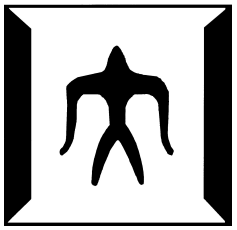
論文 / 著書情報
Article / Book Information

題目(和文)	色間位置合わせと柔軟なモザイキングのための非剛体面像レジストレーション
Title(English)	Non-Rigid Image Registration for Inter-Color Alignment and Flexible Mosaicing
著者(和文)	C.SOUZARAFAEL H.
Author(English)	RAFAEL HENRIQUE C DE SOUZA
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9586号, 授与年月日:2014年4月30日, 学位の種別:課程博士, 審査員:奥富 正敏,蜂屋 弘之,大山 真司,塚越 秀行,倉林 大輔
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9586号, Conferred date:2014/4/30, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Non-Rigid Image Registration for Inter-Color Alignment and Flexible Mosaicing



Rafael Henrique Castanheira de Souza

Department of Mechanical and Control Engineering

Okutomi and Tanaka Lab

Tokyo Institute of Technology

A thesis submitted for the degree of

Doctor of Engineering

January 30th, 2014

Acknowledgements

First of all, I would like to express my gratitude towards my thesis supervisor, Professor Masatoshi Okutomi. His guidance was a key factor for the success of our research. I thank him for accepting me in his laboratory and for providing me an excellent working environment.

I would like to thank Assistant Professor Akihiko Torii for his guidance and orientations for the redaction of my papers and thesis.

I would like to thank Professor Masao Shimizu for guiding me in the first steps of my research.

I word like to express my gratitude towards the thesis committee members: Professor Hiroyuki Hachiya, Professor Shinji Ohyama, Professor Hideyuki Tsukagoshi, and Professor Daisuke Kurabayashi. Thank you very much for your participation.

I would like to thank all my colleagues who worked with me.

I also would like to thank the Monbu-kagaku-shō for their financial support, without which my coming to Japan would have been impossible.

Last but not least, I would like to thank my family, which always supported my education and encouraged me to come to study in Japan. Finally, I thank my beloved wife, who has been by my side supporting me since the beginning of my research.

To all, my eternal gratitude.

Abstract

Image registration is the process of aligning the coordinate systems of two or more images. This technique is used for many applications in various fields, *e.g.* medical imaging, cartography, and robotics. Performing image registration for general purposes often requires high dimensional image deformation models because the images must be significantly deformed in order to register into a single image. We refer to image registration using high dimensional deformation models as non-rigid registration. As is the case with most computer-science problems, dealing with the computation in high dimensional spaces leads to difficulties on accuracy and efficiency.

In this dissertation, we tackle these essential problems in two different scenarios by proposing novel approaches tailored for each of them. First, we aim at performing accurate registration of misaligned color channels of images captured asynchronously. A practical example of such a registration problem is the inter-color alignment of time-sampled endoscopic images for medical imaging applications, a problem which requires high accuracy of the registration. Second, we aim at the efficient registration for mosaicing of images captured under flexible photographing, i.e. we use the images captured under non-restricted camera motion w.r.t. scenes. In this topic, we focus on efficient mosaicing. We compose the registered images and produce high quality mosaics in real time.

Regarding inter-color alignment, we tackle an instance of non-rigid registration problem that appears in time-sampled images. These are RGB images generated by video, with each frame capturing only one color channel of the RGB space. Each final RGB image is composed

by interpolating adjacent frames. However, if there is camera motion, color artifacts appear due to misalignment of the color channels. We propose a method to remove the color artifacts by using an efficient formulation of joint-entropy to align the color channels. Because the joint-entropy alone presents many local optima, we combined it with an operator which uses the orientation of color gradients of the channels. Compared to related methods, our proposed method is precise while still being efficient.

Our second scenario, mosaicing, is a very popular application of image registration. Despite of the possible advantages, non-rigid registration is almost never applied due to its computational cost. We propose a registration algorithm that can alleviate this problem in different scenarios. In the case of video mosaicing, we propose a registration method that can align in real-time pairs of frames selected from the input video. Our proposed method avoids the error accumulation that may happen while performing this kind of task. The real-time performance is achieved by using a registration formulation described by a sparse linear system that can be solved efficiently. When composing a mosaic, the aligned images must be combined to generate a seamless result. This process is called *stitching*. A common algorithm used during stitching is graph cut, an approach that uses graph algorithms to decide which pixels will be part of the mosaic and which pixels will be discarded. Using this method is desirable because of the quality of the resulting mosaic but, because it works pixel wise, it is not suited for real-time mosaicing. To solve this problem, we propose an efficient graph cut formulation that acts on the mesh triangles used by the non-rigid registration model. Since the number of triangles is much smaller than the number of pixels, our proposed method is faster and can be used for real-time video processing. In addition, we propose an extension of our video mosaicing method. This registration method is capable of registering a set of images while keeping the alignment globally consistent. This method is intended to be used as post-processing.

All the methods were evaluated experimentally, and at the end of this dissertation we present our conclusions and consider possible future venues of research.

Contents

List of Figures

Chapter 1

Introduction

1.1 Image registration and deformation models

A photographer may wish to stitch together some photos to create a panorama image. A physician may wish to analyze medical images. A cameraperson may wish to stabilize a video taken from a shaking camera. A cartographer may wish to create a map from a set of aerial photos. In all cases there is a need to align images. The process of aligning two or more images is called image registration. In the case of two images, it can be conceived as trying to position photos taken from the same scene one over the other so that the overlapping regions are as similar as possible. One of the photos is kept fixed and used as reference. The shape of the other photos is deformed using some model (which depends on the class of application) in order to attain the best alignment. Figure 1.1 shows an example of image registration.

The process of image registration is normally modeled as an optimization problem. Each input image has a coordinate system, and a deformation model used to map the images into some reference coordinate system, generating the alignment. A warping function is what defines a deformation model. It is a function of the form $p' = w(\theta, p)$ that maps a coordinate p from its original coordinate system into another 2D coordinate p'. If two coordinates p_1 and p_2, in two different images represent the same scene feature, then ideally p_1 and p_2 should be warped to the same coordinate after the registration is performed. This is illustrated in Figure 1.2. Warping functions can be broadly divided into two

classes: rigid and non-rigid.

In the case of rigid functions, w has a set of warping parameters θ with a constant number of parameters. Examples of rigid warping function are translation (2 parameters), Euclidean (3 parameters), similarity (4 parameters), affine (6 parameters), and homography (8 parameters). Figure 1.3 shows examples of rigid functions.

Under some circumstances, rigid functions are not enough to guarantee a good alignment. For these cases, non-rigid warping functions were developed. Non-rigid warping functions are of the form $w\left(\theta(p), p\right)$. Note that $\theta\left(p\right)$ is a function of a coordinate p since the warping parameters can vary for each coordinate p. This non-rigid warping function allows much more flexibility. However, non-rigid warping functions have some severe limitations which our research aims to alleviate.

In order to solve the registration problem, the set or function θ that yields the minimum misalignment must be chosen. Functions that measure the misalignment are called energy function. They commonly take a given θ as input and outputs a real value that indicates how good the alignment is (the lower the better).

In this dissertation we will tackle two of the main complications related to non-rigid image registration: precision and computational cost. The cause of these problems is, in short, the high dimensionality of the non-rigid warping functions. We will tackle the precision problem applied to the alignment of color channels of RGB images. The efficiency problem will be tackled in mosaicing, a classical image registration application.

The next section discusses in more details application specific problems that arise in image registration.

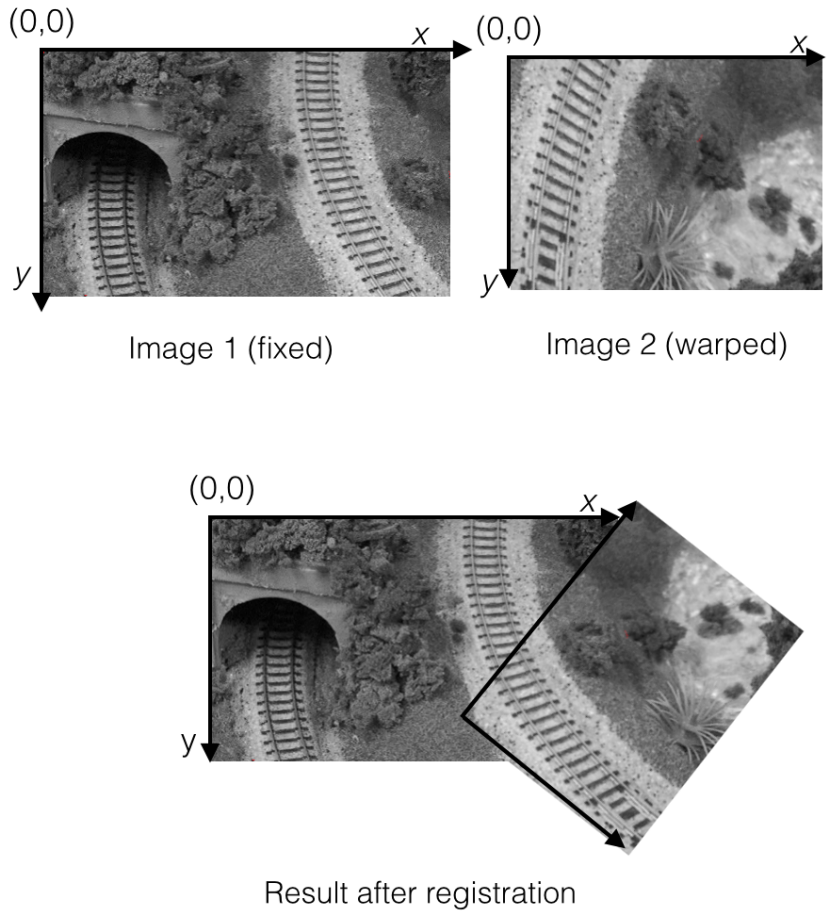


Figure 1.1: Example of image registration. The image in the left is kept fixed while the image in the right is transformed. The objective is to align the regions present in both images.

1.2 Application-specific image registration approaches

As previously explained, we must define a warping function and an energy function in order to perform image registration. The design of these functions, however, is application specific: There is no energy or warping functions that will have optimal performance for all possible applications.

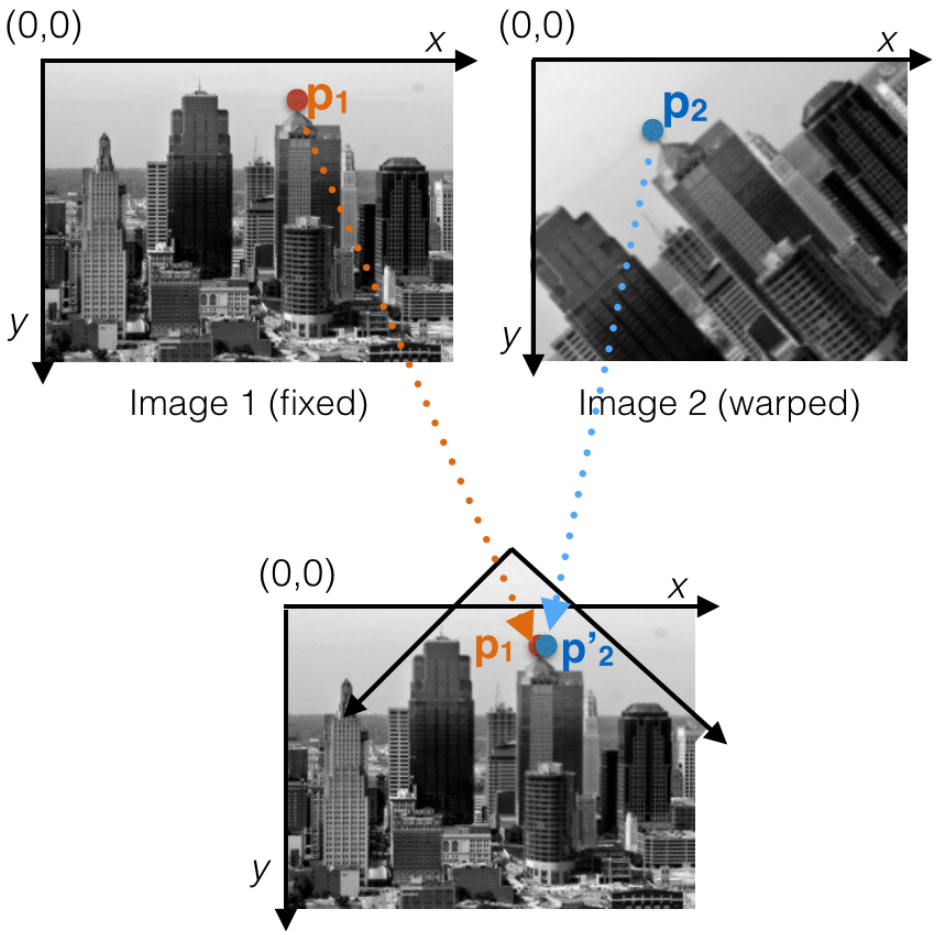


Figure 1.2: Two coordinates, p_1 and p_2, represent the same scene feature (the top of a building). After a successful registration, p_2 is warped into p_1.

The most adequate class of warping function depends mostly on the geometric relationship between the input images. Consider, for example, the case of a set of photos of the night sky that are to be aligned to create a star map. Since the celestial bodies are far away, the photos are related to each other by similarity transformation: two photos with overlapping regions can be aligned by scaling, rotation, and translation alone. Now, when generating a panorama image with photos from a camera rotating around its optical center, any two overlapping images can be aligned into each other by homography. Both similarity and homography are rigid warping functions. Now consider a medical researcher who

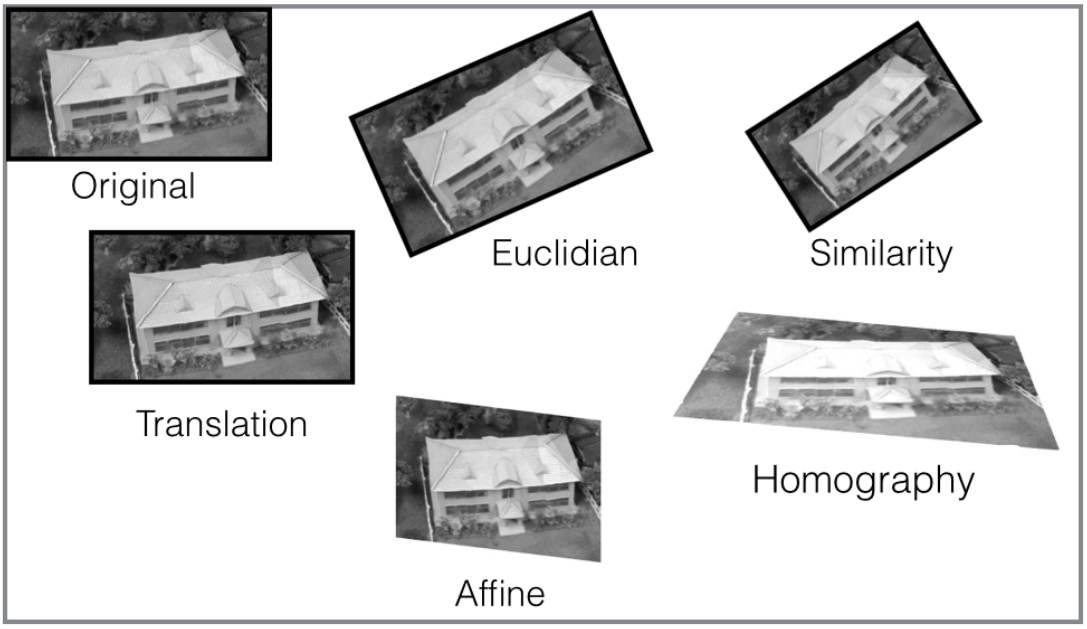


Figure 1.3: Example of rigid warping functions.

needs to align brains of different people in order to compare them. The brains cannot be aligned by any rigid function, because the brains of different people have regions with different sizes. Thus, in this case, non-rigid deformation models must be applied.

In the case of energy functions, the selection depends mostly on the nature of the images being registered. Energy functions can be broadly divided into two groups: area-based and feature-based.

Area-based energy functions work by analyzing aligned pixels (*i.e.*, pixels sharing the same coordinate after registration). How this analysis is done depends on the characteristics of the images. If the images have the same spectral band (*e.g.* photos taken using the same class of sensor), then a simple sum of squared differences (*SSD*) Szeliski [2006] can be used to evaluate the quality of the alignment. This is possible because the aligned pixels will have approximately the same value. However, when the images are captured in different spectral bands, aligned pixels may have totally different values. In this case, correlation methods are commonly applied, because the aligned pixel values will show strong correlation when the

images are properly registered.

Instead of comparing pixels, feature-based energy functions work by comparing feature points of the input images. Feature points are coordinates that represent scene features which are easy to detect and identify. Initially, feature points are detected in both input images. This is done using a feature detector algorithm, such as SURF Bay et al. [2006]. The feature detector outputs the coordinates and the descriptor vector of each detected feature. The features of one image are then matched to the features of the other by comparing their descriptors. For doing so, a variety of nearest-neighbor method is applied. The matched features are then used to evaluate the registration result. As shown in Figure 1.2, two matched features representing the same scene structure in both images will be, ideally, mapped to the same coordinate after registration. This information can be used to evaluate the quality of the alignment, by evaluating the distance between the matched features after registration. This is illustrated in Figure 1.4.

Again, the selection between area-based and feature-based energy functions is application dependent. Area-based functions, when compared to feature-based, are slow. Also, the registration process using area-based functions need an initial solution close to the optimum. Feature-based functions tend to be fast (because the number of features of an image is much smaller than the number of pixels), and in most cases there is no need of an initial solution during registration. However, feature-based energy functions will not perform well when the images have regions without features, and also if the features come from repetitive patterns (since they will not be unique and the correct match will be difficult). In these scenarios, area-based functions may achieve more precision, specially in the case of non-rigid registration.

As mentioned, non-rigid warping functions are more flexible than rigid warping functions, but they also have disadvantages. Because the number of parameters is much greater than that of a rigid function (for comparison, rigid functions have generally less than 10 parameters. while the non-rigid functions used in this dissertation could reach more than 200), non-rigid registration generally undergo the *curse of dimensionality* Friedman [1997]: the solution space of the registration problem grows exponentially regarding the number of parameters. Due to

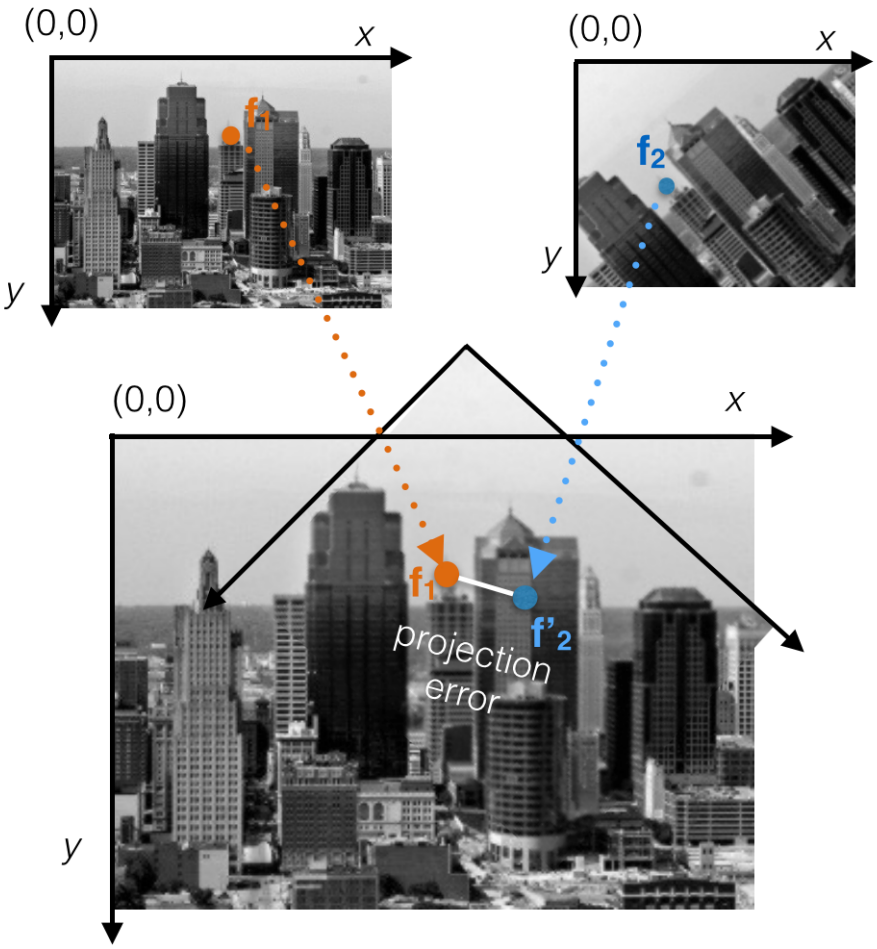


Figure 1.4: Two coordinates, p_1 and p_2, represent the same scene feature (the top of a building). After registration, p_2 is warped into p_2'. The distance between p_2' and p_1 is called *projection error*. The better the registration, the smaller the projection errors are.

this growth, the number of local optima (*i.e.*, points that are optimal in a given neighborhood but are not necessarily the desired global optimum solution) also grows exponentially, rendering the optimization problem harder to solve. Also, due to the sheer size of the problem, solving it may take much longer. Due to these problems, a regularization term must be added to the energy functions when dealing with non-rigid registration. A regularization term is a function used to smooth the search space, removing local optima.

Our objective in this dissertation is trying to develop methods to circumvent the problems with non-rigid registration methods so that their advantages may outweigh their disadvantages. Our attempts will be detailed in the following sections.

1.3 Objectives and contributions

Non-rigid image registration suffers from the problems described in the last section: lack of precision, due to the size of the solution space, and high computational cost, due to the great number of parameters. In this dissertation we will tackle some instances of this problem, to overcome some of these limitations and make non-rigid warping functions more useful.

We will tackle two classes of problems: inter-color alignment of time-sampled images and efficient image mosaicing. In the domain of inter-color alignment, we will touch the subject of multispectral image registration, because each color channel of an *RGB* image has its own spectral band. We will focus in the precision of the method without neglecting efficiency. In the domain of mosaicing, we will tackle the main problem preventing a wider use of non-rigid registration: efficiency. Before our proposed method, to the best of our knowledge, only one method used non-rigid registration for mosaicing. However, it was not designed for real-time processing (Lin et al. [2011]). All the other mosaicing systems used some form of rigid warping function. We believe that non-rigid warping functions, thanks to their flexibility, can help generate better mosaics, and we will endeavor to demonstrate it.

