Friedrich-Alexander-University Erlangen-Nuremberg

Chair of Multimedia Communications and Signal Processing

Prof. Dr.-Ing. André Kaup

Feature Matching Based High Order Motion Compensation in Medical Image Encoding

by Wu Xiuzhe

July 2017

Supervisors:

Nils Genser, M.Sc.
Karina Jaskolka, M.Sc.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ort, Datum

Unterschrift

Contents

Abstract

The huge data size of original medical image limits its practical application. Similar to video coding, motion compensation can also be applied to decrease the data size in medical image coding, especially in scanned medical image coding. But the shortage of traditional video coding is so obvious that it just concerns translation transformation. In this paper, we propose to employee high order transformation based motion compensation to improve medical image coding performance and then use feature detection and feature matching methods to estimate the optimal transformation matrix. Several feature detection algorithms are tested to find the optimal one. Furthermore, RANSAC is applied to eliminate the noise.

Chapter 1

Introduction

Original medical image data size is exceedingly huge, which results in delays in remote diagnosis system and other real-time applications. Therefore, medical image compression without significantly information loss has become more and more popular. Medical image data includes various kinds of images which are generated by different kinds of medical technology, like Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography, Ultrasound and so on. Take CT image for example. CT scan will scan organs in one direction to examine whether they work normally. It will generate a series of scanned images. The scanned image set is just like the video sequence, because the video also consists of a series of images. The only difference is that the images of video are redundant in temporal domain while the scanned images are redundant in spatial domain. Therefore, the encoding of scanned medical images can refer to video coding process. In conventional video coding, motion compensation is one of the most important parts. Hence in this report, we aim to improve motion compensation in order to boost the efficiency of medical image coding. The rest of this paper is organized as follows. In chapter 2, we will explain what is Motion Compensation and how to enhance Motion Compensation. Furthermore, new technique will be introduced to improve the efficiency. Then the comparative experimental results are presented in chapter 3. Finally, chapter 4 concludes the report.

Chapter 2

Enhanced Motion Compensation

2.1 Motion Description

Motion Vector (MV) and Motion Estimation (ME) are two fundamental techniques in prediction based video and image coding. MV measures the changes between two images in details in order to eliminate the significant redundancy in temporal domain and the target of ME is to find the optimal MV in each unit. Motion Compensation (MC) describes how to predict current frame data based on previously encoded frame. Then encoding the prediction error instead of the whole frame will considerably decrease the data size. Conventional MC just simply concentrates on translational motion while in practical application there are several kinds of transformation, such as rotation, scaling and zooming. Hence it can not be able to describe complex motions efficiently.

In order to solve this problem, affine transformation is proposed to take the place of translation transformation. Affine transformation includes translation, scaling, rotation and so on. Also, the composition of any of them is also a kind of affine transformation. All the transformations can be described as a transformation matrix. For example, the rotation transformation matrix can be defined as

$$\begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

while θ represents the rotation degree. Also, affine transformation matrix can be defined as

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

Therefore, the key problem turns into how to estimate the transformation matrix properly.

2.2 Feature Matching

Feature describes a concrete object in image, like edge, corner, blobs and so on. Feature detection checks every point to determine whether there exists a given type of feature. There are several types of feature description algorithm, like Maximally Stable Extremal Regions (MSER), Binary Robust Invariant Scalable Keypoints (BRISK), Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) features. Also, Features from Accelerated Segment Test (FAST) algorithm, Harris-Stephens algorithm (Harris) and minimum eigenvalue algorithm (MinEigen) are three popular corner detection methods.

Take SIFT and SURF for example. The key problem of SIFT algorithm is to find the keypoints over all scales and locations and calculate their scale, orientation, and location. Then this information will be used to describe those keypoints. The keypoints picked out by SIFT algorithm are distinctive and invariant to image scale and rotation, so it performs well in image matching. Specifically, the first step is to detect extrema and then eliminate those keypoints with low contrast or are poorly localized along an edge. Next, assign the orientation to the rest keypoints and then generate the keypoint descriptors based on the region around the keypoints. Besides, there is another feature detection algorithm called SURF. Inspired by SIFT, SURF is proposed to detect local features. But different from SIFT, SURF makes use of Haar wavelet responses and integer images, so it significantly improves the detection speed.

In image matching, after features of two images are generated, the next step is to compare every feature of one image to all the features of the other image to find the optimal matching based on Euclidean distance. With all the keypoints matched between two images, the transformation matrix can be easily estimated.

