the traditional distortion model (3.18) is used to distort the synthetic points, while in (b)(d) Brito's variant of the distortion model is used.

From Figures 3.6a and 3.6b, whilst all methods deteriorate when the Gaussian noise magnitude on the point coordinates increases, Brito's method appear to be highly affected by noise, i.e., the rate of increase in the Sampson error is much higher than that of MDLT or Kukelova's method. Under different COD displacements, as shown in Figures 3.6c and 3.6d, the performance of MDLT is the best among all methods and is also very consistent. Kukelova *et al.*'s method is also consistent, but not as accurate as the proposed approach. The performance of Brito *et al.*'s two-sided fundamental matrix varies tremendously depending on the level of distortion λ and amount of COD displacement. This possibly indicates a distortion model mismatch issue.



(a) Traditional distortion model (3.18).

(b) Brito's variant of distortion model.

(c) Traditional distortion model (3.18).

(d) Brito's variant of distortion model.

FIGURE 3.6: Synthetic data experiment with two-sided radial distortion. (a)(b) The standard deviation of Gaussian noise is increased from 0.0 to 5.0, while the COD is placed at the image centre. (c)(d) The standard deviation of Gaussian noise is fixed at 2.0, while the COD is displaced by the amount controlled by d following the equation $d(width/2)$. Note that in (a)(c) the traditional distortion model (3.18) is used to distort the synthetic points, while in (b)(d) Brito's variant of the distortion model is used.

3.6.2 Real image data experiments

Next, the performance of proposed point correspondence validation method based on MDLT, specifically Algorithm 3.1, was investigated. Publicly available images exhibiting various levels of radial distortion[3] were used; see Rows 1 and 2 in Table 3.1. Specifically, three classes of distortion are available (10%, 25% and 45%), and the distortion exists in both images. To extract and match interest points in the images in an automatic manner, the SIFT feature [93] implemented in the VLFeat package [131] was employed.

[3] http://arthronav.isr.uc.pt/ mlourenco/srdsift/dataset.html

The set of point correspondences produced by SIFT contain mismatches or outliers which were then identified using MDLT.

Algorithm 3.1 is compared against Brito's two-sided radial fundamental matrix [19, 18], Kukelova's method [70] and Fitzgibbon's undistortion model [44]. Previously these methods have been embedded in a RANSAC framework to enable robust estimation of epipolar geometry under radial distortion. The same operation has been performed here for outlier identification. For all methods, a correspondence is declared an inlier if the Sampson distance calculated based on the estimated fundamental matrix is less than the threshold β (the same β is used for all methods).

To compare the accuracy of the methods, the ground truth label of all the SIFT point correspondences are obtained through manual inspection. Given the labelling result of a particular method on a specific image pair, the following information is obtained:

1. N_t: true number of inlier correspondences in the data.
2. N_i: number of correspondences labelled as inliers.
3. N_c: number of correspondences correctly labelled as inliers.

The precision and recall rate of the labelling result are then calculated as

$$PR = 100\% \times \frac{N_c}{N_i}, \quad RR = 100\% \times \frac{N_c}{N_t}. \tag{3.19}$$

The results are presented in Table 3.1. It can be observed that at all distortion levels, the precision and recall rates of MDLT are higher than all the competitors. Further, as the distortion level is increased, the precision and recall rates of the competitors reduce much faster than that of MDLT. This points to the superior accuracy and robustness of MDLT towards different levels of radial distortion.

To simulate real life cases where the COD of the radial distortion may not placed at the image centre, following [19, 18] the images are cropped at appropriate subwindows. Specifically, the image is cropped in a manner such that the true COD is shifted horizontally to the right of the image centre; see Rows 1 and 2 in Table 3.2, where the CODs are marked with yellow crosses while the image centers are marked with red crosses. From the results in Table 3.2, again it can be seen that at all distortion levels, MDLT is able to obtained many more inliers (higher recall) while maintaining a smaller number of false positives (higher precision) compared to the competitors.

Failure cases have been observed when there are insufficient feature matches. In regions without sufficient feature matches, MDLT falls to standard epipolar geometry, thus loses its model flexibility.

3.7 Summary

In this chapter, a novel approach has been proposed to point correspondence validation under unknown radial distortions. The proposed method was inspired by MLS surface approximation, which was extended to enable epipolar geometry estimation. The new estimation procedure called MDLT adjusts the standard epipolar geometry model in a data-driven manner, such that radial distortions can be accounted for. This adjustment is also conducted without assuming any particular radial distortion model. The proposed algorithm is simply and involves nothing more than solving a sequence of linear least squares subproblems. Experimental results demonstrated that the proposed method has superior performance than state-of-the-art methods, in terms of high recall and precision rates for point correspondence validation.

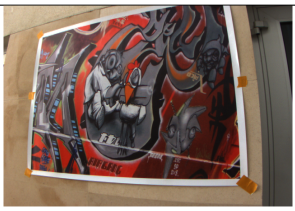
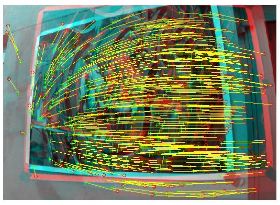
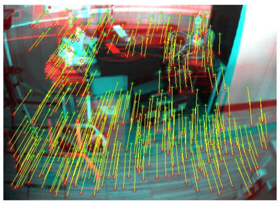
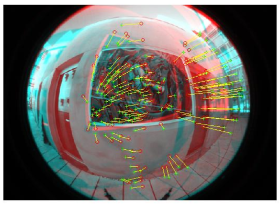
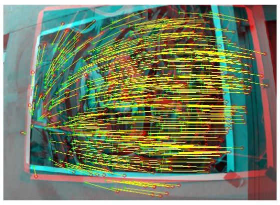
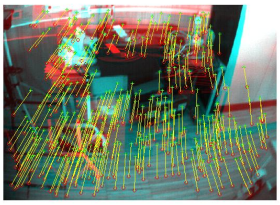
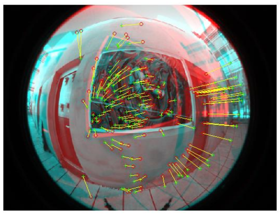
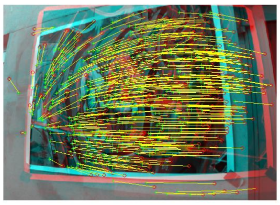
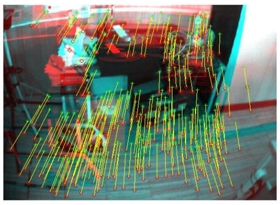
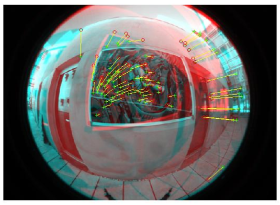
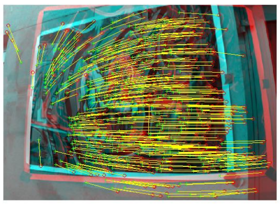
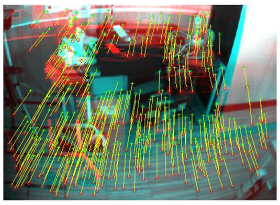
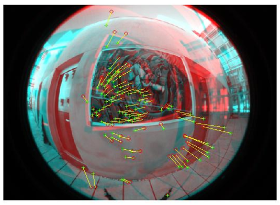
	RD 10%	RD 25%	RD 45%
Input image 1			
Input image 2			
MDLT	 $N_t = 577$ $N_i = 584$ $N_c = 574$ RR = 99.48% PR = 98.29%	 $N_t = 379$ $N_i = 381$ $N_c = 376$ RR = 99.21% PR = 98.69%	 $N_t = 165$ $N_i = 145$ $N_c = 141$ RR = 85.45% PR = 97.24%
Brito's [19, 18]	 $N_t = 577$ $N_i = 479$ $N_c = 474$ RR = 82.15% PR = 98.96%	 $N_t = 379$ $N_i = 363$ $N_c = 361$ RR = 95.25% PR = 99.45%	 $N_t = 165$ $N_i = 138$ $N_c = 136$ RR = 82.42% PR = 98.55%
Fitzgibbon's [44]	 $N_t = 577$ $N_i = 492$ $N_c = 489$ RR = 84.75% PR = 99.39%	 $N_t = 379$ $N_i = 300$ $N_c = 300$ RR = 79.16% PR = 99.99%	 $N_t = 165$ $N_i = 72$ $N_c = 71$ RR = 43.03% PR = 98.61%
Kukelova's [70]	 $N_t = 577$ $N_i = 479$ $N_c = 474$ RR = 82.15% PR = 98.96%	 $N_t = 379$ $N_i = 362$ $N_c = 361$ RR = 95.25% PR = 99.72%	 $N_t = 165$ $N_i = 91$ $N_c = 89$ RR = 53.94% PR = 97.80%

TABLE 3.1: Real image tests.

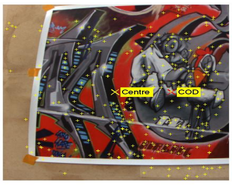
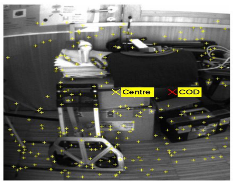
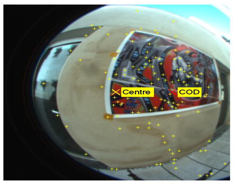
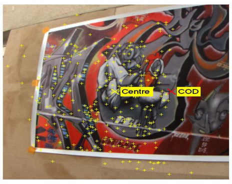
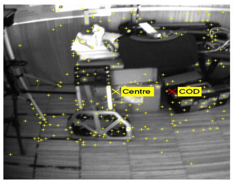
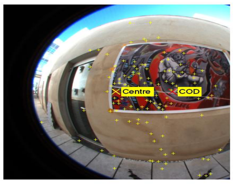
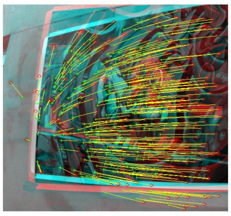
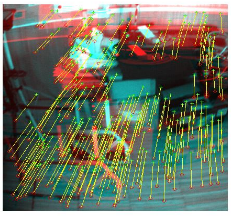
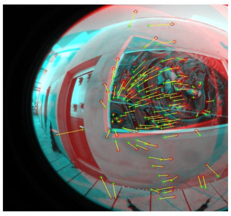
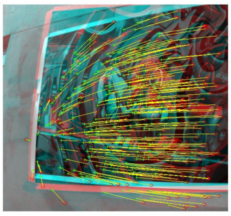
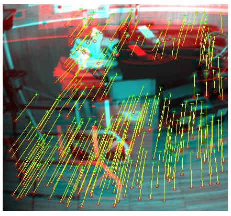
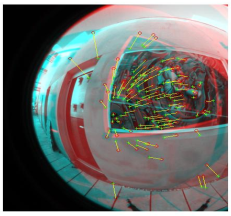
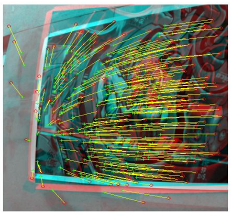
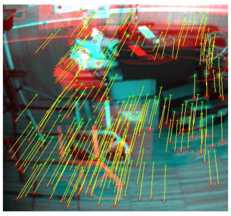
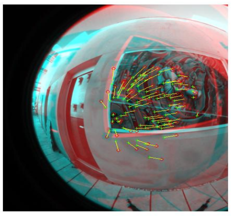
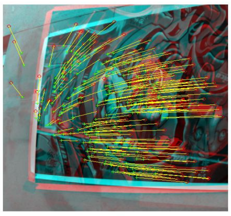
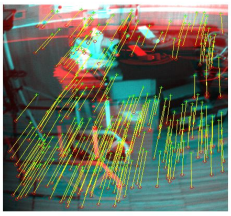
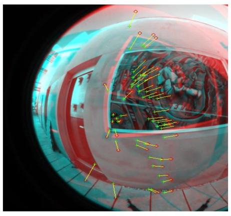
	RD 10%	RD 25%	RD 45%
Input image 1			
Input image 2			
MDLT	 $N_t = 312$ $N_i = 310$ $N_c = 301$ RR = 96.47% PR = 97.10%	 $N_t = 275$ $N_i = 275$ $N_c = 272$ RR = 98.55% PR = 98.91%	 $N_t = 119$ $N_i = 115$ $N_c = 108$ RR = 90.76% PR = 93.91%
Brito's [19, 18]	 $N_t = 312$ $N_i = 290$ $N_c = 287$ RR = 91.99% PR = 98.97%	 $N_t = 275$ $N_i = 263$ $N_c = 261$ RR = 94.91% PR = 99.24%	 $N_t = 119$ $N_i = 97$ $N_c = 97$ RR = 81.51% PR = 99.99%
Fitzgibbon's [44]	 $N_t = 312$ $N_i = 293$ $N_c = 292$ RR = 93.59% PR = 99.67%	 $N_t = 275$ $N_i = 234$ $N_c = 234$ RR = 85.09% PR = 99.99%	 $N_t = 119$ $N_i = 76$ $N_c = 76$ RR = 63.87% PR = 99.99%
Kukelova's [70]	 $N_t = 312$ $N_i = 292$ $N_c = 288$ RR = 92.31% PR = 98.63%	 $N_t = 275$ $N_i = 259$ $N_c = 259$ RR = 94.18% PR = 99.99%	 $N_t = 119$ $N_i = 54$ $N_c = 54$ RR = 45.38% PR = 99.99%

TABLE 3.2: Real image tests, with COD displacement 50%.

Chapter 4

Correspondence Insertion for APAP Image Stitching

4.1 Introduction

4.1.1 Basic homographic stitching

Given input images as shown in Figure 4.1a, the standard pipeline of image stitching [124, 25] begins with detecting and matching feature points across input images. A robust technique such as RANSAC is invoked to remove outliers and establish feature point correspondences; see Figure 4.1b. Based on feature correspondences, the projective transformation (i.e., homography) can be estimated.

Let $\mathbf{x} = [\ x \quad y\ ]^T$ and $\mathbf{x}' = [\ x' \quad y'\ ]^T$ be corresponding points across overlapping images I and I'. The homography $\mathbf{H}$ transforms $\mathbf{x}$ to $\mathbf{x}'$ following the relation

$$\tilde{\mathbf{x}}' \sim \mathbf{H}\tilde{\mathbf{x}} \Leftrightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{4.1}$$

where $\tilde{\mathbf{x}} = [\ x \quad y \quad 1\ ]^T$ is $\mathbf{x}$ in homogeneous coordinates, and $\sim$ indicates equality up to scale.

In inhomogeneous coordinates,

$$x' = \frac{xH_{11} + yH_{12} + H_{13}}{xH_{31} + yH_{32} + H_{33}}, \tag{4.2}$$

$$y' = \frac{xH_{21} + yH_{22} + H_{23}}{xH_{31} + yH_{32} + H_{33}}. \tag{4.3}$$

Rewrite (4.2) and (4.3) and obtain

$$-xH_{11} - yH_{12} - H_{13} + x'xH_{31} + x'yH_{32} + x'H_{33} = 0, \tag{4.4}$$

$$-xH_{21} - yH_{22} - H_{23} + y'xH_{31} + y'yH_{32} + y'H_{33} = 0, \tag{4.5}$$

which can be rearranged as

$$\mathbf{m}\mathbf{h} = \mathbf{0}, \tag{4.6}$$

where $\mathbf{m}$ contains monomials for $\{\mathbf{x}, \mathbf{x}'\}$ of the form

$$\mathbf{m} = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix}, \tag{4.7}$$

and $\mathbf{h} = [\ H_{11} \quad H_{12} \quad H_{13} \quad H_{21} \quad H_{22} \quad H_{23} \quad H_{31} \quad H_{32} \quad H_{33}]^T$.

Given n corresponding points, the following system of linear equations can be constructed

$$\mathbf{M}\mathbf{h} = \mathbf{0}, \tag{4.8}$$

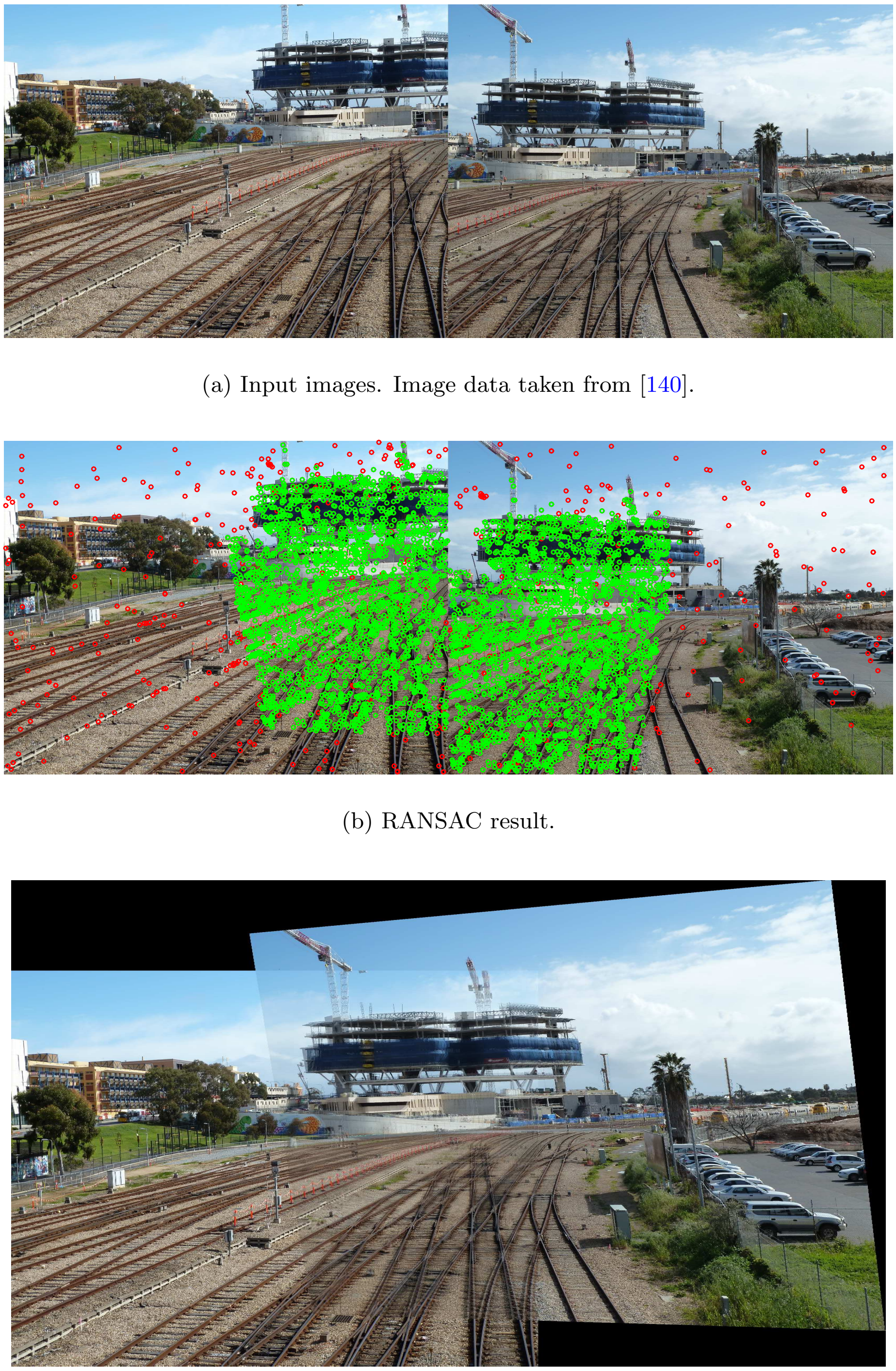
where

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}, \tag{4.9}$$

and $\mathbf{m}_i$ is (4.7) defined for the i-th point match $\{\mathbf{x}_i, \mathbf{x}'_i\}$. By imposing the constraint $\|\mathbf{h}\| = 1$, the problem is formulated as the least square problem

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmin}} \ \|\mathbf{M}\mathbf{h}\|^2. \tag{4.10}$$

The solution $\hat{\mathbf{h}}$ is the least significant right singular vector of $\mathbf{M}$.



(a) Input images. Image data taken from [140].

(b) RANSAC result.

(c) Warping result using single homography.

FIGURE 4.1: The standard pipeline of homographic stitching. (a) Feature points are detected and matched across the input images. (b) RANSAC is invoked to identify inliers (in green) from outliers (in red) and estimate the homography that brings the overlapping images into alignment. (c) The warping result is generated by compositing the aligned images onto a common canvas.

Given the estimated $\mathbf{H}$ (reconstructed from $\hat{\mathbf{h}}$), to align the images onto a common canvas, an arbitrary pixel $\mathbf{x}_*$ in the source image I is warped to the position $\mathbf{x}'_*$ in the target image I' by

$$\tilde{\mathbf{x}}'_* \sim \mathbf{H}\tilde{\mathbf{x}}_*. \tag{4.11}$$

However, image stitching with homographic warps carries the assumptions that the images were taken under pure rotational motions, or that the scene is sufficiently far away such that it is effectively planar. However, such conditions are unlikely to be satisfied in casual photography. As a result, misalignment effects or "ghosting" inevitably occur; see regions around cranes and railways in the homographic stitching result in Figure 4.1c.

4.1.2 As-projective-as-possible image stitching

Due to the limitations of homographic warps, spatially varying warps have been proposed as alternatives [86, 49, 140, 31]. Such warps can better account for the effects of parallax when aligning the overlap regions. In particular, as-projective-as-possible (APAP) warps [140] interpolate the data flexibly, while maintaining a global projective trend so as to extrapolate correctly.

The APAP warp maps each $\mathbf{x}_*$ using an input-dependent homography

$$\tilde{\mathbf{x}}'_* \sim \mathbf{H}_*\tilde{\mathbf{x}}_*, \tag{4.12}$$

where $\mathbf{H}_*$ is estimated from the weighted function

$$\mathbf{h}_* = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} \|w_*^i \mathbf{m}_i \mathbf{h}\|^2, \quad \text{s.t.} \quad \|\mathbf{h}\| = 1. \tag{4.13}$$

The non-stationary weights $\{w_*^i\}_{i=1}^N$ give higher importance to data that are closer to $\mathbf{x}_*$, and the weights are calculated as

$$w_*^i = \exp\left(-\|\mathbf{x}_* - \mathbf{x}_i\|^2/\sigma^2\right). \tag{4.14}$$

Here, σ is a scale parameter that controls the warp smoothness, and $\mathbf{x}_i$ is the point coordinate on the source image I from the i-th correspondence $\{\mathbf{x}_i, \mathbf{x}'_i\}$.

Compared to (4.11) which uses a single and global $\mathbf{H}$ for all $\mathbf{x}_*$, (4.14) assigns higher weights to data closer to $\mathbf{x}_*$. As a result, the projective warp $\mathbf{H}_*$ better respects the local structure around $\mathbf{x}_*$. Moreover, as $\mathbf{x}_*$ is moved continuously in its domain I, the warp $\mathbf{H}_*$ also varies smoothly, which is also why this method is called moving direct linear transformation (MDLT).

Equation (4.13) can be written in the matrix form

$$\mathbf{h}_* = \underset{\mathbf{h}}{\operatorname{argmin}} \left\| \mathbf{W}_* \mathbf{M} \mathbf{h} \right\|^2, \quad \text{s.t.} \ \ \|\mathbf{h}\| = 1, \tag{4.15}$$

where $\mathbf{W}_*$ is a $2N \times 2N$ diagonal matrix, and composed as

$$\mathbf{W}_* = diag([\ w_*^1 \quad w_*^1 \quad w_*^2 \quad w_*^2 \quad \ldots \quad w_*^N \quad w_*^N\ ]), \tag{4.16}$$

and $\mathbf{M}$ is the $2N \times 9$ from (4.9). This is a weighted linear squares problem, and the solution is simply the least significant right singular vector of $\mathbf{W}_*\mathbf{M}$.

However, solving (4.15) for each pixel position $\mathbf{x}_*$ on the source image I is unnecessarily wasteful, since neighboring positions will yield very similar weights (4.14) and hence very similar homographies. Thus, Zaragoza *et al.* uniformly partitioned I into a grid of $C_1 \times C_2$ cells, and only solved (4.15) for the center of each cell. Pixels within the same cell are then warped using the same homography. Based on [75], the C1 continuity of APAP warps can be established, but is out of the scope of this thesis. Figure 4.2 illustrates how the source image I is warped onto the target image I'. While, the APAP warping result without meshgrid is shown in Figure 4.3.

Ultimately, APAP warps are only as flexible as warranted by available feature matches. Without a sufficiently dense sampling of the underlying interpolant, the warp reduces to the baseline warp (projective [140]), thus defeating its spatially varying ability. A large number of feature matches are thus required to obtain good alignment, especially in areas with parallax where the true alignment function deviates from a simple homography. There is no guarantee, however, that feature matches are produced uniformly in the overlap area. Especially, SIFT output is sometimes unpredictable. As shown in Figure 4.4, the building on the right is covered by large glossy glass panels, which makes keypoint extraction/matching difficult. As a result, there are many points in the sky while no points on the building.

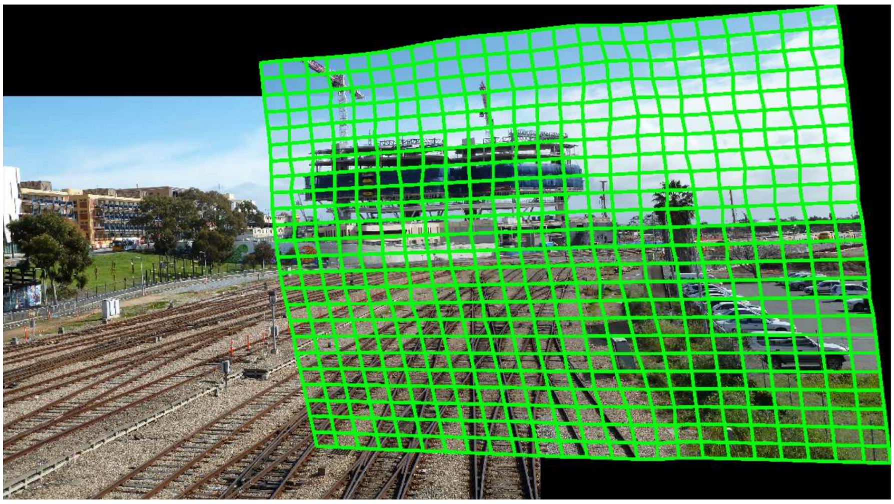


FIGURE 4.2: Aligned images with transformed cells overlaid to visualize the warp. Figure is taken from [140].