Our contributions in these two classes of problems are detailed below.

1.3.1 Accurate registration for inter-color alignment of time-sampled endoscopic images

We will tackle the problem of inter-color alignment of time-sampled images. As will be explained in Chapter 2, time-sampled images have their color channels misaligned. The misalignment, depends on the local geometry of the scene in a

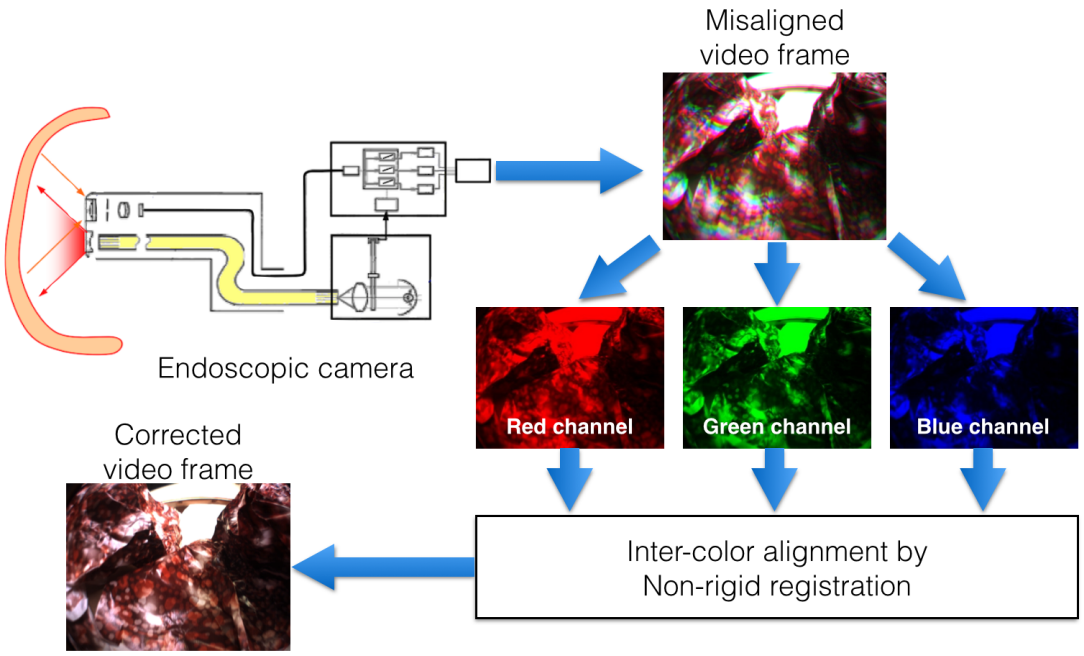


Figure 1.5: Summary of our proposed inter-color alignment method. Images are captured by an endoscopic camera. Due to the sampling process, the color channels present misalignment, resulting in color artifacts. The color channels are aligned back by non-rigid registration.

way that can not be represented by rigid warping functions. Non-rigid registration is applied for this reason. Also, we have to apply correlation based energy functions since each color channel presents a different spectral band. As will be demonstrated in Section 3.2, correlation based energy functions may be hard to optimize because their global optimum is difficult to find.

This being the case, we propose an energy function that combines an efficient computation of entropy (used to measure the correlation of pixel values) with a new energy term that compares the divergence in gradient direction of aligned pixels. The addition of this extra term, named *squared cosine term*, renders the global optimum of the energy function easier to find. Consequently, the registration results become more precise. A brief summary of the problem and the proposed method is given in Figure 1.5.

These contributions will be detailed in Chapter 2.

1.3.2 Efficient non-rigid registration for image mosaicing under flexible photographing

Nowadays, mosaicing has become a popular application of image registration. Even mobile phones are able to create mosaics. For this reason, the mosaicing system must be fast to offer a good user experience. Commonly, since non-rigid warping functions tend to be slow to optimize, rigid warping functions are used for mosaicing. Also, feature-based energy functions are used due to their robustness and efficiency Szeliski [2006]. The standard method of solving the registration with feature-based energy functions is by *RANSAC*.

RANSAC (Fischler and Bolles [1981]) is a very important method that can be used to estimate parameters of mathematical models in the presence of noise (*outliers*). The model parameters are estimated iteratively, and the optimum can be found even if the noise is very strong. However, the number of iterations, which depends on the number of parameters being estimated, grows unacceptably high when the amount of noise is significant and the number of parameters is large (as in our case). For this reason we propose a fast non-rigid image registration method that does not rely on *RANSAC*.

Our proposed method focused on real-time mosaicing of video streams. It will be capable of generating seamless mosaics in real-time using non-rigid registration. The proposed method will create mosaics in real-time by stitching together key-frames previously selected from the input stream. Our video mosaicing method will run in real-time, even though it uses non-rigid registration. To achieve real-time performance, we propose a fast formulation for the non-rigid registration problem, a fast method for selecting the video key-frames, and also an efficient graph cut formulation for stitching the key-frames into the mosaic. This graph cut formulation is tailored for non-rigid registration, being more efficient than classical formulations. Finally we propose a registration method capable of registering a set of images while keeping the alignment globally consistent. This method is intended to be used as post-processing. A brief summary of the problem and the proposed method is given in Figure 1.6.

These contributions will be detailed in Chapter 4.

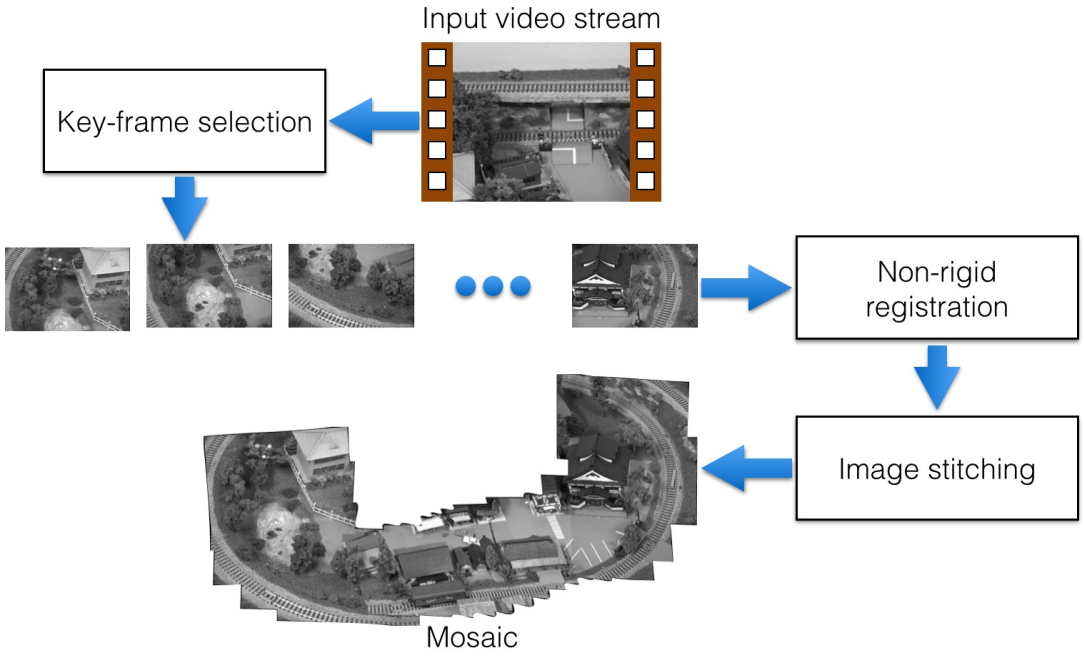


Figure 1.6: Summary of our proposed mosaicing system. Key-frames from the input video stream are selected and aligned by non-rigid registration. The aligned frames are stitched into a mosaic.

1.4 Thesis organization

The remaining of this dissertation is organized as follows.

Chapter 2, *Accurate registration for inter-color alignment of time-sampled endoscopic images*, presents the method for accurately aligning endoscopic images. We explain the details of time-sampled endoscopic imaging, the challenges of performing registration of this type of image, and the energy functions commonly used for this problem. We then explain our proposed energy function that combines joint entropy and color gradient for a more precise registration.

Chapter 3, *Experiments on accurate registration for inter-color alignment*, shows the experimental results on the image registration of time-sampled endoscopic images. We first describe the experimental setup for image acquisition of endoscopic images and the implementation details. Then, we compare our method using the proposed energy function to the baseline method. Our results validate that the inter-color alignment, according to the proposed energy func-

tion, yields robust and precise results, comparable with the best related method analyzed in our work, with significantly reduced computational cost.

Chapter 4 , *Efficient non-rigid registration for image mosaicing under flexible photographing*, presents the proposed non-rigid registration method for creating mosaics from video streams. First we introduce related methods, the scope of our proposed method, and the non-rigid warping function our method uses. Then, we present our proposed non-rigid registration system. Finally, we present our efficient stitching formulation followed by the global registration method for post-processing.

Chapter 5, *Experiments on efficient registration for image mosaicing*, shows the experimental results on the image registration for a set of images (both video key-frames and photos) captured under flexible photographing. We first describe the experimental setup for image/video acquisition and the implementation details. Next, for video stream inputs, we demonstrate that our proposed mosaicing method is capable of running in real-time while performing a more precise registration than related methods with rigid deformation models. In addition, we qualitatively validate that our proposed post-processing methods improves the quality of the final mosaics.

Finally, Chapter 6, *Conclusions and future work*, presents the contributions of this dissertation, together with some possible future venues of research. First, we discuss a possible improvement of our inter-color alignment method. Instead of using a general optimization method, as previously done in the experiments, we could investigate better optimization methods that explore the characteristics of the proposed energy function. Second, we discuss possible extensions to our proposed mosaicing method. Our system only creates planar mosaics. We could extend the mosaic creation to other shapes, such as cylinders and spheres. Also, our research dealt with 2D mosaics. Another possible extension would be creating 3D mosaics.

Chapter 2

Accurate registration for inter-color alignment of time-sampled endoscopic images

2.1 Introduction

An *RGB* image contains three different spectral bands, one for each channel. There are different ways of capturing an *RGB* image, mainly among them: color separation, spatial sampling, and time-sampling.

- Color separation works by splitting the light into 3 different spectral bands by using a prism (Figure 2.1(a)). Each one of the bands is then captured by a different *CCD* sensor. High quality images can be recorded, but the method is expensive because it requires a prism and three *CCD* sensors.

- Spatial color sampling works by detecting different spectral bands at different points on a single sensor (Figure 2.1(b)). The full *RGB* image is reconstructed then by interpolation (*demosaicing* in this context). Since it only uses one sensor, the quality is inferior to that of color separation, but the low cost makes this the most popular sampling method for commercial products.

- Time-sampling works by using a single *CCD* sensor by sampling different spectral bands at different instants. An example of application of time-

sampling can be seen in endoscopic camera (Figure 2.2). The camera illuminates the inside of the body with a different light for each video frame. The color of the light is changed by a rotating color filter. The single color frame is captured by a CCD sensor and sent to the the video processor. There, it is combined with the previous single color frames to create the RGB frames, as shown in Figure 2.3. Time-sampling can generate high resolution images, but if the camera or the scene moves, the color channels become misaligned and color artifacts may appear. Time-sampling is applied when the user wants images with both high-resolution and quality in the color information but does not want the high cost of using color separation.

In this chapter we will explain how to overcome the limitations of time-sampling methods by proposing a new energy function. This function will be used for inter-color alignment to remove the color artifacts. Since the misalignment of the color channels depends on the local geometry of the scene, non-rigid registration must be used. Feature-based energy functions are not suited to register endoscopic images. This happens because the tissues of our digestive system

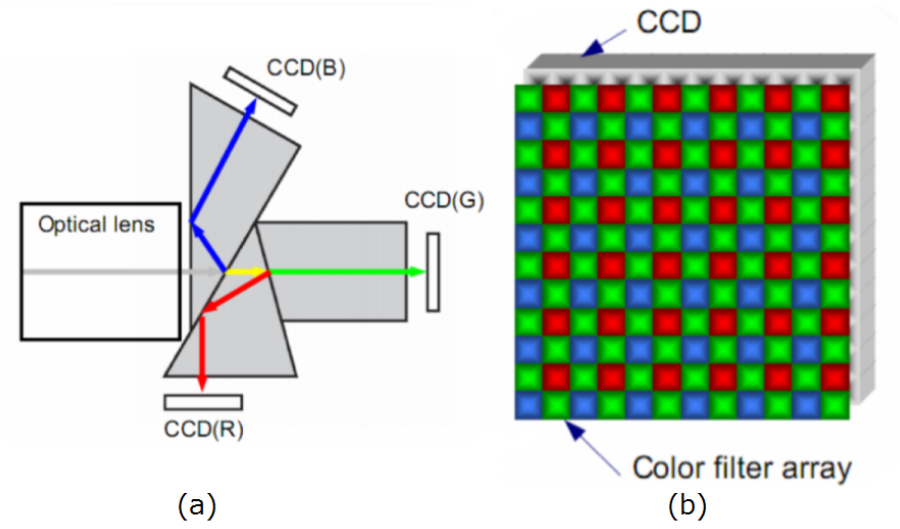


Figure 2.1: Two of the most common methods of RGB sampling: color separation (a) and spatial color sampling (b).

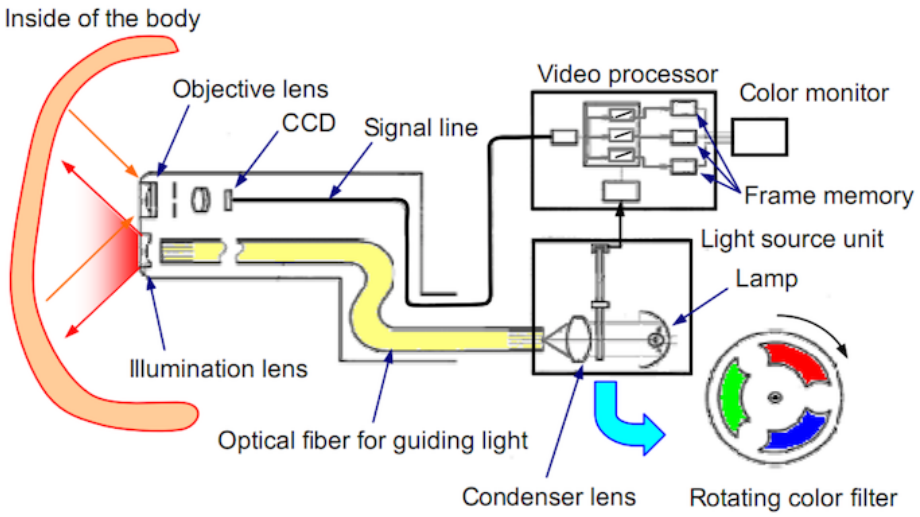


Figure 2.2: Scheme of an endoscopic camera that captures RGB images using time-sampling.

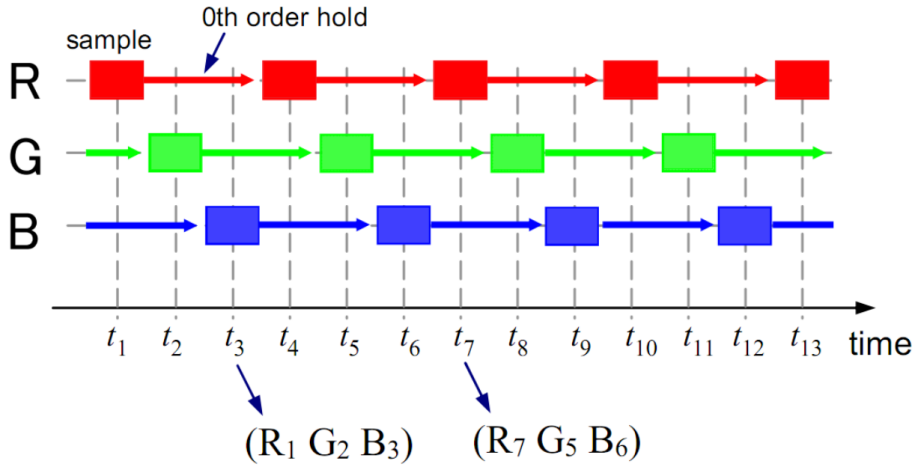


Figure 2.3: How time-sampling works. For each frame, only one channel is sampled. When composing the final RGB image, the current channel is combined with the previous samples.

are mostly smooth and without feature points. For this reason, our proposed method must be an area-based energy function. Also, since the spectral band of each color channel is different, a correlation-based methods must be used. Our proposed energy function will be based on information entropy. However, we

empirically observed that entropy is difficult to optimize, for reasons that will be explained in Chapter 3. From the literature, we understood that the orientation of the color gradient can be used as an energy function for our problem. Thus, in order to make our registration problem easier to solve, our proposed energy function will fuse entropy and gradient orientation. This will yield a more precise registration while still being efficient, as we will demonstrate.

The remaining of this chapter is organized as follows. Section 2.2 presents the related methods. Section 2.3 presents energy functions for multispectral image registration related to our research. Section 2.4 presents the mesh model that implements the warping function used by our proposed method. Finally, Section 2.5 presents our proposed energy function.

2.2 Related methods

Many applications employ the sum of squared differences (*SSD*) minimization to align gray scale images. For example, *SSD* is used for non-rigid image registration Shimizu et al. [2008]. However, *SSD* is not an appropriate measure for channels representing different spectra, because in this case the channel values for the same pixel may have great variations.

It is assumed that there is a constant relationship among pixel values in the color space. Early criteria for minimization in this scenario have been proposed in Woods et al. [1992] and Woods et al. [1993]. Furthermore, evaluation criteria for the relationship between pixel values (joint-distribution) are proposed in Ref. Hill et al. [1993]. The following metrics have been proposed as similarity evaluation criteria: distribution of the third order moment Hill et al. [1994], joint-entropy of the probability density distribution Collignon et al. [1995b]; Studholme et al. [1995], and mutual information showing the dependency of images Collignon et al. [1995a]; Maes et al. [1997].

However, methods using histograms such as mutual information present a considerable risk of the optimization algorithm being trapped in local minima. This problem appears generally due to the interpolation method used to calculate the histogram. To avoid this problem, multi-resolution hierarchical approaches

have been proposed Likar and Pernus [2001]. In the work Pluim et al. [2000], a method of combining Mutual Information and gradient orientation was proposed. As discussed in Section 2.3, this additional information gives more robustness to registration results, but ends up being harder to optimize. Image registration in medical images, a modality which uses entropy based registration massively, is presented in detail in Maintz and Viergever [1998] and Pluim et al. [2003].

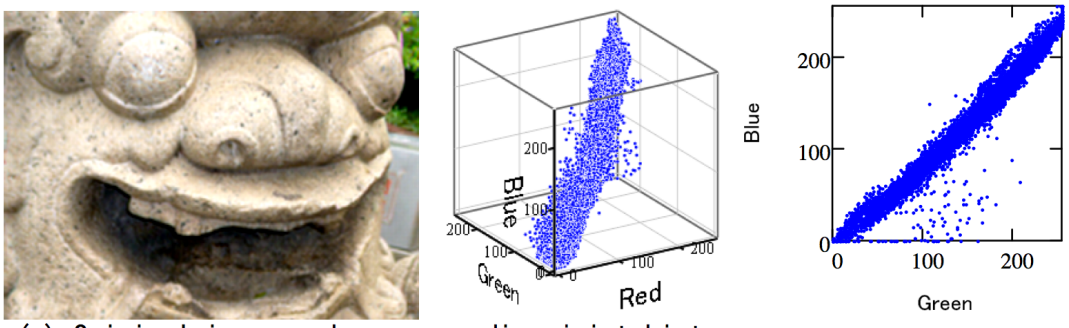
2.3 Energy functions for multispectral image registration

This section presents the joint-histogram, a way of representing the joint distribution of pixel values between images. In addition we present some common methods of measuring correlation using this histogram: joint-entropy and mutual information. A related method (introduced below), proposed an energy function that combines mutual information and gradient orientation to have a more precise registration method. This section will discuss these methods, used as base for comparison for our proposed method.

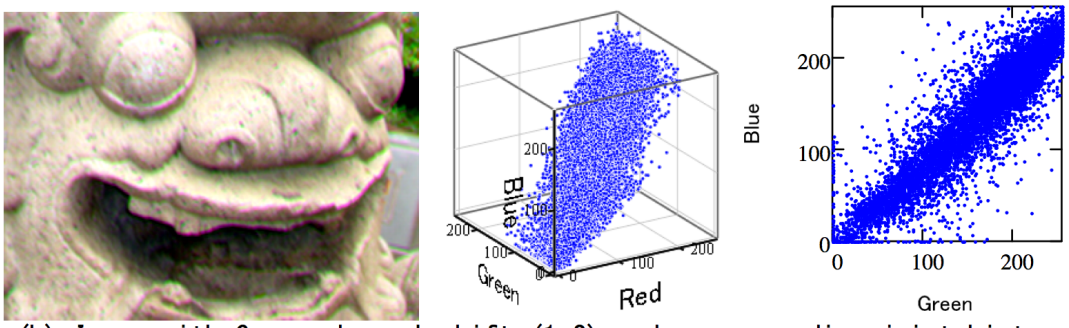
Joint-histogram of multiple images: The joint-histogram represents the joint-distribution of two or more random variables. For the *2D* joint-histogram, the pixel values of two images are used as the coordinate axis. Given two single channel images A and B, and a pixel coordinate x, $p(A(x), B(x))$ represents the probability of a pixel to have the value $A(x)$ on the image A and $B(x)$ on the image B.

The joint-distribution changes according to the transformations applied to the images. Using image registration, we seek the transformation that yields the joint distribution with the smallest entropy.

Figure 2.4 shows an example of a joint-histogram, where Figure 2.4(a) shows the original image without any displacement between the color channels, its joint-histogram in RGB space, and its projection on the plane defined by the green and blue components. Figure 2.4(b) shows the same image with the green channel



(a) Original image and corresponding joint histogram



(b) Image with Green channel shift (1, 2) and corresponding joint histogram

Figure 2.4: Color images and their joint-histograms.

shifted by $(1, 2)$ pixels, the resulting joint-distribution and its projection. In response to a tiny displacement, the joint-histogram changed significantly.

Joint-entropy of two images: We can measure the scattering of a probability density distribution by using entropy. When the distribution is uniformly scattered, the entropy is maximized. On the other hand, when it is concentrated, it reaches its minimal value. Let A and B be two images. When we calculate the entropy of a joint-histogram of A and B, $H(A, B)$, small values indicate that A and B are correlated. On the other hand, $H(A, B)$ increases as A and B are more independent;

$$H(A, B) = -\sum_{a \in A} \sum_{b \in B} p(a, b) \log p(a, b) \tag{2.1}$$

where $p(a, b)$ represents the joint-probability pixel values a and b, $a \in A$ and $b \in B$. We can calculate $p(a, b)$ from the joint-histogram.