2.3 Feature Matching with RANSAC

Although current feature matching methods demonstrate good performance in robust, there are still some possible errors which will affect the final result. There are several estimating methods applied to eliminate the noise data and optimize model parameters, among which the most famous one is Least Square method. However, Least Square method aims at minimizing residuals of all the points including inliers and outliers, which will lead to a bad performance when data set is with a lot of noise. Different from Least Square method, another algorithm called Random Sample Consensus (RANSAC) will make a distinction between inliers and outliers and then just concentrates on inliers. So in our experiment, RANSAC will be used to remove outliers to ensure the efficiency.

Chapter 3

Experimental Results

3.1 Comparison with other algorithms

In the experiments, we test six algorithms to see which one performs better, including BRISK, FAST, Harris, MinEigen, MSER and SURF. However, the first two algorithms are unstable. In several tests, they can not extract enough points. So in our experiments, they are abandoned in the first stage. Then five kinds of transformation are applied to the rest algorithms. They are rotation, translation in x-direction, translation in y-direction, scaling in x-direction, scaling in y-direction. The configuration is shown in the following table 3.1.

Table 3.1: Configuration Table

Transform type	Range	Step
Rotation	[-10, 10]	0.5
Translation in x-direction	[-50, 50]	1
Translation in y-direction	[-50, 50]	1
Scaling in x-direction	[0.9, 1.1]	0.1
Scaling in y-direction	[0.9, 1.1]	0.1

Four criteria are utilized to study the performances of algorithms, including error in x-direction, error in y-direction, absolute rotation direction and encoding time. Take rotation transformation for example. As shown above, we will test rotation transformation with degree ranging from -10 to 10. For each rotation degree, we will generate one transformed image and then estimate the matrix and calculate the errors, which means we will generate 41 images, estimate 41 matrices and calculate errors for 41 times at a time. Therefore, we use Mean Absolute Error (MAE) in errors calculation to make the comparisons easier and more direct (see table 3.2, 3.3 and 3.4).

Table 3.2: Errors in x-direction

	SURF	Harris	MinEigen	MSER
Rotation	22.6228	22.8441	22.8385	21.1132
Translation in x-direction	0.1195	0.0000	0.0000	0.0000
Translation in y-direction	0.0320	0.0000	0.0000	0.0000
Scaling in x-direction	0.1443	0.0467	0.0276	21.7863
Scaling in y-direction	0.0673	0.0588	0.0238	2.6401
Average	**4.5972**	**4.5899**	**4.5780**	**9.1079**

Table 3.3: Errors in y-direction

	SURF	Harris	MinEigen	MSER
Rotation	22.6447	22.8526	22.8401	22.2960
Translation in x-direction	0.1194	0.0000	0.0000	0.0000
Translation in y-direction	0.0320	0.0000	0.0000	0.0000
Scaling in x-direction	0.1442	0.0467	0.0276	11.4946
Scaling in y-direction	0.0673	0.0588	0.0238	2.5469
Average	**4.6015**	**4.5916**	**4.5783**	**7.2675**

From experimental results, we can see all the algorithms perform equally well except MSER in image quality. But in encoding time, SURF algorithm does much better

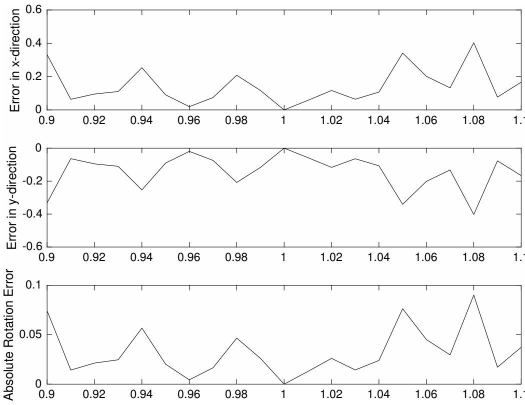
Table 3.4: Absolute Rotation Error

	SURF	Harris	MinEigen	MSER
Rotation	5.0790	5.1274	5.1254	4.8707
Translation in x-direction	0.0267	0.0000	0.0000	0.0000
Translation in y-direction	0.0072	0.0000	0.0000	0.0000
Scaling in x-direction	0.0323	0.0104	0.0062	4.0020
Scaling in y-direction	0.0151	0.0132	0.0053	0.5807
Average	**1.0321**	**1.0302**	**1.0274**	**1.8907**