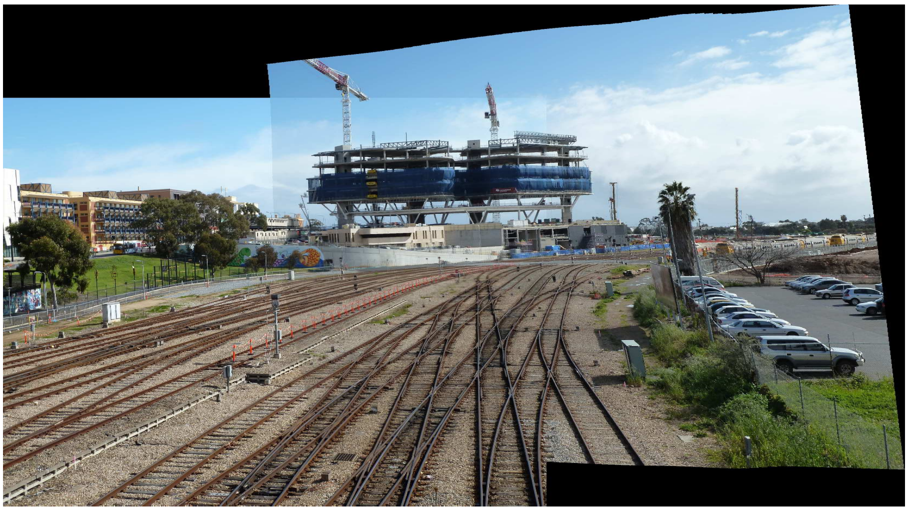
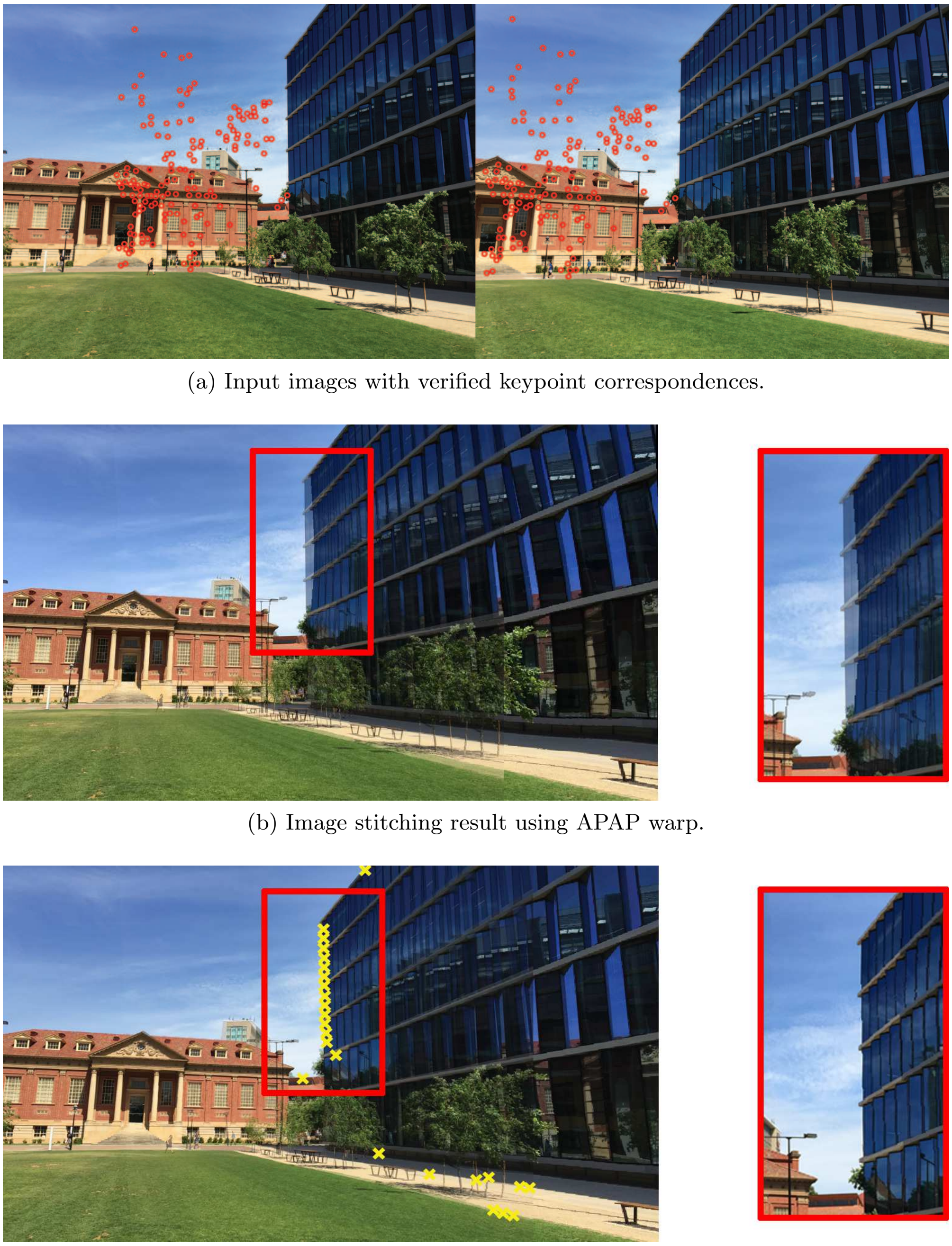


FIGURE 4.3: Result using APAP warp [140]

4.1.3 Chapter overview

Different from [141], this thesis attempts to accurately align the images throughout the overlap area before compositing. Specifically, in correspondence-poor regions, a correspondence insertion algorithm is proposed such that a good warping function can still be estimated. In the proposed method, correspondence search is accomplished for MDLT, which is the estimation method for APAP warps [140]. The simplicity of the proposed data-driven warp adaptation scheme over previous spline-based centre insertion techniques [30, 95, 11] has also been highlighted. On panoramic mosaicing problems that are challenging, the proposed approach achieves accurate alignment without being



(a) Input images with verified keypoint correspondences.

(b) Image stitching result using APAP warp.

(c) Result after automatically optimizing 25 new correspondences (indicated as yellow crosses) using the proposed method.

Figure 4.4: Overview of the proposed correspondence insertion method. (a) Two input images with verified keypoint correspondences; (b) Although APAP warp is spatially varying, without sufficient keypoint correspondences, the flexibility of the warp cannot be realised and the overlap area cannot be aligned well; (c) The proposed correspondence insertion algorithm automatically inserts and optimises new correspondences to improve the alignment.

handicapped by insufficient feature matches. Figure 4.4 gives a preview of the proposed method.

The rest of this chapter is organized as follows: Section 4.2 reviews previous work on centre insertion. Section 4.3 gives an overview of the proposed method. Section 4.4 introduces how a novel point is chosen and inserted. The location of the correspondence of the newly inserted center is optimized in Section 4.5. Then, Section 4.6 summarizes the overall data-driven warp adaptation scheme. Section 4.7 experimentally compares the proposed method with state-of-the-art stitching methods. Section 4.8 provides a summary.

4.2 Previous Work on Centre Insertion

Centre insertion has been studied extensively in spline regression [60, Chapter 5]. In particular, centre insertion has been proposed for pixel-based non-rigid object registration [30, 95, 11]. A 2D spline $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ is a function

$$f(\mathbf{x}) = \mathbf{A}^T\tilde{\mathbf{x}} + \sum_{k=1}^{K} \alpha_k \phi(\|\mathbf{x} - \mathbf{c}_k\|), \tag{4.17}$$

where $\mathbf{A} \in \mathbb{R}^{3\times 2}$ is an affine warp, $\tilde{\mathbf{x}} = [\mathbf{x}^T, 1]^T$ is $\mathbf{x}$ in augmented coordinates, $\{\alpha_k\}$ are scalar coefficients, $\{\mathbf{c}_k\}$ are 2D positions called centers, and ϕ is a radial basis function. The centers can be arbitrary (e.g., on a grid [125] over $\mathbb{R}^2$), and need not coincide with detected features.

The complexity of the warp increases with the number of centers K. If the pixels cannot be aligned well due to insufficient warp flexibility, one may consider adding new centers $\mathbf{c}_*$. Each insertion requires deciding where to place $\mathbf{c}_*$, and how to update the parameters $\{\mathbf{A}, \alpha_1, \ldots, \alpha_K, \alpha_*\}$. W.r.t. the latter, in [11] the Gauss-Newton algorithm is used to adjust the parameters to further minimize the intensity difference in the overlap area. Note that the spline parameters are not independent, e.g., the coefficients in thin plate spline (TPS) must satisfy the side condition $\sum_k \alpha_k = 0$. Thus, the updates get costlier as more centers are inserted.

Note that if $\mathbf{x}$ is sufficiently far away from all $\{\mathbf{c}_k\}$, the side condition and the monotonically decreasing radial basis function (RBF) ensures that $f(\mathbf{x})$ reduces to the affinity $\mathbf{A}$.

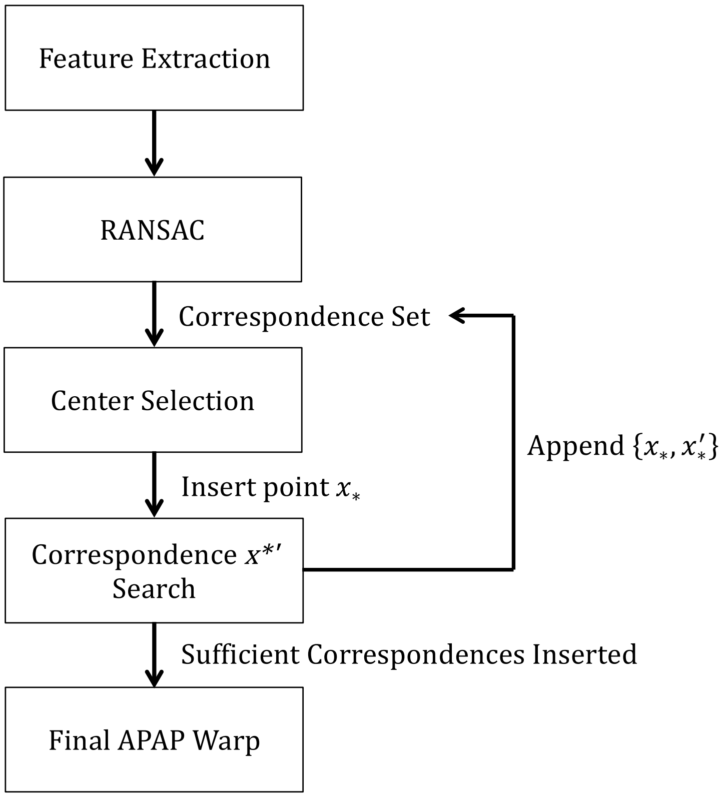


FIGURE 4.5: Correspondence insertion method overview.

This implies that splines are unsuitable for image stitching, since ideally the warp should revert to a homography in the extrapolation areas [140]. While there exist splines with a projective baseline [10], the fact remains that parameter updating can be relatively non-trivial. While the equivalent optimization on moving least squares (MLS) regression is much simpler and more efficient, which will be shown in the rest of this chapter.

4.3 Proposed Method Overview

The goal of this chapter is to find a warping function $f(\mathbf{x})$ that maps pixels from the source image I to the target image I'. The pipeline of the proposed method is given in Figure 4.5. Like most registration algorithms, the proposed method starts with feature point extraction; here, SIFT feature is used. RANSAC embedded with a standard homography (4.1) is then used to initialise the correspondence set. Thus, a set of point-wise matches $\mathcal{X} = \{\mathbf{x}_i, \mathbf{x}_i'\}_{i=1}^{N}$ are established across I and I', where $\mathbf{x}_i = [p_i, q_i]^T$ and $\mathbf{x}_i' = [p_i', q_i']^T$. The matches provide a sample of the true underlying warp, from which $f(\mathbf{x})$ is estimated. In regions where $\mathcal{X}$ undersamples the true warp (e.g., insufficient point matches), the accuracy of $f(\mathbf{x})$ in approximating the true warp is limited. The proposed method strives to construct a method to generate new correspondences $\{\mathbf{x}_*, \mathbf{x}'_*\}$ to improve $f(\mathbf{x})$, given that $f(\mathbf{x})$ is modeled as an APAP warp [140]. The step of center selection inserts a novel point $\mathbf{x}_*$ in the context of panoramic stitching. Then, the correspondence search algorithm is employed to optimize the corresponding point $\mathbf{x}'_*$ for the newly inserted $\mathbf{x}_*$. The above novel point selection and correspondence search process is repeated until the overlap area is sufficiently covered by correspondences.

The proposed method improves upon the original APAP warps by inserting correspondences in correspondence-poor regions. There are other guided feature matching methods [47][69] have been studied. However, [47] is introduced for the purpose of 3D reconstruction. Employing such complex 3D methods is a bit of "overkill" for the purpose of image stitching and only works for scene points in the overlapping area. The running time mentioned in [47] varies from 20 minutes to a few hours, which also backs up the claims. Despite employed 3D method and running time in [69], the underlying geometric model is affine transformation. While, homography is used in the proposed method, which is more suitable for the purpose of panoramic image stitching.

4.4 Center Selection

Given the current correspondence set $\mathcal{X} = \{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^{N}$, an APAP warp $f(\mathbf{x})$ (4.12) is first estimated and used to warp the source image I to align with the target image I'. Naturally correspondences should be added in regions with high alignment errors. This is provided by the absolute intensity difference map R. Since warp I is warped to align with I', it is natural to put R in the same frame as I'. Pixels with error less than ϵ (default ϵ is 100) are ignored by zeroing the corresponding values in R. See Figure 4.6a.

The proposed approach relies on seam cut [2] for pixel selection during compositing; see Figure 4.6b. Therefore, since pixels that will have their color copied (more appropriately, retained) from I', as marked in red in Figure 4.6b, are not subjected to misalignment errors, the corresponding values in R are zeroed. See Figure 4.6c.

Misalignments in regions with less structured textures (e.g., sky, trees, white board) are less obvious, thus it is less essential to introduce new correspondences in such locations. To realize this intuition, the visual saliency map of I' is computed using the method of [56]; see Figure 4.6d. Values in R corresponding to pixels with saliency less than η (default η is 0.5) are zeroed (recall that R has the same coordinate frame as I'); see Figure 4.6e.

At this stage, an error map is obtained to guide the insertion of new correspondences. Additional constraints are given by the existing correspondence set $\mathcal{X}$. Specifically, new correspondences should be inserted in regions that are not too near to $\mathcal{X}$, so as to avoid inserting redundant correspondences, and also not too far from $\mathcal{X}$, so as to ensure that correspondence search can be bootstrapped effectively by the existing $f(\mathbf{x})$. These constraints are realised by computing the distance transform D on the current set of features $\mathcal{L}$ in I'. Values of D that are less than ρ (default ρ is 15) are set to ∞; see Figure 4.6f.

Given D and R, the position $\mathbf{x}'_{\text{min}}$ that has the lowest value in $D./R$ is sought, where "./" indicates element-wise division. The new point $\mathbf{x}_*$ is then obtained as $f^{-1}(\mathbf{x}'_{\text{min}})$. To calculate the inverse APAP warp $f^{-1}(\mathbf{x}'_{\text{min}})$, the proposed technique finds the nearest neighbor of $\mathbf{x}'_{\text{min}}$ in $\{f(\mathbf{x}_i)\}_{i=1}^{N}$, then warps $\mathbf{x}'_{\text{min}}$ to I using the inverse $\mathbf{H}^{-1}(\mathbf{x})$ of the input-dependent homography (4.12) of the nearest neighbor point.

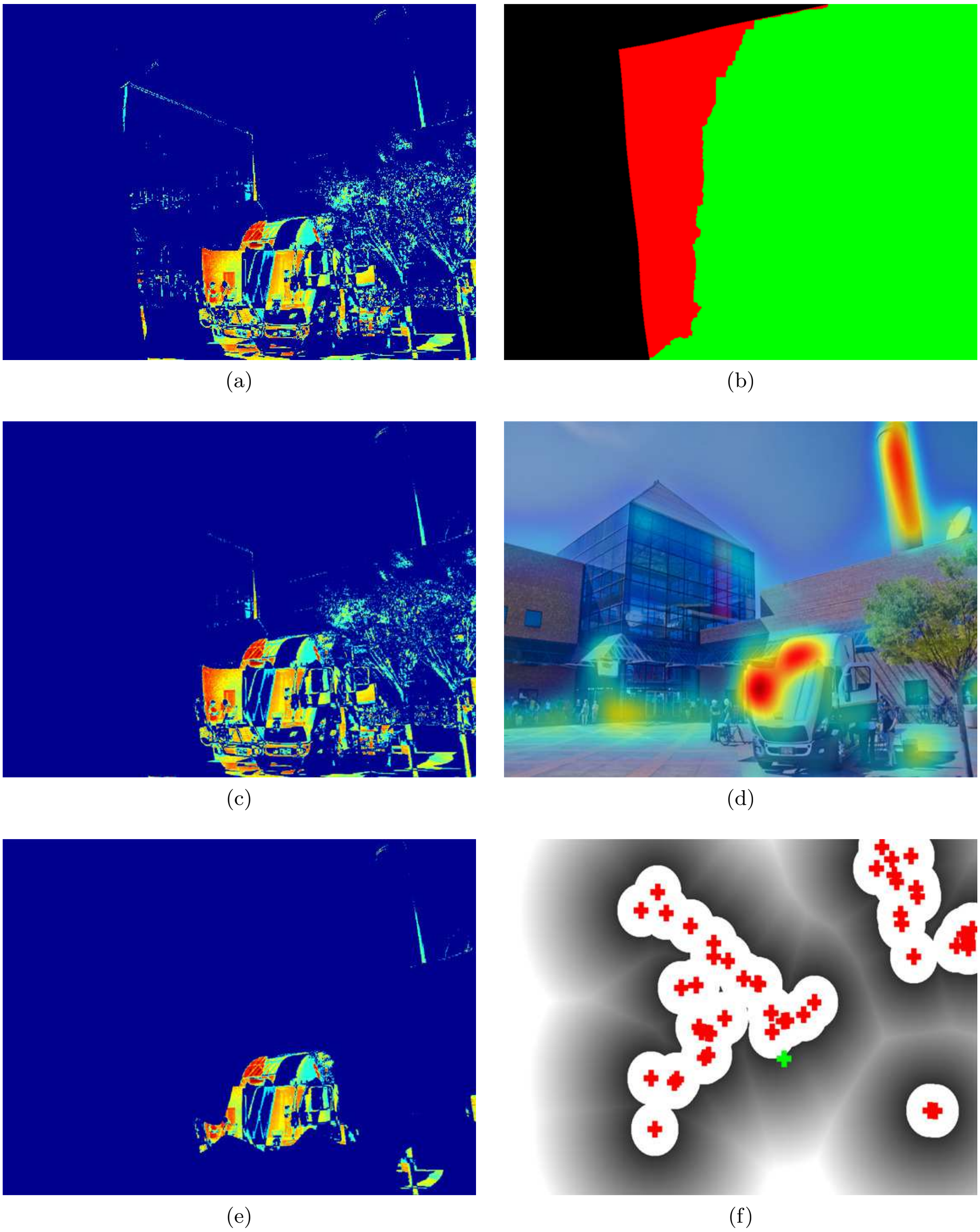


FIGURE 4.6: Center selection. (a) Absolute difference map R with values $< \epsilon$ are zeroed; (b) Optimized seam for the current $f(\mathbf{x})$; (c) Values in R corresponding to pixels selected from I' are zeroed; (d) Visual saliency map of I'; (e) Values in R corresponding to pixels with saliency $< \eta$ are zeroed; (f) Distance transform D on $\{f(\mathbf{x}_i)\}_{i=1}^{N}$ with values $< \rho$ are set to ∞ (darker areas here mean lower D values). Green cross indicates the $\mathbf{x}'_{\min}$ in this iteration.

4.5 Correspondence Search

Once the novel point $\mathbf{x}_*$ is initialised, correspondence search method is invoked to optimize the corresponding point $\mathbf{x}'_*$. Details are in the following.

4.5.1 Objective function and minimization

In regions with sparse correspondences, $\mathbf{x}$ is equally far (relative to σ) from all $\{\mathbf{x}_i\}_{i=1}^N$, and $f(\mathbf{x})$ reduces to a "rigid" projective warp, thus losing its spatially varying ability. Let $\mathbf{x}_*$ be the newly inserted point in I to raise the flexibility of $f(\mathbf{x})$. In the absence of geometric information, a matching point $\mathbf{x}'_*$ in I' is found based on pixel intensity values. To this end, the intensity matching cost is defined as

$$E(\mathbf{x}'_*) = \sum_{\mathbf{x}\in\mathbb{D}} \left[I'(f(\mathbf{x}|\mathbf{x}'_*)) - I(\mathbf{x})\right]^2, \tag{4.18}$$

where $I(\mathbf{x})$ is the pixel intensity at $\mathbf{x}$, $\mathbb{D}$ is a region in I (by default, $\mathbb{D}$ is a 31×31 subwindow), and the warp $f(\mathbf{x}|\mathbf{x}'_*)$ is now dependent on $\mathbf{x}'_*$. Specifically, the input dependent homography is now obtained as

$$\begin{aligned} \mathbf{h}(\mathbf{x}|\mathbf{x}'_*) = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} w_i(\mathbf{x})\|\mathbf{m}_i\mathbf{h}\|^2 + w_*(\mathbf{x})\|\mathbf{m}_*(\mathbf{x}'_*)\mathbf{h}\|^2, \\ \text{s.t. } \|\mathbf{h}\| = 1, \end{aligned} \tag{4.19}$$

where $\mathbf{m}_*(\mathbf{x}'_*)$ contains the monomials for $\{\mathbf{x}_*, \mathbf{x}'_*\}$ following (4.7), and

$$w_*(\mathbf{x}) = \exp\left(-\|\mathbf{x} - \mathbf{x}_*\|^2/\sigma^2\right). \tag{4.20}$$

In matrix form, (4.19) can be rewritten as

$$\begin{aligned} \mathbf{h}(\mathbf{x}|\mathbf{x}'_*) = \underset{\mathbf{h}}{\operatorname{argmin}} \left\|\mathbf{W}_*(\mathbf{x})\mathbf{M}(\mathbf{x}'_*)\mathbf{h}\right\|^2, \\ \text{s.t. } \|\mathbf{h}\| = 1, \end{aligned} \tag{4.21}$$

where $\mathbf{W}_*(\mathbf{x})$ is $\mathbf{W}_*$ diagonally extended with two $w_*(\mathbf{x})$ values, and $\mathbf{M}(\mathbf{x}'_*)$ is $\mathbf{M}$ vertically appended with $\mathbf{m}_*(\mathbf{x}'_*)$. Note that only $\mathbf{M}(\mathbf{x}'_*)$ contains the variable $\mathbf{x}'_*$.

Since the aim is to find $\mathbf{x}'_*$ by minimizing (4.18), the well-known Lucas-Kanade (LK) technique [7] is applied. A first-order Taylor expansion is applied on $E(\mathbf{x}'_* + \Delta\mathbf{x}'_*)$ to yield

$$\sum_{\mathbf{x}\in\mathbb{D}} \left[I'(f(\mathbf{x}|\mathbf{x}'_*)) + \nabla I'(f(\mathbf{x}|\mathbf{x}'_*)) \frac{\partial f(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*} \Delta\mathbf{x}'_* - I(\mathbf{x}) \right], \tag{4.22}$$

where image gradient $\nabla I'$ is computed using finite differencing. Differentiating against $\mathbf{x}'_*$ and equating to 0 yields

$$\Delta\mathbf{x}'_* = \mathbf{F}^{-1} \sum_{\mathbf{x}\in\mathbb{D}} \left[\nabla I'(f(\mathbf{x}|\mathbf{x}'_*)) \frac{\partial f(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*} \right]^T \left[I(\mathbf{x}) - I'(f(\mathbf{x}|\mathbf{x}'_*)) \right] \tag{4.23}$$

where $\mathbf{F}$ is the (approximated) Hessian

$$\mathbf{F} = \sum_{\mathbf{x}\in\mathbb{D}} \left[\nabla I'(f(\mathbf{x}|\mathbf{x}'_*)) \frac{\partial f(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*} \right]^T \left[\nabla I'(f(\mathbf{x}|\mathbf{x}'_*)) \frac{\partial f(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*} \right].$$

The current value for $\mathbf{x}'_*$ is then updated by $\Delta\mathbf{x}'_*$, and the steps are repeated until convergence. Refer to [7] for more details.

$\mathbf{x}'_*$ is initialized by mapping $\mathbf{x}_*$ with the $f(\mathbf{x})$ prior to correspondence insertion. It is crucial to note that $\mathbf{h}(\mathbf{x}|\mathbf{x}'_*)$ changes for different $\mathbf{x} \in \mathbb{D}$. Essentially a unique homography is estimated for each $\mathbf{x} \in \mathbb{D}$ given $\mathbf{x}'_*$, thus realizing a spatially varying warp. This differs from the standard LK approach for "frame global" projective registration [7].

Brightness constancy assumption. The objective function (4.18) assumes brightness constancy. In another word, this means corresponding pixels across input images have the same color/brightness. This assumption may not hold in general, especially if the images are taken using cameras with various color auto-correction routines. To ensure the applicability of the proposed method, color normalization techniques can be applied on the input images prior to stitching [138].