Mutual information: As described in Pluim et al. [2003], the joint-entropy is computed from the overlapping portion of the images; it is sensitive to the size and contents of the overlap. The joint entropy can take a low value for a complete misalignment. Mutual information (MI) has been proposed as a stable method that can deal with this problem.

$$\begin{aligned} I(A, B) &= H(A) + H(B) - H(A, B) \\ &= \sum_{a \in A} \sum_{b \in B} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \end{aligned} \tag{2.2}$$

Mutual information is a measure that shows the degree of dependence of two random variables. If two images are correlated, their mutual information will be high. A more stable way of calculating MI has been proposed in Studholme et al. [1999].

Adding color gradient information: Registration is in general an ill-posed problem: The similarity functions present many local optima and it is not trivial to determine which one is the global optimum. This problem brings serious effects in entropy based functions, which present more local optima than the SSD based ones. As we show in Section 3.2, SSD function presents a global optimum with a wider basin, easier to find, but with an ambiguous optimum solution. On the other hand, mutual information presents a global optimum with a more narrow and sharper basin, harder to find but with a very unambiguous optimum solution. The joint histogram entropy also presents a global optimum with basin narrower than MI and SSD similarity functions, making the optimization process harder.

A way of tackling this problem is to add new information into the similarity function, so that its global optimum is easier to find during optimization. The idea behind this method is to combine two or more functions, defined over the

same search space, whose global optimum is the same. This can accentuate the global optimum and eliminate local optima, facilitating the optimization. One way of adding such a type of information is by using gradient information. For the case of RGB inter-color alignment, even when the pixel value changes in different channels, the gradient direction keeps a high correlation. This is demonstrated in Figure 2.5. The method described in Pluim et al. [2000] analyzes the color gradient of the input images. It uses an energy function which compares the similarity in module and orientation of the gradient fields. When these images are aligned, their gradient fields are expected to have maximum similarity and the mutual information is also expected to be maximal.

Therefore, color gradient information can be used to make the similarity functions easier to optimize. However, one problem must be considered when computing the similarity between gradient fields from images with different spectra. Even when the fields are perfectly aligned, the gradient vector for a given coordinate may have different modules, due to spectrum differences. Also, the vectors may present opposite directions. This is illustrated in Figure 2.6. In this example, we can see that the gradient fields are aligned in the red and green channels. However, in the borders between the red and green regions, the gradient presents opposite directions.

Here we give a brief description of the formulation proposed in Pluim et al. [2000].

The angle α between two gradient vectors can be computed using their inner product: $\alpha_{x,x'}(\sigma) = \cos^{-1} \frac{\nabla x(\sigma) \cdot \nabla x'(\sigma)}{|\nabla x(\sigma)||\nabla x'(\sigma)|}$, where A and B are two input images, x represents a pixel in A, x' the warped position of x in B, and σ represents the scale of the filter used to estimate the gradient. The ∇ represents an operator that takes the gradient of the image intensity. The following equation represents the weight of a given angle.

$$w(\alpha) = \frac{\cos(2\alpha) + 1}{2} \tag{2.3}$$

This weight function favors angles near zero or near 180 degrees, as can be seen in Figure 2.7, making this gradient term (GRD) able to account for different spectra. The next equation shows a way to combine this term with the classical

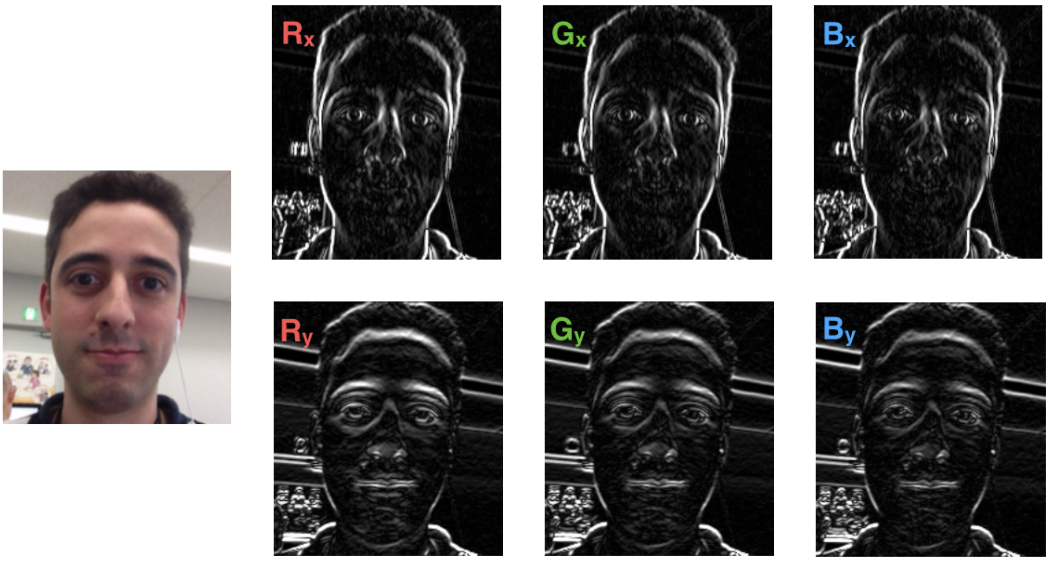


Figure 2.5: Gradient direction of the red, green, and blue channels of a color image. Even though the pixel values of each channel are different, the three gradient fields show a strong correlation.

mutual information metric.

$$I_{new}(A, B) = G(A, B)I(A, B) \tag{2.4}$$

$$G(A, B) = \sum_{(x,x')\in(A\cap B)} w(\alpha_{x,x'}(\sigma))min(|\nabla x(\sigma)|, |\nabla x'(\sigma)|) \tag{2.5}$$

The weight function is multiplied by the minimum gradient module. This enforces that only gradient vectors present in both input images will be used, thus eliminating a possible source of artifacts.

However, this method presents some problems. First, the $\cos^{-1}(\cdot)$ and $\cos(\cdot)$ are computationally expensive functions. Second, it uses the $\min(\cdot)$ function, which is numerically hard to optimize. Finally, it is not trivial how to generalize this term for more than a pair of images while minimizing the growth in computational time. In Section 2.5.2 we propose an alternative method of using the

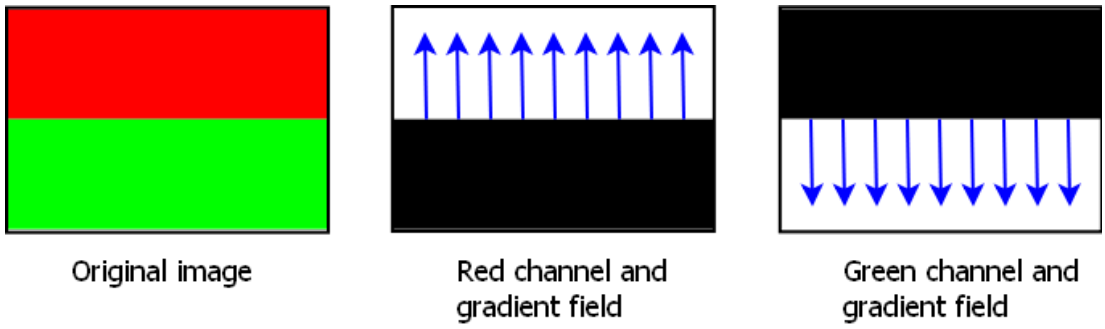


Figure 2.6: Example of aligned gradient fields displaying opposite directions.

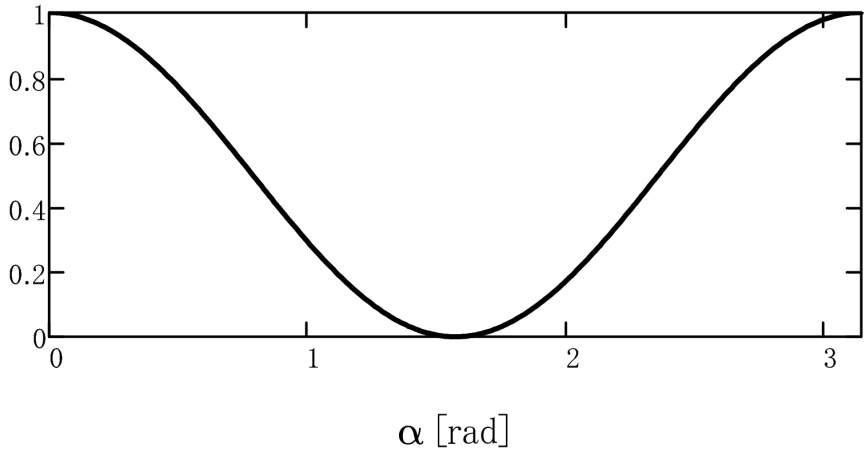


Figure 2.7: Weighting function for gradient angles.

gradient information.

2.4 Mesh model for non-rigid warping

Using B-splines, the displacement at given coordinates is obtainable from the weighted sum of displacements of a set of control points. The weights are computed using *B-spline* base functions. The vector
$\theta = [\Delta\hat{x}_1, \cdots, \Delta\hat{x}_n, \Delta\hat{y}_1, \cdots, \Delta\hat{y}_n]^\top$ is defined as the deformation parameters. The components of the vector are the horizontal and vertical displacement from the initial positions of n control points. The warped coordinates from the original

position $\mathbf{p}$ are representable as follows, given that the control point positions are equally spaced Kybic et al. [2000]:

$$\mathbf{w}(\mathbf{p};\theta) = \mathbf{p} + \mathbf{J}(\mathbf{p})\,\theta \tag{2.6}$$

Where $\mathbf{J}(\mathbf{p})$ denotes the following *B-spline* basis function matrix for the coordinate $\mathbf{p}$, computed as follows.

$$\mathbf{J}(\mathbf{p}) = \begin{bmatrix} c_1 & \cdots & c_n & 0 & \cdots & 0 \\ 0 & \cdots & 0 & c_1 & \cdots & c_n \end{bmatrix} \tag{2.7}$$

The *B-spline* basis c_i are representable using the initial positions of the control points $\hat{\mathbf{p}}_{0i} = [\hat{x}_{0i}, \hat{y}_{0i}]^\top$ and the interval between the control points (h_x, h_y):

$$c_i = \beta\left(\frac{x - \hat{x}_{0i}}{h_x}\right)\beta\left(\frac{y - \hat{y}_{0i}}{h_y}\right) \tag{2.8}$$

The cubic function β is described below.

$$\beta(t) = \begin{cases} 2/3 - (1 - |t|/2)t^2 & \text{, if } 0 \le |t| \le 1 \\ (2 - |t|)^3/6 & \text{, if } 1 < |t| < 2 \\ 0 & \text{, otherwise} \end{cases}$$

2.5 Proposed method

Our proposed method for inter-color alignment of time-sampled images is presented in this section. First, Section 2.5.1 will present an efficient method of computing the entropy of the *RGB* color distribution of 3 color channels. Section 2.5.2 will present a novel method of calculating the alignment of multispectral images using the color gradient information. This method will be combined with our efficient joint entropy to give it more precision. Details are given in the following sections.

2.5.1 Entropy of a color subspace

As shown in Figure 2.4, the 3D distribution of RGB values can be used to evaluate the alignment of the color channels.

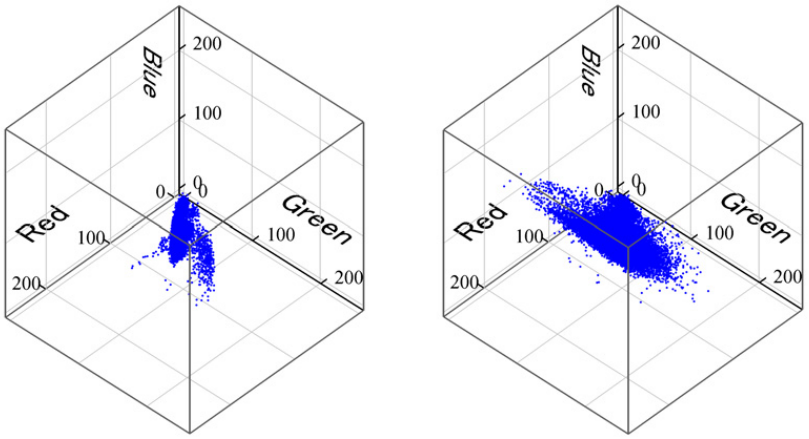
A naive approach would have a *3D* joint histogram to represent the distribution, but this can be improved in the following way. Through the analysis of natural color images, we can observe that, by using *principal component analysis*, the principal component of the joint distribution of the *RGB* color space corresponds to brightness. Comparing normal natural images with natural images whose color channels have been misaligned (as in Figure 2.4), we can observe that the subspace representing brightness has not changed significantly, only the subspace representing the color has changed. Thus, the joint entropy of the three color channels of an image can be computed from the subspace representing the color distributions, ignoring the brightness component. This implies that we do not need to use a *3D* joint-histogram, because a *2D* one is enough. This is explored by our proposed method.

The entropy of a joint-histogram in a *2D* subspace is described as follows.

$$\begin{aligned} E &= -\sum_{\xi \in \Omega} p(\xi) \log p(\xi), \\ \xi &= \mathbf{C}\mathbf{s}, \end{aligned} \tag{2.9}$$

where $\mathbf{C} : R^3 \rightarrow R^2$ is a projection matrix from the *3D* joint-histogram onto one of its subspaces. The vector $\mathbf{s} = (r, g, b)^\top$ represents the RGB values of a pixel on the 3 color channels and Ω represents the region of interest. The matrix $\mathbf{C}$ can be computed from the principal component analysis of the set of vectors $\mathbf{s}$.

Figure 2.8 shows the joint-histogram of Figure 2.4. The histogram is projected on the plane which has the normal vector of the first principal component $(0.62, 0.58, 0.53)$. Figure 2.8(a) shows the projection of Figure 2.4(a), where there is no displacement of color channels. Figure 2.8(b) shows the correspondent projection of Figure 2.4(b), where the green channel is displaced. It can be seen that the distribution is greatly spread over the projected subspace.



(a) Original image (b) Green channel shifted image

Figure 2.8: Projection of the joint-histogram (Figure 2.4) onto a 2D subspace.

The subspace $CbCr$ in the $YCbCr$ color space can be used as a good approximation. The normal vector of the $CbCr$ subspace is $(0.577, 0.577, 0.577)$, which is slightly different from the first principal component. However, the projected joint-histogram is almost the same.

In our experiments, the projected joint-histogram is estimated as the probability density $p(\xi)$ in a discretized subspace. The color value set $\mathbf{s}(\mathbf{x})$ at position $\mathbf{x}$ in the images will be projected onto a position $\xi(\mathbf{x}) = \mathbf{C}\mathbf{s}(\mathbf{x})$. In order to calculate the joint-histogram of this discretized projected subspace, the real valued coordinate $\xi(\mathbf{x})$ must also be discretized into bins. The share of the real coordinates into a set of bins is computed using *B-Spline* base functions described below.

$$D_p(\xi_b(n)) = \sum_{\mathbf{x}} \frac{1}{N} \beta \left(\frac{\xi(\mathbf{x}) - \xi_b(n)}{h_b} \right) \tag{2.10}$$

$$\beta(t) = \begin{cases} 2/3 - (1 - |t|/2)t^2 & \text{, if } 0 \leq |t| \leq 1 \\ (2 - |t|)^3/6 & \text{, if } 1 < |t| < 2 \\ 0 & \text{, otherwise} \end{cases}$$

where, $D_p(\xi_b(n))$ represents a distribution value of the probability density for the n-th discretized position $\xi_b(n)$, h_b is the bin width, and N represents the total number of pixels inside the region of interest.

For registering three color channels, for the case of RGB images, the vector s in Equation 2.9 is a *3D* vector. The projection matrix C is $R^3 \rightarrow R^2$ and the projected subspace is *2D*. The proposed method needs only one computation of entropy for registering the three channels, while the conventional registration requires two independent registrations evaluating two *MIs*.

2.5.2 Adding gradient information using the squared cosine term

This section presents a new approach for combining color gradient information with the joint-entropy for the registration of multiple images. We call it the squared cosine term (*CosSqr*).

Let $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ be a set of vectors and $\hat{\mathbf{v}}$ an estimation of the correct gradient direction (the reference vector). The $CosSqr(\mathbf{V}, \hat{\mathbf{v}})$ is defined as:

$$CosSqr(\mathbf{V}, \hat{\mathbf{v}}) = \sum_{\mathbf{v}_i \in \mathbf{V}} \cos^2(\hat{\mathbf{v}}, \mathbf{v}_i) = \sum_{\mathbf{v}_i \in \mathbf{V}} \frac{(\hat{\mathbf{v}}.\mathbf{v}_i)^2}{\|\hat{\mathbf{v}}\|^2 \|\mathbf{v}_i\|^2} \tag{2.11}$$

Let $\mathbf{p}_a$, $\mathbf{p}_b$, and $\mathbf{p}_c$ be, respectively, $\mathbf{W}(\mathbf{p}; \mathbf{P}_a)$, $\mathbf{W}(\mathbf{p}; \mathbf{P}_b)$, and $\mathbf{W}(\mathbf{p}; \mathbf{P}_c)$ (B-spline deformation model detailed in Section 2.4). Let $V_p = \{\nabla \mathbf{p}_A, \nabla \mathbf{p}_B, \nabla \mathbf{p}_C\}$ be the vector of gradients of the color channels A, B, and C. As reference vector $\hat{\mathbf{v}}_\mathbf{p}$, we used the vector of the color channel that is kept fixed during the registration, $\nabla \mathbf{p}_A$, $\nabla \mathbf{p}_B$, or $\nabla \mathbf{p}_C$, depending on the case. The $CosSqr$ term is combined with the joint-entropy similarity function by the following definition.

$$E_G = W^{-1} E \tag{2.12}$$

$$W = \sum_{\mathbf{p} \in \Omega} cosSqr(\mathbf{V}_\mathbf{p}, \hat{\mathbf{v}}_\mathbf{p}) \tag{2.13}$$

where E denotes the entropy of Equation 2.9.

This new similarity measure combines joint-entropy (Equation 2.9) and the summation of the *CosSqr* of all pixels in the region of interest.

Since the *CosSqr* of vectors pointing to opposite directions is the same, this method is robust to differences in the spectral bands of the images. Also, it is trivial to add new images (or channels) to the registration with linearly growing extra cost. The function only uses simple mathematical operations and its derivatives are not hard to compute.

Chapter 3

Experiments on accurate registration for inter-color alignment

As the experimental validation of our proposed method, we performed three classes of experiments. We compared the registration using: sum of squared differences (*SSD*), mutual information (*MI*), entropy of the projected joint-distribution (*JE*), *MI* combined with the gradient term presented in Pluim et al. [2000] (*MI+GRD*), and *JE* combined with the squared cosine term presented in the previous section (*JE+COS2*).

In the first set of experiments, we evaluated the behavior of these similarity functions using a set of simple global transformations. In the second set of experiments, the same methods were applied to time-sampled image sequences presenting non-rigid deformations. Since we simulated the time-sampling, we had the ground truth needed to quantitatively compare the previous approaches with our proposed method. In the final set of experiments, we calculated the computational cost of our proposed method by comparing it with the other methods we studied.

The remaining of this chapter is organized as follows. Section 3.1 explains implementation details concerning the experimental setup. Section 3.2 presents the experimental results regarding the three classes of experiments listed above.

3.1 Implementation details

When estimating the deformation parameters for three images A, B, and C (the color channels), the equations previously presented remain identical, except for the parameter vector $\mathbf{P}$, rewritten as $\mathbf{P} = (\mathbf{p_a}^\top|\mathbf{p_b}^\top|\mathbf{p_c}^\top)^\top$. One of the three images is used as reference and its parameters are set to 0. The other calculations are identical to the ones previously introduced.

The optimization method used in this project is the coordinate descent Abatzoglou and O'Donnell [1982], applied for all evaluation functions being compared. We used Brent's method Brent [1973] to the *1D* optimizations required by the coordinate descent algorithm. In our implementation, for each iteration, Brent's method is used to optimize all elements of $\mathbf{P}$, *i.e.*, each control point displacement, while the other deformation parameters are kept fix. Since the displacement of a single control point only affects the solution locally thanks to the spline deformation model, the similarity metric value can be recomputed efficiently from the previous calculations. Another reason for using the coordinate descent method is its robustness. In our experiments, it always converged to a good solution (except for *SSD*). Also, good solutions could be achieved even when the displacement among color channels were big (*i.e.*, 10 or more pixels).

In more details, our implementation used the following settings. The convergence criteria was that the maximum displacement from the previous estimation. When the displacement reaches a value smaller than 0.001 pixel, the optimization was stopped. The parameters are optimized in raster order, related to the control points coordinates. For every control point, first the row coordinates of image A is optimized in *1D*, then the row coordinates of image C. After, the column coordinates are optimized in the same order. The entire project was programed in C language using *OpenCV*. Some of the numerical methods were implemented using the *GNU Scientific Library*. It was programmed to execute within a single thread. The computer we used for the experiments had a *Core 2 Quad 3GHz* processor and 3 Gbyte of *RAM* memory.

3.2 Results

This section will present the experiments used to evaluate our proposed method. We started by comparing the behavior of energy functions commonly used for multispectral registration, including our proposed method. The next set of experiments were conducted to quantitatively evaluate the performance of our proposed method regarding alignment error. Finally, the last set of experiments were conducted to evaluate the computational complexity of our proposed method compared to other related methods. All three sets of experiments are described below.

3.2.1 Behavior of the energy functions

In the first set of experiments, we applied a set of rigid transformations into an image and plotted the resulting energy functions. Figure 3.1 shows the image used for this purpose. We applied rotation and translation.The rotation range was $[-15, 15]$ degrees. The translation range was $[-15, 15]$ pixels. For *SSD*, *MI*, and *MI+GRD*, we compared the red and blue channels of the test image. For our proposed methods (*JE* and *JE+COS2*), we compared all three channels. The image size was 406×304.

The results of our comparison are shown in Figure 3.2. As we can see, *SSD* presents a wide basin around the global optimum, making it easy to converge. However, its optimum position is ambiguous. *MI* has a narrower basin around the optimum, rendering it harder to optimize. However, the global optimum position is very unambiguous because the optimum basin is much sharper. Observing the results of *JE*, it can be seen that the optimum basin is much narrower than the others, making this similarity function very hard to optimize (unless the displacement is very small). *MI+GRD* has a wider optimum basin. Although it is ambiguous for the rotating transformation, this similarity function results easier to optimize than *MI* and *JE*. Finally, the proposed similarity function (*JE+COS2*) shows a behavior similar to that of *JE*. However, it presents a wider optimum basin, making it easier to optimize.

