Abstract

The aim of this project is to create a new single view of text for OCR, which is extracted from video frame sequences. Two challenges are faced in this project. One is about how to rectify perspective distortion of text scene. The other one is about how to improve image quality. For the first problem, we employ the Hough transformation based text boundaries detection method to help rectify perspective distortion. For the second problem, three super resolution algorithms are implemented including two basic super resolution algorithms and Irani and Peleg's iterated super resolution algorithm. All of them are able to improve the image quality. Experimental result is given in the final part, which analyse these three super resolution algorithm's OCR accuracy and computation performance. Finally, we reach a conclusion that choosing an accuracy and performance balanced solution is recommended.

Content

Chapter 1

Introduction

1.1 Introduction and context

The pervasive using of cheap hand-held cameras has led to text tracking and text recognition on digital images receiving a great deal of attention. Traditionally, text paper and documents are scanned from flatbed scanners into digital images, which is often sent to Optical Characters Recognition (OCR). The convenience of using hand-held cameras, Personal Digital Assistant (PDA) and mobile phones has brought ubiquitous digital text images everywhere. Therefore, text digital images not only come from text paper or documents, but now also from real world situations, for instance, vehicle license plates and road sign board text. However, those kind of digital images often have a different orientation, prospectively distortion effects are introduced. The reason for this it is that images are captured from a remote plan rather than the text plan. This is a big challenge for OCR to achieve high accuracy. Pre-processing these kind of images to OCR is necessary.

1.2 Aims and objectives

The primary aim of this project is to create a new single view of text for OCR, which is extracted from video frame sequences. OCR is sensitive to noises and perspective effects of images. For instance, a blurred text view with noises and a text view which is not a frontal view might have perspective effects. Thus, this project faces two challenges, one is to reduce noise and the other is to remove perspective distortion. These two challenges are two sub aims of this project.

Additionally, it was considered important to obtain a high accuracy and a good performance at the same time.

The objectives to meet this aim can be broken down as follows:

- Decoding the videos to get every single frame.
- Detecting highlighted pink areas containing text.
- Removing perspective distortion.
- Feature detection for computing homography matrix.
- Warping one frame to another by homography matrix.
- Aligning two adjacent frames by image registration methods.
- Implementing three super resolution algorithms, creating a high resolution frame.
- Achieving all the objectives by C++ programming.

More specifically:

Multi-view registration is a technology which merges multi frames to a single frame. There are two kinds of image registration: rigid registration and non-rigid registration. Rigid registration assumes the source and target object are in the same pose. Non-Rigid registration assumes images form from different time instances. This project is based on rigid image registration.

The perspective distortion effect is one of the main reasons that affect the OCR's ability to recognize characters. The projection profile based method and the Hough transformation based method are two popular methods to remove this distortion. The former one is often applied on text view, and the latter one is often applied on the building. This project employs the Hough transformation based method combining extra information such as document boundaries, background boundaries and page layout to rectify perspective distortion. This method is simple and effectively creates a frontal view needed for OCR without knowing the camera's parameters.

Feature detection tools are evaluated to detect corresponding feature points to estimate the Homograpy matrix. The Homograpy matrix is a perspective transformation matrix between source and destination plane. With this matrix, it becomes possible to warp one frame to the other for the purpose of aligning two frames and fuse the information. Residual error is computed for the homography transformation, according this residual error, some noise frames are ignored. Levenberg-Marquardt (LM) refinement measurement is also applied to increase accuracy of estimation homography matrix.

In order to improve quality of our result, this project will create a high resolution image from observed low resolution images. Usually high resolution images are clearer than low resolution images. Three super resolution algorithms were implemented. Two of them were very basic while the other one came from Irani and Peleg's. The simple super resolution methods were able to remove noise especially for a single point noise. Irani and Peleg's algorithm is a complicated method which is iteration with a kernel matrix to converge the true value. It begins with the initial estimate of average value of all low resolution images. Then, a kernel matrix is constructed for mapping the

relationship of each low resolution pixel to each high resolution pixel. Finally, Irani and Peleg's updated equation was employed to estimate a high resolution image.

1.3 Organization

In chapter 2, the background and literature research is summarized including image perspective distortion rectification, super resolution, feature detection and computation of homography matrix. Chapter 3 describes the designation processing. The basic models of this project are described. Development methodology is also described in brief. Chapter 4 elaborates details of each development stage. The main computation flowchart is given at the beginning, which is a clue to follow the whole computation process. Then, each step is explained according to the main flowchart in detail. Chapter 5 shows two experimental results. One is the evaluation result of SIFT and SURF tools, the other is the OCR accuracy results of the three super resolution algorithms, which is also our final result. Chapter 6 gives a brief conclusion and discusses some future work.