4.5.2 Jacobian of APAP warp

Evaluating $f(\mathbf{x}|\mathbf{x}'_*)$ and its Jacobian requires solving the weighted algebraic least squares problem (4.21) at each iteration - again, this differs from the common types of parametric motions used in LK [7]. Specifically, the solution $\mathbf{h}(\mathbf{x}|\mathbf{x}'_*)$ to (4.21) is the least significant eigenvector of

$$\mathbf{S}(\mathbf{x}|\mathbf{x}'_*) := [\mathbf{W}_*(\mathbf{x})\mathbf{M}(\mathbf{x}'_*)]^T[\mathbf{W}_*(\mathbf{x})\mathbf{M}(\mathbf{x}'_*)], \tag{4.24}$$

where $\mathbf{S}(\mathbf{x}|\mathbf{x}'_*)$ varies with $\mathbf{x}'_*$. The eigenvector satisfies

$$\left[\mathbf{S}(\mathbf{x}|\mathbf{x}'_*) - \lambda(\mathbf{x}|\mathbf{x}'_*)\right]\mathbf{h}(\mathbf{x}|\mathbf{x}'_*) = 0, \tag{4.25}$$

$$\|\mathbf{h}(\mathbf{x}|\mathbf{x}'_*)\| = 1, \tag{4.26}$$

where $\lambda(\mathbf{x}|\mathbf{x}'_*)$ is the eigenvalue. Via the chain rule,

$$\frac{\partial f(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*} = \frac{\partial f(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{h}(\mathbf{x}|\mathbf{x}'_*)}\frac{\partial \mathbf{h}(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*}. \tag{4.27}$$

The first term can be obtained by differentiating (4.12). The second term requires differentiating the eigenvector. Based on known results [94], the following expression

$$\frac{\partial \mathbf{h}(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*} = \left[\lambda(\mathbf{x}|\mathbf{x}'_*)\mathbf{I} - \mathbf{S}(\mathbf{x}|\mathbf{x}'_*)\right]^\dagger \frac{\partial \mathbf{S}(\mathbf{x}|\mathbf{x}'_*)}{\partial \mathbf{x}'_*}\mathbf{h}(\mathbf{x}|\mathbf{x}'_*) \tag{4.28}$$

can be derived, where $\mathbf{I}$ is the identity matrix. The derivative of $\mathbf{S}(\mathbf{x}|\mathbf{x}'_*)$ can in turn be obtained based on (4.24). Note that only the last-two rows of $\mathbf{M}(\mathbf{x}'_*)$ depend on $\mathbf{x}'_*$.

The proposed correspondence search procedure is summarized in Algorithm 4.1. Note that in Step 4, the eigenvector $\mathbf{h}(\mathbf{x}|\mathbf{x}'_*)$ for each $\mathbf{x} \in \mathbb{D}$ needs to be calculated. Using modern linear algebra packages, this does not represent significant computational load, even for large $\mathbf{S}(\mathbf{x}|\mathbf{x}'_*)$, e.g., 1000 × 1000. Moreover, an incremental decomposition scheme [140] can be used to further reduce computational cost.

Algorithm 4.1 Correspondence search for APAP warp.

```
Require: Images $I$ and $I'$, feature matches $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$, novel point $\mathbf{x}_*$.
1: Initialize $\mathbf{x}'_*$ by warping $\mathbf{x}_*$ using (4.12).
2: repeat
3:    for each $\mathbf{x} \in \mathbb{D}$ do
4:       Solve (4.21) to obtain $\mathbf{h}(\mathbf{x}|\mathbf{x}'_*)$ and $\lambda(\mathbf{x}|\mathbf{x}'_*)$.
5:       Calculate transformation $f(\mathbf{x}|\mathbf{x}'_*)$.
6:       Calculate warp Jacobian (4.27) for $\mathbf{x}$.
7:    end for
8:    Calculate $\Delta\mathbf{x}'_*$ (4.23) and update $\mathbf{x}'_* \leftarrow \mathbf{x}'_* + \Delta\mathbf{x}'_*$.
9: until $\mathbf{x}'_*$ converges.
```

Comparison against center insertion for splines. At this juncture, it is instructive to compare the proposed correspondence insertion algorithm with spline-based center insertion techniques (Section 4.2). The proposed warp update algorithm involves nothing more than searching for a 2D point $\mathbf{x}'_*$. This is a direct consequence of using "point set surfaces" [3] to define the warp. Contrast this to spline-based center insertion schemes, where all the warp parameters $\{\mathbf{A}, \alpha_1, \ldots, \alpha_K, \alpha_*\}$ need to be adjusted in each update.

4.6 Data-driven Warp Adaptation Scheme

Given the novel point $\mathbf{x}_*$, Algorithm 4.1 is invoked to find its correspondence $\mathbf{x}'_*$. The newly inserted correspondence $\{\mathbf{x}_*, \mathbf{x}'_*\}$ is appended to $\mathcal{X}$, if $E(\mathbf{x}'_*)$ is less than ω (default ω is 1000). Else, the new correspondence is considered unsatisfactory and discarded. In any case, $\mathbf{x}'_{\text{min}}$ is appended to $\mathcal{L}$ to prevent it from being selected again in the next iteration. The novel point selection and correspondence search steps are encapsulated in a data-driven warp adaption scheme, which iteratively inserts new correspondences until sufficient "coverage" of the overlap area is achieved. The data-driven warp adaptation scheme is summarized in Algorithm 4.2.

As an indication of runtime, invoking Algorithm 4.2 on the image pair in Figure 4.7 inserted 81 new correspondences in 65 seconds, among which 11 correspondences were accepted.

Algorithm 4.2 Data-driven warp adaptation.

```
Require: Input images $I$ and $I'$, initial correspondence set $\mathcal{X} = \{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^{N}$, error threshold $\epsilon$, saliency threshold $\eta$, distance threshold $\rho$, and acceptance threshold $\omega$.
$\mathcal{L} \leftarrow \{\mathbf{x}'_i\}_{i=1}^{N}$.
Compute visual saliency map on $I'$; see Figure 4.6d.
loop
    Estimate APAP warp $f(\mathbf{x})$ from $\mathcal{X}$.
    Warp $I$ to align with $I'$ using $f(\mathbf{x})$.
    $R \leftarrow$ absolute intensity difference map in overlap area.
    Set values in $R$ which are $< \epsilon$ to 0; see Figure 4.6a.
    Optimize seam [2] for pixel selection in overlap area; see Figure 4.6b.
    Set values in $R$ corresponding to pixels selected from $I'$ according to the seam to 0; see Figure 4.6c.
    Set values in $R$ corresponding to pixels of $I'$ with saliency $< \eta$ to 0; see Figure 4.6e.

    $D \leftarrow$ distance transform on $\mathcal{L}$ in the overlap area.
    Set values in $D$ that are $< \rho$ to $\infty$; see Figure 4.6f.
    If $D./R$ is all $\infty$, then break.
    $\mathbf{x}'_{\text{min}} \leftarrow$ location in $D./R$ with minimum value.
    $\mathbf{x}_* \leftarrow f^{-1}(\mathbf{x}'_{\text{min}})$.
    $\mathbf{x}'_* \leftarrow$ optimized correspondence from Algorithm 4.1.
    if $E(\mathbf{x}'_*) < \omega$ then
        $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathbf{x}_*, \mathbf{x}'_*\}$.
    end if
    $\mathcal{L} \leftarrow \mathcal{L} \cup \mathbf{x}'_{\text{min}}$.
end loop
```

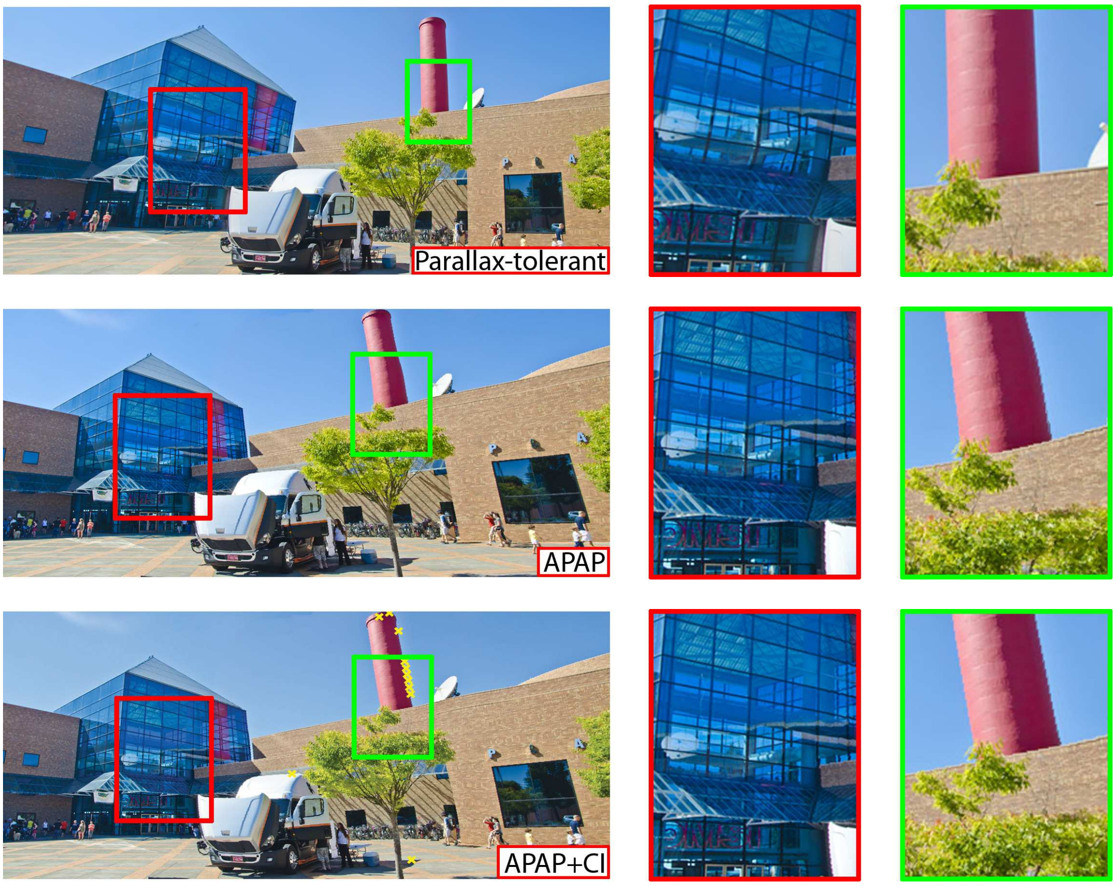


FIGURE 4.7: Comparing three methods on *truck* image pair. Inserted correspondences by APAP+CI are shown as yellow points.

4.7 Results

4.7.1 Comparisons with state-of-the-art stitching methods

In this section, results are shown using images of a scene with significant depth parallax. The proposed method is compared (abbreviated as APAP+CI) against other state-of-the-art approaches, namely the original APAP method [140] and parallax-tolerant image stitching [141]. Publicly available images by Zaragoza *et al.* and Zhang and Liu, as well as additional images collected in this thesis, are used. Most of the tested images were taken under camera poses which give rise to significant depth parallax.

For APAP warps, the code shared by Zaragoza *et al.* was used. For parallax-tolerant image stitching, this section simply reprinted the results (where available) from the project page of Zhang and Liu. For newly collected images, this section executed the thesis author's implementation of Zhang and Liu's method.

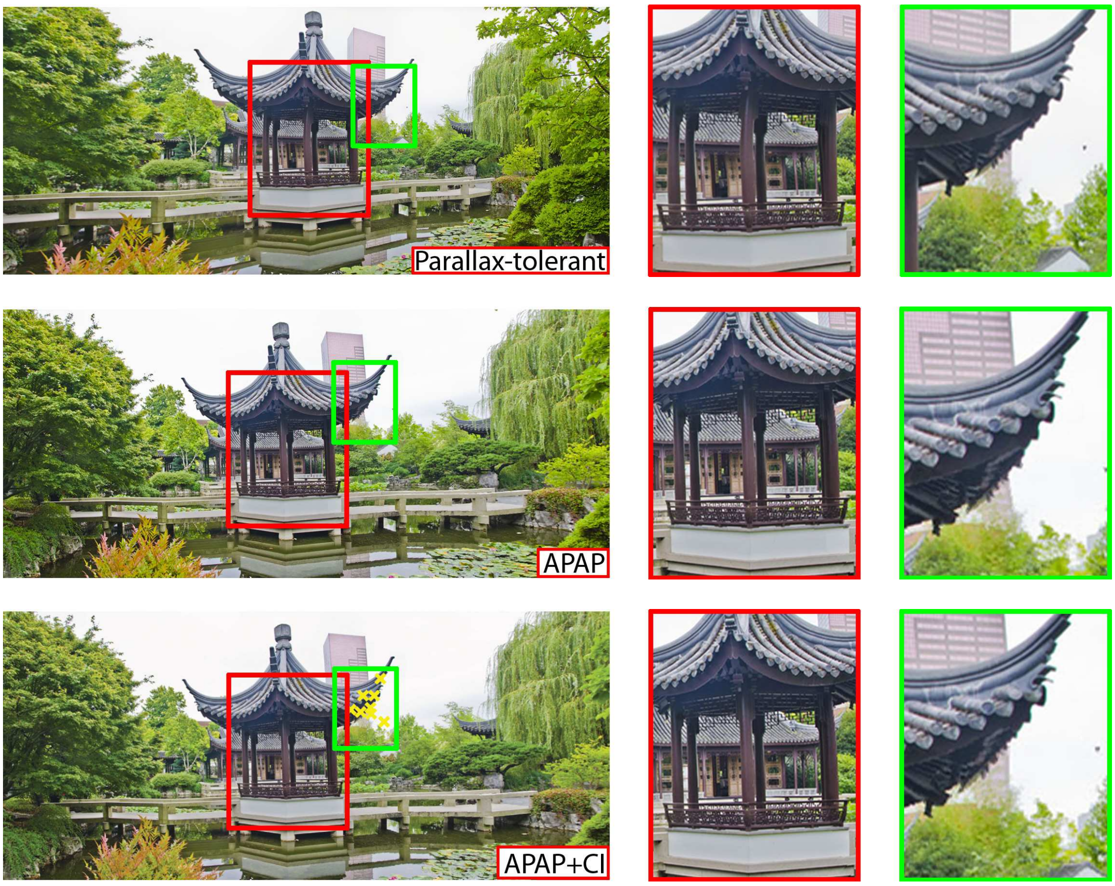


FIGURE 4.8: Comparing three methods on *temple* image pair. Inserted correspondences by APAP+CI are shown as yellow points.

Parameter settings for the proposed method are as follows: $\sigma = 8$ in (4.14), $\mathbb{D}$ in (4.18) is a 31×31 subwindow, $\epsilon = 100$, $\eta = 0.5$, $\rho = 15$, and $\omega = 1000$ in Algorithm 4.2.

In image pairs with very serious depth parallax, not all pixels have valid correspondences in the other view. Theoretically, the true warping function must "fold over" or be discontinuous to correctly align the images. Such characteristics are not supported by APAP or content preserving warps (CPW) [86] used in parallax-tolerant image stitching. Following Zhang and Liu, seam cut is employed to composite the images and remove ghosting.

Figures 4.7 and 4.8 show results on two image pairs used by Zhang and Liu. In Figure 4.7, parallax-tolerant image stitching produced significant distortions on the glass building. This was likely due to the concentration of the local homography on the major building to the right, and neglecting the other regions not lying on the same plane (cf. Figure 2.4). In contrast, APAP warp was more capable of globally aligning the images; notice that the glass building was not distorted. However, unpleasant distortions exist around the

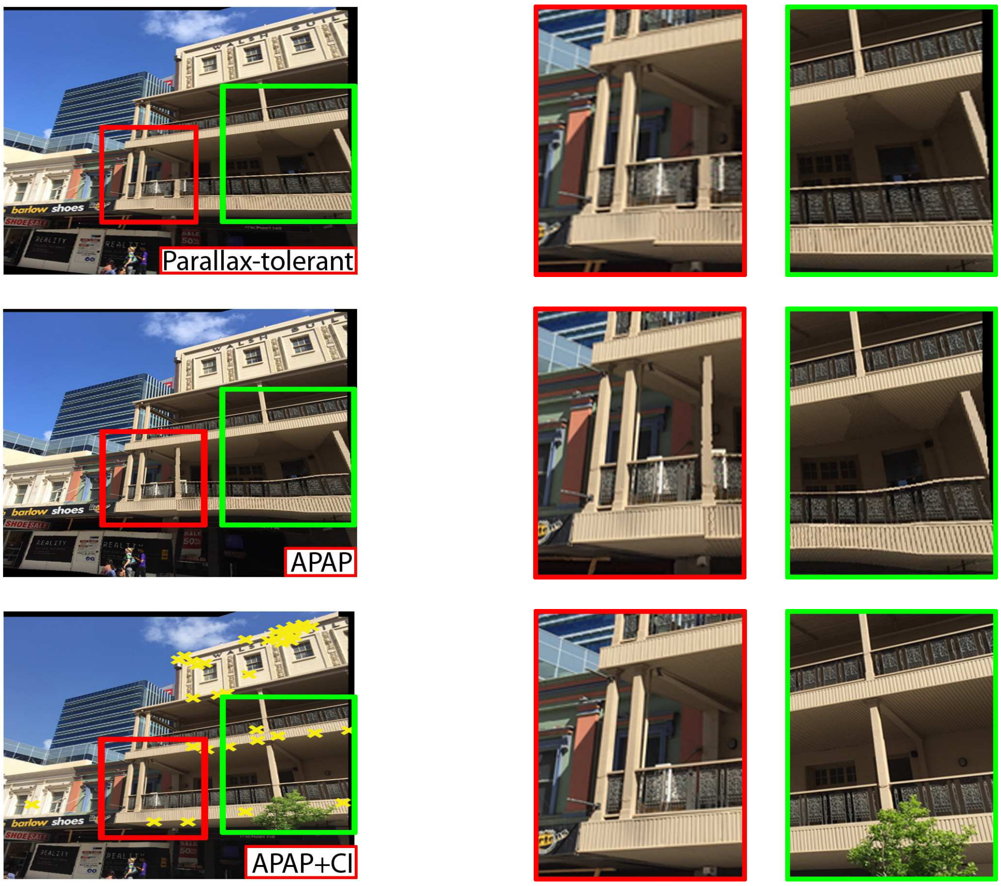


FIGURE 4.9: Comparing three methods on *shopfront* image pair. Inserted correspondences by APAP+CI are shown as yellow points.

smokestack - due to a lack of feature matches in this region, the warp was "dragged away" by existing feature matches on the lower building. The proposed method APAP+CI rectified the distortion by inserting new correspondences in the appropriate positions. In Figure 4.8, observe the distortions on the pavilion produced by parallax-tolerant image stitching. Overall, APAP warp accurately aligned the whole image, however, due to the lack of feature correspondences, the tower in the background appeared discontinuous. This was rectified by APAP+CI with the insertion of new correspondences.

Similar results on two more challenging image pairs are shown in Figures 4.9 and 4.10; these are newly collected data. Results show that by inserting new correspondences to adapt the warp, the proposed method rectifies the weakness of APAP warp.

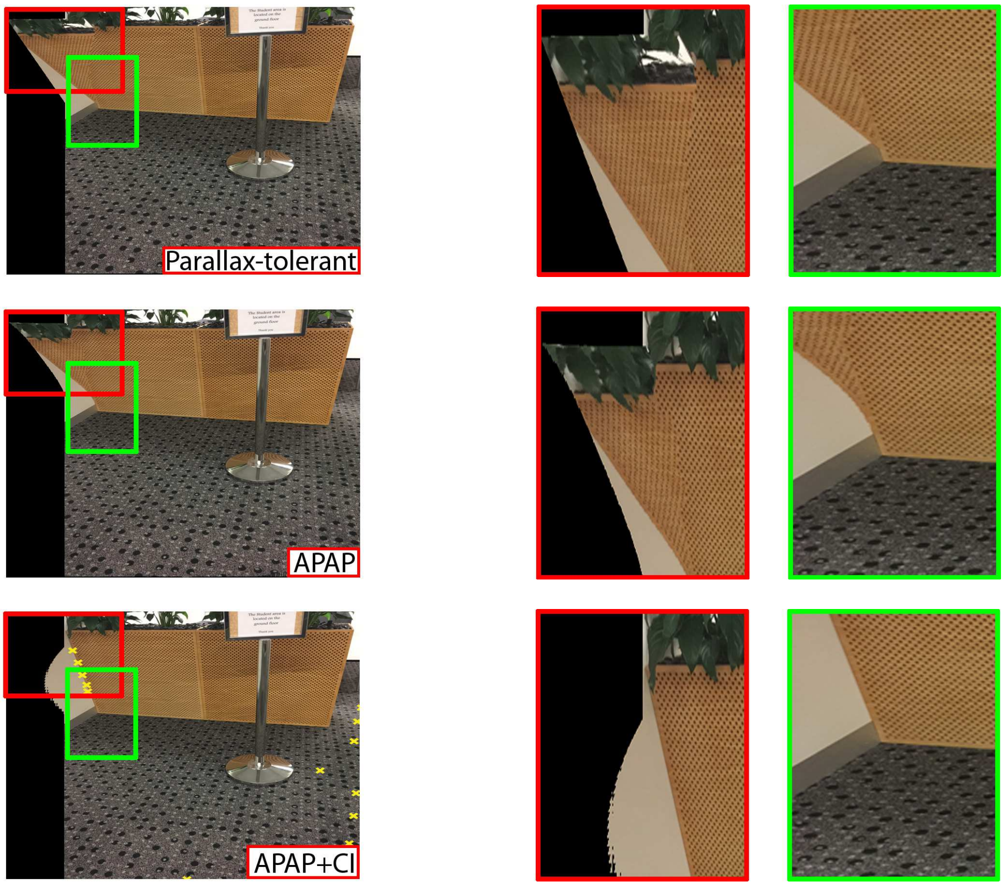


FIGURE 4.10: Comparing three methods on *lobby* image pair. Inserted correspondences by APAP+CI are shown as yellow points.

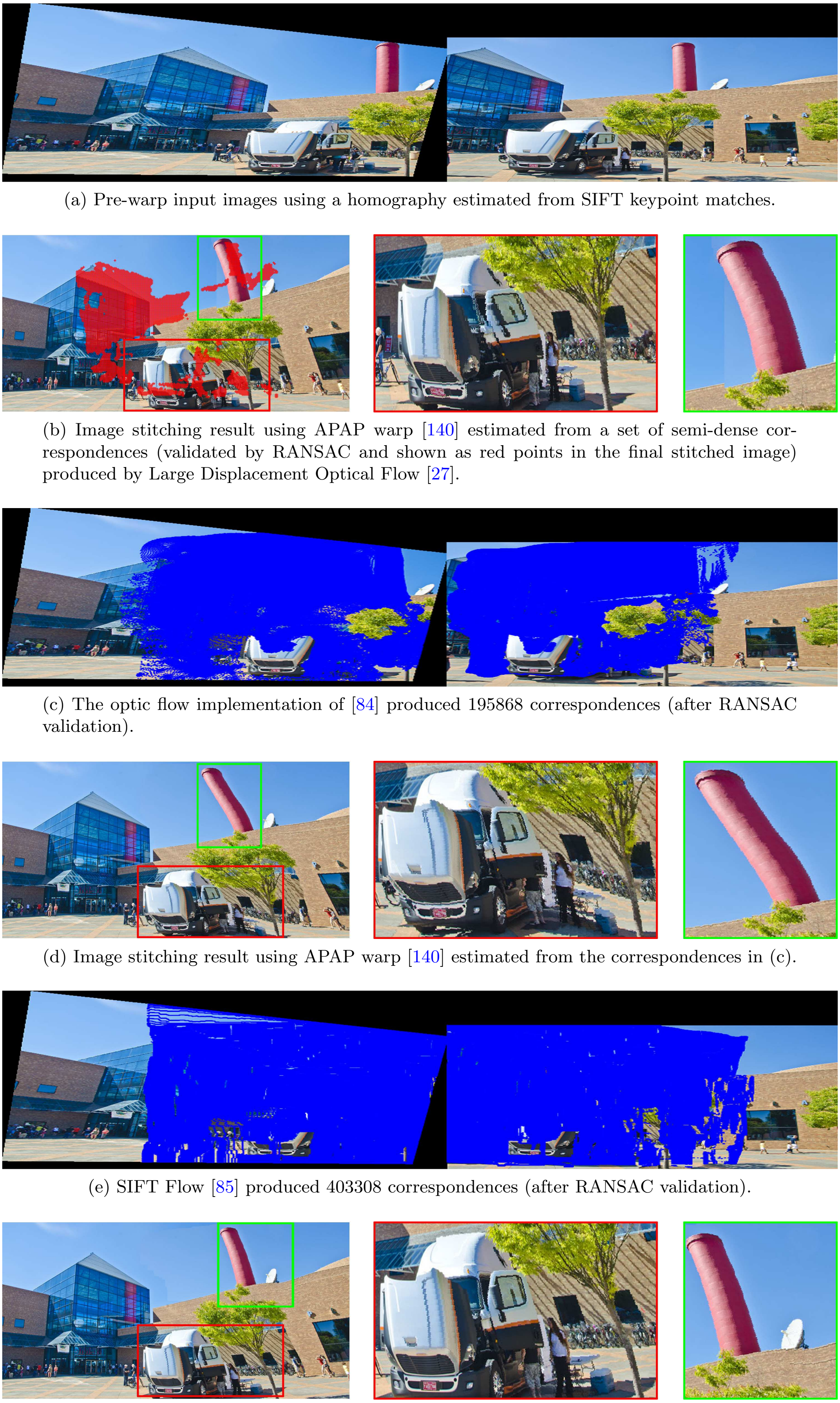
4.7.2 Evaluation of flow-based methods

This section evaluates the quality of the dense correspondences from flow-based methods for image stitching. All image pairs from Section 4.7.1 are used. Three state-of-the-art flow-based dense and semi-dense correspondence methods were evaluated [84, 85, 27]. Before obtaining the dense correspondences, one image of each pair was pre-warped using a homography estimated from sparse SIFT keypoint matches. This served to simplify the problem for the flow-based methods. Further, RANSAC was invoked with a tight inlier threshold (1 pixel) to ensure high-quality correspondences, before APAP warp [140] (the baseline) was estimated.

Despite the above precautions, the stitching results exhibit significant local distortions. This indicates that many of the correspondences are actually inaccurate. The small error tolerance of RANSAC still allowed sufficient local deviations (e.g., due to repetitive textures) that distorted the warp. While such local inaccuracies may not affect motion analysis or segmentation, they are fatal for accurate image stitching using spatially varying warps.

In Fig. 4.11, all three methods produce significant distortions on the chimney and the truck, despite the dense correspondences. In Fig. 4.12, dense correspondences were obtained around the pavilion. However, in Fig. 4.12b the pavilion is seriously distorted, and in Figs. 4.12d and 4.12f the two separate pillars of the pavilion are merged into one and the tower in the background also appears discontinuous. This points to the inaccuracies in the dense correspondences.

In Fig. 4.13, the inaccuracies in the dense correspondences are obvious, especially around top left corner of Fig. 4.13e. This leads to discontinuities in the balcony and missing pillars. The scene in Fig. 4.14 contains two apparent planes, and SIFT was able to find good sparse correspondences from only one of them (the floor). Thus, the prewarping result using a homography cannot wholly align the images well. Inevitably, this causes problems for the flow-based methods. Large displacement optical flow [27] found a large amount of correspondences on the ground; however, the region on the wall of the flower bed was not covered. The optical flow implementation of [84] and SIFT Flow [85] captured matches on the wall, but still produced stitching results with significant artifacts in Figs. 4.14d and 4.14f. This is due to the repeated textures on the wall which lead to inaccurate dense correspondences.