By these simple experiments, we concluded that the similarity functions, when

Figure 3.1: Image used in the first set of experiments.

combined with color gradient, are likely to perform better. We will demonstrate this fact by the following set of experiments.

3.2.2 Registration error

To validate the accuracy of our proposed method, we experimented with 5 sets of images presenting non-rigid transformations. The original images were used to generate time-sampled sequences. To simulate time-sampling, for each frame of the video, one of its channels was selected and the other two ignored. The selection followed the usual channel order: R, G, and then B, repeated for all images. Table 3.1 shows more details of the sequences: the number of frames used, the image size, and the number of control points used in the registration. We compared the same five methods we used for the last experiment.

Table 3.2 shows the experimental results. For each test sequence, we calculated the average RMS error over all images. The error is calculated from the

Test Sequence	# of frames	size	control points
1	10	448×336	11×8
2	6	360×283	9×7
3	5	430×284	11×7
4	5	391×412	10×10
5	5	460×333	12×9

Table 3.1: Details about the image sequences.

Test Sequence	SSD	MI	JE	MI+GRD	JE+COS2
1	9.640	4.336	6.420	2.949	3.432
2	5.346	2.640	3.029	2.630	2.716
3	6.173	3.888	4.871	3.462	3.430
4	7.700	3.686	3.440	2.748	2.684
5	5.563	4.098	3.439	2.378	2.302

Table 3.2: Results of the average RMSE over the pixel values.

difference between the *RGB* values given by the ground truth images and the estimated results. Figure 3.3 shows samples of each sequence and the average RMSE (Root Mean Square Error). The error of each image in a sequence is broken down in Figure 3.4. As we could predict, *SSD* had the worst results because the color channels present different spectra. Since *JE* has an optimum basin narrower than *MI*, the performance of the former was generally worse than that of the latter. Figures 3.5 and 3.6 show some examples of the results obtained by our proposed method. The fist column shows the images before the color channel registration; the second column shows the registered images; the last column shows the ground truth. The comparison between the methods that use color gradient information and the methods that do not indicate that gradient information has a positive effect in the similarity function performance. As can be seen in Figure 3.2, the result of using color gradient is the extension of the global optimum basin, making the optimization process easier, and the average error smaller. Both methods, *MI+GRD* and *JE+C0S2*, have statistically similar average error. Figure 3.7 shows an example comparing the results of all energy functions.

3.2.3 Computational complexity

Finally, this set of experiments is used to evaluate computational complexity.

First, the similarity function over images with different sizes is computed. For each image and each similarity function, 1000 computations were made. After that, the average execution time was calculated. We took the time to calculate the similarity between the red and blue channels comparing to green one.

Figure 3.8 shows the results for each method. As can be seen, the fastest similarity function is *SSD*. The proposed similarity function (*JE*) performed faster than *MI*. This happened because this function only takes information of a smaller subspace of the joint distribution, ignoring useless information. Among the methods using color gradient, the proposed method (*JE+COS2*) proved to be much faster than *MI+GRD*, as predicted.

Second, we computed the average computational time and number of iterations for the entire optimization process, using the studied similarity functions. These experiments were conducted because even if one similarity function is faster to be evaluated than another, it does not imply that the optimization process using the former metric will be faster than using the latter. This may happen because, generally, the metric computation is a small proportion of all the optimization computational cost.

The results for each similarity metric and each image sequence is shown in Tables 3.3 and 3.4.

We can see that the proposed method (*JE+COS2*) performs always quicker and also needs a smaller number of iterations to converge than *MI+GRD*.

In conclusion, these experiments show that the methods which use color gradient information are slower than methods that do not, but they are more precise. Also, the proposed method showed results that, comparing by precision, are similar to the results of the related method presented in Pluim et al. [2000]. Finally, it was verified that the proposed method needs a smaller number of iterations to converge. Also, the total computation time is smaller than the related method.

Test Sequence	SSD	MI	JE	MI+GRD	JE+COS2
1	21	11	11	19	11
2	20	9	10	11	7
3	22	17	9	23	16
4	56	22	12	11	9
5	21	65	21	34	19

Table 3.3: Average number of iterations of the optimization process.

Test Sequence	SSD	MI	JE	MI+GRD	JE+COS2
1	4255	2894	5240	10289	6536
2	2819	1635	3163	3546	2532
3	2050	3323	4452	8684	7024
4	7354	6334	6845	6592	6394
5	1620	13638	10144	14356	10523

Table 3.4: Average execution time of all the optimization process, in seconds.

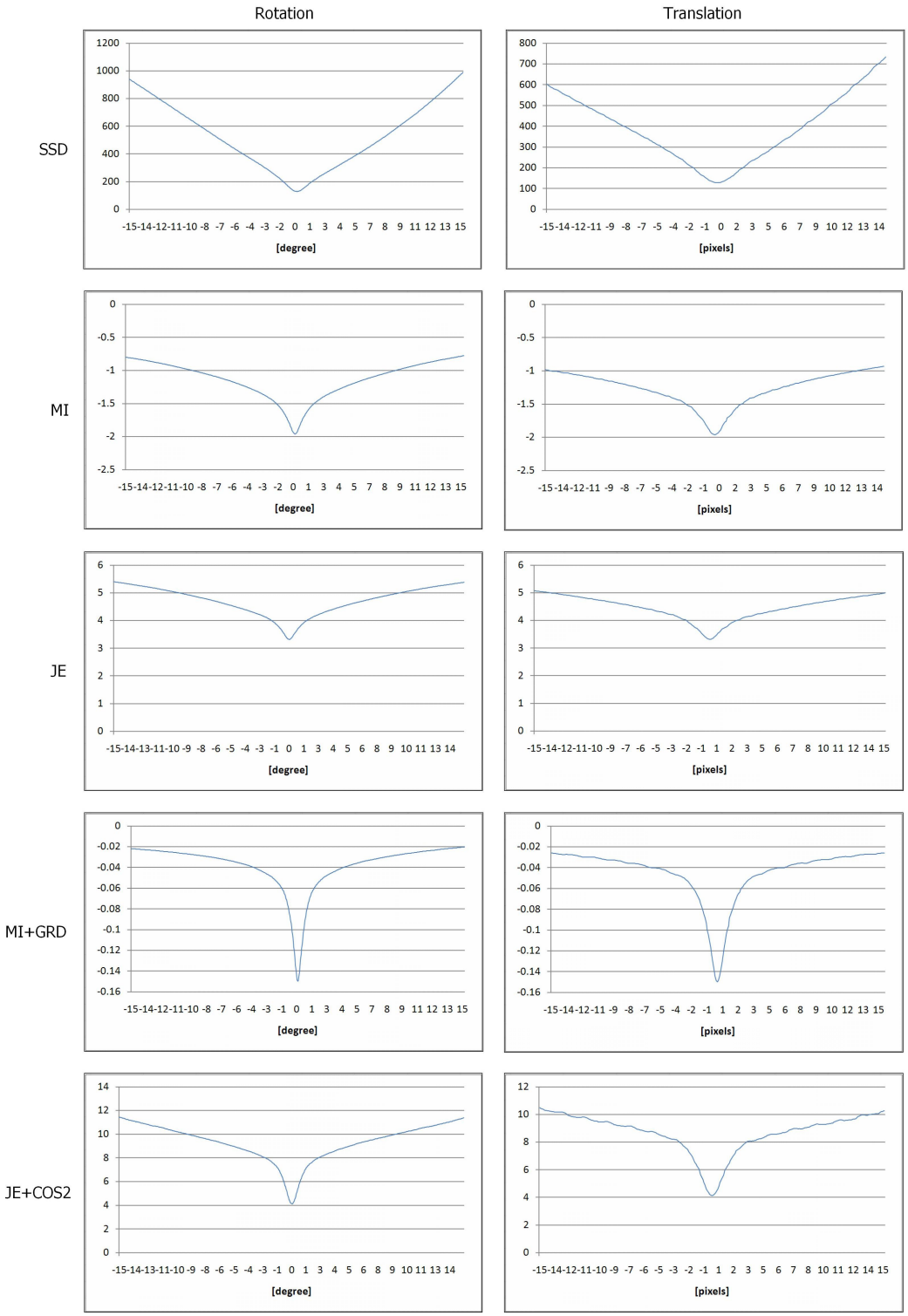


Figure 3.2: Behavior of the studied similarity functions under *1D* transformations.

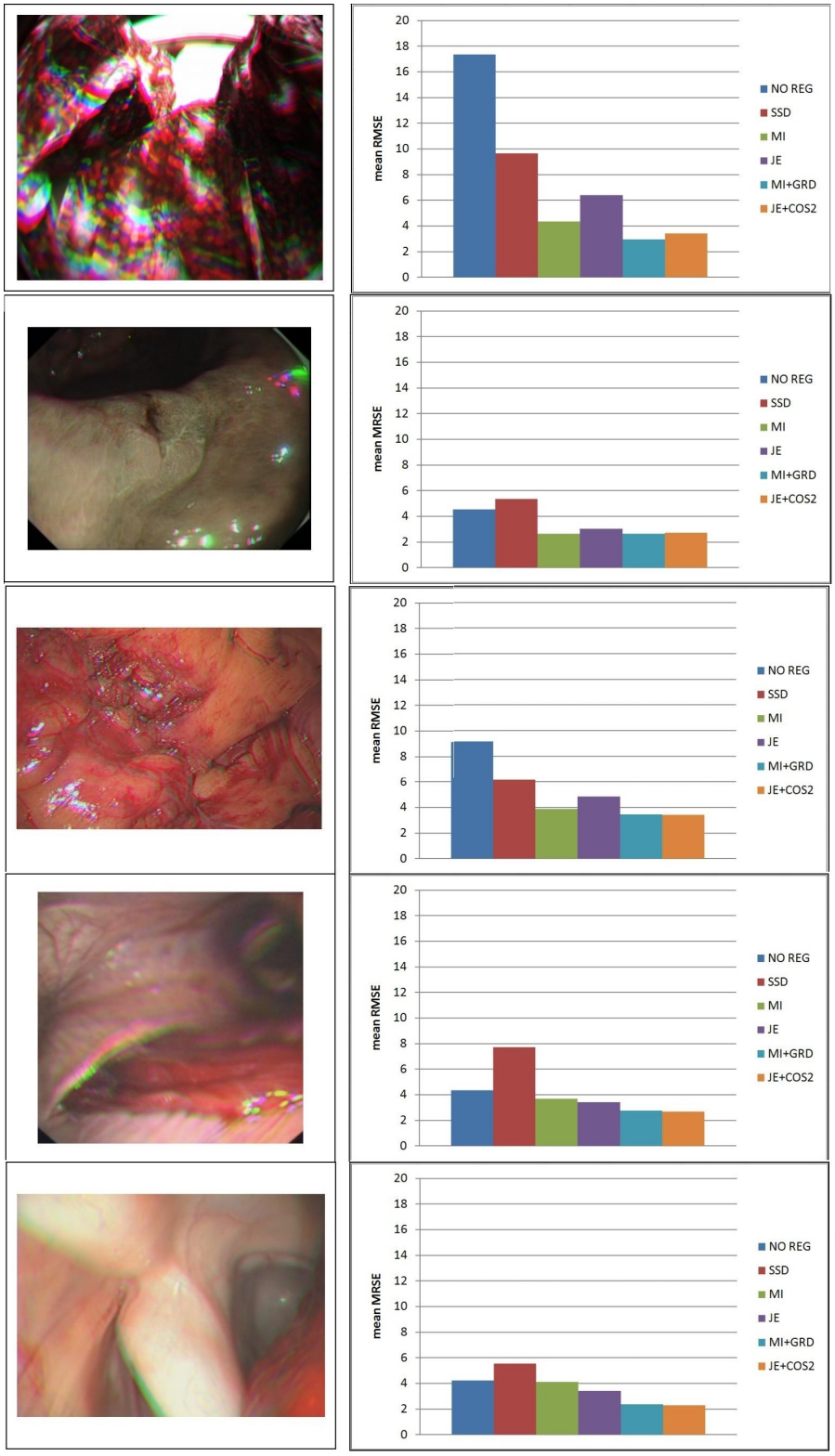


Figure 3.3: Sample of each video sequence and the comparison of performance for all studied methods.

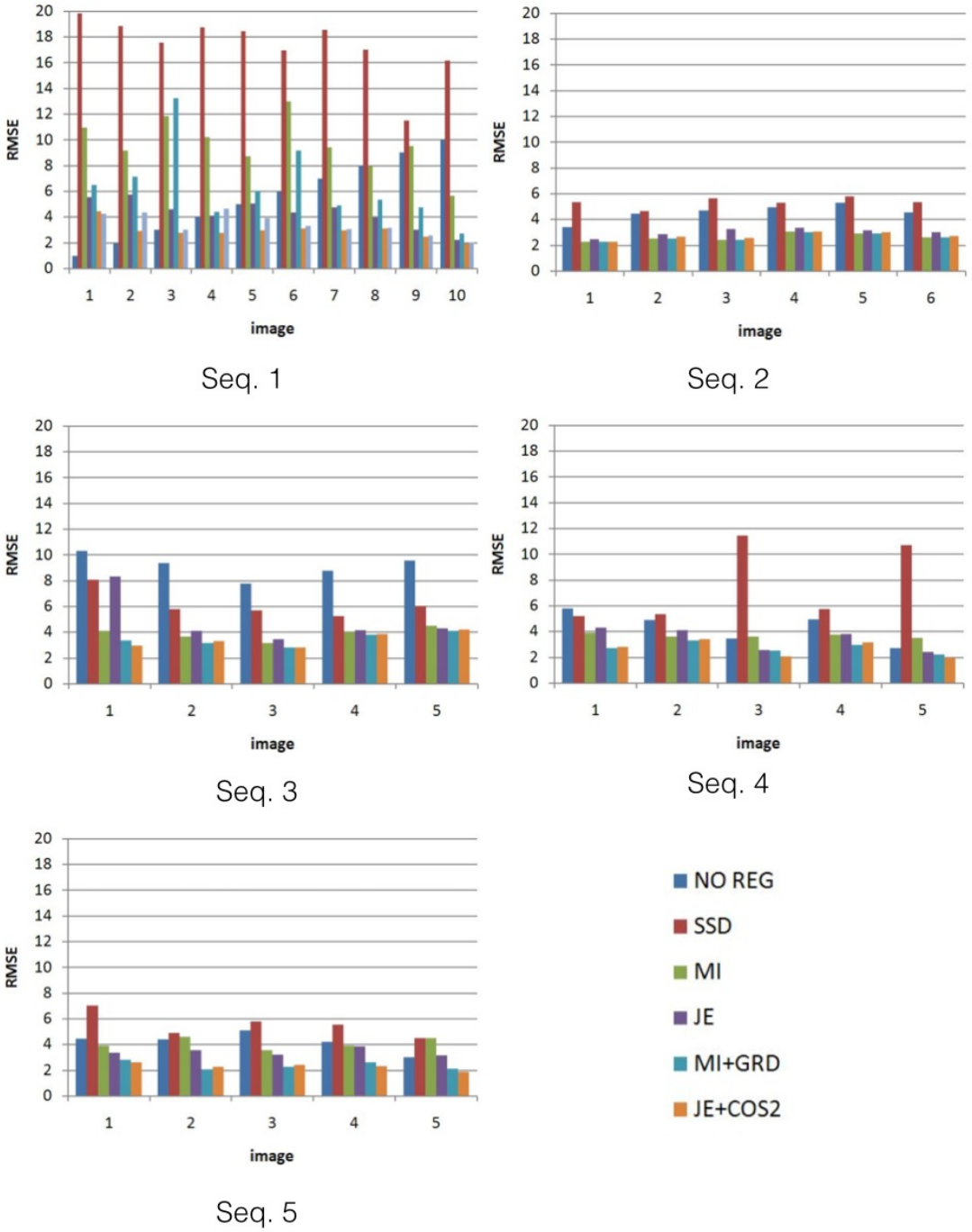


Figure 3.4: Registration error broken down by image sample.

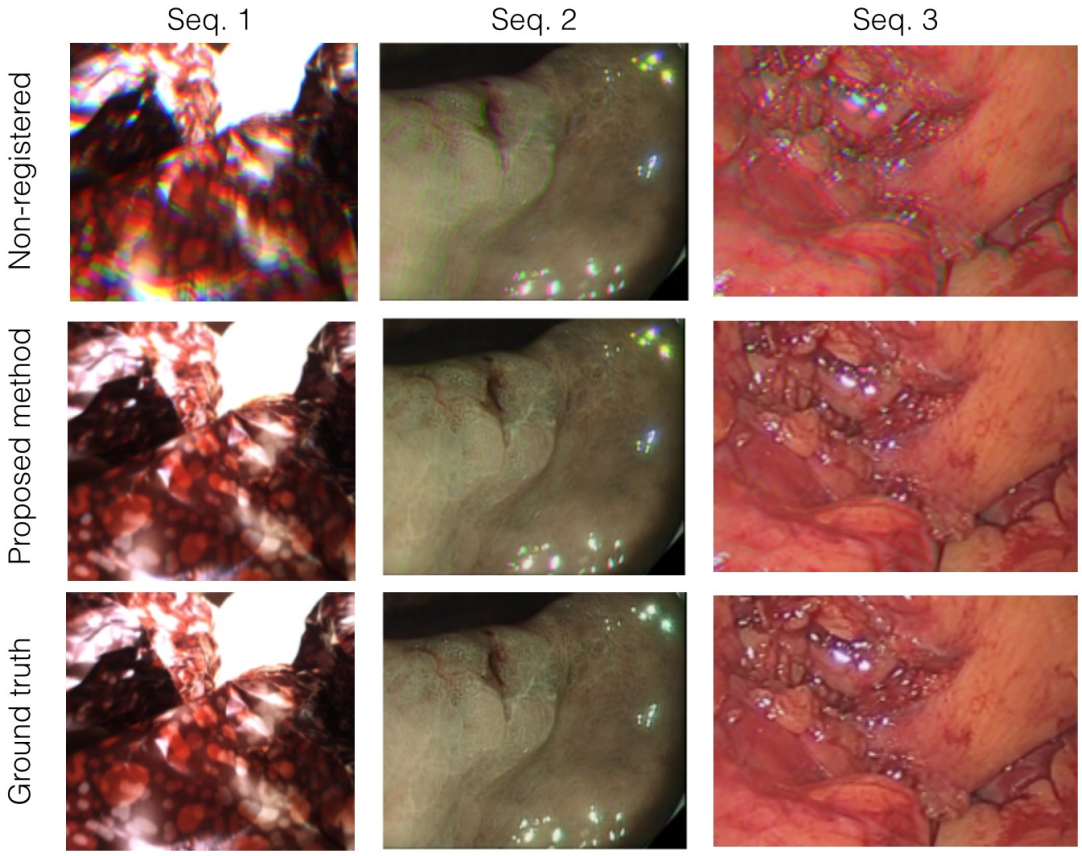


Figure 3.5: Examples of results comparing the time-sampled images, the results of our proposed method, and the ground truth (sequences 1, 2, and 3).

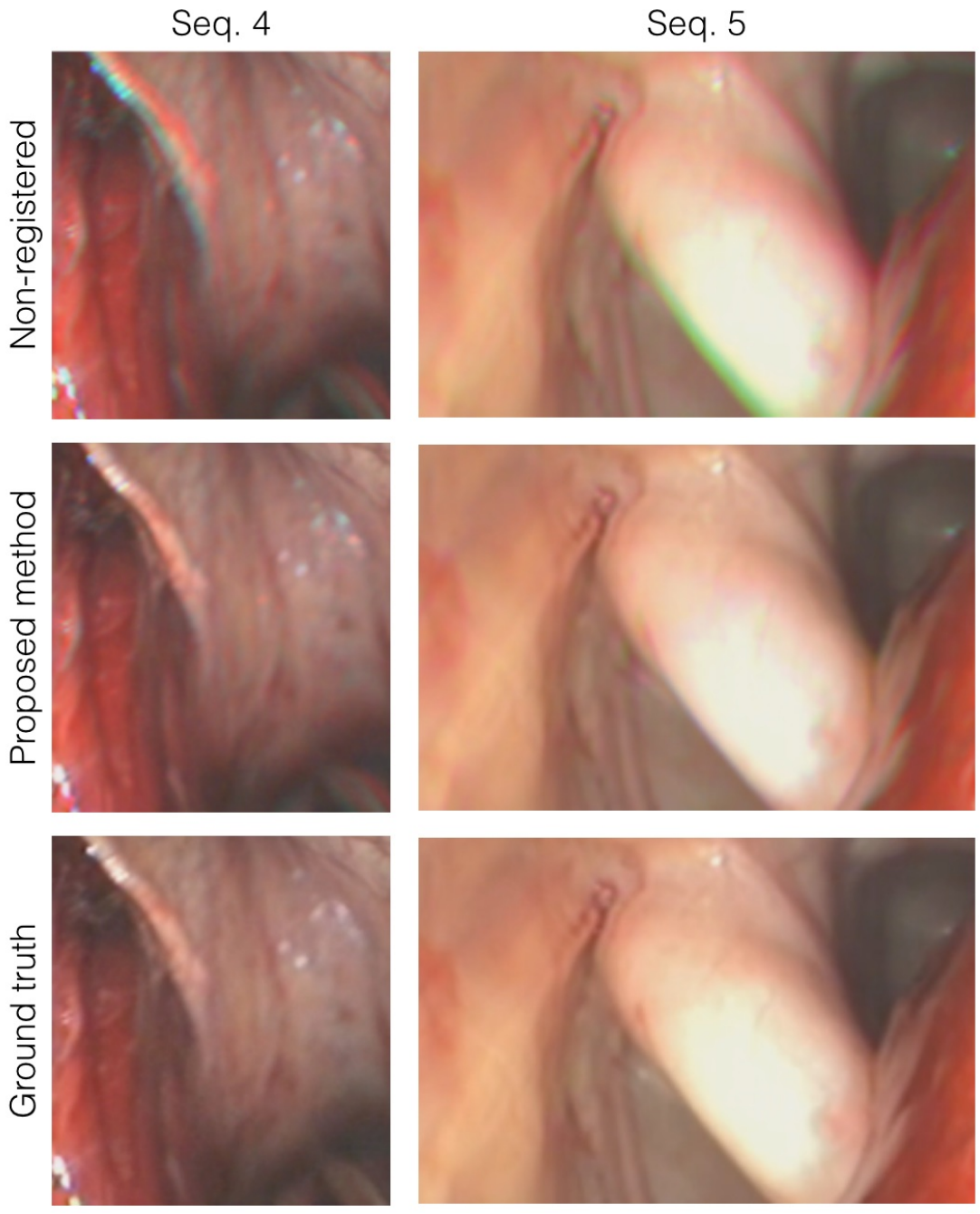


Figure 3.6: Examples of results comparing the time-sampled images, the results of our proposed method, and the ground truth (sequences 4 and 5).