Chapter 2

Background and Literature Research

This chapter consists four parts supporting to this project. The first part is image perspective distortion rectification which aims to remove perspective distortion. This kind of distortion is the main reason for achieving low accuracy of OCR of camera-based text scene. The second part is known as super resolution which is a measure of increasing quality of image interpolated from Multi-frames. Several super resolution methods are examined in this section and all of them are based on machine learning algorithms. The third part is called feature detection, which is also a critical element of this project. It provides corresponding points to build relationship between two different adjacent frames. The research of feature detection is contributed to make decision of choosing feature detection algorithms. The fourth part describes how to compute homography matrix methods.

2.1 Image perspective distortion rectification

Camera-based images are suffered from perspective distortion which severe affects the accuracy of OCR. Many efforts have been made for removing the perspective distortion and rectify the distorted image into a frontal-parallel view. Gaussian sphere based, Hough transform based method and projected profile based method are three popular methods to rectify perspective distortion. The aim of those three algorithms is to find vanishing points or infinity line to rectify perspective distortion, which are perspective transformation variant features.

The understanding of 2D transformation and non-homogenous coordinate is necessary for the later explaining. 2D transformations of the plane can be represented by 2 x 2 matrices in non-homogeneous and 3 x 3 in homogeneous. The task of rectification perspective distortion is to compute such a matrix 2 x 2 in non-homogeneous or 3 x 2 in homogeneous reflecting a certain 2D transformation between observed text scene and frontal-view text scene. 2D transformation includes similarity transformation, rotation transformation, affine transformation and perspective transformation. Usually, A point

in non-homogeneous is represented $x=(x,y)^T$, the same point in homogeneous is represented by a 3 column vector, which made by adding an extra dimension with value 1 in the end of non-homogeneous representation. For example $x=(x,y,1)^T$, is a homogeneous representation of x. The reason for introducing a homogeneous is that most of kind transformations can be represented by linear transformation on the homogeneous coordinates as well as in non-homogeneous correspondingly but not vice versa. The convenience of homogeneous coordinates shows the benefit of computing a 3 x 3 perspective matrix is our purpose. E.g. perspective 2D transformation can be represented in homogeneous coordinates by linear transformation but can't be represented non homogeneous coordinates by linear transformation.

2.1.1 Camera modelling and invariant geometry features

Camera modelling is also critical to describe perspective transformation in mathematically ways. Perspective transformation is a mathematical projection between a 3D object space and a 2D image. This section explains the how to select a camera model and describes some invariant geometry features under perspective transformation, because rectify process is based on these invariant or variant features.

The most popular camera model is pinhole camera model, which can be described by the formula $p = MP$, where p is the point in the image, M is the camera matrix, and P is the point in the real world. Point p and P are represented in homogeneous coordinates and defined as column vectors $p:(x,y,1)^T$ and $P:(x,y,z,1)^T$ respectively. Camera matrix is a 3 x 4 matrix, which is a perspective transformation matrix mapping real world 3D point P to image 2D point. According to this kind of transformation some geometry features exist. For instance, parallel line in the world will intersect at a point in the infinity in the perspective space. Sets of parallel line will intersect at different point, and all of them lie on a same line l_∞, which is defined as $l_\infty = (l_1, l_2, l_3)^T$. One method of removing a perspective distortion is to map this line $l_\infty = (l_1, l_2, l_3)^T$ to the line $l_\infty = (0,0,1)^T$. Vanish points are very important to compute this infinity line. At least, two vanish points are needed to compute infinity line. Usually, parallel lines of two main directions are used to compute, which are vertical and horizontal directions. By using those invariant geometry or variant features, we can estimate images in different angles with same scene further we are able to rectify the perspective distortions.

Cross Ratio is another invariant for perspective transformation. Linlin Li and Chew Lim Tan [21] employ this invariant to establish a database for all letters in all perspective angles and each input image is searched in a global similarity matrix, in order to obtain most likely perspective angles.

2.1.2 Vanishing points

This section describes how to detect vanishing points and how to use vanishing points to rectify perspective distortion.

What is vanish points? A set of parallel lines in real world will converge to a single point on the image plane, which is regarded as vanishing points. This point is formed by the effects of perspective transformation described above. Finding this point is important to reconstruction of 3D model and rectification of perspective distortion. The following part describes Gaussian sphere based, projection project based and the Hough transformation based vanishing points detection methods in details.