(a) Pre-warp input images using a homography estimated from SIFT keypoint matches.

(b) Image stitching result using APAP warp [140] estimated from a set of semi-dense correspondences (validated by RANSAC and shown as red points in the final stitched image) produced by Large Displacement Optical Flow [27].

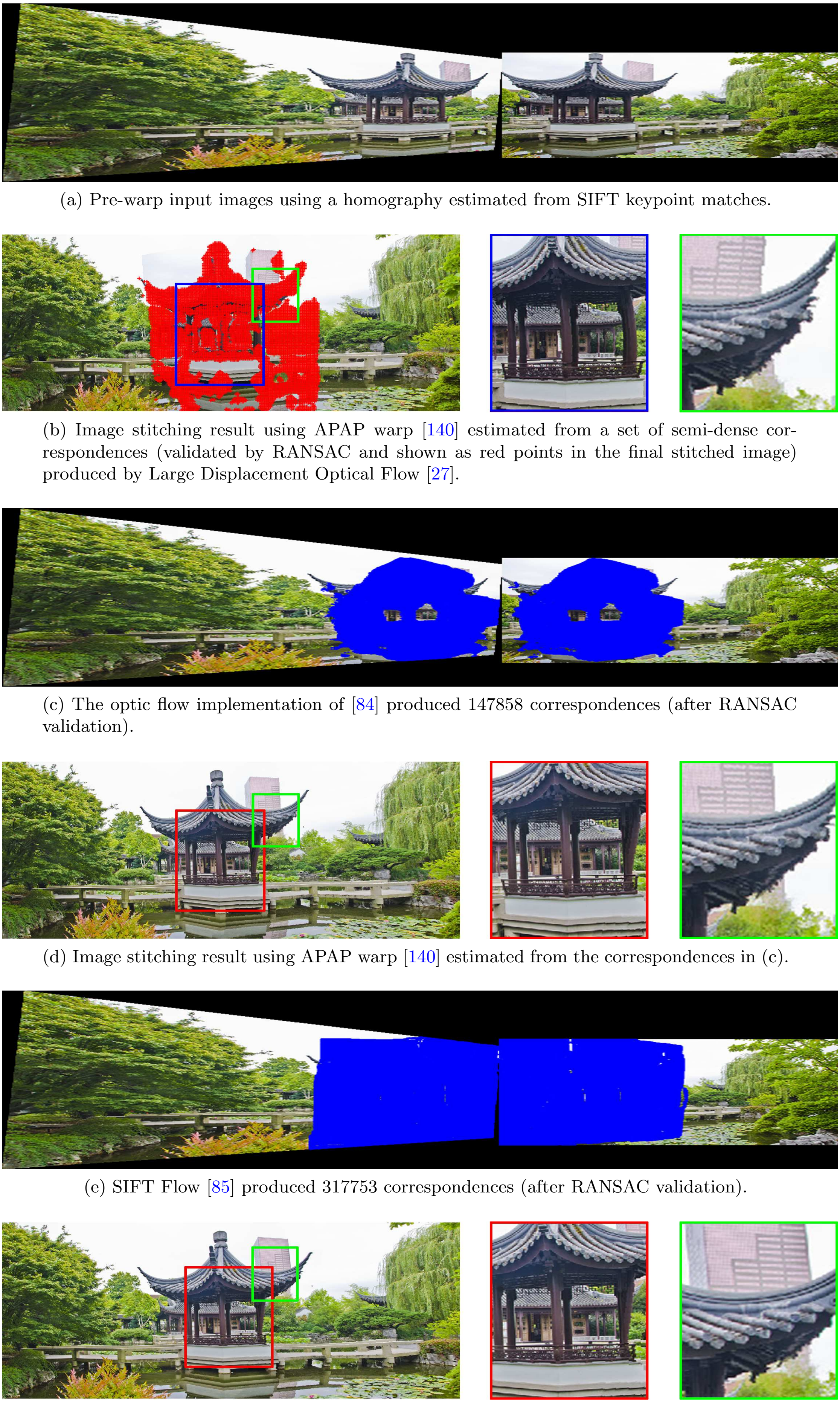
(c) The optic flow implementation of [84] produced 195868 correspondences (after RANSAC validation).

(d) Image stitching result using APAP warp [140] estimated from the correspondences in (c).

(e) SIFT Flow [85] produced 403308 correspondences (after RANSAC validation).

(f) Image stitching result using APAP warp [140] estimated from the correspondences in (e).

FIGURE 4.11: Dense correspondences and stitching results of three flow-based methods on the *truck* image pair.



(a) Pre-warp input images using a homography estimated from SIFT keypoint matches.

(b) Image stitching result using APAP warp [140] estimated from a set of semi-dense correspondences (validated by RANSAC and shown as red points in the final stitched image) produced by Large Displacement Optical Flow [27].

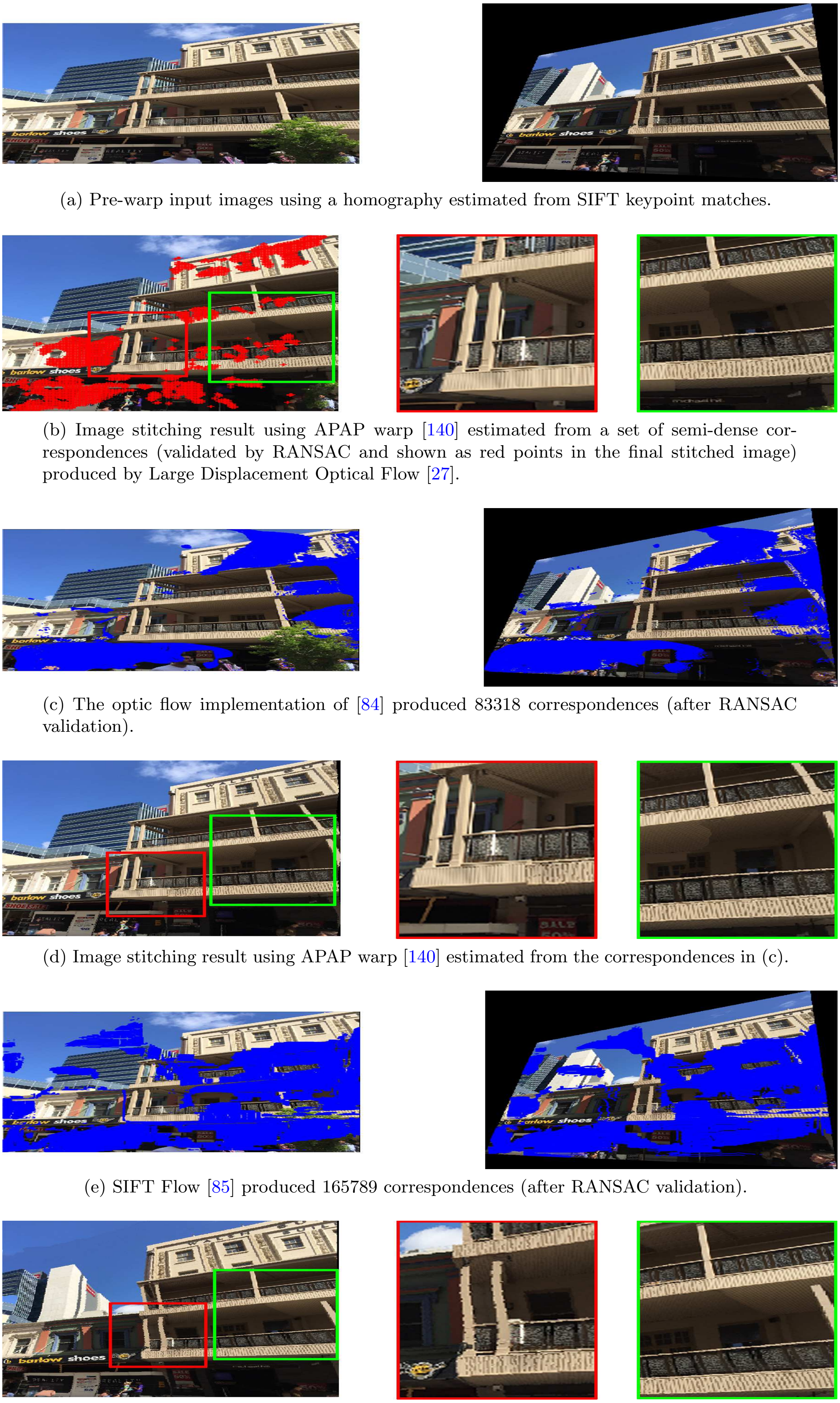
(c) The optic flow implementation of [84] produced 147858 correspondences (after RANSAC validation).

(d) Image stitching result using APAP warp [140] estimated from the correspondences in (c).

(e) SIFT Flow [85] produced 317753 correspondences (after RANSAC validation).

(f) Image stitching result using APAP warp [140] estimated from the correspondences in (e).

Figure 4.12: Dense correspondences and stitching results of three flow-based methods on the *temple* image pair.



(a) Pre-warp input images using a homography estimated from SIFT keypoint matches.

(b) Image stitching result using APAP warp [140] estimated from a set of semi-dense correspondences (validated by RANSAC and shown as red points in the final stitched image) produced by Large Displacement Optical Flow [27].

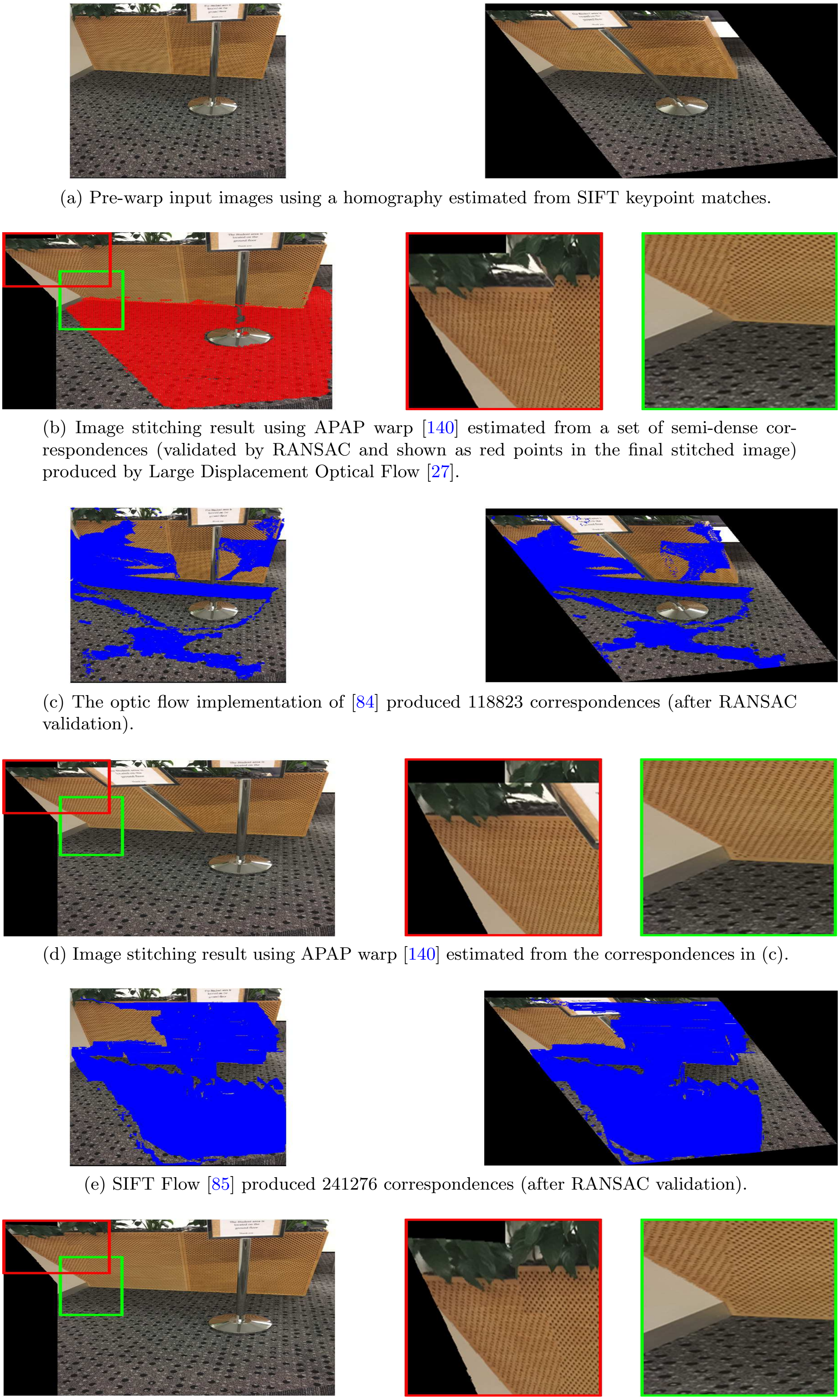
(c) The optic flow implementation of [84] produced 83318 correspondences (after RANSAC validation).

(d) Image stitching result using APAP warp [140] estimated from the correspondences in (c).

(e) SIFT Flow [85] produced 165789 correspondences (after RANSAC validation).

(f) Image stitching result using APAP warp [140] estimated from the correspondences in (e).

FIGURE 4.13: Dense correspondences and stitching results of three flow-based methods on the *shopfront* image pair.



(a) Pre-warp input images using a homography estimated from SIFT keypoint matches.

(b) Image stitching result using APAP warp [140] estimated from a set of semi-dense correspondences (validated by RANSAC and shown as red points in the final stitched image) produced by Large Displacement Optical Flow [27].

(c) The optic flow implementation of [84] produced 118823 correspondences (after RANSAC validation).

(d) Image stitching result using APAP warp [140] estimated from the correspondences in (c).

(e) SIFT Flow [85] produced 241276 correspondences (after RANSAC validation).

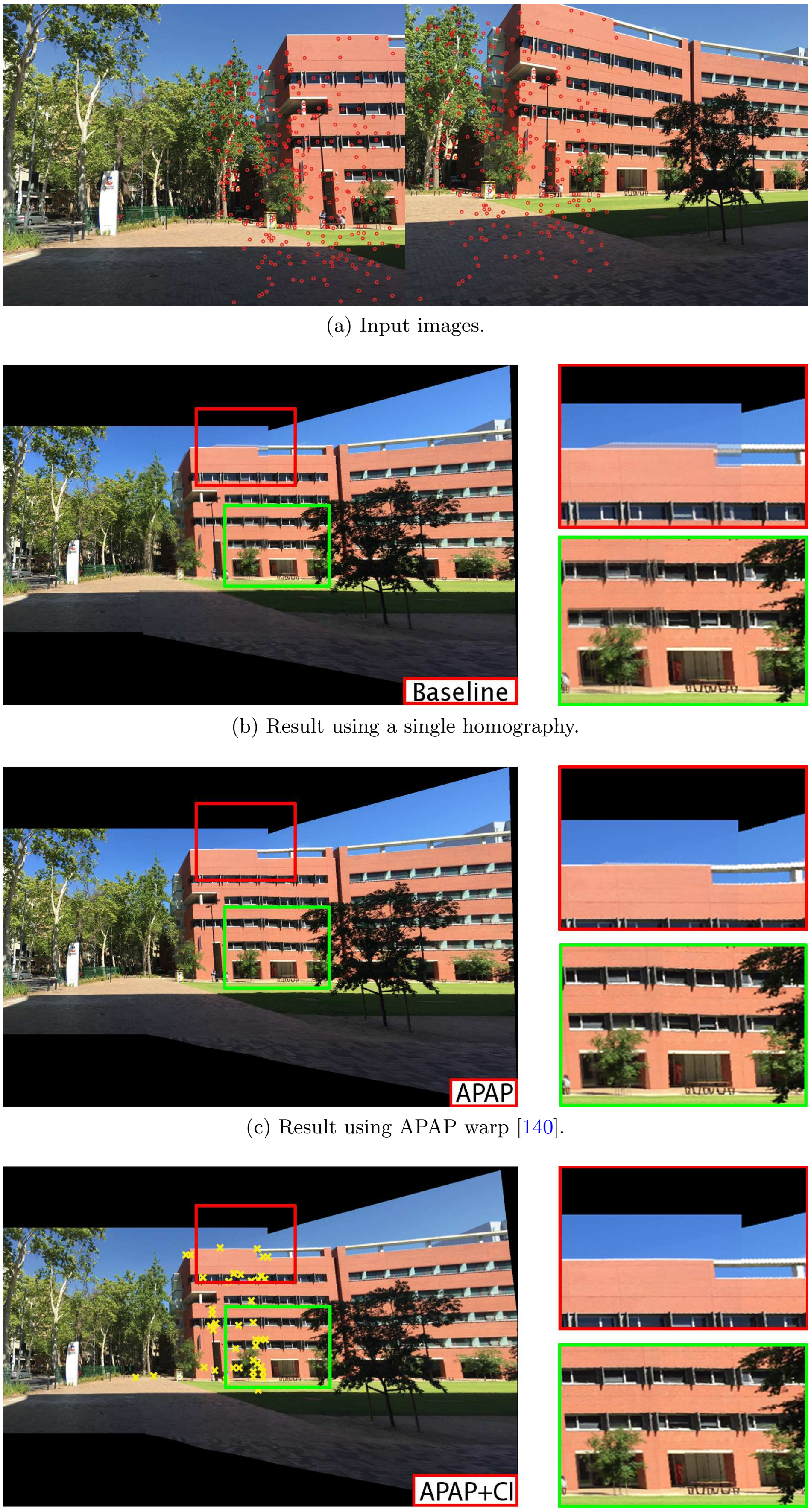
(f) Image stitching result using APAP warp [140] estimated from the correspondences in (e).

Figure 4.14: Dense correspondences and stitching results of three flow-based methods on the *lobby* image pair.

4.7.3 Comparisons on image pairs without significant parallax

In this section, results on the additional three images pairs that have taken of scenes without significant parallax are presented. Because parallax is not present, all results are generated without using seam cut blending. Seam cut, however, is an important step in parallax-tolerant image stitching [141], so the proposed method (APAP+CI) is only compared with results obtained from a single homography (baseline) and the APAP method. Since seam cut blending is not used, the proposed method is used on the overlapped image region only, which is slightly different from the Algorithm 4.2. Figs. 4.15, 4.16 and 4.17 show the results.

The single homography model works under the assumption that the images are sufficiently far away or taken with a camera undergoing pure rotational motion. For these images, this imaging condition is not satisfied and a single homography is not sufficient to align the images. As shown in Figs. 4.15b, 4.16b and 4.17b, warping with a single homography introduces significant ghosting artifacts in the stitching results. APAP warp is able to provide a more accurate alignment, but fail if there are insufficient point matches (e.g. pillar in Fig. 4.15c, arch in Fig 4.16c, and railing and eave in Fig 4.17c). The proposed method automatically adds new correspondences and rectifies the weakness of the APAP warp producing results that have a better overall alignment.



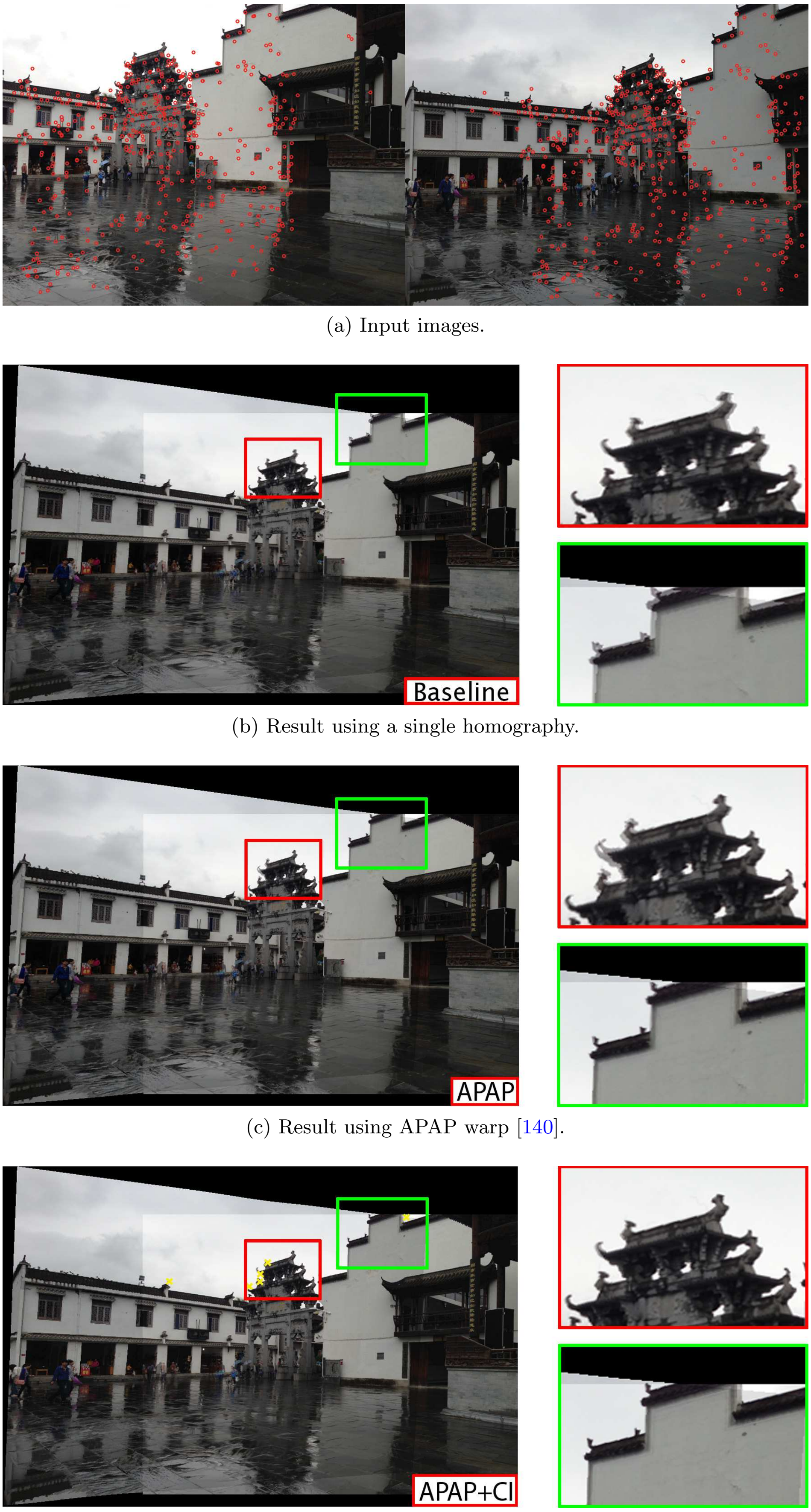
(a) Input images.

(b) Result using a single homography.

(c) Result using APAP warp [140].

(d) Image stitching result after adding 38 new correspondences using the proposed method.

FIGURE 4.15: Comparing three methods on the *building* image pair. Inserted correspondences by APAP+CI are shown as yellow points.



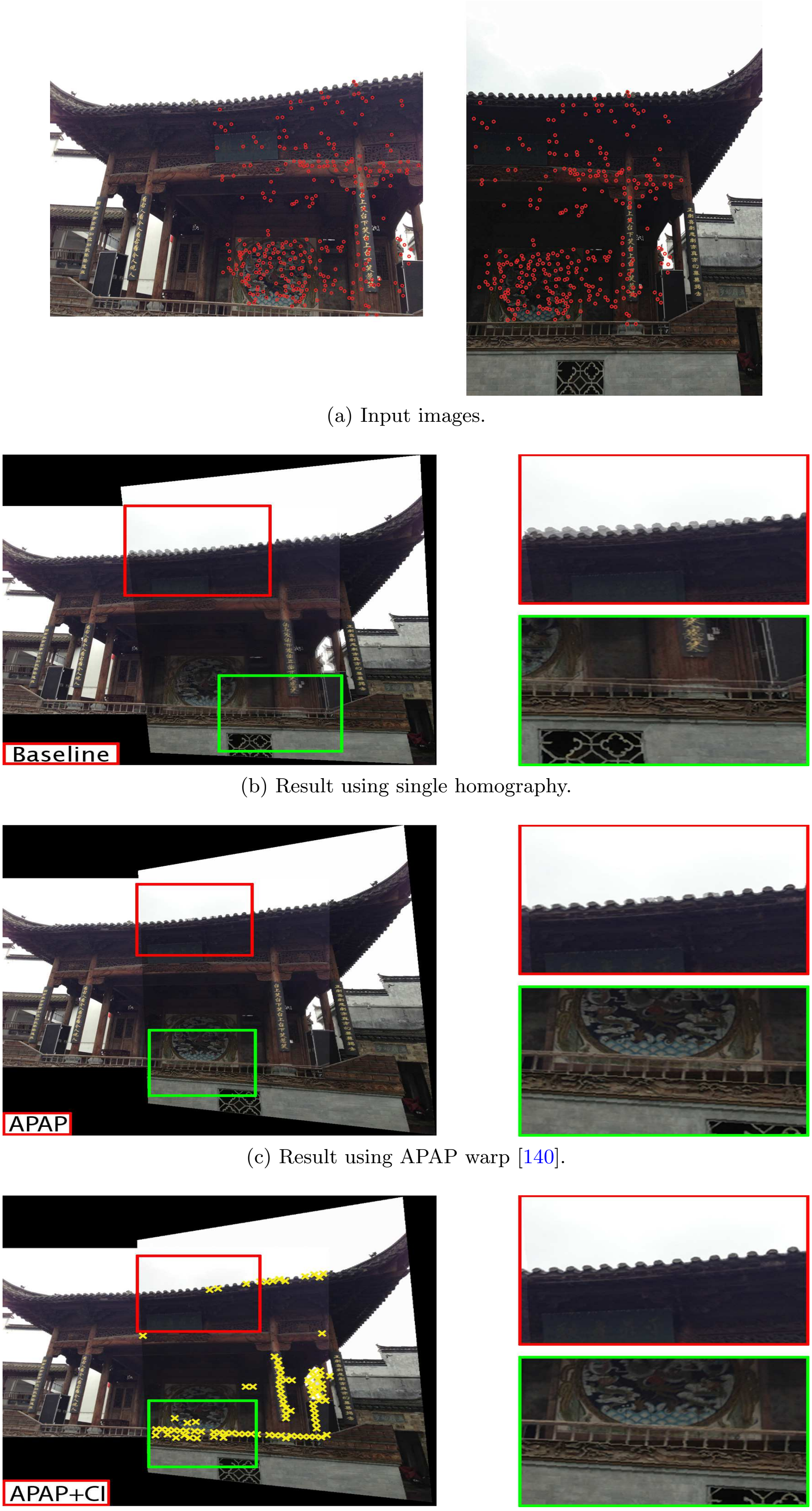
(a) Input images.