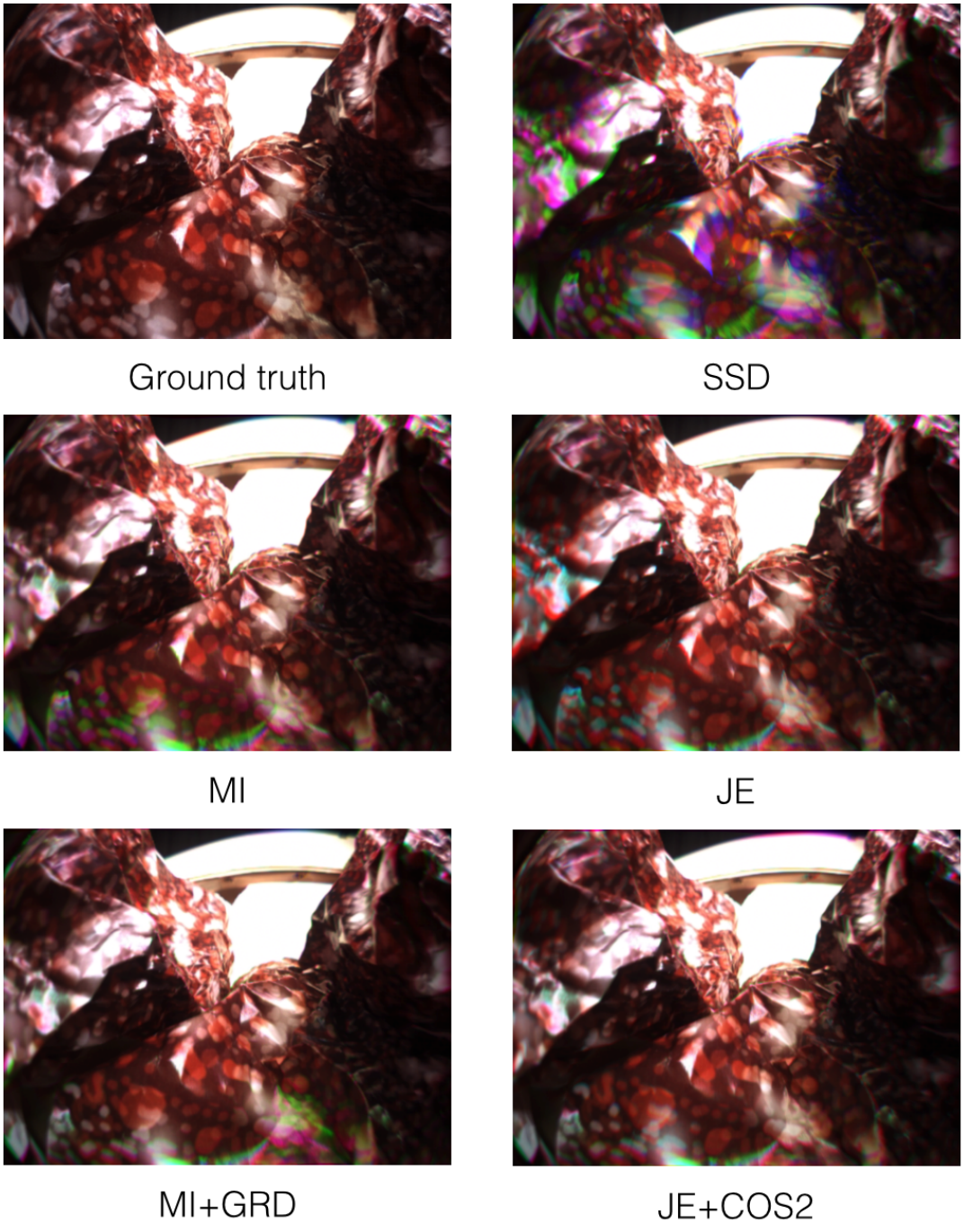


Figure 3.7: Example of inter-color alignment results. The top-left position shows the ground truth. We can see that *SSD* has the worst results. The proposed *JE* shows results inferior to *MI*, while *MI+GRD* and the proposed *JE+COS2* have similar results.

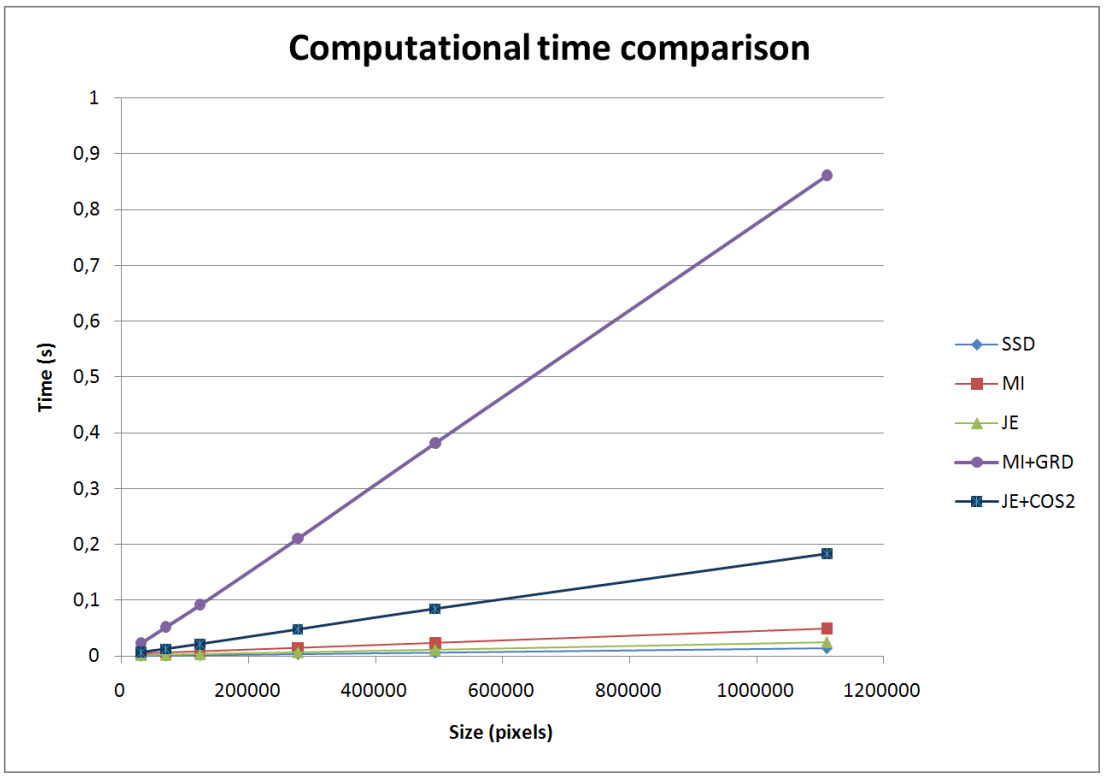


Figure 3.8: Computation times for all similarity functions studied.

Chapter 4

Efficient non-rigid registration for image mosaicing under flexible photographing

4.1 Introduction

This chapter presents the proposed non-rigid registration method for creating mosaics from images captured without restrictions in camera motion or in the setup of the scenes.

First, we propose a real-time mosaicing method that uses video streams as input. The method is online, *i.e.*, the mosaic is instantly updated when new images are input into the system. This method has a very efficient and flexible registration model based on a feature-based energy function and a non-rigid warping function. In addition, for accelerating the entire process, we propose a method for selecting informative key-frames by using the statistics of feature matches. A crucial point in mosaic creation is to determine how the images will be combined into the final mosaic. This process is called stitching. Stitching algorithms may be very slow, specially with high resolution images. We propose a graph cut formulation that explores the characteristics of the non-rigid warping function used by both proposed methods in this chapter to stitch the mosaic very efficiently.

Finally we propose a post-processing method capable of performing registration while keeping the alignment globally consistent. This method is intended

to be used to improve the mosaics generated by other methods (*e.g.* our video mosaicing method). This registration model is an extension of our proposed video mosaicing method that allows globally consistent alignment of the input images while still being computationally efficient.

These contributions will be presented in the following sections.

The remaining of this chapter is organized as follows. Section 4.2 presents the related methods. Section 4.3 presents the scope of our proposed methods. Section 4.4 presents the triangular mesh model used to implement the warping function used by our proposed methods. Finally, Section 4.5 presents our proposed methods.

4.2 Related methods

Since mosaicing is a well studied area of computer vision, there are many approaches to 2D mosaicing. These works can be grouped into three classes: (1) offline methods that use homography or lower degree functions, (2) offline methods that use higher degree transformations, and (3) online methods. The group (1) includes the works, Brown and Lowe [2007]; Deng and Zhang [2003]; Hsu et al. [2002]; Sawhney et al. [1998]; Vercauteren et al. [2005], which are based on rigid warping functions such as homography. The group (2) includes the works Can et al. [2000]; Chaiyasarn et al. [2009], which model the deformation as quadratic functions. The work in Lin et al. [2011] handle non-planar scenes by using non-rigid registration of image pairs. This method, however, was not designed to handle either real-time or global registration.

The group (3), includes the works of Crispell et al. [2008]; Peleg et al. [2000]. The work in Crispell et al. [2008] uses 3D information for registering aerial images using a non real-time algorithm. The method in Peleg et al. [2000] is online and avoids the problem of over deformation by using fixed camera movements (*e.g.* translation, forward motion).

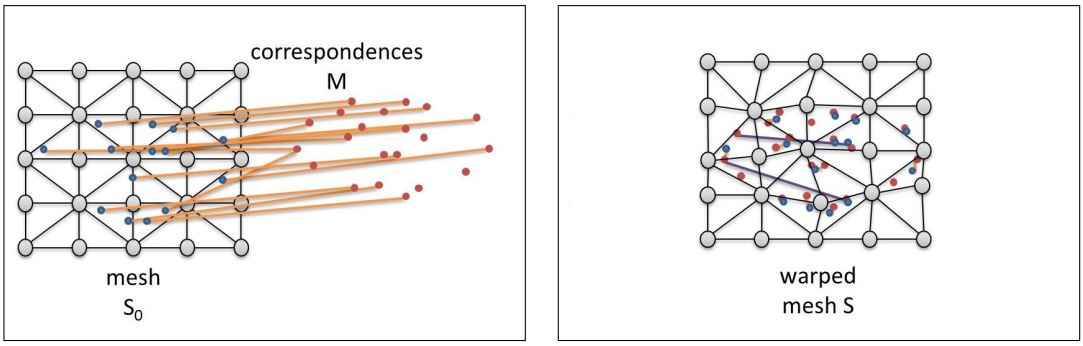
Although most of the presented works dealing with mosaicing make use of rigid transformations such as homography or affine, there are more general registration methods that use non-rigid deformation. Some of them are feature based, *e.g.*

Chui and Rangarajan [2003]; Pilet et al. [2005]; Zhu et al. [2009]. Feature based methods are generally more computationally efficient than area based methods Szeliski [2006], specially in the case of non-rigid image registration. The method in Zhu et al. [2009] can register pairs of images correctly, even in the presence of large amounts of outliers, in real-time. However, this method is designed for pairs of images only. In this dissertation, this method was extended and used for mosaicing.

4.3 Scope and limitations of the proposed methods

As showed in the previous section, most mosaicing systems use some sort of rigid warping function, the most common being homography. However, homography can only align images that were taken from a camera rotating around its optical center or whether the scene being registered is a plane. If there is any deviation from these constraints, the alignment will not be precise. Our proposed methods go beyond these limitations. We aim at creating mosaics without restrictions in camera motion or in the setup of the scenes.

However, our proposed methods have some limitations. In the case of the real-time video mosaicing system, it will only be able to register pairs of images, unlike our global registration method for post-processing. For this reason, loops (the camera records a location, moves somewhere else and then returns to the previous location again) will not be properly registered. Also, due to the nature of the energy function, the non-rigid warping functions cannot have discontinuities. This means, for example, that occluded objects will not register well. Also, scenes presenting strong parallax will also fail to yield a precise registration result. A final limitation comes from the characteristics feature-based energy functions. The scene must not have a big number of repeating patterns, since the feature points in these regions will be very similar and hard to match correctly.



(a) (b)

Figure 4.1: Deformation using a mesh model. (a) Identity mesh S_0. (b) Mesh S warped to reduce the projection error of the matched features.

4.4 Mesh model for non-rigid warping

A $2D$ triangular mesh with M control points is used to express the non-rigid transformations of our feature-based registration methods. Each control point v has its coordinates (x, y). The control point coordinates for the mesh i are written as $\theta_i = (X_i, Y_i)$, where X_i is a vector containing the x coordinates of the control points and Y_i the vector containing the y coordinates (in the case of pairwise registration as presented in Section 4.5.1, there is only one mesh being used at a given time). The warp of any point p inside a mesh triangle defined by the control points v_a, v_b, and v_c can be calculated using the barycentric coordinates of p: $w(p, \theta_i) = \sum_{k \in \{a,b,c\}} B(p, v_l) [X_{i,k}, Y_{i,k}]^{\mathrm{T}}$, where $B(p, v_k)$ is the barycentric coordinate of p in relation to $v_l \in \{v_a, v_b, v_c\}$ (computed in relation to the identity warping). This warping function can be written as:

$$w(p, \theta_i) = t_P^{\mathrm{T}} \theta_i, \tag{4.1}$$

where t_P^{T} is a vector with M dimensions, set to zero except in the positions a, b, and c, where it is set to $B(p, v_a)$, $B(p, v_b)$, and $B(p, v_c)$, respectively. Figure 4.1 illustrates the basic principle of this kind of transformation.

4.5 Proposed methods

We propose three methods that tackle the main problem faced when performing non-rigid registration for mosaicing: efficiency.

First, we present in Section 4.5.1 a real-time video mosaicing method, a method that can create mosaics in real time using video streams as input. We propose in Section 4.5.2 an efficient graph cut formulation for pixel selection, specially tailored for non-rigid registration. Finally, we present in Section 4.5.3 a post-processing registration method which performs global alignment of the input images. All three contributions are listed in the following sections.

4.5.1 Flexible real-time non-rigid registration for video mosaicing

The first method discussed in this chapter is a real-time mosaicing system. This class of system must be able to increase the mosaic iteratively, avoid error accumulation during the registration, and the algorithm must be efficient, despite the complexity of the non-rigid model. Our proposed method has these characteristics.

The mosaicing system implementing the proposed method consists of 4 modules, detailed in Figure 4.2: frame selection, feature matching, registration, and mosaic rendering. The frame selection module reads the input video stream and selects which key-frames will be used to create the mosaic. It also manages the detection of the feature points to be used during registration. The feature matching module matches the feature points in the newly selected frame to the features in the previously selected frame. The non-rigid registration module receives the set of matched features and registers the newly selected frame into the previously selected one. The registered frame is then sent to the mosaic rendering module, where it is stitched to the mosaic and displayed. The procedure is repeated again, until the end of the video. The modules are explained in more details in the following sections.

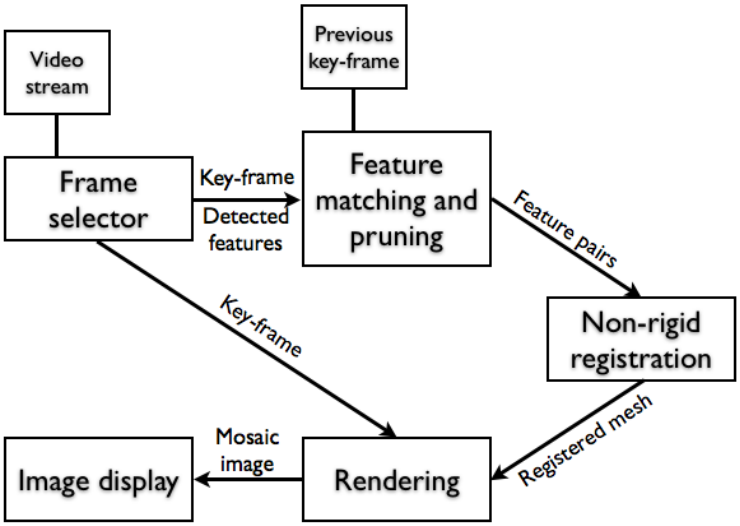


Figure 4.2: Real-time image registration system diagram.

Frame Selection: In order to create mosaics efficiently, only a small subset of the video frames must be selected. This key-frame set must be as sparse as possible, to reduce the number of registrations performed. At the same time, it must contain overlapping key-frames so that a mosaic can be composed out of them. To fulfill these requirements, it is necessary to estimate the overlap of pairs of frames. This is achieved by the following proposed algorithm: (1) the features in both frames being compared are detected using *SURF* descriptors Bay et al. [2006]; (2) the nearest-neighbor matching of the features is computed; (3) the distribution of the distance between the matched descriptors is approximated by a histogram; (4) the overlap measure (OM) is computed. The OM was defined as follows:

$$OM(H) = \sum_{j=1}^{n_{Bin}} G\left((j-0.5)h_{size}, \varsigma\right) H_j \ , \tag{4.2}$$

where n_{Bin} is the number of bins in the histogram, h_{size} is the size of each bin, $(j-0.5)\,h_{size}$ is the average range of the bin j, G is a Gaussian weighting function with 0 mean and standard deviation ς. This weighting function assigns larger

weights to values near zero, and the weight decays quickly, so that the bins which probably contain correct matches receive a larger weight than the bins with wrong matches. Thus, using the *OM*, the key-frames are selected by the following algorithm: (1) the first video frame is selected and used as reference; (2) the next frame whose *OM* (comparing with the reference frame) is smaller than a given threshold is selected and becomes the new reference. Step (2) is repeated until the end of the video.

It was experimentally observed that the probability distribution of the descriptor distances changes according to the intersection size between the image pair. Figure 4.3(a) shows two frames with a small overlap. The descriptor distance has a bell-shape like distribution (Figure 4.3(c), red). Figure 4.3(b) shows two frames with a larger overlap. The distribution becomes bimodal (Figure 4.3(c) blue). The smallest peak represents the inliers among the matched features. Figure 4.3 shows the variation of *OM* over time, in a video recorded by a translating camera. The value of *OM* decreases as the intersection becomes smaller and rises again when a new frame is selected.

Registration Formulation: Now we explain the registration formulation used in the proposed method. Two features are desirable: First, the mosaic must be as seamless as possible; second, over-deformation must be avoided, as in Figure 4.4. For doing so, the proposed method applies a non-rigid deformation model that uses triangle meshes (Section 4.4). It also performs feature-based registration by using the feature points detected during the frame selection procedure and pruned by *RANSAC* Fischler and Bolles [1981].

The initial model of pairwise non-rigid image registration was drawn from Zhu et al. [2009], which was based on Pilet et al. [2005]. It is summarized by the equation below:

$$E\left(S\right) = E_{C}\left(S\right) + \lambda E_{\mathrm{Sm}}\left(S\right) \quad , \tag{4.3}$$

where E_{C} is the correspondence energy function and E_{Sm} is the smoothness energy. The constant λ balances the trade-off between precision and mesh smoothness. The registration is solved by finding the mesh S which minimizes $E\left(S\right)$. The correspondence energy is proportional to the projection error of the warped

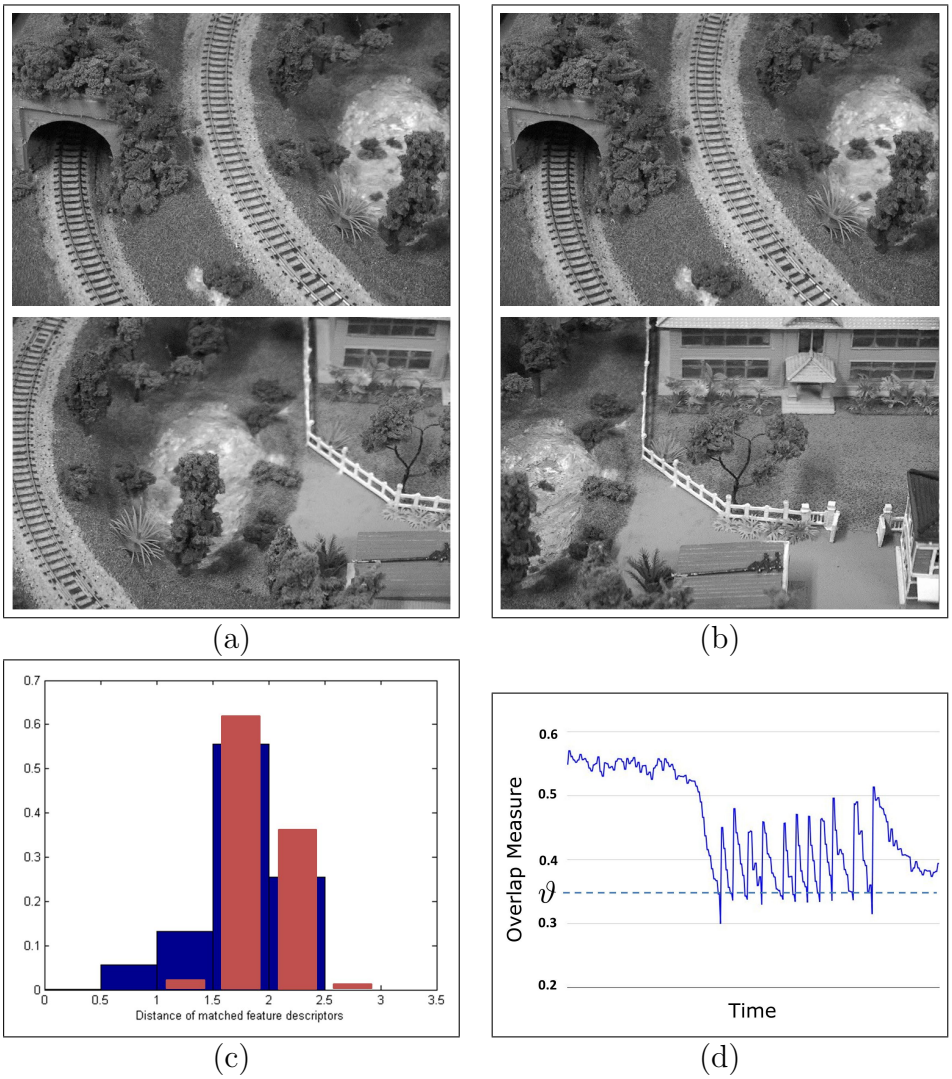
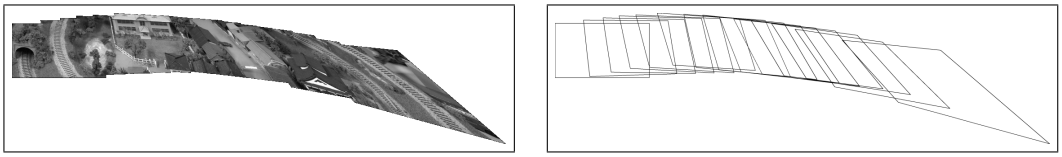


Figure 4.3: Frame selection. (a) Pair of frames with a large overlap. (b) Pair of frames with a small overlap. (c) Histogram of the distance of matched descriptors: the blue bars represent pair (a) and the red bars the pair (b). (d) Variation of the overlap measure over time.

features, while the smoothness energy measures the discontinuities on S; this energy is important to remove wrong feature matchings. However, the initial formulation described by Equation 4.3 is suitable for pairwise image registration. The registration of sequences of images poses some additional problems. If only pairwise registration is used to align a sequence of images, over-deformation due



(a) (b)

Figure 4.4: Error accumulation using homography. (a) Rendered mosaic. (b) Projected frame borders. The last frame is the most deformed.

to error accumulation may occur (Figure 4.4).