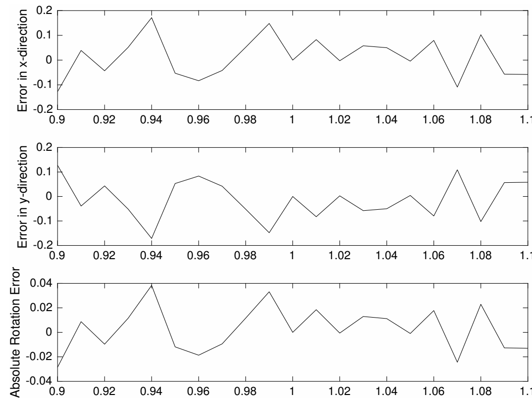
Table 3.5: Encoding Time

	SURF	Harris	MinEigen	MSER
Rotation	4.1332	7.5518	7.9936	6.1552
Translation in x-direction	9.7934	18.1545	19.0536	14.4355
Translation in y-direction	9.7886	18.3776	18.6649	14.4880
Scaling in x-direction	2.2310	3.7534	3.9251	3.1005
Scaling in y-direction	2.0900	3.8913	3.9543	3.1913
Average	**5.6072**	**10.3457**	**10.7183**	**8.2741**

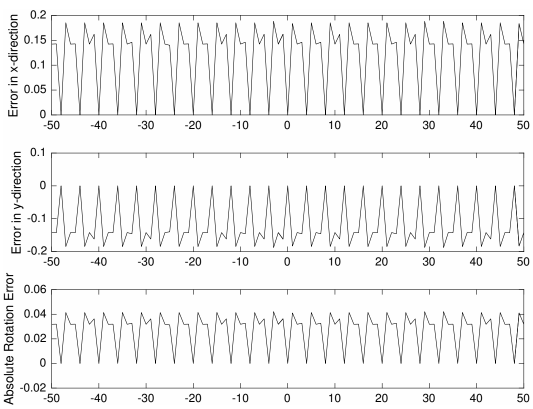
than all the other algorithms. Hence, we decide to use SURF for further tests. The following five figures are details about SURF in five kinds of transformation.



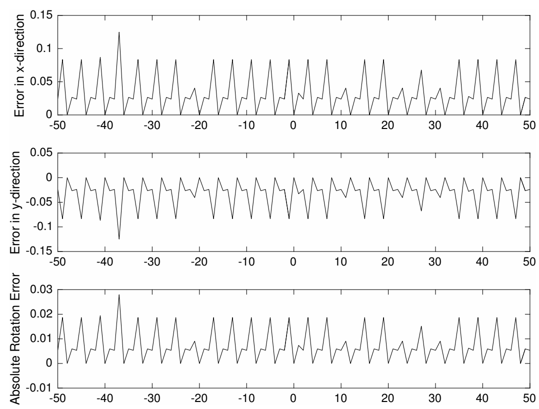
(a) Scaling in x-direction



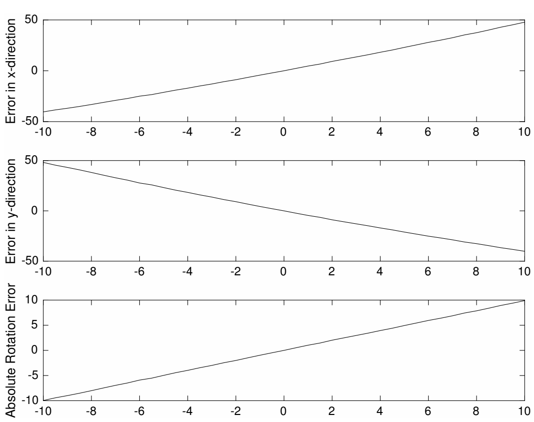
(b) Scaling in y-direction



(c) Translation in x-direction



(d) Translation in y-direction



(e) Rotation

Figure 3.1: Figures

Chapter 4

Conclusion

In this report, we propose to use high-order motion compensation to take the place of conventional translation-only motion compensation in medical image coding. To accomplish this target, we employee feature detection and feature matching techniques to estimate the transformation matrix. To guarantee the efficiency, we also apply RANSAC to eliminate the effect of outliers. Finally, we test six famous feature detection algorithms and find the optimal algorithm, which is called SURF. In further research, we can use it in HEVC or some else areas about video and image coding.