(b) Result using a single homography.

(c) Result using APAP warp [140].

(d) Image stitching result after adding 6 new correspondences using the proposed method.

FIGURE 4.16: Comparing three methods on the *arch* image pair. Inserted correspondences by APAP+CI are shown as yellow points.



(a) Input images.

(b) Result using single homography.

(c) Result using APAP warp [140].

(d) Image stitching result after adding 93 new correspondences using proposed the method.

FIGURE 4.17: Comparing three methods on the *stage* image pair. Inserted correspondences by APAP+CI are shown as yellow points.

4.8 Summary

Wide-baseline image stitching is a challenging problem. Flow-based methods often fail to produce dense and accurate correspondences, while spatially varying warps are only flexible up to the sparse set of keypoint matches given. This chapter presented a novel data-driven warp adaption scheme for APAP image stitching. A core step in the proposed algorithm is a correspondence insertion technique. The proposed method improves upon the original APAP warps, which fails when the overlap region is correspondence-poor. Generated results also show that it is crucial to accurately align the images throughout the overlap area, even if sophisticated compositing is used.

Chapter 5

Video Stabilization Using Homography Fields

5.1 Introduction

5.1.1 2D video stabilization

2D video stabilization methods rely on simple 2D image transforms (e.g., affine or projective), which are very efficient to estimate. Given input frame sequence $I_1, I_2, \ldots, I_T$, transform $\mathbf{T}_i$ is first estimated from feature matches across two successive (or neighboring) frames (e.g., via SIFT matching), as shown in Figure 5.1.

By applying the transform estimation for every pair of adjacent frames, a transformation chain can be obtained. Most methods build up the transformation chain by cumulating transforms with an anchoring frame (normally the first frame I_1) to describe the camera motion; see Figure 5.2, original camera trajectory is plotted in dotted lines.

Camera motion is then filtered to perform stabilization and remove high-frequency jitters; see Figure 5.2, stabilized camera motion path appears continuous and smooth. From the smoothed trajectory, update transforms $\mathbf{B}_i$ are constructed, which can be applied on each frame of the video to "undo" the jerky motions.

Lastly, using the obtained update transforms $\mathbf{B}_i$, full-frame warps are applied on original frames to render stabilized frame sequence $I'_1, I'_2, \ldots, I'_T$; see Figure 5.1.

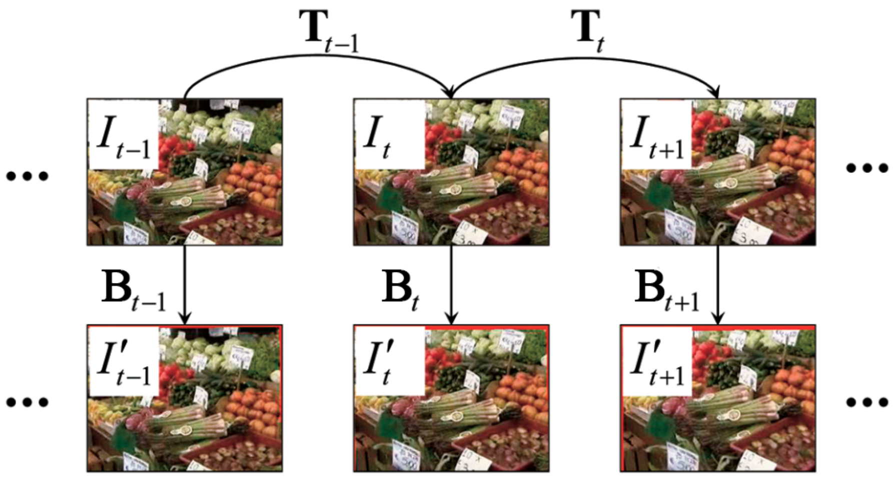


Figure 5.1: 2D video stabilization methods estimate 2D image transforms $\mathbf{T}_i$ across two successive frames to capture the camera motion. Camera motion is then smoothed to remove high-frequency motions. From the smoothed trajectory, construct update transforms $\mathbf{B}_i$ that can be applied on each frame of the video to "undo" the jerky motions. The stabilized frames $I_i{}'$ are rendered as the bottom frame sequence. Figure taken from [98].

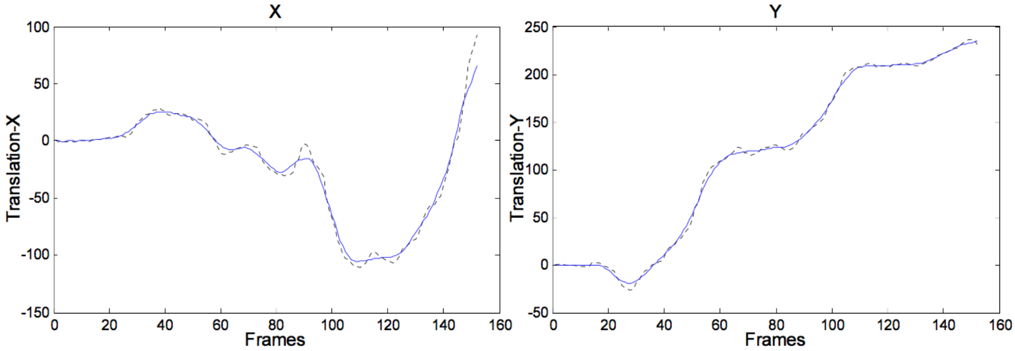


Figure 5.2: Original camera trajectory along X and Y directions (dotted line) are displayed. The estimated camera motion is filtered to perform stabilization. The smoothed motion path is shown with solid lines. Figure taken from [98].

2D video stabilization methods employ simple 2D image transforms to estimate camera trajectory and define update transform, which leads to the fact that standard 2D stabilization methods can only provide very limited amount of stabilization because the 2D motion model is restricted. This approach can only achieve good results if the camera motion is nearly purely rotational, or the scene is nearly purely planar. However, in most cases, camera motions are much more complicated than that. Thus, there is typically a large gap between 2D video stabilization and the desired outputs.

5.1.2 Chapter overview

Contrary to the recent works that have proposed more complex motion models and smoothing algorithms, this chapter focuses on constructing better update transforms. Given that update transforms will be eventually applied on all the frames to obtain the final result, it is more fruitful to design warping functions that can adjust the frames without introducing undesirable artifacts. On the other hand, to estimate the camera trajectory for smoothing, it is sufficient to capture the main trend of motion, rather than precisely characterize the movement of each pixel - in this thesis, following Gleicher and Liu [51], simple 2D homographies are used for motion modeling and smoothing. For the all-important update transform, homography fields, which are spatially varying warps that are regularized to be as projective as possible, have been proposed. This enables flexible and accurate warping that adheres closely to the underlying scene geometry. The obtained update transform is powerful enough to eliminate unwanted jerky motions, while at the same time prevent the warped sequence of frames from appearing wobbly or distorted. Crucially, homography fields can be integrated closely with any homography-based smoothing algorithm; this realizes a video stabilization pipeline that smooths globally and warps locally.

The ability of the proposed warping function extends beyond removing shakiness in videos. This chapter shows how rolling shutter effects (distortions due to non-uniform CMOS sensor readout) can be simultaneously modeled and compensated for using homography fields. The proposed method is also able to solve the problem of video inpainting to fill in blank regions in the output video resulting from cropping the adjusted frames which are non-rectangular. It will be shown how inter-frame motions can accurately be modeled based on homography fields, such that blank regions can be filled in using pixels from neighboring frames without creating undesirable misalignment errors. In contrast to previous works that have tackled the above issues separately using different methodologies and algorithms, the proposed approach is the first that treats trajectory smoothing, rolling shutter removal and video inpainting under a single unified video stabilization pipeline.

The rest of the chapter is organized as follows. Section 5.2 gives an overview of the proposed shaky video postprocessing pipeline. Section 5.3 describes the proposed video

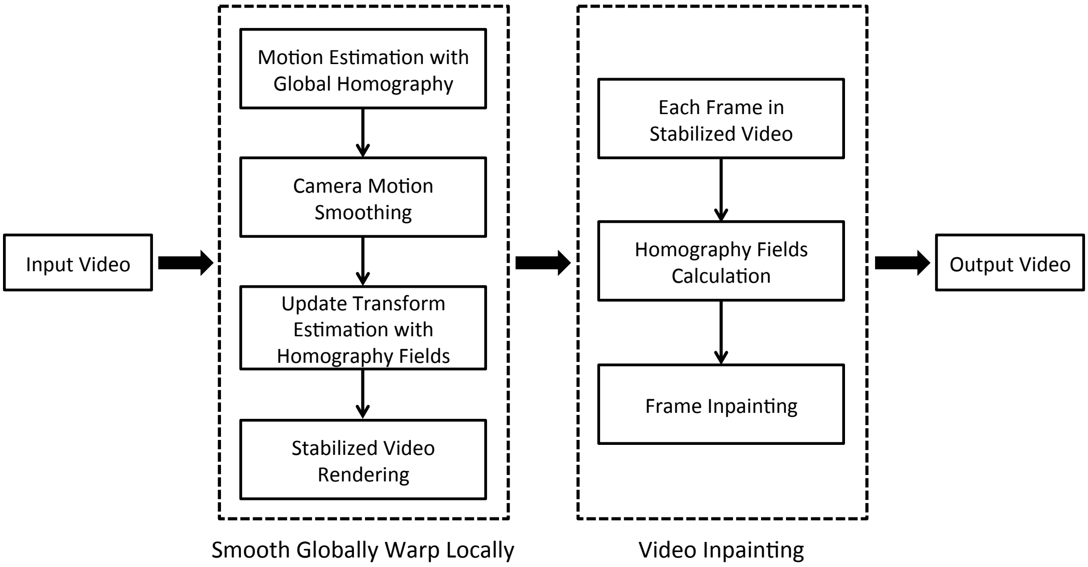

Input Video
Motion Estimation with Global Homography
Camera Motion Smoothing
Update Transform Estimation with Homography Fields
Stabilized Video Rendering
Smooth Globally Warp Locally
Each Frame in Stabilized Video
Homography Fields Calculation
Frame Inpainting
Video Inpainting
Output Video

FIGURE 5.3: Overview of the proposed video processing pipeline. Given the input video, the shaky camera motion is firstly estimated using the frame global homography, and smoothed to remove high-frequency components. The update transform is then constructed with the usage of homography fields to produce the stabilized video. The unavoidable blank pixels are then filled in by the video inpainting method.

stabilization pipeline and the estimation of homography fields for frame updating. Section 5.4 shows how homography fields can naturally deal with rolling shutter effects, and demonstrates its superior efficacy relative to previous techniques. Section 5.5 proposes the video inpainting approach based on homography field warping. Results are presented in Section 5.6. Conclusions and discussions on the limitations of the proposed approach are given in Section 5.7.

5.2 Proposed Method Overview

There are two main parts in the proposed pipeline. Given the input video, the shaky camera motion is firstly estimated using the frame global homography, and smoothed to remove high-frequency components. The update transform is then constructed with the usage of homography fields, which are spatially varying warps that are regularized to be as projective as possible, to produce the stabilized video. These steps construct the proposed “smooth globally warp locally” stabilization algorithm. Due to the deviation of the smoothed camera path from the original trajectory, the existence of blank pixels in the stabilized video is unavoidable. The stabilized video is then processed through the video inpainting step. The proposed video inpainting method fills in blank regions in each frame using pixels from neighboring frames without introducing unpleasant artifacts. Figure 5.3 gives an overview of the proposed video stabilization pipeline. The proposed approach treats camera trajectory smoothing, rolling shutter removal and video inpainting under a single unified video stabilization pipeline. Each step will be introduced in detail in the following.

5.3 Smooth Globally Warp Locally

Consider the (ideal) case where the video is shot with the camera purely rotating about a point. Under the perspective camera model, each inter-frame image motion can be modeled perfectly by a homography. Chaining the homographies thus yields a representation of the camera trajectory, which can then be stabilized. Gleicher and Liu [51] interpreted the process as warping all the frames to the first frame to create a panoramic mosaic, then finding a stable sequence of cropping homographies through the mosaic.

Let the video frames be $I_1, I_2, \dots, I_T$. Denote $\mathbf{C}_t$ as the homography that warps I_t to the first frame I_1. To undo the effects of jerky motion, the frame I_t must be warped by the update transform

$$\mathbf{B}_t = \mathbf{P}_t^{-1}\mathbf{C}_t, \tag{5.1}$$

where $\mathbf{P}_t$ is a homography that warps I_t to I_1 following a smoothed camera path. Note that since both $\mathbf{P}_t$ and $\mathbf{C}_t$ are homographies, update transform $\mathbf{B}_t$ is also a homography. Furthermore, if $\mathbf{P}_t = \mathbf{C}_t$ (i.e., no smoothing), the update transform reduces to the identity mapping.

5.3.1 Motion estimation and smoothing

The transform $\mathbf{C}_t$ is recursively defined as

$$\mathbf{C}_t = \mathbf{C}_{t-1}\mathbf{H}_{t,t-1}, \tag{5.2}$$

where $\mathbf{C}_1$ is the identity matrix, and $H_{t,t-1}$ is the homography that maps points from I_t to I_{t-1}. Motion estimation is thus achieved by estimating $\mathbf{H}_{t,t-1}$ for all $t = 2, \dots, T$, then chaining them in the right order to yield the camera trajectory.

To estimate $\mathbf{H}_{t,t-1}$, a set of features across I_t and I_{t-1} are first detected and matched. To ensure more uniform coverage of features, the same strategy used in [139] of applying the SIFT technique on overlapping subwindows across I_t and I_{t-1} is employed. Alternatively, dense features can be tracked in the video which are then broken up into feature matches [122]. Given the point matches, RANSAC is used to robustly estimate the homography $\mathbf{H}_{t,t-1}$.

As mentioned earlier, any stabilization and motion planning algorithm that represents the camera trajectory as a homography chain can be used in the proposed framework. For concreteness, the single path smoothing algorithm of Liu *et al.* [89] is used. Given $\{\mathbf{C}_t\}_{t=1}^{T}$, the smoothed path $\{\mathbf{P}_t\}_{t=1}^{T}$ is obtained by minimizing

$$O(\{\mathbf{P}_t\}) = \sum_t \left(\|\mathbf{P}_t - \mathbf{C}_t\|^2 + \lambda \sum_{r \in \Omega_t} \|\mathbf{P}_t - \mathbf{P}_r\|^2 \right), \tag{5.3}$$

where Ω_t contains the index of the neighbors of I_t. In this thesis, each frame is linked to the nearest 40 frames. The second term smooths the trajectory, while the first term encourages $\mathbf{P}_t$ to be close to $\mathbf{C}_t$. Parameter λ controls the strength of smoothing by trading off the two terms. Minimizing (5.3) can be done by a Jacobi-based iterative solver [22]:

$$\mathbf{P}_t^{(\xi+1)} = \frac{1}{1+2\lambda}\mathbf{C}_t + \sum_{r \in \Omega_t, r \neq t} \frac{2\lambda}{1+2\lambda}\mathbf{P}_r^{(\xi)}, \tag{5.4}$$

where ξ is an iteration index. At initialization, set $\mathbf{P}_t^{(0)} = \mathbf{C}_t$.

5.3.2 From global to local

In practice, the camera motion is unlikely to be purely rotational, thus the homography $\mathbf{H}_{t,t-1}$ cannot perfectly align the pixels of I_t and I_{t-1}. Nonetheless, as the results will show, for the purpose of capturing and smoothing the main trajectory of the camera, a chain of homographies is sufficient to produce excellent stabilization.

However, if the update transform $\mathbf{B}_t$ is also a frame-global homographic warp, updating I_t with $\mathbf{B}_t$ will cause distortions or wobbling effects in the video. To avoid such artifacts, $\mathbf{B}_t$ should be a spatially varying warp. Specifically, for a pixel $\mathbf{p}_*$ in I_t, the proposed

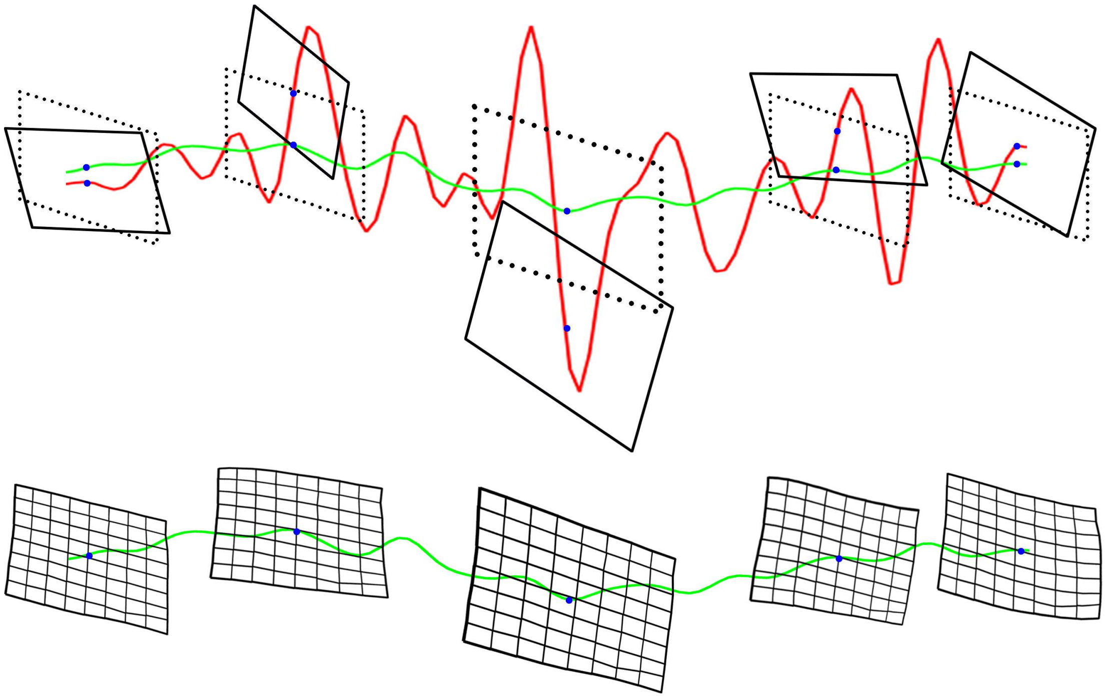


Figure 5.4: Smooth globally warp locally stabilization method. Given the input video, the camera motion is estimated as a chain of 2D homographies (red trajectory) which are then smoothed (green trajectory). Instead of applying frame-global update transforms (dotted frames), the smoothed parameters are used to estimate homography fields (shown as flexible grids), which are spatially varying warps warps that are regularized to be as projective as possible.

method warps $\mathbf{p}_*$ to the output frame using the local update transform

$$\mathbf{B}_t^* = \mathbf{P}_t^{-1}\mathbf{C}_t^*. \tag{5.5}$$

Conceptually, $\mathbf{C}_t^*$ is a localized homography that warps $\mathbf{p}_*$ to the base frame I_1. Similar to (5.2), $\mathbf{C}_t^*$ can be defined recursively as

$$\mathbf{C}_t^* = \mathbf{C}_{t-1}^*\mathbf{H}_{t,t-1}^*, \tag{5.6}$$

where $\mathbf{C}_1^*$ is the identity matrix, and $\mathbf{H}_{t,t-1}^*$ is a localized homography that maps p_* from I_t to I_{t-1}. Collectively, for all $\{\mathbf{p}_*\}$ in I_t, the overall set of $\{\mathbf{B}_t^*\}$ define the update transform for I_t. Note that in each $\mathbf{B}_t^*$, the de-shaking adjustment is still provided by applying the inverse of the smoothed global homography chain $\mathbf{P}_t$. Thus, this thesis dubs the proposed method "smooth globally warp locally"; Figure 5.4 provides an overview. The following section shows how the localized homography $\mathbf{H}_{t,t-1}^*$ can be estimated such that $\mathbf{B}_t^*$ provides an update transform that preserves the contents and geometric structures in I_t.

5.3.3 Homography field warps

Let $\{(\mathbf{p}_i, \mathbf{p}'_i)\}_{i=1}^N$ be a set of corresponding feature points across I_t and I_{t-1}, where each $\mathbf{p}_i = [x_i \; y_i]^T$ and $\mathbf{p}'_i = [x'_i \; y'_i]^T$. $\tilde{\mathbf{p}}_i = [x_i \; y_i \; 1]^T$ is $\mathbf{p}_i$ in homogeneous coordinates. At this stage, mismatches have been removed by RANSAC. The direct linear transformation (DLT) method estimates the frame-global homography $\mathbf{H}_{t,t-1}$ that maps from I_t to I_{t-1} by solving

$$\underset{\mathbf{h}}{\operatorname{argmin}} \sum_i \|\mathbf{m}_i\mathbf{h}\|^2 = \underset{\mathbf{h}}{\operatorname{argmin}} \; \|\mathbf{M}\mathbf{h}\|^2, \quad s.t. \; \|\mathbf{h}\| = 1, \tag{5.7}$$

where $\mathbf{h}$ is 9-vector obtained by vectorizing $\mathbf{H}_{t,t-1}$, and $\mathbf{m}_i$ is a 2×9 matrix resulting from linearizing the homography mapping constraint based on data $\{\mathbf{p}_i, \mathbf{p}'_i\}$. Matrix $\mathbf{M}$ is obtained by vertically stacking $\mathbf{m}_i$ for all i. The solution to (5.7) is simply the least significant right singular vector of $\mathbf{M}$. Refer to Section 4.1.1 for details on homography estimation.

To estimate a local $\mathbf{H}^*_{t,t-1}$ centered on an arbitrary $\mathbf{p}_*$ in I_t, Zaragoza *et al.* [140] proposed the moving direct linear transformation (MDLT) technique

$$\underset{\mathbf{h}}{\operatorname{argmin}} \sum_i \|w^*\mathbf{m}_i\mathbf{h}\|^2 = \underset{\mathbf{h}}{\operatorname{argmin}} \; \|\mathbf{W}^*\mathbf{M}\mathbf{h}\|^2, \quad s.t. \; \|\mathbf{h}\| = 1, \tag{5.8}$$

where the weight w^*_i is calculated as

$$w^*_i = \exp\left(-\left\|\mathbf{p}_i - \mathbf{p}_*\right\|^2 / \sigma^2\right), \tag{5.9}$$

and $\mathbf{W}^*$ is a $2N \times 2N$ matrix composed as

$$\mathbf{W}^* = \operatorname{diag}([w^*_1 \; w^*_1 \; w^*_2 \; w^*_2 \; \ldots \; w^*_N \; w^*_N]). \tag{5.10}$$

The solution to (5.8) is the least significant right singular vector of $\mathbf{W}^*\mathbf{M}$. The scalar weights $\{w^*_i\}_{i=1}^N$ assign greater values to point matches that are closer to $\mathbf{p}_*$ based on an isotropic Gaussian kernel of width σ. This allows $\mathbf{H}^*_{t,t-1}$ to adapt to the local structure around $\mathbf{p}_*$. As $\mathbf{p}_*$ is varied across the 2D domain of I_t, the set of homographies $\{\mathbf{H}^*_{t,t-1}\}$ collectively create a field of homographies.

The smoothness of the overall warp is ensured by the spatial chaining of the homographies $\{\mathbf{H}^*_{t,t-1}\}$ by (5.9). Specifically, any two pixels in I_t that are spatially close will yield similar local homographies, since their respective set of weights will be similar. To regularize the overall warp, an offset γ is applied

$$w^*_i = \max\left(\exp\left(-\left\|\mathbf{p}_i - \mathbf{p}_*\right\|^2/\sigma^2\right), \gamma\right), \tag{5.11}$$

whereby if γ approaches 1, the homography field warp $\{\mathbf{H}^*_{t,t-1}\}$ reduces to the frame-global homography $\mathbf{H}_{t,t-1}$.

5.3.4 Calculating homography fields

While the homography field is defined over all pixels in I_t, in practice, closely neighboring pixels yield very similar homographies. Following Zaragoza *et al.* [140], this thesis solves (5.8) on a $X \times Y$ grid on I_t, and warps a pixel from I_t using its closest local homography. This also implies that the corresponding update transform $\{\mathbf{B}^*_t\}$ is defined over the same grid. It is crucial to note that each $\mathbf{H}^*_{t,t-1}$ can be solved independently, thus the effort for estimating a homography field scales linearly with the number of local homographies. For $X \times Y = 2500$ and $N = 2000$, Matlab can solve for all local homographies in $\approx 1s$.

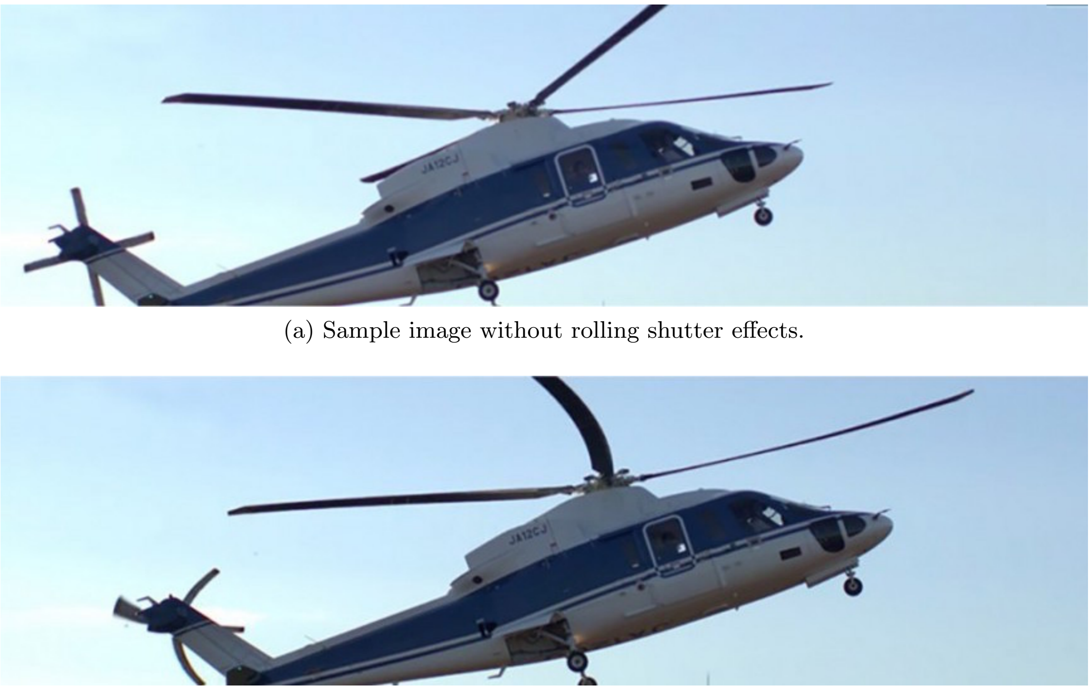
$\mathbf{B}^*_t$ can be further simplified by computing the homography chain as

$$\mathbf{C}^*_t = \mathbf{C}^*_{t-1}\mathbf{H}^*_{t,t-1} \approx \mathbf{C}_{t-1}\mathbf{H}^*_{t,t-1}, \tag{5.12}$$

i.e., the frame-global homography chain $\mathbf{C}_{t-1}$ is used to propagate the homography field between I_t and I_{t-1} to the base frame. In practice, this yields little noticeable difference in the warping results. Computationally, this provides significant savings since only a single homography chain (i.e., the camera trajectory) needs to be maintained. Contrast the proposed approach to Liu *et al.* [89] who need to estimate and smooth multiple homography chains.