To avoid error accumulation, we propose a modified version of the previous energy function. The new term, $E_{\text{Ref}}(S - S_{\text{Ref}})$ is named *reference mesh energy*. The mesh S_{Ref} represents a model of how the mesh S should look like without over-deformation. Alternatively, it may roughly represent how the user of the mosaic system would expect the image (warped by S) to look like. The constant μ regulates the reference mesh energy weight. The new formulation is presented below:

$$E'(S) = E_C(S) + \lambda E_{\text{Sm}}(S) + \mu E_{\text{Ref}}\left(S - S_{\text{Ref}}\right) \quad . \tag{4.4}$$

Correspondence Energy: The correspondence energy $E_{\text{C}}(S)$ is a function of the projection error of the matched features. The matched feature set is represented by M. The matched feature pair $c \in M$ is composed of two features (c_0, c_1), where c_0 is a feature found in the target image and c_1 is its paired feature found in the image being registered. The warp function is denoted by $w(c_1, S)$. The function υ is the same robust estimator used by Zhu et al. [2009]. It is defined below:

$$E_{\text{C}}(S) = \sum_{c \in M} \upsilon\left(c_0 - w(c_1, S), \sigma\right); \quad \upsilon(\delta, \sigma) = \begin{cases} \frac{||\delta||^2}{\sigma^n} & \text{if } ||\delta|| \leq \sigma \\ \\ \sigma^{2-n} & \text{otherwise} \end{cases} \tag{4.5}$$

The function υ has two parameters: the projection error δ and the radius of tolerance σ. The matches whose projection errors are greater than the radius of tolerance are considered outliers and penalized. The radius of tolerance σ dictates

which matched feature pairs will be considered outliers, conferring robustness to the registration procedure.

Smoothness Energy: The correspondence energy, if used alone, is sensitive to outliers among the matched features. A smoothness constraint is added to the model in order to avoid this problem. The proposed method uses the same smoothness constraint found in Zhu et al. [2009] and Pilet et al. [2005]. This energy is the sum of the approximate second derivative of the mesh S. Let E be the set of all collinear control points in S that define two adjacent edges. The smoothness energy is defined below:

$$E_{\text{Sm}}(S) = \sum_{i,j,k \in E} (-x_i + 2x_j - x_k)^2 + (-y_i + 2y_j - y_k)^2 = X^{\text{T}} K X + Y^{\text{T}} K Y \ , \tag{4.6}$$

where $K = K'^{\text{T}} K'$, and K' is a matrix containing one row per triplet in E and one column per mesh vertex. The row corresponding to the triplet (i, j, k) has all of its values zero except by values in columns i, j, and k, that have values -1, 2, and -1, respectively Pilet et al. [2005].

Reference Mesh Energy: The registration using the energy function in Equation 4.3 is only suited for pairwise registration, because registration error may accumulate, as shown in Figure 4.4. The role of the reference mesh energy is to alleviate this problem. This energy is proportional to the L_2 distance between the mesh S and the reference mesh S_{Ref}. The former is the registration solution and the latter is an approximation of how S should be if it had no over-deformation. The criteria selected to generate S_{Ref} was to make it look similar to the original captured image. S_{Ref} is defined as the similarity transformation (*i.e.*, rotation, translation and scaling) that minimizes the correspondence energy. This kind of rigid transformation was selected because it preserves the proportions of the original key-frame. The reference mesh can be computed efficiently by minimizing the projection error using the similarity transformations combined

with RANSAC (Section 1.3.2). The reference mesh energy is defined below:

$$E_{\text{Ref}}\left(S - S_{\text{Ref}}\right) = \frac{1}{2}||S - S_{\text{Ref}}||^2 \quad . \tag{4.7}$$

During the optimization process, the reference mesh energy is stronger in the regions of the mesh S where there are no features. While the region with features is deformed to minimize the projection error, the region without features is deformed by similarity transformations. These local differences in the deformation field are not possible for rigid deformation models.

Optimization Routine: As pointed in Zhu et al. [2009], the projection error δ can be written as a linear system. Given that: $c_0 = (c_{0x}, c_{0y})$, $c_1 = (c_{1x}, c_{1y})$:

$$||\delta||^2 \quad = \left(c_{0x} - t^{\text{T}}x\right)^2 + \left(c_{0y} - t^{\text{T}}y\right)^2 \quad , \tag{4.8}$$

where x and y are the coordinates of the mesh and $t_{c_1} \in R^N$ is a vector (N is the number of control points) representing the barycentric coordinates of the feature point c_1, which is inside the triangle defined by $v_i, v_j, v_k \in S_0$, (calculated in the identity mesh). The vector t_{c_1} has all its values 0, except in the coordinates i, j, and k, where the barycentric coordinates of c_1 in relation to v_i, v_j, and v_k are set, respectively. Using Equation 4.6 and Equation 4.7, the energy $E'(S)$ in Equation 4.4 can be rewritten as:

$$E'(S) = \tfrac{1}{\sigma^n}\textstyle\sum_{c\in M_{Inl}} \left(c_{0x}^2 + c_{0y}^2 - 2 \begin{bmatrix} c_{0x}t \\ c_{0y}t \end{bmatrix}^{\text{T}} S + S^{\text{T}} \begin{bmatrix} tt^{\text{T}} & 0 \\ 0 & tt^{\text{T}} \end{bmatrix} S \right) + |M_{Out}|\sigma^{2-n} + \lambda\left(X^{\text{T}}KX + Y^{\text{T}}KY\right) + \tfrac{\mu}{2}||S - S_{\text{Ref}}||^2 \quad , \tag{4.9}$$

where M_{Inl} is the set of inlier matches, M_{Out} is the set of outlier matches. The following definitions are done for simplification: $A = \frac{1}{\sigma^n}\sum_{c\in M_{Inl}} tt^{\text{T}}$, and $b = \begin{bmatrix} b_x \\ b_y \end{bmatrix} = \frac{1}{\sigma^n}\sum_{c\in M_{Inl}} \begin{bmatrix} c_{0x}t \\ c_{0y}t \end{bmatrix}$. Computing the gradient of E' and setting it to

zero, the mesh S can be found by solving a linear system:

$$S = \begin{bmatrix} \lambda K + A + \mu I & 0 \\ 0 & \lambda K + A + \mu I \end{bmatrix}^{-1} \left(b + \mu S_{\text{Ref}}\right) \quad . \tag{4.10}$$

The optimization is repeated varying the value of σ, which decreases during the optimization procedure. In the beginning, σ is large, allowing many possible outliers to influence the result of the optimization process. However, since the module of the derivative of the E_{C} is small when σ is large, E_{Sm} and E_{Ref} have a larger weight and they initially guide the optimization. As the value of σ decreases, the weight of E_{C} increases, guiding the optimization to minimize the projection error of the remaining inliers. In this way, this registration method is robust to outliers. The process stops when σ is smaller than a given threshold.

In order to display the results, the registered images are stitched together using the method described in the next section.

4.5.2 Efficient stitching algorithm for non-rigid image registration

It is generally impossible to guarantee that the alignment of all regions of an image will be equally good. This is specially true in the particular case of non-rigid registration, where regions with few features are often misaligned. For this reason, the key-frames must be carefully stitched into the mosaic. The standard approaches for this problem consist of selecting which pixels of the new image will be used in the mosaic. This is generally done by defining a stitching line. Pixels in one side of the stitching line are added to the mosaic, and the pixels in the other side are ignored. A common criteria for selecting the stitching line is to make it pass through pixels which are well aligned to the mosaic. By this, the pixels in both sides of the stitch (the pixels of the new registered key-frame and the pixels already in the mosaic) will be similar and the final composite mosaic will be seamless. The methods in Kwatra et al. [2003], for example, use this approach. However, pixel selection is too slow for a real-time method, specially

dealing with high resolution images.

Our proposed solution to this problem is to select mesh triangles instead of pixels. Even in a mesh with a high degree of freedom, there are considerably fewer triangles than pixels in the key-frame. In our approach, we define a stitching line that passes through triangles that are well aligned. The number of inlier features (*i.e.*, feature points correctly aligned to the mosaic) inside a triangle was used to evaluate its alignment. It was observed that triangles with inlier features inside are much better aligned than triangles without them. The stitching line is selected by solving a graph cut formulation, which is presented below.

Let $S_0, S_1, S_2, ..., S_{n-1}$ be the meshes already added to the mosaic. They are represented in gray in Fig 4(a). Let $T_0, T_1, T_2, ..., T_{n-1}$ be the set of triangles that compose these meshes. Each one of these sets is defined as $T_i = \{\tau_i^0, \tau_i^1, ..., \tau_i^m\}$, where m is the number of triangles inside each mesh (m is constant). Each triangle is defined by three non-identical control points: $\tau_i^j = \{v_a, v_b, v_c\}; v_a, v_b, v_c \in S_i$. Now, let $T'_0, T'_1, T'_2, ..., T'_{n-1}$ be the set of triangles, for all previously added meshes, which were selected to be included into the mosaic. Therefore, $T'_i \subseteq T_i$.

Let S_n be the next mesh to be inserted into the mosaic (represented in red in 4(a), and C_n the matched features of the last key-frame F_n. Let $\omega : T_n \rightarrow \mathbb{N}$ be a function that receives as parameter a triangle τ_n^j from the mesh S_n and returns the number of aligned features which lie inside τ_n^j.

To use graph cut, a graph and an edge-weight function must be defined. The graph $\{U, E\}$ is constructed based on S_n. The set of vertices U has one vertex u_j for each triangle τ_n^j in S_n. The set of edges E has one edge for each pair of adjacent triangles in S_n (two triangles are adjacent if they share an edge). To follow the graph cut formulation, two special vertices, $\mathfrak{s}$ and $\mathfrak{t}$ (source and sink) are added. Let Tb_n be the set of triangles that are on the borders of S_n. For each one of the triangles in Tb_n one extra edge is added to E: if a triangle $\tau_b \in Tb_n$, represented in U by the vertex u_j, has at least one intersection with one of the triangles already in the mosaic, the edge $(\mathfrak{s}, u_j)$ is added. These triangles will definitely not be added to the mosaic. Otherwise, the edge $(u_j, \mathfrak{t})$ is added to E. It means that the border triangles that do not intersect the mosaic will definitely be added to the mosaic. Now let $w : E \rightarrow \mathbb{R}$ be an edge-weight function. This function is defined as follows:

$$w\left(u_i, u_j\right) = \begin{cases} \left(\omega\left(\tau_n^i\right) + \omega\left(\tau_n^j\right) + \varepsilon\right)^{-1} & \text{For } u_i \neq \mathfrak{s} \text{ and } u_j \neq \mathfrak{t} \\ \infty & \text{otherwise} \end{cases} \tag{4.11}$$

where ε is a small value used to avoid divisions by zero. This function gives weights that are inversely proportional to the number of inlier features inside the adjacent triangles, enforcing that the minimum cut must pass through the triangles with the most inlier features. The edges from the sink and source vertices are given infinite weight.

The graph generated by this process is illustrated in Fig. 4(b). Using these definitions, the optimal stitching can be computed using the max-flow min-cut algorithm as defined in Y.Boykov and Jolly [2001], for example. The triangles τ_n^j whose vertices u_j end up in the side of $\mathfrak{s}$ (regarding the optimal cut) are not added to the mosaic. The other triangles are selected and added to ${T'}_n$. Figure 4(c) shows the minimum cut obtained by the graph-cut algorithm.

4.5.3 Efficient non-rigid image registration with globally consistent alignment for flexible image mosaicing

The next proposed method is similar to the mosaicing system presented in Section 4.5.1, except that it takes unordered image sets as input. It is intended to be used as a post-processing method used to improve the final mosaic. It can be used together with our video-mosaicing system or any other registration method. Since all images are already available, this method can perform global registration, *i.e.*, all images are registered at the same time, taking all other images into account. However, the problems with efficiency become more serious, since the number of variables of the deformation model is much greater. This section will present a registration method that can address these problems.

The proposed registration system, detailed in Figure 4.6, is composed of 6 important modules. The input image module is responsible for reading the input image set and detecting the feature points in each image. The topology infer-

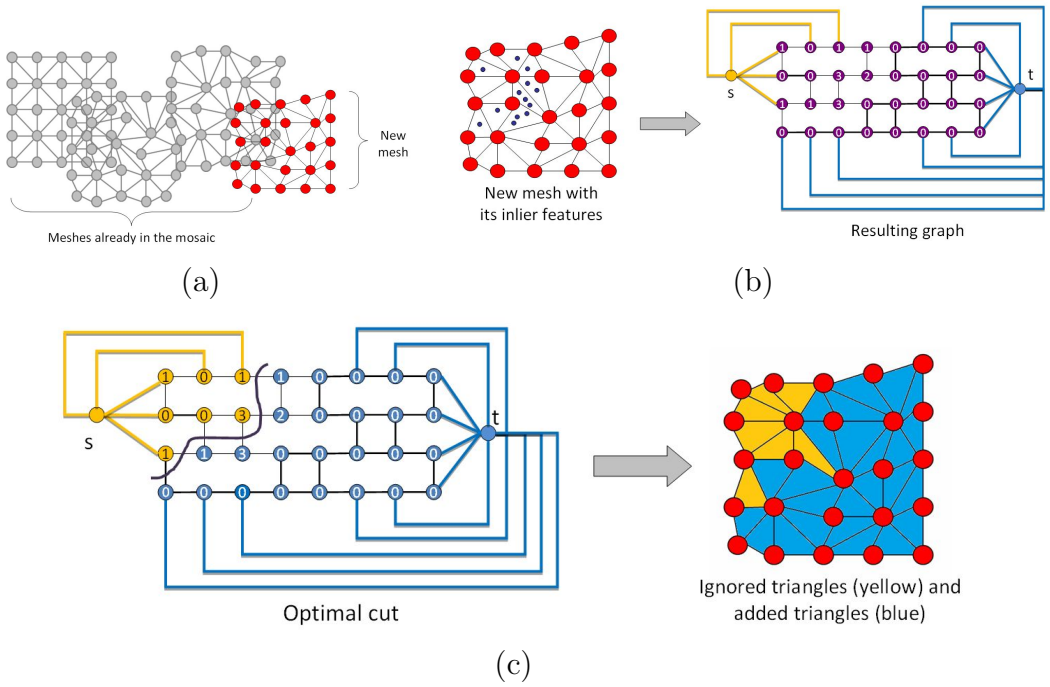


Figure 4.5: Proposed triangle-wise graph cut algorithm. (a) New mesh added into the mosaic. (b) A graph is created to represent the new mesh; vertices representing the border triangles which overlap with the mosaic receive a $\mathfrak{s}$ edge and the other vertices representing border triangles receive a $\mathfrak{t}$ edge; the weight of each vertex is inversely proportional to the number of aligned features inside its corresponding triangle. (c) The minimum cut is computed; the triangles whose vertices are in the side of s will not be added to the mosaic, while the other triangles will.

ence module detects, using the feature points, the relationship between the input images. It is responsible for creating the image graph (explained in the next section). The feature matching module receives the feature pairs and the image graph and uses them to create a list of matched features, an information used by the registration and rendering modules. The rigid registration model is responsible for generating a rough initial global registration. This initial solution is then passed to the non-rigid registration module, which creates the final registration that will be used by the rendering module to generate the final mosaic.

Details of the procedures described above are given in the following sections.

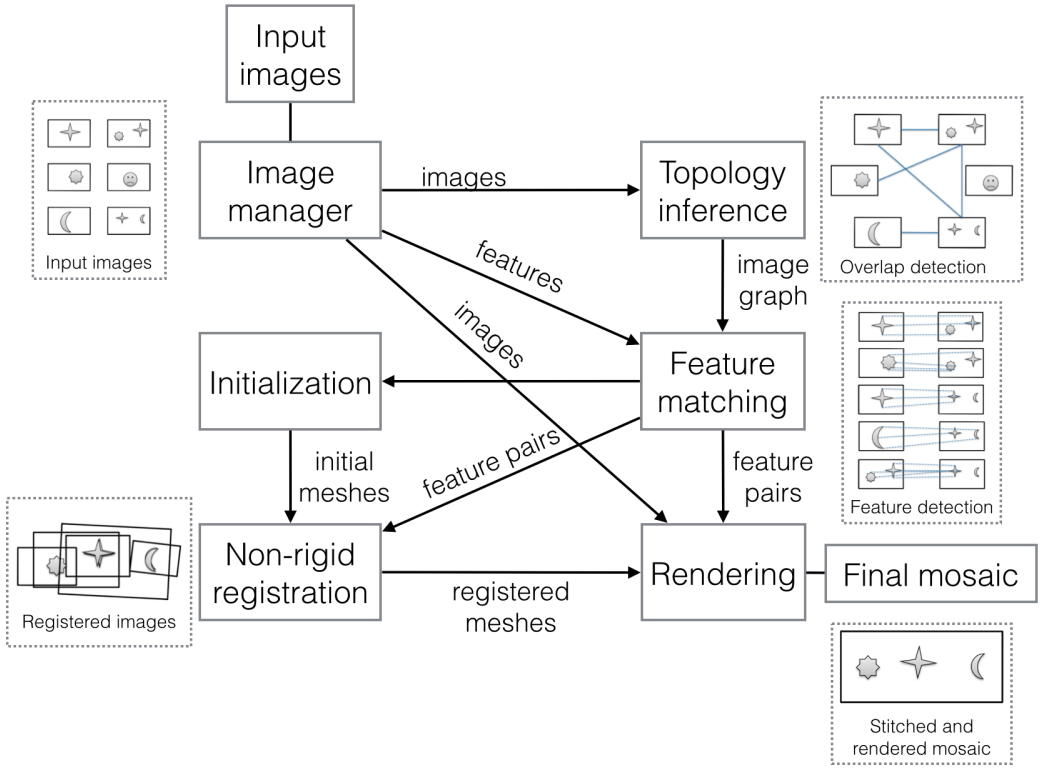


Figure 4.6: Globally consistent registration system diagram.

Building an image graph with efficient image similarity computation: First, in order to efficiently represent the relationship of the input images (*topology inference*), we build an image graph. In the image graph, each vertex corresponds to an input image. A pair of vertices is linked by an edge only if the corresponding images have visual overlap. We consider that a pair of images overlaps if there is a sufficient number of inlier feature point matches after geometric verification Hartley and Zisserman [2004]; Philbin et al. [2007]. Since finding feature correspondences by matching all pairs of input images is extremely costly, it is necessary to select subsets of reasonable candidates to perform actual feature matching. Such candidates are efficiently selected by using image similarities computed as cosines of angles of bag-of-features (BoF) image vectors Sivic and Zisserman [2006]. In our implementation, we give at most 6 candidates to each node for performing actual feature matching. This way of creating the image

graph is, in essence, the same as the method proposed in Agarwal et al. [2009].

Initialization by global rigid registration: The problem of global non-rigid image registration is extremely high dimensional. Therefore, to achieve good results, a proper initialization must be performed. This initialization is done using similarity functions, *i.e.*. translation, rotation, and scaling. Based on Brown and Lowe [2007], the process is as follows:

1. The image with most overlaps is the first to be added to the mosaic;
2. an input image which overlaps at least one image in the mosaic is added;
3. the warping of all images in the mosaic is updated.

Steps 2 and 3 are repeated until the mosaic is complete.

Other methods can also be used to generate the initial global registration.

Deformation model: Given the input image set $I = \{I_0, I_1, ..., I_{N-1}\}$, each image I_i has its corresponding warping parameters $\theta_i \in \theta$. Also, let F_i be the set of feature points in image I_i, and $F = \{F_0, F_1, ..., F_{N-1}\}$.

Energy function: The registration energy is the generalization for multiple images of the function defined in de Souza et al. [2012]; Pilet et al. [2005]; Zhu et al. [2009]. These are the same functions applied to the problem of real-time video mosaicing (Section 4.5.1), generalized to global registration. This function is defined as follows:

$$E(\theta) = E_C(\theta) + \lambda E_{Sm}(\theta) + \mu E_{Ref}(\theta - \hat{\theta}).$$

where E_C is the correspondence energy function, E_{SM} is the smoothness energy and E_{Ref} is the reference mesh energy.

The correspondence energy $E_C(\theta)$ is a function of the projection error of the corresponding features. Each correspondence $c \in C$ is composed of 4 values: (i, j, k, l), used as indices of the features F_{ik} and F_{jl} (feature k of image i, and vice versa). The warping function is represented by $w(F_{ik}, \theta_i)$. The function v

is the same robust estimator used in Section 4.5.1. The functions E_C and υ are defined below:

$$E_C(\theta) = \sum_{(i,j,k,l)\in C} \upsilon\left(w\left(F_{ik}, \theta_i\right) - w\left(F_{jl}, \theta_j\right), \sigma\right); \tag{4.12}$$

$$\upsilon(\delta, \sigma) = \begin{cases} \frac{||\delta||^2}{\sigma^n} & \text{if } ||\delta|| \leq \sigma \\ \\ \sigma^{2-n} & \text{otherwise} \end{cases}$$

Again, the function υ has two parameters: the projection error δ and the radius of tolerance σ. The matches whose projection error are greater than the radius of tolerance are considered outliers and penalized.

The smoothness energy E_{Sm} is the sum of the modules of the second derivative of θ_i. It is defined below:

$$E_{Sm}(\theta) = \frac{1}{2} \sum_{i=0}^{N-1} X_i^{\mathrm{T}} K X_i + Y_i^{\mathrm{T}} K Y_i,$$

where $K = K'^{\mathrm{T}} K'$, and K' is a matrix containing one row per pair of adjacent mesh edges $[(a, b), (b, c)]$ and one column per control point. The row corresponding to the triplet (a, b, c) has all of its values zero except in columns a, b, and c, that have values -1, 2, and -1.