- Gaussian sphere based vanishing points detection

A mapping of surface of an image onto the unit sphere which forms Gaussian sphere. Tail lies at the center of the Gaussian sphere. Head lies on the surface of the Gaussian sphere. Figure 2-1 shows the model of Gaussian sphere, two parallel lines in image plane map to Gaussian sphere which always intersect in one point referred to vanishing point. Virginio [28] employ this model to estimate vanishing points. The benefit of Gaussian sphere is it not only represents finite vanishing points but also represents infinite vanishing points. After mapping image line segment to Gaussian sphere, Virginio [28] provides two methods to compute vanishing points, one is a statistical approach which employ least square method the other is Hough transformation based method.

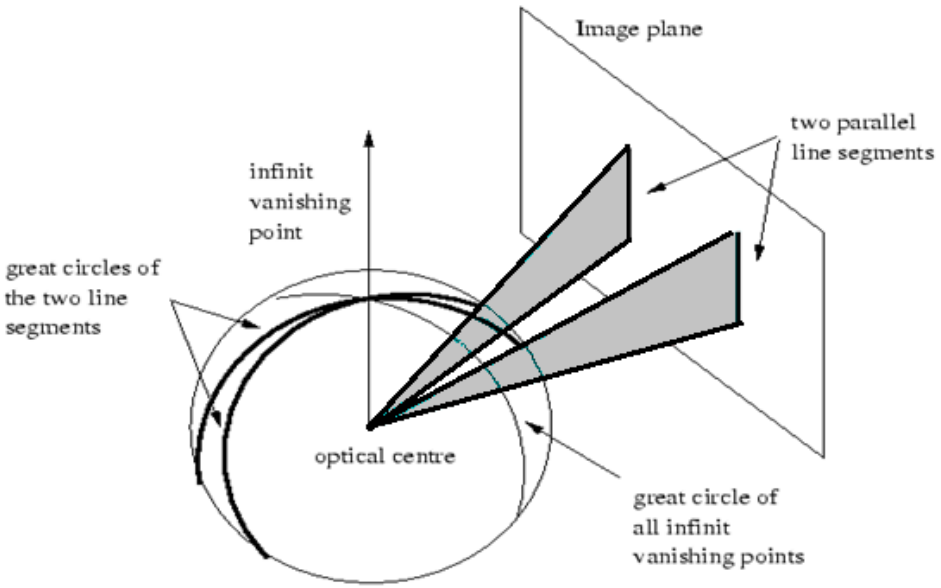

Figure 2-1 The Gaussian sphere as an accumulator space
http://www.nada.kth.se/~carstenr/CVonline/node1.html

- Projection profile based vanishing points detection

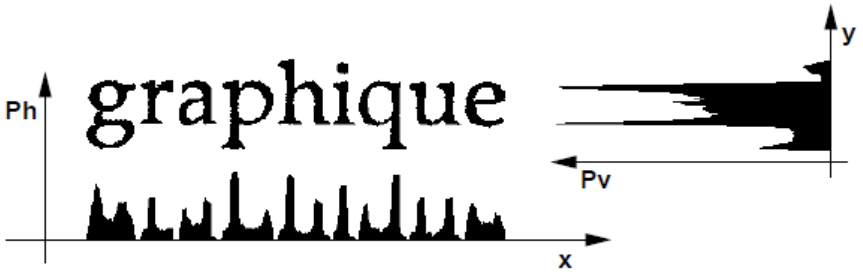

Figure 2-2 vertical and horizontal projection profiles [1]

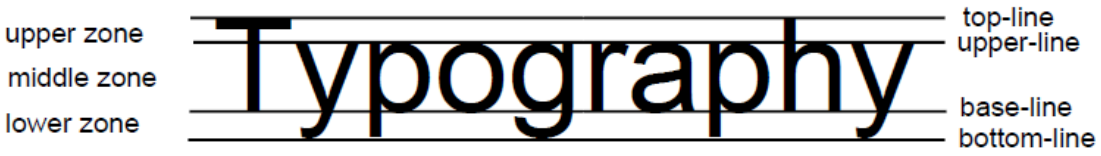

Figure 2-3 structure of text line [1]

Projection profile is an intensity statistics feature of text view. Figure 2-2 is an example of a text view of projection profile. Vertical projection profile and horizontal projection profiles is plot as histogram. Figure 2-3 shows a feature for Latin words, each word consists four lines: top-line, upper-line, base-line, and bottom-line. The vertical projection profile reflects these four lines clearly. Maximum intensity value of vertical projection indicates one line of these four feature lines. According to the figure 2-2, the intensify value of upper line and base line is significant for identifying them out. Horizontal projection profile also reflects a feature of a Latin word. The intensity intervals reflect the intervals of letters. Projection profile is a stable statistic feature of text view, but is has the limitation that it is only significant in Latin word, which is not significant in all languages.