5.4 Removing Rolling Shutter Effects

[1] http://www.premiumbeat.com/blog/know-the-basics-of-global-shutter-vs-rolling-shutter/



(a) Sample image without rolling shutter effects.

(b) Same image with rolling shutter effects.

FIGURE 5.5: Example of rolling shutter effects[1]. Rolling shutter issues can damage a photo in a number of ways, but most often by causing a horizontal skew, also known as the "jello effect", when helicopter propellers panning in (b).

Rolling shutter distortions can occur prominently in videos captured using CMOS sensors (e.g., digital single-lens reflex cameras (DSLRs) or cameras available on smartphones and tablet devices); see Figure 5.5 for examples. Interpreting video stabilization as panoramic mosaicing again, rolling shutter effects can be compensated if all the video frames can be aligned accurately with the base frame, which is assumed to be free of rolling shutter distortions. In the context of the proposed approach, it is required that each pixel-centered homography chain $\mathbf{C}_t^*$ can precisely warp $\mathbf{p}_*$ in I_t to its rightful position in I_1. This amounts to accurately aligning each pair of I_t and I_{t-1}, to account for rolling shutter effects.

Grundmann *et al.* [55] proposed a 2D image motion model whereby each frame is divided vertically into K strips that are readout at different times. The goal is thus to align the corresponding strips between neighboring frames; see Figure 2.7. Grundmann *et al.*

proposed a homography mixture warp, which has the constraint

$$\underbrace{\begin{bmatrix}\mathbf{W}_1\mathbf{M} & \mathbf{W}_2\mathbf{M} & \cdots & \mathbf{W}_K\mathbf{M}\end{bmatrix}}_{\mathbf{\Pi}\in\mathbb{R}^{2N\times 9K}} \underbrace{\begin{bmatrix}\mathbf{h}_1\\ \mathbf{h}_2\\ \vdots\\ \mathbf{h}_K\end{bmatrix}}_{\boldsymbol{\eta}\in\mathbb{R}^{9K}} = 0, \tag{5.13}$$

where matrix $\mathbf{M}$ is obtained from linearizing the homography constraint for N point matches following (4.9), $\mathbf{W}_k$ is a diagonal matrix which weights each point match based on their distance from the mid-line of each strip. The K homographies are solved for simultaneously via DLT, i.e., the solution $\boldsymbol{\eta}$ is the least significant right singular vector of $\mathbf{\Pi}$.

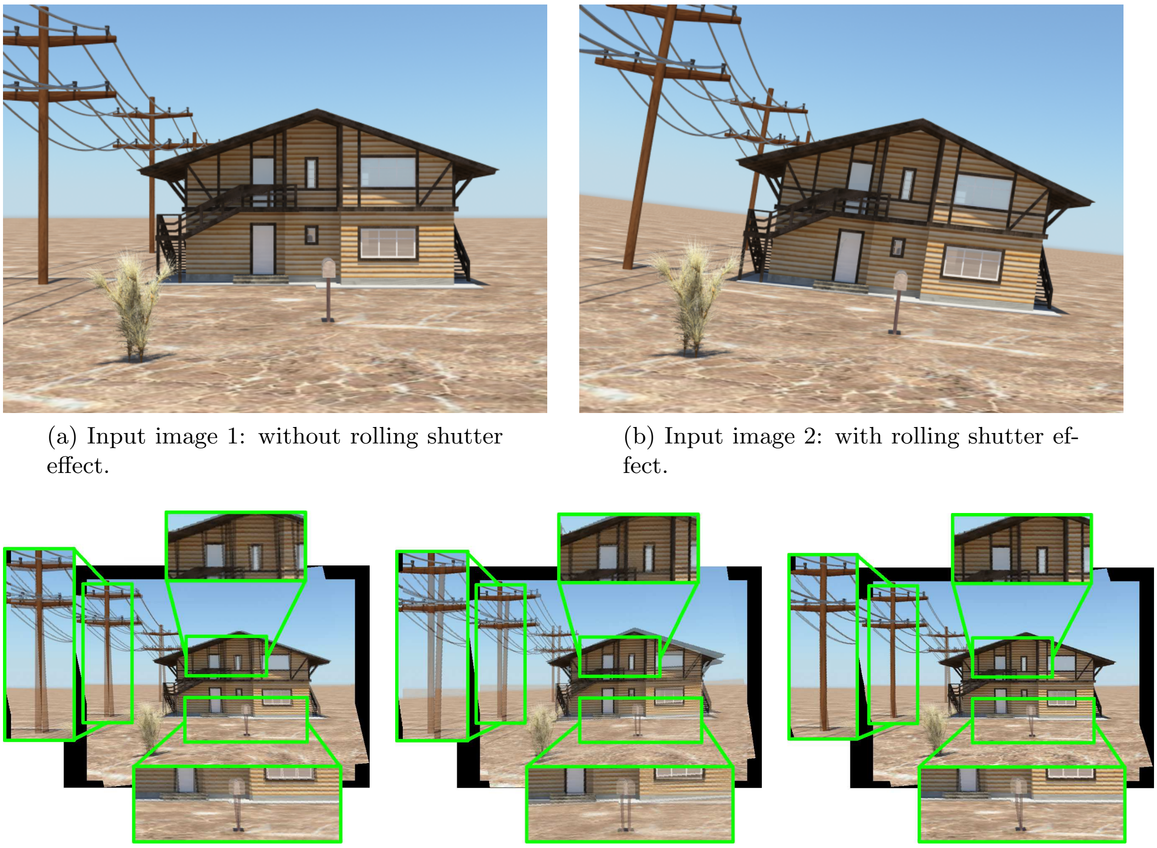
A weakness of the model, however, is that it ultimately assumes that the camera motion is purely rotational. The flexibility in the homography mixture occurs only along the vertical dimension. If there is distortion along the horizontal direction, e.g., depth parallax due to non-pure rotational camera motion, the strip-to-strip alignment will be poor; Figure 5.6 shows a concrete example. This in turn leads to distortions and wobbling effects in the smoothed video.

To improve the robustness of Grundmann *et al.*'s model with respect to distortions that it can handle while aligning images, the proposed approach replaces the spatial weighting function (5.9) with a non-isotropic (horizontal) Gaussian kernel

$$w_i^* = \exp\left(-\left(\frac{(x_i - x_*)^2}{2\sigma_x^2} + \frac{(y_i - y_*)^2}{2\sigma_y^2}\right)\right), \tag{5.14}$$

where $\mathbf{p}_* = [x_* \; y_*]^T$. By setting $\sigma_x > \sigma_y$ (usually $\sigma_x = 2\sigma_y$), the proposed method encourages the homography fields warp to increase its vertical flexibility to account for rolling shutter, without neglecting possible distortions in the horizontal directions. Figure 5.6 shows the result on the same data, and Section 5.6 will provide more comparisons.

More fundamentally, Grundmann *et al.* estimate the K homographies in (5.13) simultaneously to ensure overall smoothness. Using DLT also implies the constraint $\|\boldsymbol{\eta}\| = 1$, which does not prevent any $\mathbf{h}_k$ from becoming degenerate, thus further regularization



(a) Input image 1: without rolling shutter effect.

(b) Input image 2: with rolling shutter effect.

(c) Warping result using Global Homography

(d) Warping result using Homography Mixture [55]

(e) Warping result using Homography Field

FIGURE 5.6: Figures (a) and (b) are two input images taken from [45], with rolling shutter effect in the second one. Global Homography is not able to model rolling shutter effect and will introduce pixel misalignment, as shown in (c). Although Homography Mixture [55] corrects skewed lines to be straight in (d), distortion along the non-vertical direction, e.g., depth parallax due to non-pure rotational camera motion, will not to be corrected. Warping result generated using Homography Field is much better, as shown in (e).

and refinement of $\boldsymbol{\eta}$ is necessary. In contrast, the proposed approach chains the homographies via the spatial weighting (5.14), and all local homographies are estimated independently without fear of degeneracy.

5.5 Inpainting with homography fields

Taking video stabilization as panoramic mosaicing again, one can stitch neighboring frames to the current frame to fill in the missing values. However, instead of using standard homographies (which cannot deal with depth parallax and rolling shutter effects), homography fields are used to conduct the warping. In contrast to Matsushita *et* *al.*'s [98] method, homography field warps can depend on a projective regularizer that is supported by the epipolar constraint. Although the projective mapping is valid only if the camera purely rotates, in the absence of any information in the faraway blank regions, a homography field warp reduces to a projective warp, which limits the amount of distortion in the inpainting region. The proposed novel inpainting technique is summarized in Algorithm 5.1. The following subsections provide details of major steps of the algorithm, while Figure 5.7 shows an actual iteration.

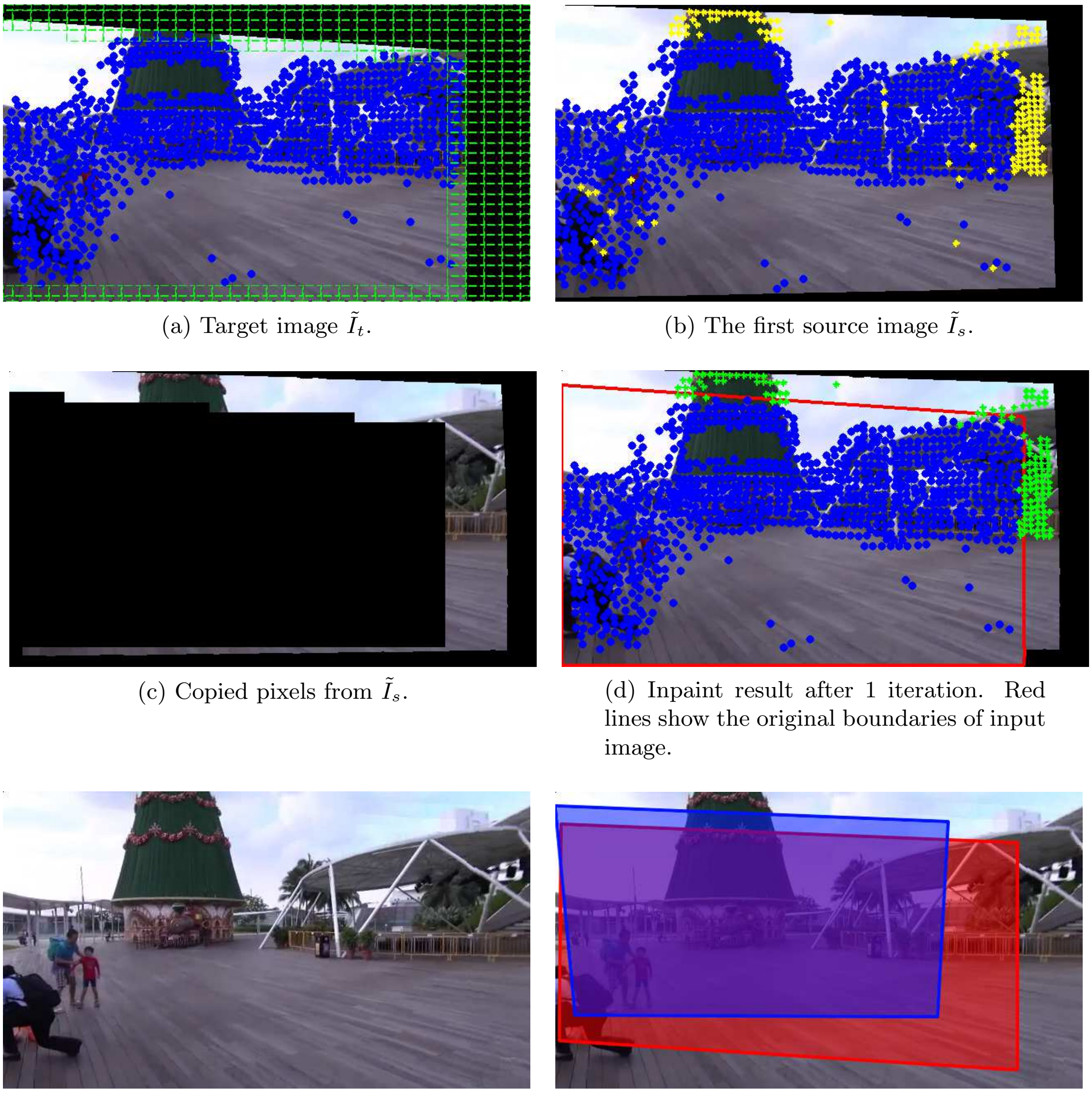
Algorithm 5.1 Video inpainting based on homography field warps.

```
Require: Target frame $\tilde{I}_t$, source frames $\mathcal{S} = \{\tilde{I}_s \mid s \neq t\}$, original feature matches
    between all frames.
while there exist blank pixels in $\tilde{I}_t$ and $\mathcal{S}$ is not empty do
    Remove from $\mathcal{S}$ the frame $\tilde{I}_s$ closest in time to $\tilde{I}_t$.
    Compute additional feature matches between $\tilde{I}_t$ and $\tilde{I}_s$.
    Conduct sliding window RANSAC between $\tilde{I}_t$ and $\tilde{I}_s$ to verify new matches.
    for each blank pixel $\mathbf{p}_*$ in $\tilde{I}_t$ do
        Compute $\mathbf{H}^*_{t,s}$ that is centered on $\mathbf{p}_*$ by (5.8).
        Warp $\mathbf{p}_*$ to $\tilde{I}_s$ by calculating $\tilde{\mathbf{p}}'_* \sim \mathbf{H}^*_{t,s}\tilde{\mathbf{p}}_*$.
        if $\tilde{I}_s(\mathbf{p}'_*)$ is not blank or out of bounds then
            Copy pixel colors from $\tilde{I}_s(\mathbf{p}'_*)$ to $\tilde{I}_t(\mathbf{p}_*)$.
            if $\mathbf{p}'_*$ is a detected feature in $\tilde{I}_s$ then
                Designate $\mathbf{p}_*$ as a feature, and match $\mathbf{p}_*$ to the correspondences of $\mathbf{p}'_*$.
            end if
        end if
    end for
end while
```

5.5.1 Sliding window RANSAC

At this stage, the input video frames have been updated $\tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_T$ to remove jerky motions and rolling shutter distortions. Each $\tilde{I}_t$ is stored as a 2D image with blank regions. The feature matches used in motion estimation and smoothing have also been warped to the updated frames.

Given the current target frame $\tilde{I}_t$ to inpaint, the unused source frame $\tilde{I}_s$ that is closest in time to $\tilde{I}_t$ is chosen. Apart from the point matches inherited from motion estimation, new features across $\tilde{I}_t$ and $\tilde{I}_s$ are also detected and matched. RANSAC is then conducted in a sliding window fashion to remove outliers. Specifically, given a common subwindow of $\tilde{I}_t$ and $\tilde{I}_s$, RANSAC is applied to estimate a standard homography using the feature matches in the subwindow. Once all subwindows have been processed, any match that



(a) Target image $\tilde{I}_t$.

(b) The first source image $\tilde{I}_s$.

(c) Copied pixels from $\tilde{I}_s$.

(d) Inpaint result after 1 iteration. Red lines show the original boundaries of input image.

(e) Ipainting result after 20 iterations.

(f) The blue shaded and red shaded windows show the output frames from Grundmann *et al.* [55] and Liu *et al.* [89].

FIGURE 5.7: Given a target image I_t in (a), blue points are tracked feature points. For the missing area with green meshgrid, the homography fields connected with source image in (b) are calculated. Blue points are corresponding feature points, while yellow points are feature points which appear on source image but not on target image. With the calculated homography field, the part warped from source onto target image is shown in (c). After copying pixels fall on the missing area, the warping result after one iteration is given in (d). Green points are propagated feature points as explained in Section 5.5.2. The final inpainting result is given in (e). In (f), the output frame of the proposed method is compared with results of Grundmann *et al.* [55] and Liu *et al.* [89].

is not deemed an inlier in a subwindow is discarded. The rationale for this procedure is that a frame-global homography is not flexible enough to describe the motion between $\tilde{I}_t$ and $\tilde{I}_s$, thus many genuine matches will be lost if standard RANSAC is applied. For a local region, however, a homography can adequately characterize the feature motions.

5.5.2 Feature propagation

For each blank pixel p_* in $\tilde{I}_t$, a local homography $\mathbf{H}^*_{t,s}$ is estimated between $\tilde{I}_t$ and $\tilde{I}_s$ following (5.8). The source pixel $\mathbf{p}'_*$ is then obtained by warping $\mathbf{p}_*$ to $\tilde{I}_s$. If $\mathbf{p}'_*$ is a defined pixel, its color is copied to $\mathbf{p}_*$. To increase efficiency, $\tilde{I}_t$ can be divided into $X \times Y$ cells, and (5.8) is invoked on the center of the cells. A blank pixel is then warped using the local homography of the cell to which it belongs.

Similar to all feature-based methods, homography fields require good feature matches to produce accurate alignment. This means, however, that blank pixels far away from the defined regions (and available feature matches) may not be mapped optimally to the source image. While projective regularization in homography fields avoids excessively bad warps, ideally there should be feature matches in or close to the blank pixels.

Fortunately, due to the camera movement and the time-based order of choosing source frames in Algorithm 5.1, the blank regions in $\tilde{I}_t$ are usually inpainted in the order of their proximity to the defined regions; see Figure 5.7c. The proposed method exploits this to progressively grow the set of feature matches between the blank regions and other source frames. If a copied pixel $\mathbf{p}'_*$ from $\tilde{I}_s$ is a detected feature, the inpainted pixel $\mathbf{p}_*$ will also be designated as one. The correspondences of $\mathbf{p}'_*$ in the other frames will then be matched to $\mathbf{p}_*$. This ensures that a target blank pixel will always be close to feature matches; see Figure 5.7d.

The proposed feature propagation step is analogous to Matsushita *et al.*'s optical flow-based motion inpainting. Thus, even though feature matches can be transferred to the blank regions as a byproduct, they will not benefit the subsequent motion propagation steps, since the optic flow vectors will not change. Section 5.6.3 will compare both approaches.

5.6 Results

To evaluate the performance of the proposed method, this section has conducted experiments on the video clips used in Grundmann *et al.* [55][2] and Liu *et al.* [89][3]. A variety of scenes and camera motions are contained in these videos. The reader is referred to the supplementary material[4] for the full video results.

Parameters settings for the proposed approach are as follows: the number of neighboring frames in Ω_t in (5.3) is 40; for calculating homography field warps for frame updating (Section 5.3.4) and video inpainting (Section 5.5), the grid size is $X \times Y = 20 \times 20$. Generally $\sigma = 8$ and $\gamma = 0.05$ are used. For video with rolling shutter effect, $\sigma_x = 12$, $\sigma_y = 6$ are used in stead. In the proposed pipeline, motion estimation, smoothing and frame updating take about 0.8s per frame, while inpainting takes $\approx$ 6.4s per frame.

In the following sections, the various components of the proposed pipeline are benchmarked against several state-of-the-art video stabilization approaches: Matsushita *et al.* [98], Gleicher and Liu [51], Liu *et al.* [86], Grundmann *et al.* [55] and Liu *et al.* [89].

5.6.1 Smooth globally warp locally

An underlying premise of the proposed approach is that 2D homographies are sufficient for motion estimation and smoothing. This follows the practice of influential works such as Matsushita *et al.* [98] and Gleicher and Liu [51]. However, the update transform must be more flexible than a standard homography warp, in order to deal with violations to the assumption of pure rotational motions and other distortions. To validate the hypothesis, the following videos have been generated (see supplementary material):

- Video 1: stabilization by temporal local method [98] and frame updating via frame-global homography (5.1).
- Video 2: stabilization by temporal local method [98] and frame updating via homography field (5.5).

[2] http://www.cc.gatech.edu/cpl/projects/rollingshutter/

[3] http://liushuaicheng.org/SIGGRAPH2013/database.html

[4] https://www.youtube.com/playlist?list=PLXqTNhVpuRQhzl6J0_2jBVBlN4wIZuZjt

- Video 3: stabilization by single path smoothing (5.3) and frame updating via frame-global homography (5.1).
- Video 4: stabilization by single path smoothing (5.3) and frame updating via homography field (5.5).

Observe that significant distortions and wobbliness remain in Videos 1 and 3, more so in Video 1 since the temporal local method is not as prolific in smoothing very shaky videos. Good results are obtained in Videos 2 and 4, even though the same homography-based stabilization approaches are used. This supports the intuition that homography chains are sufficient to characterize the camera trajectory, and that the "trick" for successful video stabilization is good frame updating.

Liu *et al.* [89] proposed an approach where a video is divided into a uniform grid of subwindows, and a 2D homography chain is estimated for each subwindow. The multiple chains are then smoothed in a bundled manner. Content preserving warps (CPW) is adapted to conduct frame updating - in short, their approach can be considered "smooth locally warp locally". Video 5 shows the result of [89] on the same input video above. Comparing Videos 4 and 5, it is evident that both the proposed approach can provide the same quality. Practically, however, the proposed approach is simpler and more efficient since only one homography chain needs to be estimated and stabilized. Note that since the proposed pipeline is not tied to a specific stabilization and motion planning algorithm, any homography-based technique can be exploited by the proposed pipeline.

5.6.2 Removing rolling shutter effects

Under the proposed approach, the successful removal of rolling shutter effects depends on the accurate alignment of two neighboring frames I_t and I_{t-1} in the video. This section compares the ability of several spatially varying warps to align neighboring frames from videos captured under rolling shutter. Figures 5.8 – 5.11 show results from standard global homography, CPW [86], homography mixtures [55] and homography fields with horizontal kernel (5.14), labeled as GH, CPW, MH and H-field respectively. It can be observed that, on average, homography fields provide more accurate alignment. This stems from the fact that in most real life videos, the inter-frame camera motion is not pure-rotational, thus the warping function should also handle the potential occurrence

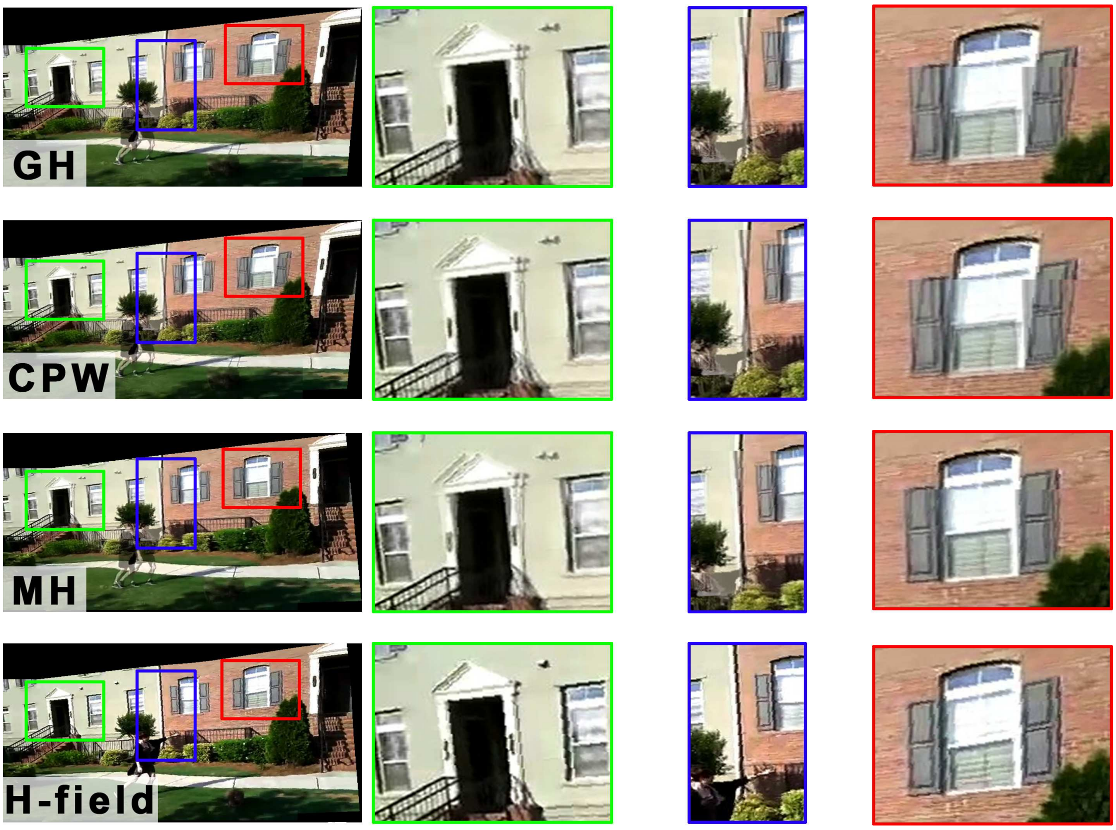


Figure 5.8: Rolling shutter warping result 1.

of depth parallax. The horizontal kernel is proven to be adequate for this purpose. Although homography mixtures can rectify slanted vertical structures (due to rolling shutter), it fails to account for more general distortions. The allowance of non-rigid warping enables CPW to also handle rolling shutter distortions. However, the aim of preserving the rigidity of an overlaid grid structure seems to limit the flexibility of CPW to handle significant parallax and distortions.