The third term is the reference mesh energy. The reference mesh set $\hat{\theta}$ represents a rough approximation of how the images are expected to be aligned. This term is used to make the mosaic images not to deviate from their approximate optimum position. In fact, if the reference mesh energy is removed, the mosaic will collapse to a point, since a point would be an optimum solution of all other terms. The reference mesh energy is proportional to the L_2 distance between the

mesh θ_i and the reference mesh $\hat{\theta}_i$. It is defined below:

$$E_{Ref}(\theta - \hat{\theta}) = \frac{1}{2}\sum_{i=0}^{N-1} ||\theta_i - \hat{\theta}_i||^2.$$

The reference mesh parameters are initialized with the rigid mosaic results (Section 4.5.3) and are updated during the optimization (as explained below). The constants λ and μ regulate the energy weights.

Optimization routine: From Equation 4.1, the projection error of a pair of features F_{ik} and F_{jl} can be written as:

$$\begin{aligned} ||\delta||^2 &= \left(t_{ik}^{\mathrm{T}} X_i - t_{jl}^{\mathrm{T}} X_j\right)^2 + \left(t_{ik}^{\mathrm{T}} Y_i - t_{jl}^{\mathrm{T}} Y_j\right)^2 \\ &= \left(X_i^{\mathrm{T}} t_{ik} t_{ik}^{\mathrm{T}} X_i - 2X_i^{\mathrm{T}} t_{ik} t_{jl}^{\mathrm{T}} X_j + X_j^{\mathrm{T}} t_{jl} t_{jl}^{\mathrm{T}} X_j\right) \\ &+ \left(Y_i^{\mathrm{T}} t_{ik} t_{ik}^{\mathrm{T}} Y_i - 2Y_i^{\mathrm{T}} t_{ik} t_{jl}^{\mathrm{T}} Y_j + Y_j^{\mathrm{T}} t_{jl} t_{jl}^{\mathrm{T}} Y_j\right) \end{aligned}$$

Let T_{ik}^2, T_{jl}^2 and $T_{ik}T_{jl}$ be $\begin{pmatrix} t_{ik}t_{ik}^{\mathrm{T}} & 0 \\ 0 & t_{ik}t_{ik}^{\mathrm{T}} \end{pmatrix}$, $\begin{pmatrix} t_{jl}t_{jl}^{\mathrm{T}} & 0 \\ 0 & t_{jl}t_{jl}^{\mathrm{T}} \end{pmatrix}$, and $\begin{pmatrix} t_{ik}t_{jl}^{\mathrm{T}} & 0 \\ 0 & t_{ik}t_{jl}^{\mathrm{T}} \end{pmatrix}$, respectively. (4.12) can be rewritten as:

$$\begin{aligned} E_C(\theta) = \tfrac{1}{\sigma^n} \sum_{c \in C_{inl}} &\left(\theta_i^{\mathrm{T}} T_{ik}^2 \theta_i - 2\theta_i^{\mathrm{T}} T_{ik} T_{jl} \theta_j + \theta_j^{\mathrm{T}} T_{jl}^2 \theta_j\right) \\ &+ |C_{out}| \sigma^{2-n}, \end{aligned}$$

where c is a correspondence, C_{inl} and C_{out} are the set of inlier and outlier correspondences, according to σ. The derivatives regarding θ_i can be computed as:

$$\frac{\partial E_C}{\partial \theta_i} = \frac{1}{\sigma^n} \sum_{c \in C_{inl}^{(i)}} \left(2T_{ik}^2 \theta_i - 2T_{ik} T_{jl} \theta_j\right),$$

where $C_{inl}^{(i)}$ are all inlier correspondences which has one of its feature points in F_i. Since θ_i is constant in the summation, the equation can be rewritten as:

$$\frac{\partial E_C}{\partial \theta_i} = \frac{2}{\sigma^n} \sum_{c \in C_{inl}^{(i)}} \left(T_{ik}^2\right) \theta_i - \frac{2}{\sigma^n} \sum_{c \in C_{inl}^{(i)}} \left(T_{ik} T_{jl} \theta_j\right).$$

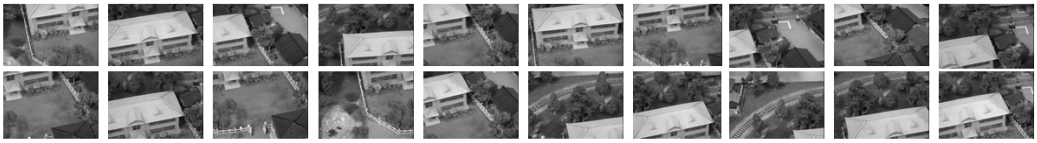
The derivatives of the smoothness and reference mesh energies can be computed as:

$$\frac{\partial E_{Sm}}{\partial \theta_i} = \lambda \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \theta_i, \quad \frac{\partial E_{Ref}}{\partial \theta_i} = \mu \left(\theta_i - \hat{\theta}_i\right).$$

Combining the three derivatives:

$$\begin{aligned} \frac{\partial E}{\partial \theta_i} = \; & \frac{2}{\sigma^n} \left(\sum_{c \in C_{inl}^{(i)}} \left(T_{ik}^2\right) + \lambda \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} + \mu I \right) \theta_i \\ & - \frac{2}{\sigma^n} \sum_{c \in C_{inl}^{(i)}} \left(T_{ik} T_{jl} \theta_j\right) - \mu \hat{\theta}_i. \end{aligned}$$

Setting the derivative to zero, θ_i can be computed by a linear system which is solved iteratively. In each iteration σ is reduced by ν. This increasingly gives more weight to the correspondence energy and rejects more outlier correspondences. Also, after each iteration, $\hat{\theta}$ is updated. Each $\hat{\theta}_i$ is set to be the mesh deformed by similarity transformations which is the closest to θ_i. This can be quickly calculated by least squares regression. Figure 4.7 shows an example of a result obtained by our proposed method.



(a)

(b)

Figure 4.7: Example of registration results. (a) 20 input images. (b) Globally consistent mosaic created by our proposed method.

Chapter 5

Experiments on efficient registration for image mosaicing

This chapter presents the experimental results for our mosaicing methods. First, the implementation details carried out for the experiments are presented, followed by three sections detailing the results for each one of our proposed methods: real-time mosaicing, globally consistent mosaicing, and fast stitching.

This chapter shows the experimental results for our mosaicing methods. We first describe the experimental setup for image/video acquisition and the implementation details. Next, for video stream inputs, we demonstrate that our mosaicing by the proposed non-rigid registration is capable for running in real-time while performing more precise registration than standard methods with the rigid deformation models. We qualitatively evaluate that our proposed graph cut algorithm improves the quality of the final mosaics.

The remaining of this chapter is organized as follows. Section 5.1 explains implementation details concerning the experimental setup. Section 5.2 presents the experimental results of our real-time video mosaicing method. Section 5.3 presents the qualitative results of our fast stitching method. Finally, Section 5.4 presents the qualitative results of our globally consistent registration method.

Param.	Value	Description
ϑ	0.4	Frame selection threshold.
ς	1.0	Frame selection weight function std. deviation.
λ	10^{-6}	Smoothness energy parameter.
μ	10^{-7}	Reference mesh energy parameter.
n	4	Correspondence energy parameter.
σ_0	32	Registration parameter; initial radius of tolerance.
σ_{min}	3	Minimum radius of tolerance; *i.e.*, projection error.
η	0.5	Radius of tolerance decay rate.

Table 5.1: Parameter settings for the proposed method.

5.1 Implementation details

The project was run in a computer with Intel(R) Core(TM) i7 CPU (2.93 GHz) and 4GB of RAM. The proposed method was implemented using the OpenCV library. The parameter setting is presented in Table 5.1.

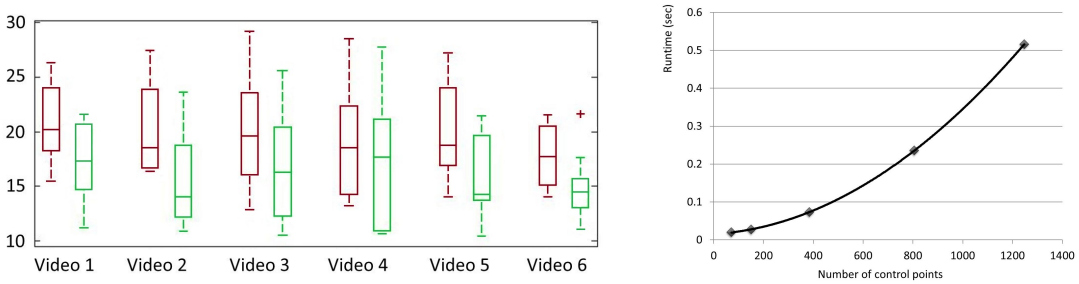
For the reference mesh computation, the precision of RANSAC is set to 99% in the presence of 70% of outliers. The size of the mesh was 19×28 control points. The videos used on the experiments had a resolution of 720×480.

In the case of our proposed globally consistent mosaicing method, the experimental setup is as follows. The non-rigid energy parameters λ and μ were set to 10^{-5} and 10^{-9}, respectively. The decay rate ν of σ is set to 1.30, its initial value is set to 50 and its minimum value is set to 7. The optimization takes 8 iterations to finish. Each mesh has 280 control points.

5.2 Results: video mosaicing

5.2.1 Registration Precision

This experiment presents the comparison between homography and non-rigid transformations concerning precision. The experiments use the mean appearance error, defined as the mean absolute difference between all aligned pixels. The experiments were conducted by registering pairs of images. Figure 5.1(a) shows



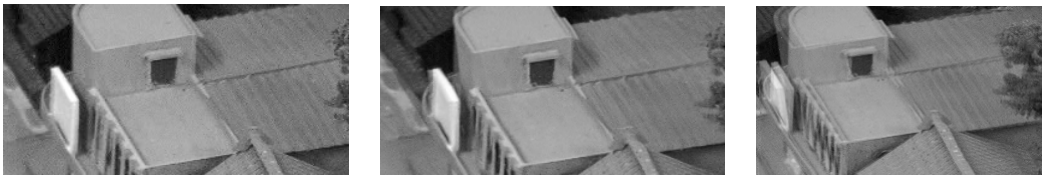
(a) (b)

Figure 5.1: Quantitative experiment results. (a) Appearance error with homography and non-rigid transformations. The error is measured as the mean absolute difference between pixel gray-scale values of aligned pixels, in a set of videos. The red boxes show the results obtained by homography, and the green boxes represent the results of the proposed method. (b) Execution time (seconds) in relation to number of control points.

the results of the average error of pair-wise registration over different video sequences. Figure 5.2 shows a detail of a pair of registered frames (the averaged image). As can be seen, the results achieved by the the proposed method are always more precise than the results using homography. This happens because the deformation field between the pairs of images can not be precisely described by a global transformation like projection, since the displacement field depends on the geometry of the scene.

5.2.2 Over-deformation avoidance

This set of experiments compares mosaics done by the proposed method and non-rigid image registration as described by Zhu et al. [2009]. The comparisons are done regarding over-deformation. Figure 5.3 shows the results. Both methods use the same set of frames. As previously shown in Figure 4.4, using homography, the registration error tends to build up and cause the frames to over-deform. When using only non-rigid registration, without the reference mesh energy, error accumulation also happens, even though the alignment error is small when compared to homography. The proposed method, using the reference mesh energy, mini-



(a) (b) (c)

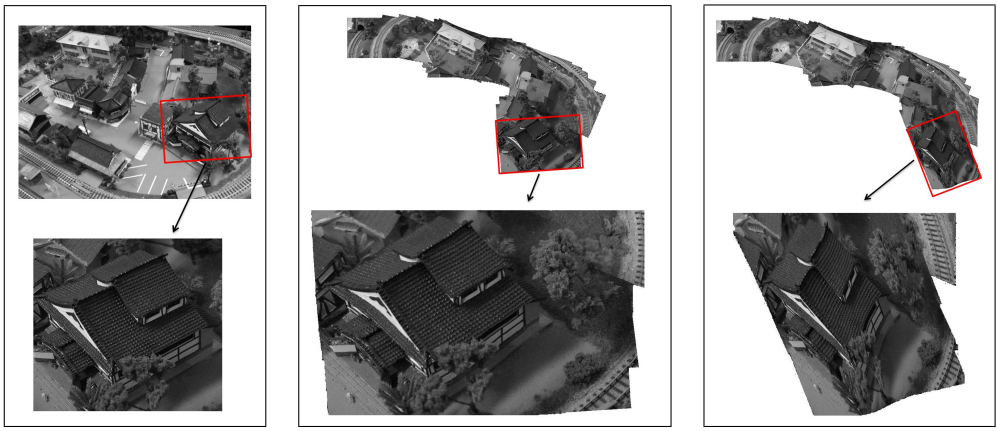
Figure 5.2: Pairwise registration precision. (a) Original video frame. (b) Detail of a pair of registered frames, aligned by the proposed method. (c) Detail of a pair of registered frames, aligned using homography. The overlapping region of the registered frames was averaged.

mizes the amount of over-deformation. This result may be achieved by related methods using bundle adjustment, but the proposed method achieves the same by only doing pair-wise registration.

5.2.3 Comparison with a standard method

In this set of experiments, the proposed method was compared to a standard method, implemented by Microsoft Image Composite Editor (ICE), version 1.3.5. Using ICE, the user can choose different camera movements. The one which yielded the best result was selected. The proposed method used the parameters described in Section 5.1. ICE and the proposed method used the same set of key-frames. Figure 5.4 shows the mosaic created from a video taken by a camera moving over a city model.

Figure 5.4(a) shows the results obtained by the proposed method. Figure 5.4(b) shows the results obtained by ICE. As can be seen, the results obtained by the proposed method are more complete than the results given by ICE. This happens because of the complex camera movement and the non-planar surface, which violate the projection constraints used by ICE.



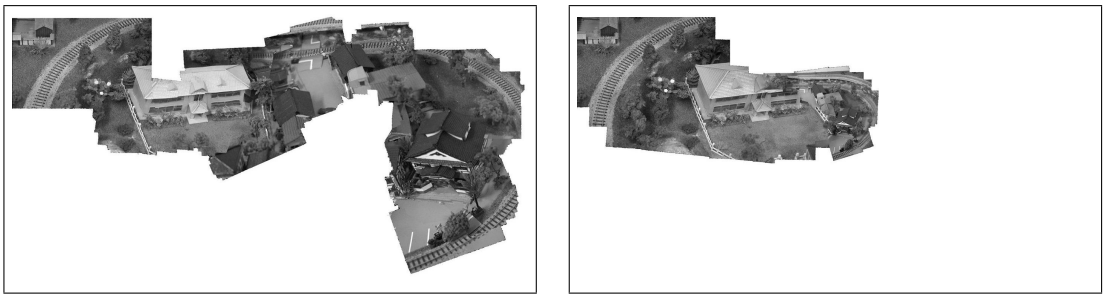
(a) (b) (c)

Figure 5.3: Mosaicing results, regarding overdeformation. (a) City model used in the experiments, showing the expected undeformed frame. (b) Result obtained by the proposed method. (c) Results obtained using only non-rigid registration without the reference mesh energy. The result generated by the proposed method shows less deformation.

5.2.4 Computational complexity

The current implementation of the proposed method runs in about 32 frames per second with a rate of 2 frames selected per second, reasonable for videos where the camera movement is not excessively fast.

Each iteration of frame selection takes approximately 0.031 seconds, so the frame rate is about 32 frames per second, enough for most videos. Figure 5.1(b) shows runtime regarding only the registration procedure. It was executed 10 times for each quantity of control points (the computation of the reference mesh is included). As can be seen from the experiments, registration runtime grows slowly. This happens because the implementation uses sparse matrices to represent the registration model. The runtime of the frame selection and mosaic creation procedure was also computed. Using approximately 1000 triangles, the registration can be done in about 3 frames per second. Regarding the mosaic creation, each frame takes on average 0.4 seconds to be added into the mosaic, a tax of nearly 2 frames per second.



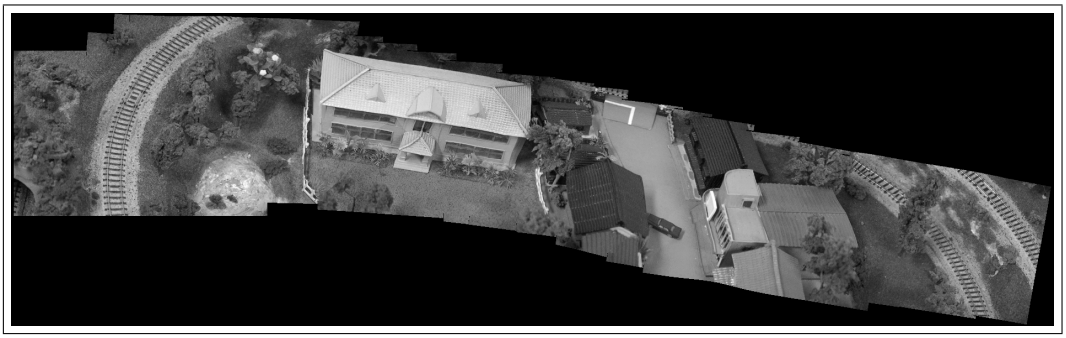
(a) (b)

Figure 5.4: Comparison between the proposed method and a standard method; (a) shows the result of the proposed method; (b) shows the result of the standard method. The proposed method created a more complete mosaic since it can handle more complex camera movements.

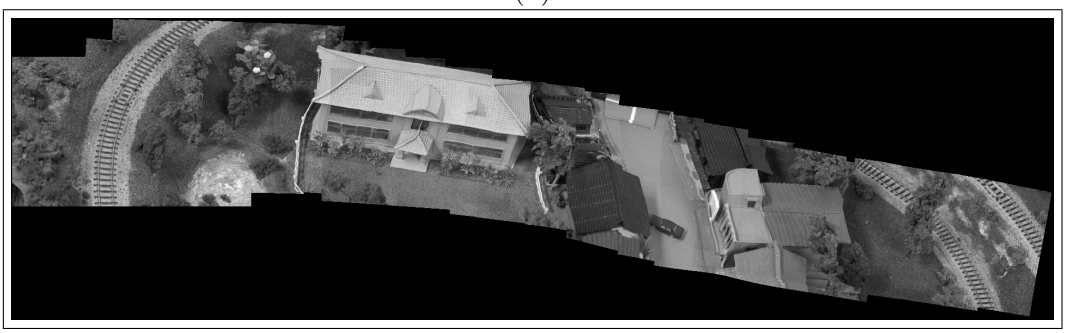
The conclusion is that the proposed method can run in real-time, given the conditions above. Further optimization on the method may be performed in the future.

5.3 Results: fast image stitching

This set of experiments is done to evaluate the improvements in the final mosaic when applying the triangle-wise graph cut algorithm presented in Section 4.5.2. The comparisons were done using two input videos. The proposed stitching method was compared to the mosaics created overlapping consecutively key-frames. Both results use the same set of input images and the same registered feature points. Figure 5.5 shows the results from the first video, and Figure 5.6 shows the results for the second video. Figure 5.5(a) and Figure 5.6(a) show the results of the proposed method, while Figure 5.5(b) and Figure 5.6(b) show the results of mosaic created by overlapping the key-frames. Figure 5.7 and 5.8 show details of these mosaics. Figure 5.9 shows the triangles selected by the proposed method to be included into the final mosaics. As can be seen in these results, the proposed stitching scheme can ignore most registration errors which occur in regions without inlier matched features.



(a)



(b)

Figure 5.5: Mosaic stitching results. (a) Results of the proposed image stitching method. (b) Results obtained by overlapping the selected key-frames. The mosaic generated by the proposed method presents much less seam marks.

5.4 Results: globally consistent image mosaicing

We demonstrate the advantages of the proposed method through experiments on several sets of real images.

Figure 5.10(a) shows an example of input images, which capture a (non-planar) city georama model. An example of our final result is shown in Figure 5.10(b). We compare the results on each set of images with the registration approaches used in state of the art softwares. We chose the global registration using affine model as the baseline for the comparison, which is used by Microsoft Image Composite Editor (ICE). Figure 5.10(c) and (d), show the results of affine registration and the mosaic created by ICE, respectively. *Note that the result created*

Table 5.2: Registration error (RMSE) of overlapping pixels. The proposed method is consistently more precise than affine registration.

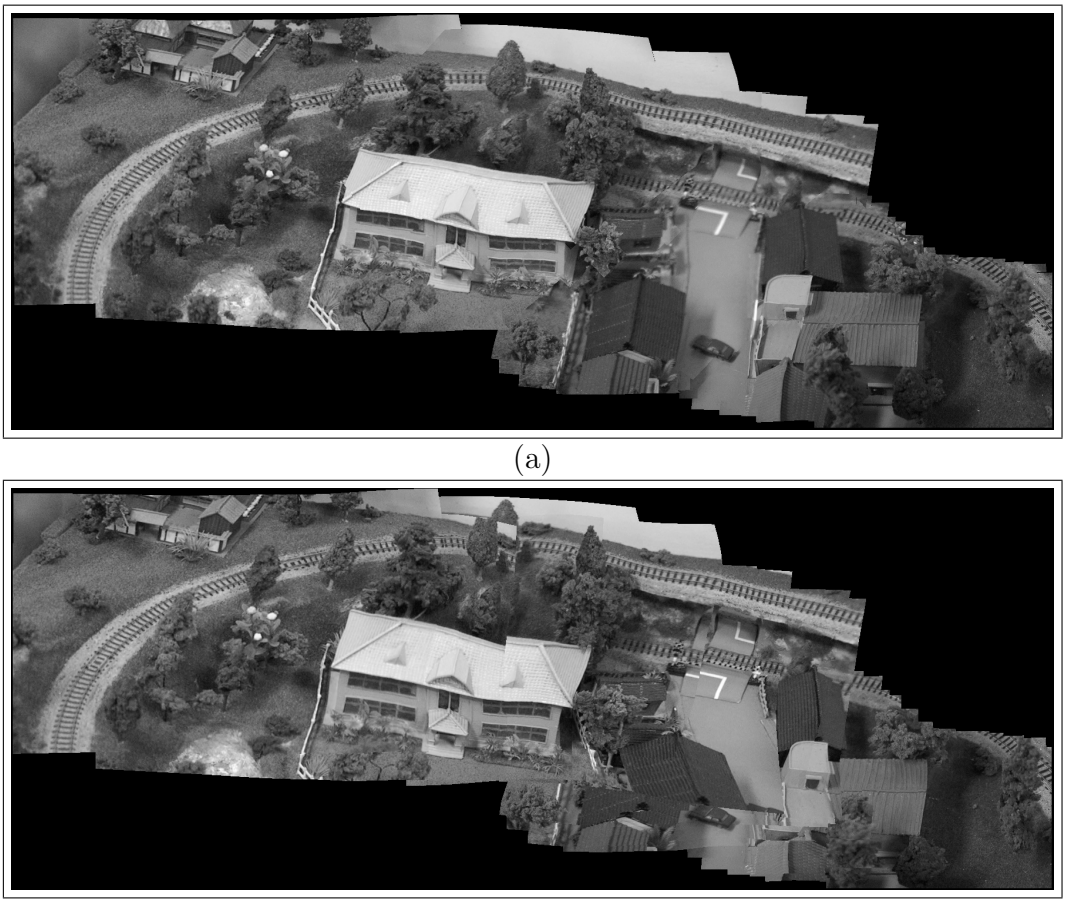
Dataset	1	2	3	4	5	6	7
Proposed	20.5	18.6	22.9	23.8	12.3	27.3	15.5
Affine	26.8	28.3	26.8	26.8	18.6	32.2	16.7

by ICE is post processed; this is out of the scope of these experiments. Since our result in Figure 5.10(b) registers fairly well compared to the other methods, post processing would most likely generate a seamless mosaic.