If text scene is distorted by different angles of view, the projection profile is different. By computing the minimum entropy of projection profile from different angles, horizontal vanishing points can be found, because the minimum of entropy indicates the main direction of four feature lines. This direction is also the direction of vanishing point, because parallel lines in 3D world always intersect in one point referred to vanishing point mentioned above. Computing projection profile starts from convert images to binary image which only contains 0 indicates background 1 indicates foreground. By computing binary image of images redundant information is ignored that leads high accuracy.

Clark[24] estimates horizontal vanishing points based on an extension of projection profile and estimate vertical vanishing points base on some paragraph formatting (PF) information. Text line spacing variation when paragraphs with perspective distortion.

The limitation of this method is that this method only to apply on the paragraph text images. However, for single or double lines text scene this method is unable to work successfully. All data source of this project are from real world scene, most of them only contain one or two lines of words. So, Clark's method is only able to compute one vanishing point for this project, but it is not enough to rectify perspective distortion.

- The Hough transformation based vanish points detection

The Hough transformation was invented by Richard Duda and Peter Hart in 1972, which transform points from orthogonal coordinates to polar coordinates by the Formula 2-1.Traditionally, Hough transformation was used to identify lines in the image, later the Hough transformation has been extended to identify arbitrary shapes.

$$r = x\cos\theta + y\sin\theta$$

Formula 2-1

Google Int. [27] employ cascaded Hough transformation helps to detect straight lines, vanishing points and vanishing lines. Three layers Hough transformation supports the detection. The first layer Hough transformation helps to detect straight lines, which is indicated by a point. The second Hough transformation is applied to the first layer and only peak values remains. In the result of second layer Hough transformation points indicate intersection points, and lines indicate collinear intersection points. Then the Hough transformation is applied to the second layer, again only peak values remains in the result of the third layer. Points in the result of the third the Hough transformation indicate lines of intersection points. The third Hough detects the collinear line intersections, it sufficient to find vanishing points and vanishing lines.

The Hough transformation based vanish points finding are widely used in real world scene, especially, in architectural environments, because man made architectural environments have relatively regular shapes. So, vertical and horizontal dominant directions can be detected by the Hough transformation easily.

- Summary

Projection profile is a stable statics method to estimate vanishing points, but it is unable to estimate horizontal vanishing points. It needs extra information from paragraph to estimate horizontal vanishing points. So it is not suitable for this project. The Hough transformation based vanishing points is usually employed in real world scene. Our approach is based on Hough transformation, because the scene of this project is from real text scene rather than document or text pages. The computation of Gaussian sphere is complicated, so we abandon this method and this method is seldom used to detect vanishing point of text scene.

2.2 Super resolution algorithm

Super-resolution (SR) is a technique that enhances the resolution of an image. Single-frame super resolution creates high-resolution image from a single frame which often need training set to train a model, and then apply this model to the single frame obtaining a high-resolution image. Multiple-frame supper resolution often use machine learning algorithm to merge several frames to create and estimate a high-resolution image. This project is based on multiple-frame super resolution. The creation of a high-resolution image is a process of fusing information from several number of low resolution images called observers.

Image super resolution is a well-studied topic. This chapter examines two multi-frame super resolution methods both of them are based on machine learning algorithm. One is an iterated method to converge estimated high resolution values. The other one is maximum likelihood algorithm which is a more sophisticated than iterated method.

2.2.1 Irani and Peleg's super resolution

The main idea of estimating a high-resolution image of Irani and Peleg's method is "compare simulated and observed low-resolution images" [22]. Conventionally, the high resolution image is an estimated image and low-resolution images are observes. Figure 2-7 is an illustration of Irani and peleg's super resolution. This algorithm is an iterated method to converge true high resolution values. Formula 2 is the updating function. $f^{(n+1)}\left(\underset{x}{\rightarrow}\right)$ is the approximation of f obtained after n iteration, which is from initial guess $f^{(0)}\left(\underset{x}{\rightarrow}\right)$ referred to average of all observers indicated in irani and peleg's paper. Updated value is the difference of simulated and observed low resolution images. Formula 1 is the function describing how to simulate the n^{th} low resolution image from the n^{th} high resolution image, where $g^{(n)}\left(\underset{y}{\rightarrow}\right)$ is the low resolution image obtained by applying the simulated imaging process to $f^{(n)}$. Basically, a pixel in low resolution image is estimated by a sum weighted function of a group of high resolution pixel. The group of high resolution pixels are the neighbours of centre of the receptive field of low resolution pixel in high-resolution image. The weighted function is referred to point spread function (PSF), which reflects the essential relationship between low and high resolution images.

What is PSF? "The point spread function (PSF) describes the response of an imaging system to a point source or point object. In many contexts can be thought of as the extended blob in an image that represents an unresolved object".