Videos 6 to 9 show the usage of the proposed video stabilization pipeline to successfully rectify rolling shutter effects. The input videos were used in Grundmann *et al.*'s work.

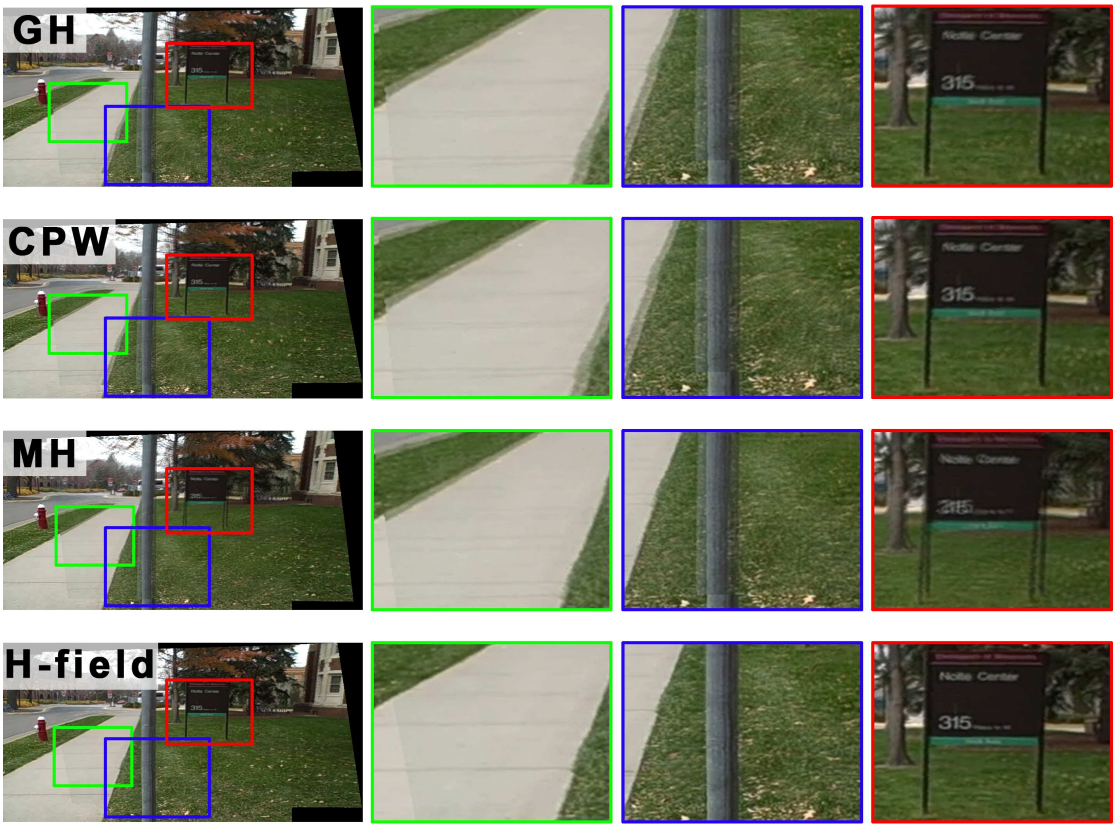


Figure 5.9: Rolling shutter warping result 2.

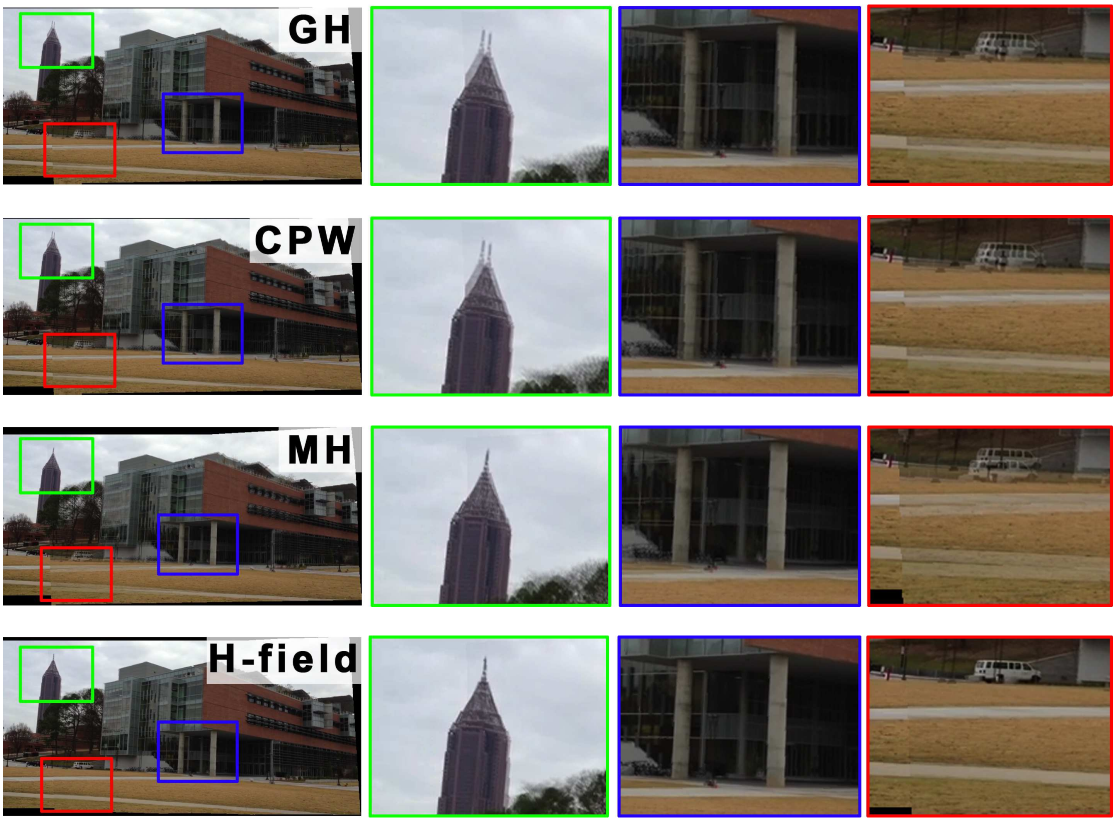


FIGURE 5.10: Rolling shutter warping result 3.

5.6.3 Video inpainting

Since most recent video stabilization approaches do not conduct video inpainting, Matsushita *et al.* [98] remains the state-of-the-art. First, the proposed method is compared with methods that do not conduct inpainting at all [55, 89]. As shown on the top right of Figures 5.12–5.17, the stabilized output frames of such methods must sacrifice significant image contents in order to avoid blank regions in the video. This can clearly show the image area which has been abandoned due to the cropping step and the frame size of the proposed inpainting method. Observe that the amount of discarded contents is larger if the video is stabilized more aggressively (Figures 5.16 and 5.17). This is expected since the stabilized path deviate more from the original trajectory.

If the frame to be inpainted has small blank regions (Figure 5.12; see also the target frames in [98]), both Matsushita *et al.* and the proposed approach can satisfactorily inpaint the video. The target frames in Figures 5.13–5.17 have large blank regions - again, this condition occurs if the video is very shaky and significant amounts of smoothing must be applied. On such challenging cases, it can be observed that Matsushita *et al.*'s

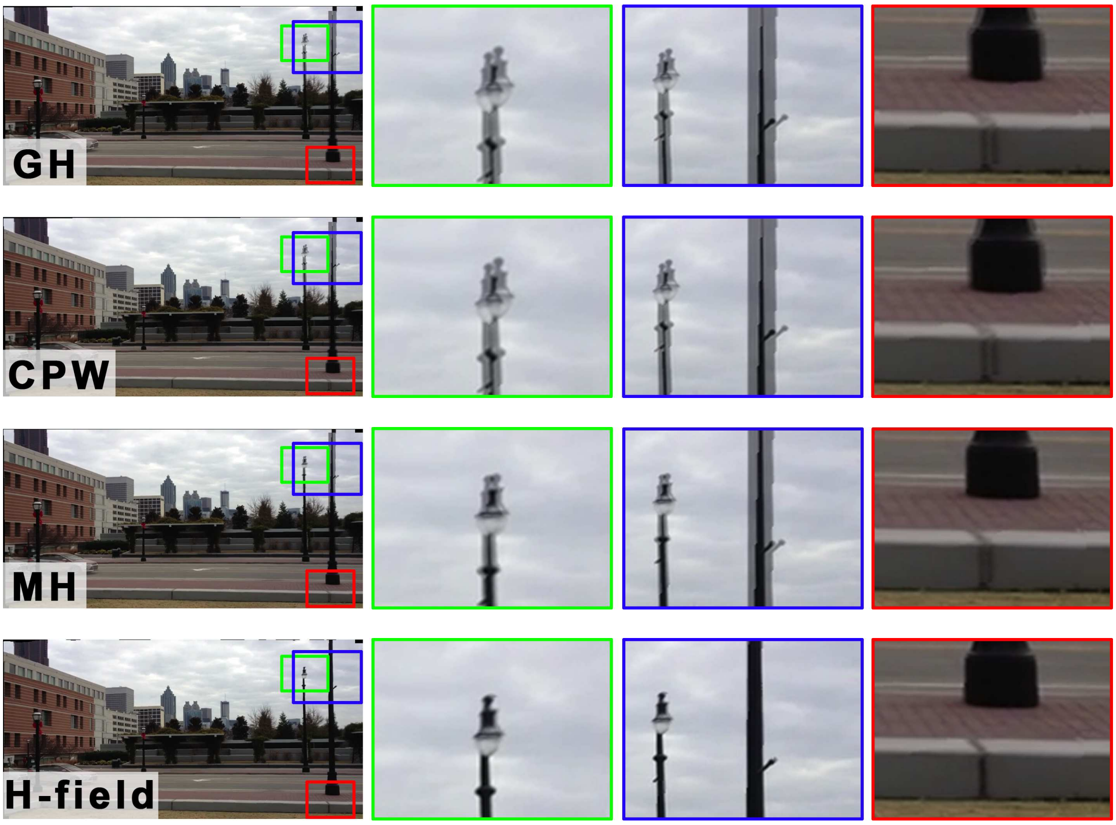


Figure 5.11: Rolling shutter warping result 4.

motion propagation often introduces artifacts in locations far away from the originally defined pixels. Note that the default number of neighboring frames used in [98] is 12, which can inpaint very limited blank region. Instead, neighboring size of 40 was used. In contrast, homography fields can depend on projective regularization and feature propagation for guidance to more accurately fill-in large blank regions.

5.6.4 Overall results

Videos 10 to 26 in the supplementary material are the results generated with the overall video stabilization pipeline. Note that the blue boundaries in the videos mark the image region of stabilized frames before inpainting. To compare results of the proposed method with results of the state-of-the-art methods, please refer to the videos of [89] (also included in the supplementary material[5]) and [54, 55][6] (the latter has been implemented as the video stabilizer on YouTube).

[5] https://www.youtube.com/playlist?list=PLhRzYMNYGcCehIMvvY9vl-y3FOMOlKOmJ

[6] https://www.youtube.com/playlist?list=PLhRzYMNYGcCdF4LPbuiHmMBYhxerNMGB_

However, failure cases have been observed. When the camera is spinning too fast while shooting, or when the scene is too homogeneous, no/insufficient feature tracks can be extracted which leads to the breakage of homography chain. Under such circumstance, the proposed method fails to smooth shaky camera motion.

5.7 Summary

In this chapter, a new 2D video stabilization method that stabilizes globally and warps locally has been proposed. The hypothesis is that global homography is sufficient for motion representation and camera path stabilization. The key to effective video stabilization is the construction of accurate update transforms. To this end, this chapter proposes the usage of homography fields which is a kind of spatially varying warp. Compared with the state-of-the-art methods, the proposed method can generate equally good results with a much simpler pipeline. Based on homography fields, a video inpainting method has been proposed for stabilized videos. With the help of inpainting, users do not need to worry about the cropping ratio and limit the strength of stabilization.

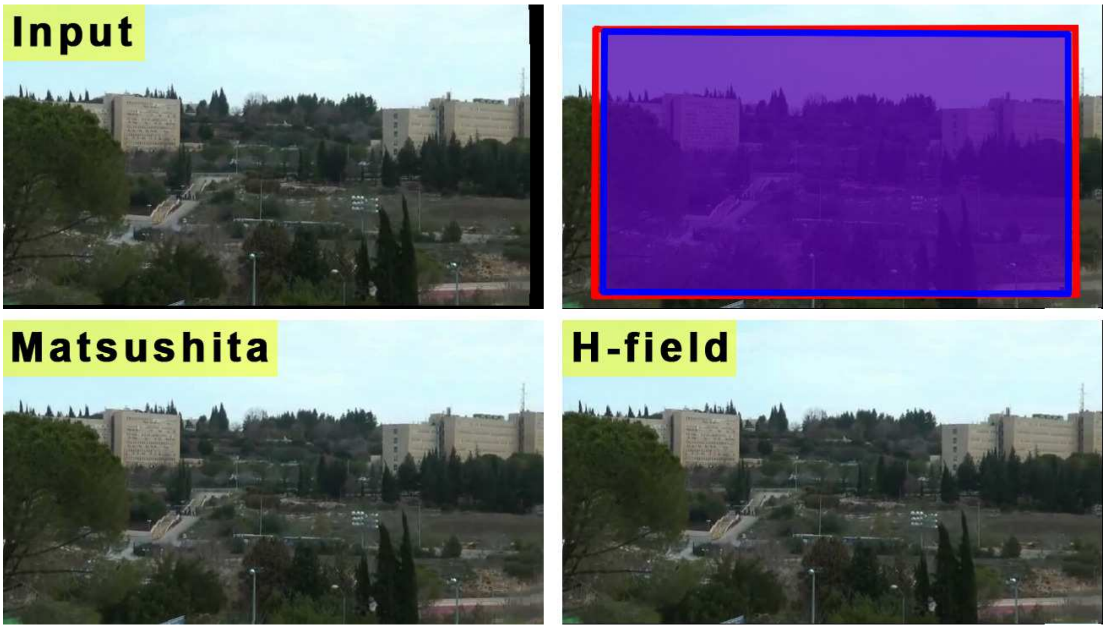


FIGURE 5.12: Selected video inpainting result 1. (top left) Updated input frame $\tilde{I}_t$; (top right) Blue and red shaded windows respectively indicate equivalent final output frames of Grundmann *et al.* [55] and Liu *et al.* [89] who do not conduct video inpainting; (bottom left) Matsushita *et al.* [98]'s result using motion inpainting; (bottom right) Result using homography fields.

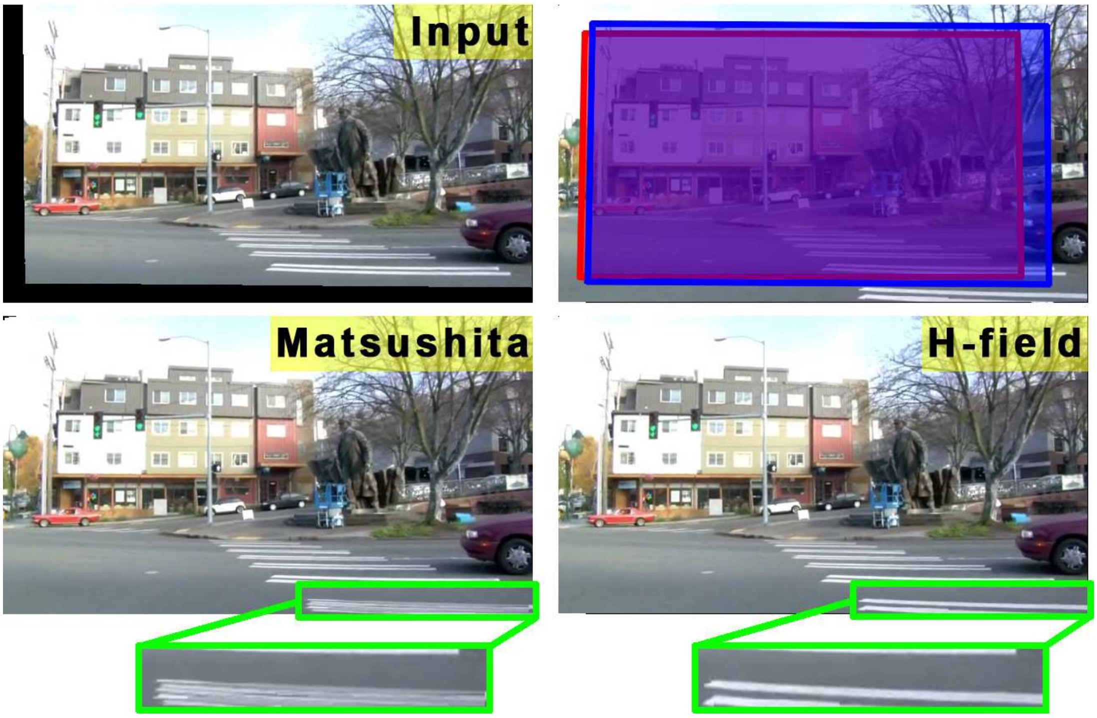


FIGURE 5.13: Selected video inpainting result 2. (top left) Updated input frame $\tilde{I}_t$; (top right) Blue and red shaded windows respectively indicate equivalent final output frames of Grundmann *et al.* [55] and Liu *et al.* [89] who do not conduct video inpainting; (bottom left) Matsushita *et al.* [98]'s result using motion inpainting; (bottom right) Result using homography fields.

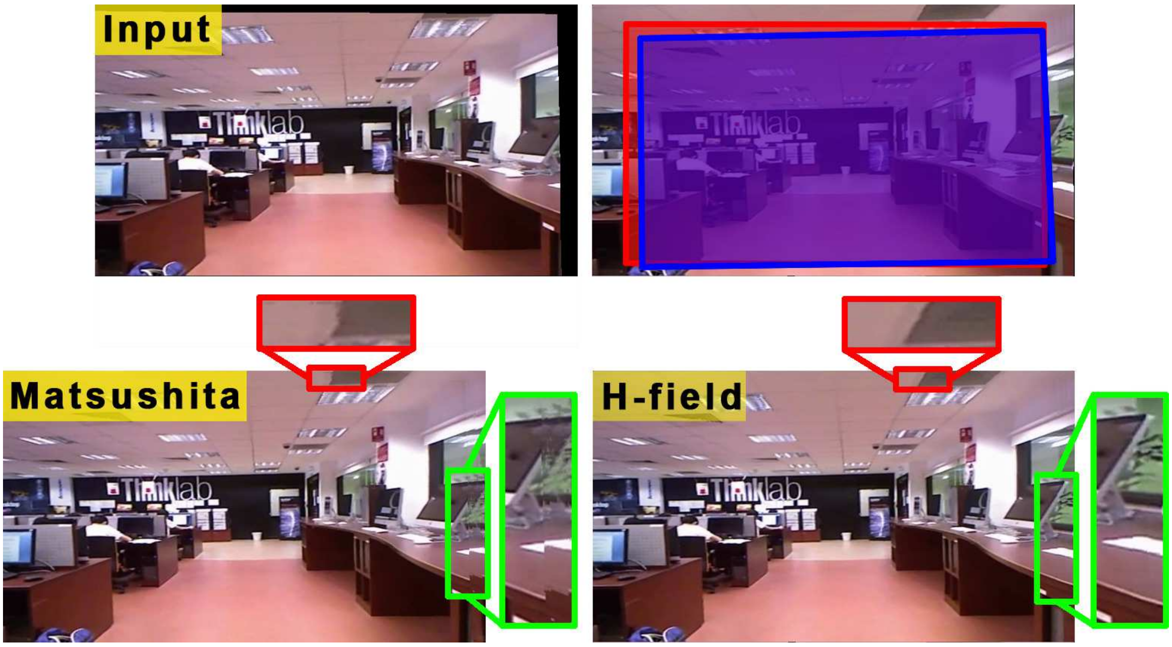


FIGURE 5.14: Selected video inpainting result 3. (top left) Updated input frame $\tilde{I}_t$; (top right) Blue and red shaded windows respectively indicate equivalent final output frames of Grundmann *et al.* [55] and Liu *et al.* [89] who do not conduct video inpainting; (bottom left) Matsushita *et al.* [98]'s result using motion inpainting; (bottom right) Result using homography fields.

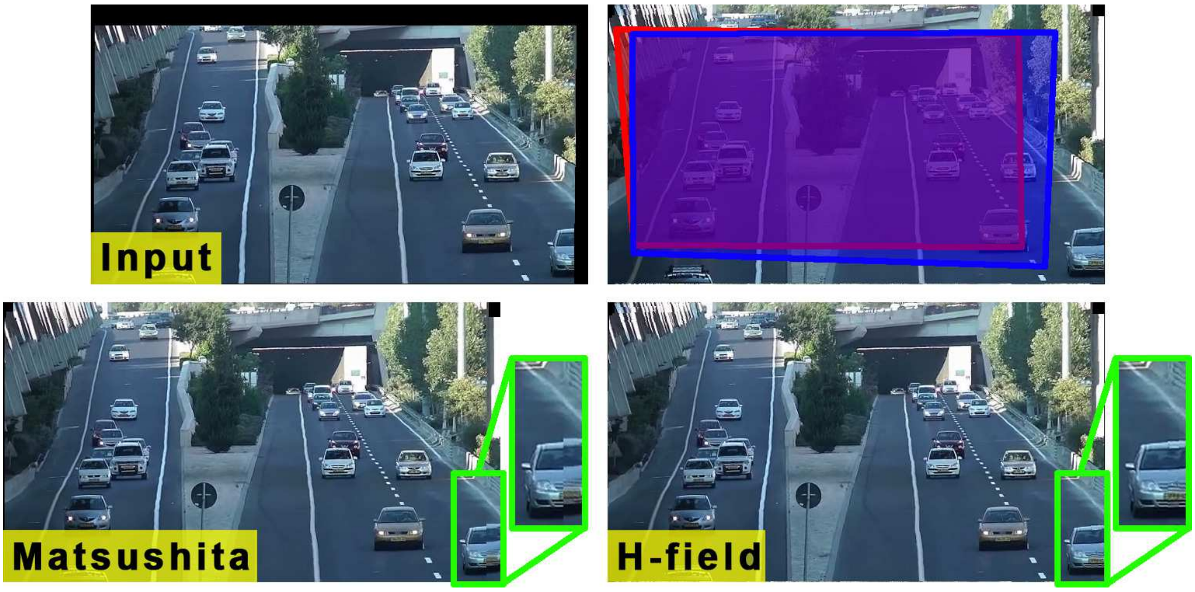


FIGURE 5.15: Selected video inpainting result 4. (top left) Updated input frame $\tilde{I}_t$; (top right) Blue and red shaded windows respectively indicate equivalent final output frames of Grundmann *et al.* [55] and Liu *et al.* [89] who do not conduct video inpainting; (bottom left) Matsushita *et al.* [98]'s result using motion inpainting; (bottom right) Result using homography fields.

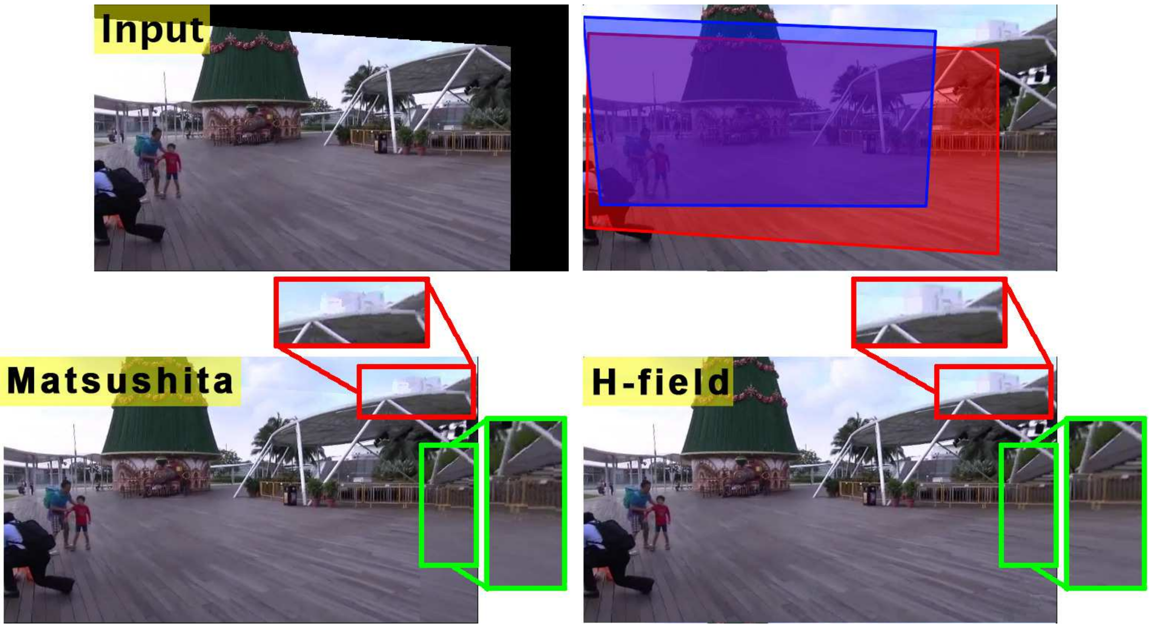


FIGURE 5.16: Selected video inpainting result 5 (same data used in Figure 5.7). (top left) Updated input frame $\tilde{I}_t$; (top right) Blue and red shaded windows respectively indicate equivalent final output frames of Grundmann *et al.* [55] and Liu *et al.* [89] who do not conduct video inpainting; (bottom left) Matsushita *et al.* [98]'s result using motion inpainting; (bottom right) Result using homography fields.