Figures 5.11, 5.12, and 5.13 show more examples of the results obtainable with our proposed method. Figure 5.11 shows a mosaic globally registered using a rigid warping function (affine), while Figure 5.12 shows the same images being aligned using our proposed method. The alignment using our proposed method is more consistent. Figure 5.13 shows the comparison between a mosaic created using our video mosaicing method (left) and our globally consistent registration method (right). This shows global registration can be used as post-processing to improve the mosaics created by our video mosaicing method.

As a quantitative evaluation, the registration error was measured on each set of images as the *RMSE* of intensity differences in overlapping pixels, for all pairs of overlapping images. The results are summarized in Table 5.2. As we can see, the proposed method consistently yields less error than global affine registration.

We also conducted experiments to evaluate the computational efficiency and scalability of our method. They were run in a 2.00GHz Core Duo notebook with 2.00GB RAM. Figure 5.14 shows the results of iteration time in relation to the number of images. As we can see, the time grows linearly, even though the size of the linear system grows quadratically. This is due to the sparsity of the linear systems.



(a)

(b)

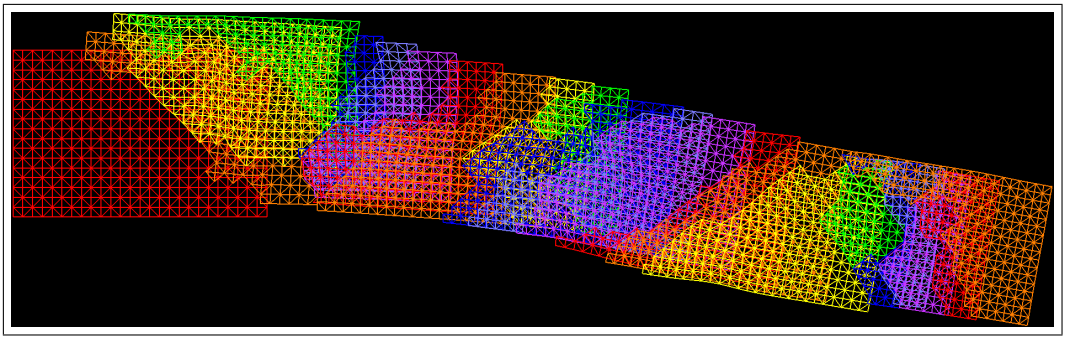
Figure 5.6: Mosaic stitching results. (a) Results of the proposed image stitching method. (b) Results of just overlapping the selected key-frames. The mosaic generated by the proposed method presents much less seam marks.

(a) (b) (c)

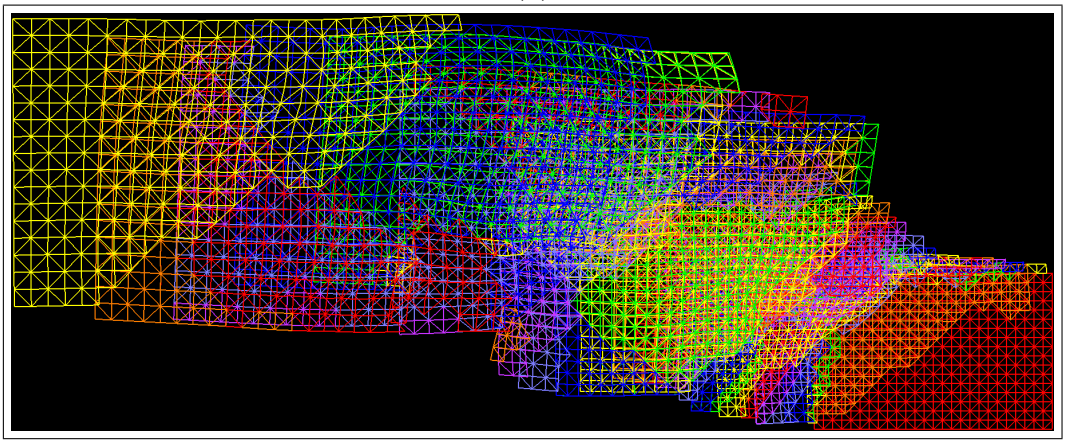
Figure 5.7: Details of the mosaic in Figure 5.5. (a) Video key-frame. (b) Result of the proposed stitching method. (c) Result of overlapping the key-frames. The seam marks are nearly invisible.

(a) (b) (c)

Figure 5.8: Details of the mosaic in Figure 5.6. (a) Video key-frame. (b) Result of the proposed stitching method. (c) Result of overlapping the key-frames. The seam marks are nearly invisible.

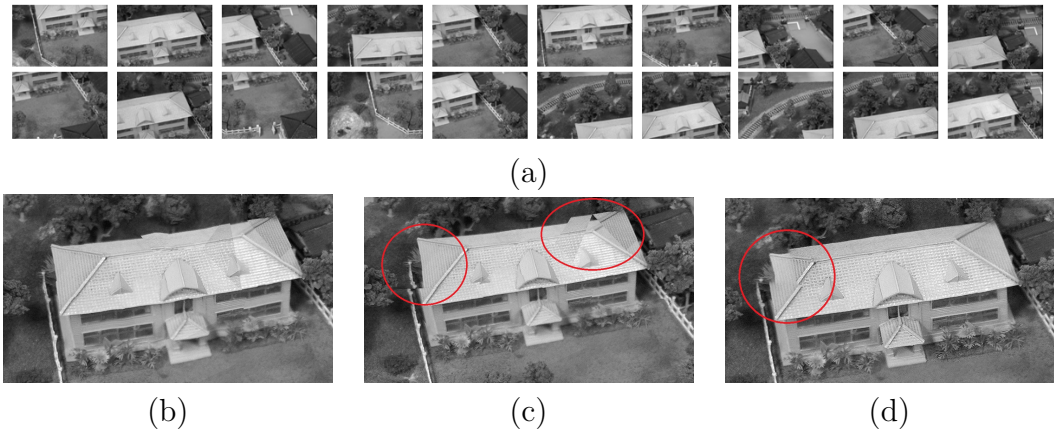


(a)



(b)

Figure 5.9: Triangles selected by the proposed stitching method in (a) Figure 5.5 and (b) Fig. 5.6. The triangles are selected in order to minimize the amount of seam marks.



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

Figure 5.10: Example of registration results. (a) 20 input images. (b) Proposed method. (c) Affine global registration result. (d) Microsoft Image Composite Editor (ICE), using homography. The regions with strong misalignment are circled. Note that Microsoft ICE uses complex post-processing methods. We can evaluate that the proposed method generates a result more geometrically consistent.

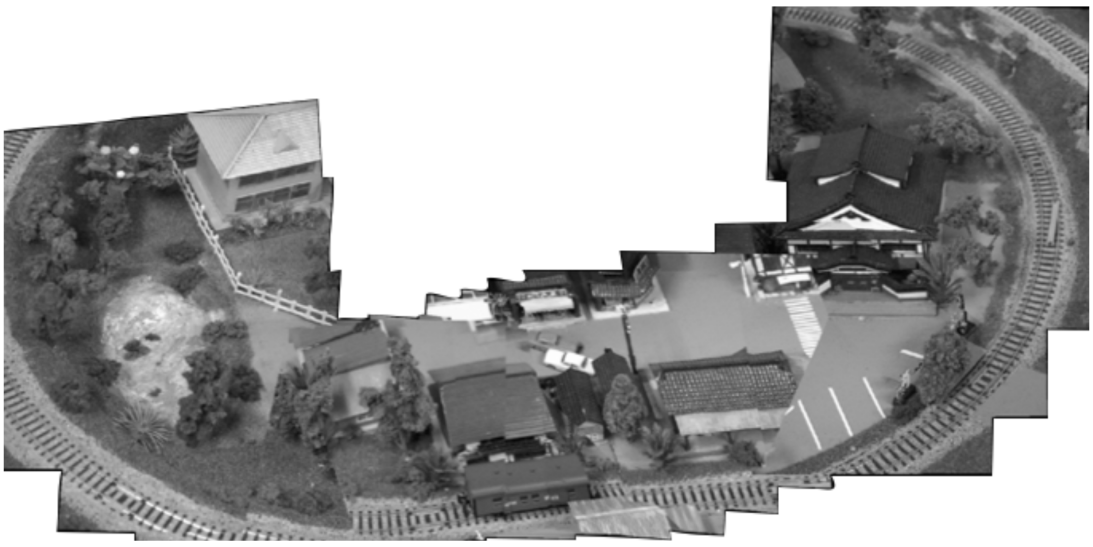


Figure 5.11: Mosaic created by global registration using affine transformations.

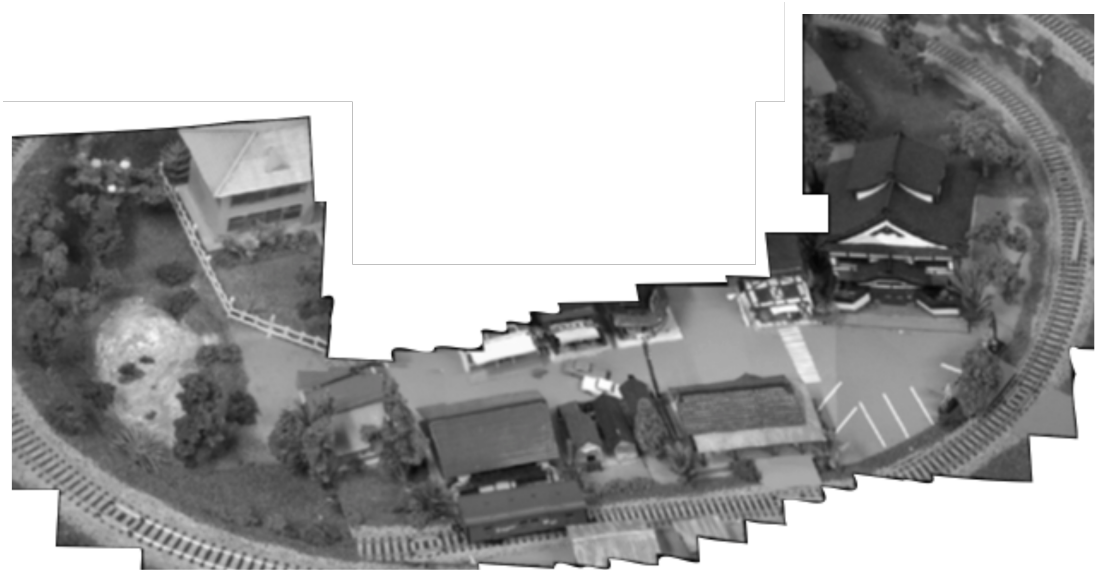


Figure 5.12: Mosaic created by global registration using our proposed method.

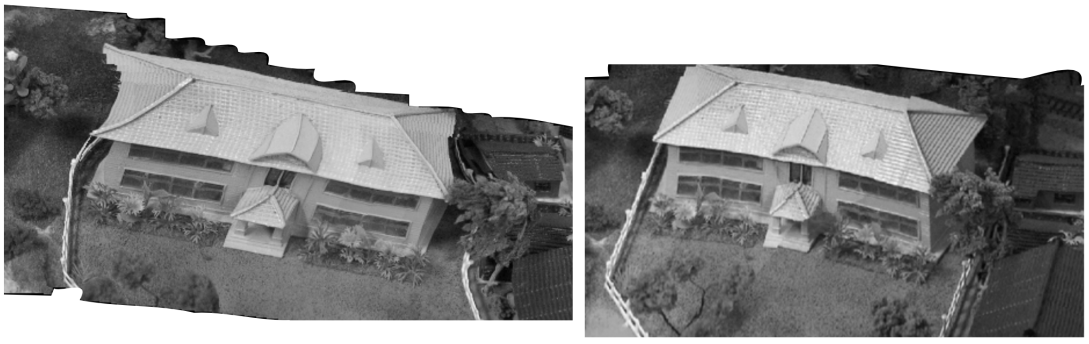


Figure 5.13: Detail of a mosaic created by our proposed video mosaicing method (left) and by our globally consistent registration method (right). The result shows that global registration can be useful for post-processing.

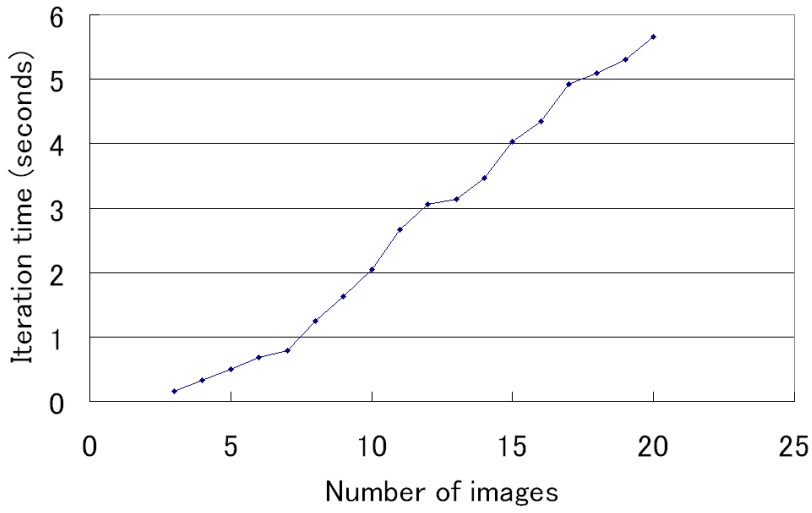


Figure 5.14: Computational time. Although the matrix sizes grow quadratically, the iteration time grows linearly.

Chapter 6

Conclusions and future work

6.1 Conclusions

This dissertation presented contributions to non-rigid image registration, particularly in the area of inter-color alignment and flexible mosaicing.

Regarding inter-color alignment, this dissertation presented a method for registration of color channels which uses a non-rigid deformation model. We presented a new energy function. This function was based on the entropy calculated over a subspace of the joint-distribution of the color channels being registered. Also, we presented a method to combine this similarity function and color gradient information. Our results validated that the inter-color alignment, according to the proposed energy function, yielded robust and precise results, comparable with the best related method analyzed in our work, with significantly reduced computational cost. The proposed method is also applicable to other multimodal registration problems.

Regarding mosaicing, the contributions presented were threefold: a method for real-time non-rigid mosaicing creation, a stitching algorithm optimized for non-rigid triangular meshes, and a fast non-rigid registration algorithm for global image alignment. The presented real-time non-rigid mosaic creation algorithm was designed to deal with the problem of lack of precision of the traditional methods when dealing with the mosaicing of non-planar surfaces. It also could handle the error accumulation that may occur when images are added sequentially into the mosaic. For this purpose, the so-called *Reference Mesh Energy* was presented.

An efficient method of key-frame selection, created to achieve real-time performance was also presented. To complement this method, a customized graph-cut algorithm for mosaic stitching was proposed. This algorithm was optimized for triangular meshes and registration by feature matches. It was able to generate mosaics with reduced stitching errors efficiently. The proposed video mosaicing method presents a serious restriction: Since it uses no global alignment, the generated mosaic is prone to error if a region of the scene is recorded more than once (loop). In order to alleviate this problem, a fast globally consistent non-rigid registration algorithm was designed. This new global registration technique was an extension of our video mosaicing method. It was designed as a post-processing method to improve the mosaics created by other methods. The results were precise while preserving computational efficiency.

6.2 Future Work

As possible extensions of the work presented in this dissertation, the presented inter-color alignment objective function needs a more efficient optimization method. Such algorithm may be the target of future research.

Regarding real-time image mosaicing and also the proposed global registration method, large discontinuities in the registration warping fields are heavily penalized by the regularization terms. Allowing discontinuities would generate much better mosaics in the case of non-planar surfaces with sharp discontinuous regions. This would be an important research direction.

The real-time mosaicing and the global registration methods both use $2D$ meshes. The extension to $3D$ meshes seems like a researching path with great possibilities, with applications in $3D$ video recording and improvement of $3D$ models generated by structure from motion systems.

References

T. Abatzoglou and B. O'Donnell. Minimization by coordinate descent. *Journal of Optimization Theory and Applications*, 36:163–174, 1982. 29

S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV'09*, pages 72–79, 2009. 58

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin / Heidelberg, 2006. URL http://dx.doi.org/10.1007/11744023_32. 6, 47

R. P. Brent. *Algorithms for minimization without derivatives*. Courier Dover Publications, 1973. 29

M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74:59–73, 2007. ISSN 0920-5691. 43, 58

A. Can, C. V. Stewart, B. Roysam, and H. L. Tanenbaum. A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: application to mosaicing the curved human retina. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 585–591, 2000. doi: 10.1109/CVPR.2000.854924. 43

K. Chaiyasarn, T. Kim, F. Viola, R. Cipolla, and K. Soga. Image mosaicing via quadric surface estimation with priors for tunnel inspection. In *Image*

Processing (ICIP), 2009 16th IEEE International Conference on, pages 537–540, 2009. doi: 10.1109/ICIP.2009.5413902. 43

H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003. ISSN 1077-3142. doi: DOI:10.1016/S1077-3142(03)00009-2. 44

A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. *Proc. of International Conference on Information Processing in Medical Imaging*, pages 263–274, 1995a. 16

A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal. 3d multi-modality medical image registration using feature space clustering. *Proc. of International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, 905: 195–204, 1995b. 16

D. Crispell, J. Mundy, and G. Taubin. Parallax-free registration of aerial video. In *Proc. British Machine Vision Conf.*, 2008. 43

R. H. C. de Souza, M. Okutomi, and A. Torii. Real-time image mosaicing using non-rigid registration. In *PSIVT*, volume 7087, pages 311–322, 2012. 58

Y. Deng and T. Zhang. Generating panorama photos. In *Proc. of SPIE Internet Multimedia Management Systems IV*, 2003. 43

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/358669.358692. 10, 48

J. H. Friedman. On bias, variance, 0/1loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997. 6

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 57

D. L. G. Hill, D. J. Hawkes, N. A. Harrison, and C. F. Ruff. A strategy for automated multimodality image registration incorporating anatomical knowledge and imager characteristics. *Proc. of International Conference on Information Processing in Medical Imaging*, 687:182–196, 1993. 16

D. L. G. Hill, C. Studholme, and D. J. Hawkes. Voxel similarity measures for automated image registration. *Visualization in Biomedical Computing, Proc. of SPIE*, 2359:205–216, 1994. 16

Steve Hsu, Harpreet S. Sawhney, and Rakesh Kumar. Automated mosaics via topology inference. *IEEE Computer Graphics and Applications*, 22(2):44–54, 2002. doi: 10.1109/38.988746. 43

V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3): 277–286, 2003. doi: DOI:10.1145/882262.882264. 53

J. Kybic, P. Thevenaz, A. Nirkko, and M. Unser. Unwarping of unidirectionally distorted epi images. *IEEE Trans. on Medical Imaging*, 19:80–93, 2000. 23

B. Likar and F. Pernus. A hierarchical approach to elastic registration based on mutual information. *Image and Vision Computing*, 19:33–44, 2001. 17

W.-Y. Lin, S. Liu, Y. Matsushita, T. Tsong Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *CVPR*, 2011. 8, 43

F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Trans. on Medical Imaging*, 16, 1997. 16

J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2:1–36, 1998. 17

S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22 (10):1144–1154, October 2000. ISSN 0162-8828. doi: 10.1109/34.879794. 43

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, pages 1 –8, june 2007. doi: 10.1109/CVPR.2007.383172. 57

J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *in Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 822–828, 2005. 44, 48, 51, 58

J. P. W. Pluim, J. B. Antoine Maintz, and M. A. Viergever. Image registration by maximization of combined mutual information and gradient information. *Lecture Notes In Computer Science*, 1935:452–461, 2000. 17, 20, 28, 33

J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Trans. on Medical Imaging*, 22:986–1004, 2003. 17, 19

H. S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Computer VisionECCV98*, pages 103–119. Springer, 1998. 43

M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. *Computer Vision and Pattern Recognition, IEEE Conference on*, pages DVD–Rom, 2008. 16

J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In *CLOR*, pages 127–144, 2006. 57

C. Studholme, D. L. G. Hill, and D. J. Hawkes. Multiresolution voxel similarity measures for mr-pet registration. *Proc. of International Conference on Information Processing in Medical Imaging*, pages 287–298, 1995. 16

C. Studholme, D. L. G. Hill, and D. J. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32:71–86, 1999. 19

R. Szeliski. Image alignment and stitching: A tutorial. *Found. Trends. Comput. Graph. Vis.*, 2:1–104, January 2006. ISSN 1572-2740. doi: 10.1561/0600000009. 5, 10, 44

T. Vercauteren, A. Perchant, X. Pennec, and N. Ayache. Mosaicing of confocal microscopic in vivo soft tissue video sequences. In *MICCAI 2005*, volume 3749, pages 753–760, 2005. 43

R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid automated algorithm for aligning and reslicing pet images. *Journal of Computer Assisted Tomography*, 16:620–633, 1992. 16

R. P. Woods, J. C. Mazziotta, and S. R. Cherry. Mri-pet registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17:536–546, 1993. 16

Y. Y.Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105 –112 vol.1, 2001. doi: 10.1109/ICCV.2001.937505. 55

J. Zhu, M. R. Lyu, and T. S. Huang. A fast 2d shape recovery approach by fusing features and appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1210–1224, 2009. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.151. 44, 48, 50, 51, 52, 58, 65