(http://en.wikipedia.org/wiki/Point_spread_function)

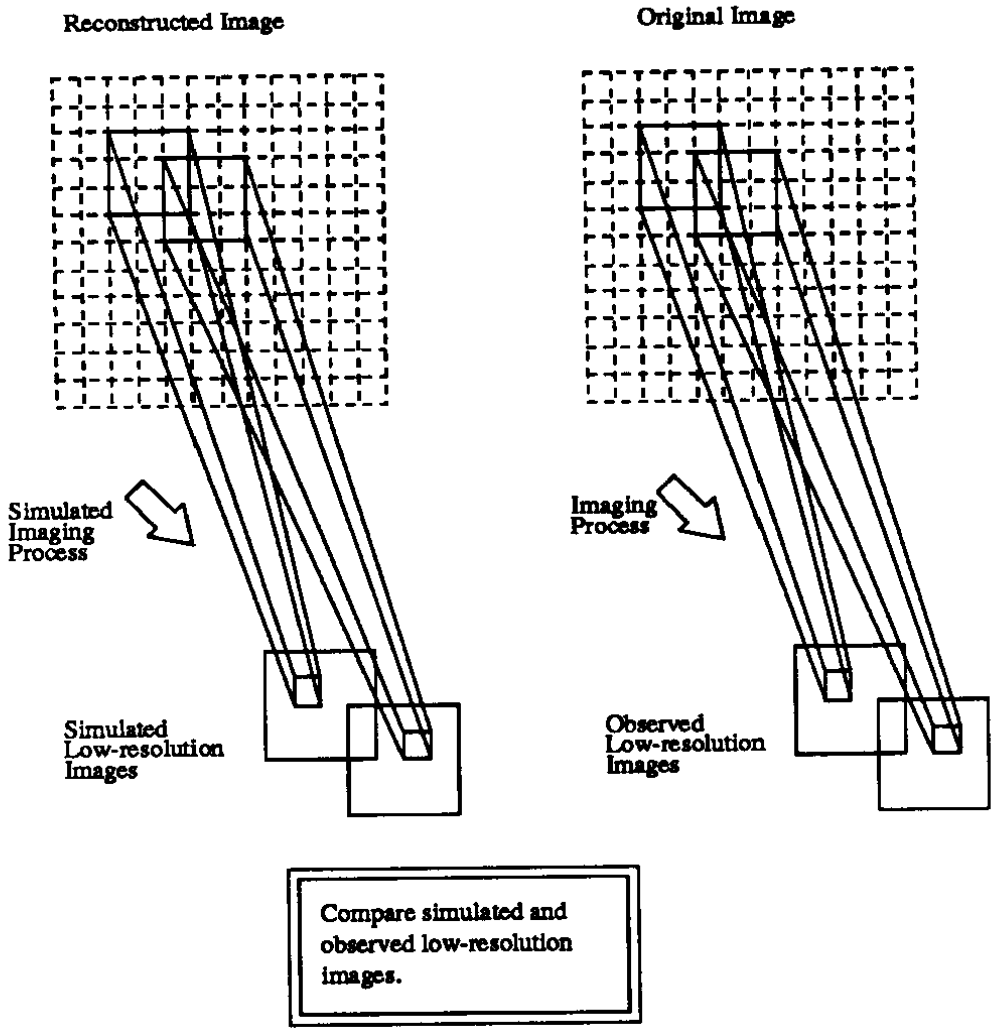

Figure 2-4 Schematic diagram of the super resolution algorithm. A reconstructed image is sought such that after simulating the imaging process, the simulated low-resolution images are closest to the observed low-resolution images. The simulation of the imaging process is expressed by formula 2-2 [22]

$$g^{(n)}\left(\underset{y}{\rightarrow}\right) = \sum_{\underset{x}{\rightarrow}} f^{(n)}\, h^{PSF}\left(\underset{x}{\rightarrow} - \underset{z_g}{\rightarrow}\right)$$

Formula 2-2 [22]

$$f^{(n+1)}\left(\underset{x}{\rightarrow}\right) = f^{(n)}\left(\underset{x}{\rightarrow}\right) + \sum_{\underset{y}{\rightarrow}\ \in \cup_k Y_{k,x}} \left(g_k\left(\underset{y}{\rightarrow}\right) - g_k^{(n)}\left(\underset{y}{\rightarrow}\right)\right) \frac{\left(\mathrm{h}^{BP}_{\underset{x}{\rightarrow}\underset{y}{\rightarrow}}\right)^2}{\mathrm{c}\sum_{\underset{y}{\rightarrow}\ \in \cup_k Y_{k,x}} \mathrm{h}^{BP}_{\underset{x}{\rightarrow}\underset{y}{\rightarrow}}}$$

Formula 2-3 [22]