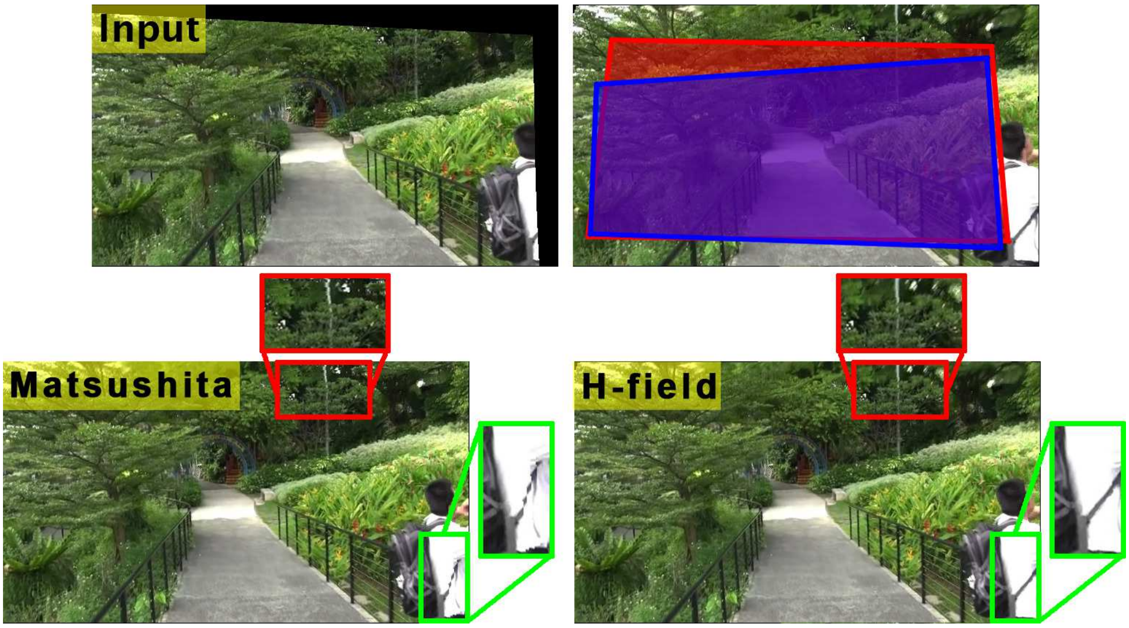


FIGURE 5.17: Selected video inpainting result 6. (top left) Updated input frame $\tilde{I}_t$; (top right) Blue and red shaded windows respectively indicate equivalent final output frames of Grundmann *et al.* [55] and Liu *et al.* [89] who do not conduct video inpainting; (bottom left) Matsushita *et al.* [98]'s result using motion inpainting; (bottom right) Result using homography fields.

Chapter 6

Conclusions and Future Work

Registration is a fundamental task in Computer Vision. It is essential for (a) integrating information taken from different sensors, (b) finding changes in images taken under different conditions, (c) extracting three dimensional information from images, and (d) for model-based object recognition. Among many related research topics, this thesis investigated three of them in Chapter 2: radial distortion estimation (Section 2.3), image stitching (Section 2.4) and video stabilization (Section 2.5).

Apart from background research and literature review, this thesis also made several contributions. First, Chapter 3 treats radial distortion as a violation to the basic epipolar geometry equation and adjusts the epipolar geometry using the framework of moving least squares (MLS). MLS method is extended to allow for epipolar estimation, and is combined with M-estimators to enable robust point matching under severe radial distortion.

Secondly, a correspondence insertion algorithm is proposed for as-projective-as-possible (APAP) warps [140]. The proposed method automatically identifies misaligned regions, and inserts appropriate point correspondences to increase the flexibility of the warp and improve alignment. Chapter 4 has shown how correspondence search can be accomplished for moving direct linear transformation (MDLT). On panoramic mosaicing problems that are challenging, the proposed approach achieves accurate alignment without being handicapped by insufficient feature matches.

Lastly, this thesis proposes homography fields for video stabilization. A homography field is a spatially varying warp that is regularized to be as projective as possible, so as to enable accurate warping while adhering closely to the underlying geometric constraints. Chapter 5 has shown that homography fields are powerful enough to meet the various warping needs of video stabilization, not just in the core step of stabilization, but also in video inpainting. This enables relatively simple algorithms to be used for motion modeling and smoothing.

6.1 Future Work

6.1.1 Radial distortion correction

Due to radial distortion, epipolar lines meant to be straight are "bent" to be curves. Brito *et al.* [18] look for straight lines in all possible epipolar curves and argue that the center of distortion (COD) must be on such a line. Then, the COD can be recovered from the intersection point of straight epipolar lines. Based on this knowledge, the proposed approach can be extended to allow for COD extraction and radial distortion correction, such that accurate 3D information can be extracted from the input images.

6.1.2 Quantitative evaluations on image stitching methods

Chapter 4 has improved upon the original APAP warps, which fails when the overlap region is correspondence-poor. In the experimental result section, qualitative visual evaluations have been conducted, while quantitative evaluations are needed to back up the ability of the proposed method. However, it has been noticed that the quantitative benchmarking technique employed in [140] is not suitable for the proposed method due to the indeterminacy of the number of correspondences. Meanwhile, quantitative evaluation methods tested in [36] also lead to some confusion as visually better results may have worse evaluation scores. Thus, this thesis realizes the need of a robust and abroad suitable quantitative evaluation method on image stitching techniques. Such research will be studied in future work.

6.1.3 Extensions for video stabilization

The proposed video stabilization and inpainting pipeline relies on the existence of sufficient feature matches across successive frames. With denser feature points, the proposed method is able to produce stabilized frames closer to the real scene and inpainted images with less distortion. However, if the feature points are not widely spread across the entire image area, especially around the area with clear and definite geometric structure, the proposed method will not produce very satisfactory results. Although a center insertion method designed for wide-baseline image stitching, with a focus on panoramic stitching, has been presented in Chapter 4, it is not suitable for video stabilization due to the processing speed. To alleviate this problem, possible solutions will be sought for by investigating spatially smooth optic flow methods [88].

Another issue of inpainting in some input videos is the occasional sudden exposure changes between neighboring frames. This yields unsightly “strips” in the inpainted regions. To deal with exposure changes, the exposure normalization schemes [63] will be investigated.

Bibliography

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. “Photographing Long Scenes with Multi-viewpoint Panoramas”. In: *Transactions on Graphics* (2006).

[2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. “Interactive Digital Photomontage”. In: *Transactions on Graphics* (2004).

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. “Computing and Rendering Point Set Surfaces”. In: *Transactions on Visualization and Computer Graphics* (2003).

[4] H. S. Alhichri and M. Kamel. “Virtual Circles: A New Set of Features for Fast Image Registration”. In: *Pattern Recognition Letters* (2003).

[5] N. Amenta and Y. J. Kil. “Defining Point-set Surfaces”. In: *Transactions on Graphics* (2004).

[6] P. R. Andresen and M. Nielsen. “Non-rigid Registration by Geometry-constrained Diffusion”. In: *Medical Image Analysis* (2001).

[7] S. Baker and I. Matthews. “Lucas-Kanade 20 years on: a unifying framework”. In: *International Journal of Computer Vision* (2004).

[8] S. Baker, E. P. Bennett, S. B. Kang, and R. Szeliski. “Removing Rolling Shutter Wobble”. In: *Computer Vision and Pattern Recognition.* 2010.

[9] J. P. Barreto and K. Daniilidis. “Fundamental Matrix for Cameras with Radial Distortion”. In: *International Conference on Computer Vision.* 2005.

[10] A. Bartoli, M. Perriollat, and S. Chambon. “Generalized Thin-plate Spline Warps”. In: *International Journal of Computer Vision* (2010).

[11] A. Bartoli and A. Zisserman. “Direct Estimation of Non-rigid Registration”. In: *British Machine Vision Conference.* 2004.

[12] A. Baumberg. “Reliable Feature Matching Across Widely Separated Views”. In: *Computer Vision and Pattern Recognition.* 2000.

[13] J. S. Beis and D. G. Lowe. “Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces”. In: *Computer Vision and Pattern Recognition.* 1997.

[14] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. “Meshless Methods: An Overview and Recent Developments”. In: *Computer Methods in Applied Mechanics and Engineering* (1996).

[15] Y. Bentoutou, N. Taleb, M. C. E. Mezouar, M. Taleb, and L. Jetto. “An Invariant Approach for Image Registration in Digital Subtraction Angiography”. In: *Pattern Recognition* (2002).

[16] P. J. Besl and H. D. McKay. “A Method for Registration of 3-D Shapes”. In: *Transactions on Pattern Analysis and Machine Intelligence* (1992).

[17] D. Bhattacharya and S. Sinha. “Invariance of Stereo Images via the Theory of Complex Moments”. In: *Pattern Recognition* (1997).

[18] J. H. Brito, R. Angst, K. Köser, and M. Pollefeys. “Radial Distortion Self-Calibration”. In: *Computer Vision and Pattern Recognition.* 2013.

[19] J. Brito, R. Angst, K. Köser, C. Zach, P. Branco, M. Ferreira, and M. Pollefeys. “Unknown Radial Distortion Centers in Multiple View Geometry Problems”. In: *Asian Conference on Computer Vision.* 2013.

[20] J. Brito, C. Zach, K. Köser, M. Ferreira, and M. Pollefeys. “One-sided Radial Fundamental Matrix Estimation”. In: *British Machine Vision Conference.* 2012.

[21] M. Bro-Nielsen and C. Gramkow. “Fast Fluid Registration of Medical Images”. In: *International Conference on Visualization in Biomedical Computing.* 1996.

[22] I. Bronštejn and K. Semendyayev. *Handbook of Mathematics.* Springer Berlin Heidelberg, 1997.

[23] D. Brown. “Close-range Camera Calibration”. In: *Photogrammetric Engineering* (1971).

[24] L. G. Brown. “A Survey of Image Registration Techniques”. In: *Computing Surveys* (1992).

[25] M. Brown and D. Lowe. “Automatic panoramic image stitching using invariant features”. In: *International Journal of Computer Vision* (2007).

[26] M. Brown, R. Szeliski, and S. Winder. “Multi-image Matching Using Multi-scale Oriented Patches”. In: *Computer Vision and Pattern Recognition.* 2005.

[27] T. Brox and J. Malik. “Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2011).

[28] P. J. Burt and E. H. Adelson. “A Multiresolution Spline with Application to Image Mosaics”. In: *Transactions on Graphics* (1983).

[29] S. C. Cain, M. M. Hayat, and E. E. Armstrong. “Projection-based Image Registration in the Presence of Fixed-pattern Noise”. In: *Transactions on Image Processing* (2001).

[30] T.-J. Cham and R. Cipolla. “Automated B-spline Curve Representation Incorporating MDL and Error-minimizing Control Point Insertion Strategies”. In: *Transactions on Pattern Analysis and Machine Intelligence* (1999).

[31] C.-H. Chang, Y. Sato, and Y.-Y. Chuang. “Shape-preserving Half-projective Warps for Image Stitching”. In: *Computer Vision and Pattern Recognition.* 2014.

[32] W.-H. Cho and K.-S. Hong. “Affine Motion Based CMOS Distortion Analysis and CMOS Digital Image Stabilization”. In: *Transactions on Consumer Electronics* (2007).

[33] D. Claus and A. W. Fitzgibbon. “A Rational Function Lens Distortion Model for General Cameras”. In: *Computer Vision and Pattern Recognition.* 2005.

[34] X. Dai and S. Khorram. “Development of a Feature-based Approach to Automated Image Registration for Multitemporal and Multisensor Remotely Sensed Imagery”. In: *Geoscience and Remote Sensing.* 1997.

[35] F. Devernay, O. Faugeras, and I. S. Antipolis. “Automatic Calibration and Removal of Distortion From Scenes of Structured Environments”. In: *SPIE.* 1995.

[36] V. Dissanayake, S. Herath, S. Rasnayaka, S. Seneviratne, R. Vidanaarachchi, and C. Gamage. “Quantitative and Qualitative Evaluation of Performance and Robustness of Image Stitching Algorithms”. In: *International Conference on Digital Image Computing: Techniques and Applications.* 2015.

[37] F. Dornaika and R. Chung. “Mosaicking Images with Parallax”. In: *Signal Processing: Image Communication* (2004).

[38] A. Eden, M. Uyttendaele, and R. Szeliski. “Seamless Image Stitching of Scenes with Large Motions and Exposure Differences”. In: *Computer Vision and Pattern Recognition.* 2006.

[39] W. Faig. “Close-range Precision Photogrammetry for Industrial Purposes”. In: *Photogrammetria* (1981).

[40] O. Faugeras. “What Can be Seen in Three Dimensions with an Uncalibrated Stereo Rig?” In: *European Conference on Computer Vision.* 1992.

[41] O. Faugeras and G. Toscani. “The Calibration Problem for Stereo”. In: *Computer Vision and Pattern Recognition.* 1986.

[42] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. “Camera Self-calibration: Theory and Experiments”. In: *European Conference on Computer Vision.* 1992.

[43] M. A. Fischler and R. C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* (1981).

[44] A. Fitzgibbon. “Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion”. In: *Computer Vision and Pattern Recognition.* 2001.

[45] P. E. Forssen and E. Ringaby. “Rectifying Rolling Shutter Video from Hand-held Devices”. In: *Computer Vision and Pattern Recognition.* 2010.

[46] W. Förstner. “A Feature Based Correspondence Algorithm for Image Matching”. In: *International Archive of Photogrammetry and Remote Sensing* (1986).

[47] Y. Furukawa and J. Ponce. “Accurate, Dense, and Robust Multiview Stereopsis”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2010).

[48] S. Ganapathy. “Decomposition of transformation matrices for robot vision”. In: *Robotics and Automation.* 1984.

[49] J. Gao, S. J. Kim, and M. S. Brown. “Constructing Image Panoramas Using Dual-homography Warping”. In: *Computer Vision and Pattern Recognition.* 2011.

[50] D. Gennery. “Stereo-camera calibration”. In: *Image Understanding Workshop.* 1979.

[51] M. L. Gleicher and F. Liu. “Re-cinematography: Improving the Camerawork of Casual Video”. In: *Transactions on Multimedia Computing, Communications, and Applications* (2008).

[52] A. Goldstein and R. Fattal. “Video Stabilization Using Epipolar Geometry”. In: *Transactions on Graphics* (2012).

[53] V. Govindu, C. Shekhar, and R. Chellappa. “Using Geometric Properties for Correspondence-less Image Alignment”. In: *International Conference on Pattern Recognition.* 1998.

[54] M. Grundmann, V. Kwatra, and I. Essa. “Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths”. In: *Computer Vision and Pattern Recognition.* 2011.

[55] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. “Effective Calibration Free Rolling Shutter Removal”. In: *International Conference on Computational Photography* (2012).

[56] J. Harel, C. Koch, and P. Perona. “Graph-Based Visual Saliency”. In: *Neural Information Processing Systems.* 2006.

[57] C. Harris and M. Stephens. “A Combined Corner and Edge Detector”. In: *Alvey Vision Conference.* 1988.

[58] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Second. Cambridge University Press, 2004.

[59] R. Hartley. “In Defense of the Eight-point Algorithm”. In: *Transactions on Pattern Analysis and Machine Intelligence* (1997).

[60] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* 2nd. Springer, 2008.

[61] J. Heikkila and O. Silven. “A Four-step Camera Calibration Procedure with Implicit Image Correction”. In: *Computer Vision and Pattern Recognition.* 1997.

[62] Y. C. Hsieh, D. M. McKeown, and F. P. Perlant. "Performance Evaluation of Scene Registration and Stereo Matching for Cartographic Feature Extraction". In: *Transactions on Pattern Analysis and Machine Intelligence* (1992).

[63] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim. "Color transfer using probabilistic moving least squares". In: *CVPR*. 2014.

[64] J. Jia and C.-K. Tang. "Image Stitching Using Structure Deformation". In: *Transactions on Pattern Analysis and Machine Intelligence* (2008).

[65] T. Kadir and M. Brady. "Saliency, Scale and Image Description". In: *International Journal of Computer Vision* (2001).

[66] E. Y. Kang, I. Cohen, and G. Medioni. "A Graph-based Global Registration for 2D Mosaics". In: *International Conference on Pattern Recognition*. 2000.

[67] S. B. Kang. "Catadioptric Self-calibration". In: *Computer Vision and Pattern Recognition*. 2000.

[68] S. B. Kang. *Semi-automatic Methods for Recovering Radial Distortion Parameters from a Single Image*. 1997.

[69] J. Kannala and S. S. Brandt. "Quasi-Dense Wide Baseline Matching Using Match Propagation". In: *Computer Vision and Pattern Recognition*. 2007.

[70] Z. Kukelova, M. Byröd, K. Josephson, T. Pajdla, and K. ström. "Fast and Robust Numerical Solutions to Minimal Problems for Cameras with Radial Distortion". In: *Computer Vision and Image Understanding* (2010).

[71] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting: An Introduction*. Computational Mathematics and Applications. Academic Press, 1986.

[72] P. Lancaster and K. Salkauskas. "Surfaces Generated by Moving Least Squares Methods". In: *Mathematrics of Computation* (1981).

[73] I.-K. Lee. "Curve Reconstruction from Unorganized Points". In: *Computer Aided Geometric Design* (2000).

[74] D. Levin. "Mesh-Independent Surface Interpolation". In: *Geometric Modeling for Scientific Visualization*. 2004.

[75] D. Levin. "The Approximation Power of Moving Least-squares". In: *Math. Comput.* (1998).

[76] H. Li and R. Hartley. “A Non-iterative Method for Correcting Lens Distortion from Nine-Point Correspondences”. In: *International Conference on Computer Vision Workshop*. 2005.

[77] H. Li, B. S. Manjunath, and S. K. Mitra. “A Contour-based Approach to Multisensor Image Registration”. In: *Transactions on Image Processing* (1995).

[78] S. Z. Li, J. Kittler, and M. Petrou. “Matching and Recognition of Road Networks from Aerial Images”. In: *European Conference on Computer Vision*. 1992.

[79] C.-K. Liang, L.-W. Chang, and H. H. Chen. “Analysis and Compensation of Rolling Shutter Effect”. In: *Transactions on Image Processing* (2008).

[80] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. “Smoothly Varying Affine Stitching”. In: *Computer Vision and Pattern Recognition*. 2011.

[81] T. Lindeberg. “Scale-space for Discrete Signals”. In: *Transactions on Pattern Analysis and Machine Intelligence* (1990).

[82] T. Lindeberg and J. Gårding. “Shape-adapted Smoothing in Estimation of 3-D Shape Cues from Affine Deformations of Local 2-D Brightness Structure”. In: *Image and Vision Computing* (1997).

[83] A. Litvin, J. Konrad, and W. C. Karl. “Probabilistic Video Stabilization Using Kalman Filtering and Mosaicking”. In: *SPIE*. 2003.

[84] C. Liu, W. Freeman, and E. Adelson. “Beyond Pixels: Exploring New Representations and Applications for Motion Snalysis”. PhD thesis. Massachusetts Institute of Technology, 2009.

[85] C. Liu, J. Yuen, and A. Torralba. “SIFT Flow: Dense Correspondence across Scenes and Its Applications”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2011).

[86] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. “Content-preserving Warps for 3D Video Stabilization”. In: *SIGGRAPH*. 2009.

[87] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. “Subspace Video Stabilization”. In: *Transactions on Graphics* (2011).

[88] S. Liu, Y. Ping, P. Tan, and J. Sun. “SteadyFlow: spatially smooth optical flow for video stabilization”. In: *CVPR*. 2014.

[89] S. Liu, L. Yuan, P. Tan, and J. Sun. “Bundled camera Paths for Video Stabilization”. In: *Transactions on Graphics* (2013).

[90] W. X. Liu, T.-J. Chin, G. Carneiro, and D. Suter. “Point Correspondence Validation under Unknown Radial Distortion”. In: *International Conference on Digital Image Computing: Techniques and Applications*. 2013.

[91] H. C. Longuet-Higgins. “A Computer Algorithm for Reconstructing a Scene from Two Projections”. In: *Nature*. 1981.

[92] D. G. Lowe. “Object Recognition from Local Scale-invariant Features”. In: *International Conference on Computer Vision*. 1999.

[93] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* (2004).

[94] J. R. Magnus. “On Differentiating Eigenvalues and Eigenvectors”. In: *Econometric Theory* (1985).

[95] S. Marsland and C. J. Twining. “Constructing Data-driven Optimal Representations for Iterative Pairwise Non-rigid Registration”. In: *Second International Workshop on Biometric Image Registration*. 2003.

[96] R. Marzotto, A. Fusiello, and V. Murino. “High Resolution Video Mosaicing with Global Alignment”. In: *Computer Vision and Pattern Recognition*. 2004.

[97] J. Matas, T. Obdrzalek, and O. Chum. “Local Affine Frames for Wide-baseline Stereo”. In: *International Conference on Pattern Recognition*. 2002.

[98] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. “Full-Frame Video Stabilization with Motion Inpainting”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2006).

[99] S. J. Maybank and O. D. Faugeras. “A Theory of Self-calibration of a Moving Camera”. In: *International Journal of Computer Vision* (1992).

[100] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. “A Comparison of Affine Region Detectors”. In: *International Journal of Computer Vision* (2005).

[101] K. Mikolajczyk and C. Schmid. “Scale & Affine Invariant Interest Point Detectors”. In: *International Journal of Computer Vision* (2004).

[102] C. Morimoto and R. Chellappa. "Evaluation of Image Stabilization Algorithms". In: *International Conference on Acoustics, Speech and Signal Processing.* 1998.

[103] S. Moss and E. R. Hancock. "Multiple Line-template Matching with the {EM} Algorithm". In: *Pattern Recognition Letters* (1997).

[104] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. "Point Based Animation of Elastic, Plastic and Melting Objects". In: *SIGGRAPH/Eurographics Symposium on Computer Animation.* 2004.

[105] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras.* CRC Press, Inc., 2005.

[106] S. A. Nene and S. K. Nayar. "A Simple Algorithm for Nearest Neighbor Search in High Dimensions". In: *Transactions on Pattern Analysis and Machine Intelligence* (1997).

[107] D. Nister and H. Stewenius. "Scalable Recognition with a Vocabulary Tree". In: *Computer Vision and Pattern Recognition.* 2006.

[108] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. "Multi-level Partition of Unity Implicits". In: *SIGGRAPH.* 2003.

[109] P. Pérez, M. Gangnet, and A. Blake. "Poisson Image Editing". In: *Transactions on Graphics* (2003).

[110] E. Ringaby and P.-E. Forssen. "Efficient Video Rectification and Stabilisation for Cell-Phones". In: *International Journal of Computer Vision* (2012).

[111] A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* 2nd. Academic Press, Inc., 1982.

[112] P. J. Rousseeuw and M. Hubert. "Robust Statistics for Outlier Detection". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2011).

[113] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. "Nonrigid Registration Using Free-form Deformations: Application to Breast MR Images". In: *Transactions on Medical Imaging* (1999).

[114] M. D. Sambora and R. K. Martin. "Improving the Performance of Projection-Based Image Registration". In: *Aerospace Conference.* 2008.

[115] S. Schaefer, T. McPhail, and J. Warren. "Image Deformation Using Moving Least Squares". In: *Transactions on Graphics* (2006).

[116] F. Schaffalitzky and A. Zisserman. “Multi-view Matching for Unordered Image Sets”. In: European Conference on Computer Vision. 2002.

[117] G. Shakhnarovich, P. Viola, and T. Darrell. “Fast Pose Estimation with Parameter-Sensitive Hashing”. In: *International Conference on Computer Vision.* 2003.

[118] D. Shepard. “A Two-dimensional Interpolation Function for Irregularly-spaced Data”. In: *National Conference.* 1968.

[119] J. Shi and C. Tomasi. “Good features to track”. In: *Computer Vision and Pattern Recognition.* 1994.

[120] H.-Y. Shum and R. Szeliski. “Construction of Panoramic Mosaics with Global and Local Alignment”. In: *International Journal of Computer Vision* (2000).

[121] G. Stockman, S. Kopstein, and S. Benett. “Matching Images to Models for Registration and Object Detection via Clustering”. In: *Transactions on Pattern Analysis and Machine Intelligence* (1982).

[122] N. Sundaram, T. Brox, and K. Keutzer. “Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow”. In: *European Conference on Computer Vision.* 2010.

[123] R. Swarninathan and S. K. Nayar. “Non-metric Calibration of Wide-angle Lenses and Polycameras”. In: *Computer Vision and Pattern Recognition.* 1999.

[124] R. Szeliski. *Image Alignment and Stitching: A Tutorial.* TechReport MSR-TR-2004-92. Microsoft Research, 2004.

[125] R. Szeliski and J. Coughlan. “Spline-based Image Registration”. In: *International Journal of Computer Vision* (1997).

[126] R. Szeliski. “Image Alignment and Stitching: A Tutorial”. In: *Foundations and Trends in Computer Graphics and Vision* (2006).

[127] B. Triggs. “Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes”. In: *European Conference on Computer Vision.* 2004.

[128] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. “Bundle Adjustment - A Modern Synthesis”. In: *International Conference on Computer Vision Workshop.* 2000.

[129] R. Tsai. “A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses”. In: *Journal of Robotics and Automation* (1987).

[130] T. Tuytelaars and L. Van Gool. “Matching Widely Separated Views Based on Affine Invariant Regions”. In: *International Journal of Computer Vision* (2004).

[131] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/. 2008.

[132] C. ye Wang, H. Sun, S. Yada, and A. Rosenfeld. “Some Experiments in Relaxation Image Matching Using Corner Features”. In: *Pattern Recognition* (1983).

[133] W.-H. Wang and Y.-C. Chen. “Image Registration by Control Points Pairing Using the Invariant Properties of Line Segments”. In: *Pattern Recognition Letters* (1997).

[134] G. Q. Wei and S. D. Ma. “A Complete Two-plane Camera Calibration Method and Experimental Comparisons”. In: *International Conference on Computer Vision*. 1993.

[135] P. Wen. “Medical Image Registration Based-on Points, Contour and Curves”. In: *International Conference on BioMedical Engineering and Informatics*. 2008.

[136] J. Weng, P. Cohen, and M. Herniou. “Camera Calibration with Distortion Models and Accuracy Evaluation”. In: *Transactions on Pattern Analysis and Machine Intelligence* (1992).

[137] Y. Wexler, E. Shechtman, and M. Irani. “Space-time Completion of Video”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2007).

[138] W. Xu and J. Mulligan. “Performance evaluation of color correction approaches for multi-view image and video stitching”. In: *Computer Vision and Pattern Recognition*. 2010.

[139] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi. “Robust and Fast Motion Estimation for Video Completion”. In: *Machine Vision Applications*. 2013.

[140] J. Zaragoza, T.-J. Chin, Q.-H. Tran, M. S. Brown, and D. Suter. “As-Projective-As-Possible Image Stitching with Moving DLT”. In: *Transactions on Pattern Analysis and Machine Intelligence* (2014).

[141] F. Zhang and F. Liu. “Parallax-tolerant Image Stitching”. In: *Computer Vision and Pattern Recognition.* 2014.

[142] Z. Zhang. “On the Epipolar Geometry Between Two Images With Lens Distortion”. In: *International Conference on Pattern Recognition.* 1996.

[143] B. Zitová and J. Flusser. “Image Registration Methods: A Survey”. In: *Image and Vision Computing* (2003).