Where,

- f – The target high-resolution image to be constructed.
- $f^{(n)}$ – The approximation of f obtained after n iteration
- g_k – The k_{th} observed low resolution image
- $g_k^{(n)}$ – the low resolution image obtained by applying the simulated imaging process to $f^{(n)}$
- h^{PSF} – The point spread function of the imaging blur
- h^{BP} – A back-projection kernel
- $\underset{x}{\rightarrow}$ denotes a high-resolution pixel
- $\underset{y}{\rightarrow}$ denotes a low resolution pixel
- $\underset{z_g}{\rightarrow}$ Denotes the centre of the receptive field of $\underset{y}{\rightarrow}$ in $f^{(n)}$.
- $Y_{k,x}$ denotes the set $\{\underset{y}{\rightarrow} \in g_k | \underset{y}{\rightarrow} \; is\; influenced\; by \; \underset{x}{\rightarrow}\}$
- c is a (constant) normalizing factor
- $\mathrm{h}_{\underset{x}{\rightarrow}\,\underset{y}{\rightarrow}}^{BP} = h^{BP}(\underset{x}{\rightarrow} - \underset{z_g}{\rightarrow})$

The essential part of this algorithm is points spread function which describes the relationship between low-resolution image and high-resolution image. For the consideration of computation a kernel matrix is constructed.

Rows indicates high-resolution pixels, columns indicates low resolution pixels. Each entry of this matrix is computed by 2D Gaussian function model referred to formula 2-5.

$$\widehat{W_{ij}} = \exp\left\{-\frac{\left|\left|v_j - u_i'\right|\right|}{2\sigma^2}\right\}$$

Formula 2-4

$$W_{ij} = \frac{\widehat{W_{ij}}}{\sum_{j'} W_{i\,j'}}$$

Formula 2-5

, where u_i' is the 2D location of the i^{th} pixel's centred when projected into the frame of the high-resolution image, and v_j is the 2D location of the j^{th} high-resolution pixel in its own frame. W_{ij} is the weight of PSF. With this kernel matrix, we are able to estimate high-resolution image from several low resolution images.

Another function in formula 2-3 is h^{BP}: back projection function (BPF). What is back projection function? The point spread function of back projection function is circularly symmetric. Indicated by Irani and Peleg's $(h^{BP})^2$ can be the same of h^{PSF}.After n^{th} iteration constructed image converges the value of original image.

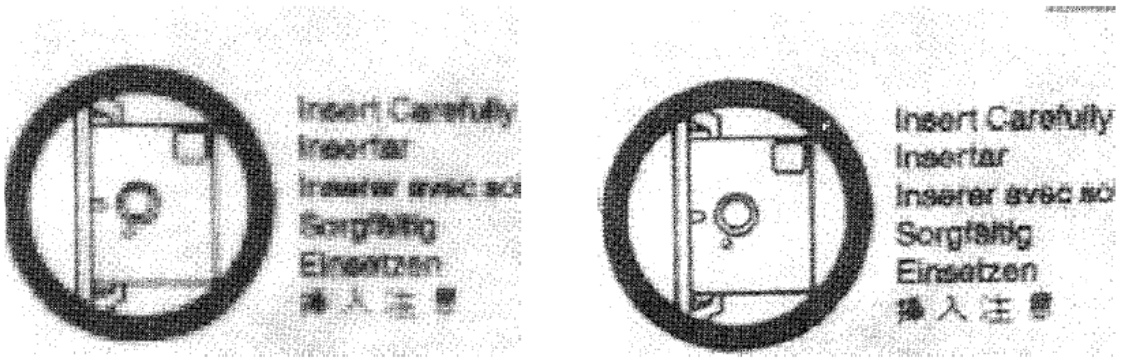

Figure 2-5 super resolution from Irani and Peleg's algorithm. Left: one of original image. Right: improved resolution image. [22]

2.2.2 Maximum likelihood super resolution algorithm

Maximum likelihood super resolution algorithm is maximizing the probability of having low resolution images. Suppose we have k low-resolution images, each low-resolution image contains M pixels and each high-resolution image contains N pixels. Formula 2-6 shows how to estimate high-resolution image.

$$x_{ML} = (W^T W)^{-1} W^T y$$

Formula 2-6

Where, x is high-resolution image, $W^{(k)}$ can be constructed by Formula 2-5.W is the KM $\times$N stack of all K of the $W^{(k)}$ matrices, and y is the KM $\times$1 stack of all the vectored low-resolution images. This matrix is large and sparse. It is hard to construct such huge matrix.

2.2.3 Summary

Irani and Peleg's super resolution algorithm is an iterated method which requires long time but less computer resources. ML super resolution algorithm is a more

sophisticated method. However, due to the limitation of computer resource we decide to implement Irani and Peleg's super resolution algorithm rather than Maximum likelihood super resolution algorithm.

2.3 Feature detection algorithms

Features points of frames are detected to compute a matrix which establishes 2D transformation relationship between two adjacent frames. Good feature detector obtains features are distinctive and invariant. Distinctive means different features' descriptor is different. Invariant means features are invariant under transformations such as rotation, affine transformation. Harris corner detector [3] is a widely used corner detector, which proposed in 1988. However, there is a limitation to Harris that is it can't detect scale-invariant corners. Later, Mikolajczyk and Schmid create a scale-invariant detector [16] in 2001, which is based on scale space. Several scale-invariant appeared in that time. In 2004, Lowe presents Scale Invariant Features Transform (SIFT) [6] algorithm. It is a set of method for corresponding points detecting and matching. In the same year Ke and Sukthankar apply PCA to normalize gradient patch instead of histograms in SIFT [14]. The PCA-SIFT reduced the feature dimension of SIFT, but the speed dose not improved. GLOH [14] is a SIFT like descriptor but the computation cost is higher than SIFT. SURF [11] (Speeded up Robust Features) was first presented by Herbert Bay in 2006. SURF's detector and descriptor is different with SIFT, which employs integer image and box filter for the hessian matrix. The merit of SURF is its speed is much higher than SIFT.

This chapter focus on three feature detection algorithm: Harris based corner detector, SIFT and SURF.

2.3.1 Harris based corner detector

Most of interest point detector methods are based on the smoothing of a local area of image. A simply smoothing method is to make use of the average of neighbour points replacing this point.

$$R_{ij} = \frac{1}{(2k+1^2)} \sum_{u=i-k}^{u=i+k} \sum_{v=j-k}^{v=j+k} F_{uv}$$

Formula 2-7

For instance, we can make use of equation Formula 2-7 to update point R (i, j). F_{uv} Is the input of image, R_{ij} is the output. In terms of this equation point R (i, j) is smoothed by its neighbourhood points (a range of area of points nearby this point). Formula 2-7

illustrates the shift-invariant property of this point. Harris based corner detector is also based on smoothing of a local area. All the interesting points detected by Harris detector have large gradients in all directions, in which large gradient means a predetermined threshold. The limitation of this algorithm is it is not sensitive for different size of images. Therefore this method is not a scale invariant interesting corner detector.

2.3.2 SIFT

The main aim of SIFT is to improve Harris corner detection algorithms' limitation that is not scale invariant. Sample scale space is employed to solve scale invariant problem. SIFT algorithm is divided into four steps.

1) Detect scale-space extrema

The first step of SIFT is to detect scale-space extrema. Images scales downsize by halve to build a scale different image pyramid. In each layer of pyramid the image is smoothed by different Gaussian function in order to compute the difference of Gaussian. Formula 2-8 describes the difference-of-Gaussian function. The purpose of this is to find gradients in each point.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

Formula 2-8

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

Formula 2-9

Where, $D(x,y,\sigma)$ is difference-of-Gaussian function.

In order to detect local extrema, SIFT detection compares centre points (black point in figure 2-6) to its nine neighbours in the scale above image, nine neighbours in the scale below image and eight neighbours in current image, which marked by in the figure 2-6 with a gray circle. Therefore, 3 * 3 * 3 -1 points are compared with target point. If this point larger than all its neighbours or less than all its neighbours it is selected as a candidate point.

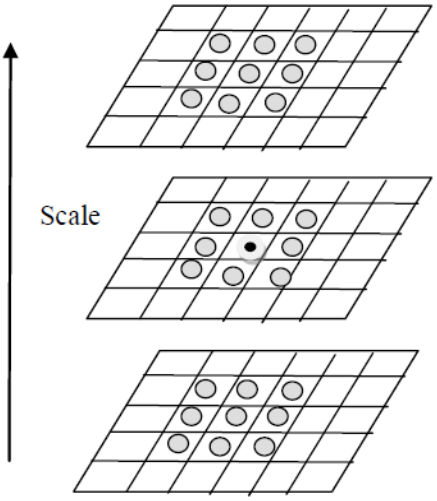

Figure 2-6

2) Accurate key points localization

Candidate points are not always stable and very sensitive to edge. SIFT removes some candidate point which around the edge.

3) Orientation assignment

Refined candidate points then are assigned orientation which ensures it invariant in image rotation. "By assigning a consistent orientation of each key point based on local image properties, the key point descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation." [6] Lows has indicated the reason why assign orientation to target points. Following formula is employed to compute orientation. $m\,(x,y)$ indicates the scale (weight) of origination, $\theta\,(x,y)$ indicates the angles of this origination. In fact, each point has eight neighbour points, thus eight orientations are assigned to one point.

$$m\,(x,y) = \sqrt{L\,(x+1,y) - L(x-1,y)^2 + L\,(x,y+1) - L(x,y-1)^2}$$

Formula 2-10

$$\theta\,(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L\,(x,y-1)}{L\,(x+1,y) - L(x-1,y)}\right)$$

Formula 2-11

4) Local image descriptor

Location, scale and orientation for each point have computed by previous works. The next step for SIFT is to declare a descriptor to contain enough information for matching. The descriptor in SIFT is a 128 feature vector for each interesting point, which contains the information of scale and orientation. In fact, the 128 feature vector is orientation

histograms summarizing the 4 X 4 sub regions. The sub region is defined by a Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point.

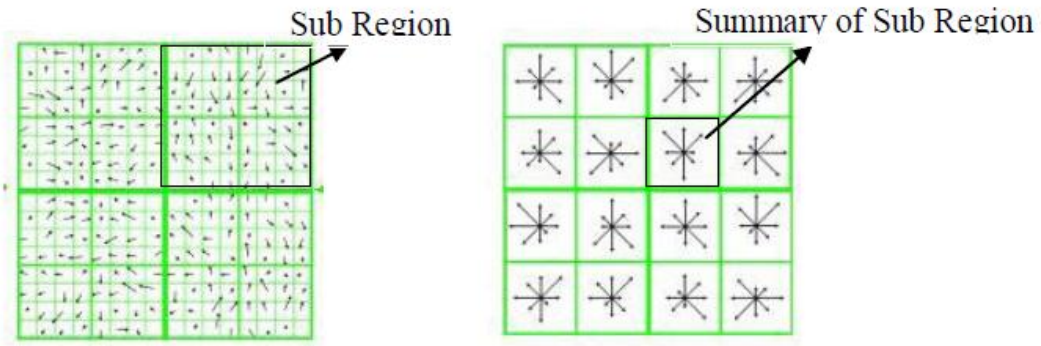

Figure 2-7 Left: image gradients Right: Key point descriptor

Figure 2-7 shows how to compute SIFT feature vector. For each key point, to compute it neighbours' orientations and then summarizing those orientations by sub region. SIFT keeps 4 X 4 summarized orientations, for each sub region includes 8 directions. Therefore, the final vector contains 4 x 4 x 8 (128) elements, which can be seen from the right figure of figure 2-7. In this figure there are 128 arrows, each of them indicates one entry of this feature vector.

2.3.3 SURF

SURF is another method for detecting and matching corresponding points, which computation performance is higher than SIFT. SIFT relies on double image but SURF depends on the integral images, which is able to reduce computational time. SURF detection algorithm is based on the Hessian matrix called Fast-Hessian Detector. Hessian matrix is the square matrix of second-order partial derivatives of a function. The Hessian matrix function in SURF is the convolution of image function and Gaussian.

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

Formula 2-12 [11]

, where $L_{xx}(x,\delta)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image function I in point x.

The detector apply box filter to approximate Gaussian second order derivatives. Figure 2-7 is an example of using 9 x 9 box filter to approximate Gaussian second order derivatives. Image pyramids are built in SIFT, in order to detect scale invariant features. SURF makes use of box filter and integral images, it don't need to iteratively reduce

image size to build scale spaces. SURF adjusts the box filter size to build scale spaces, which is faster than SIFT. For instance, SURF use 9 x 9, 15 x 15, 21 x 21, 27 x 27 as different box filters size. If Gaussian second order derivatives δ=1.2 is corresponding to 9 x 9 box filters, then 27 x 27 box filter is related to δ=3 x 1.2 = 3.6. To locate interest point 3 x 3 x 3 neighbourhood is used to find maxima of the determinant of the Hessian matrix.

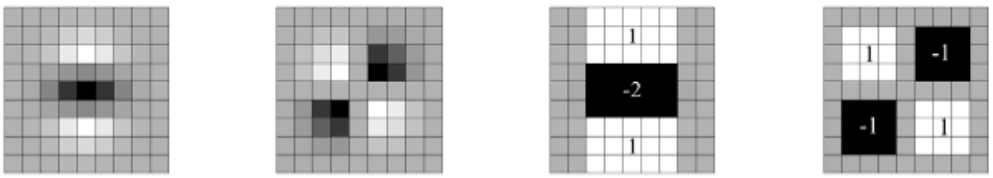

Figure 2-7 Left to right: the (discredited and cropped) Gaussian second order partial derivatives in y-direction and x y-direction, and our approximations thereof using box filters. The grey regions are equal to zero. [11]

In order to detect the rotation invariant features, SURF's orientation is assigned by computing the Haar-wavelet response in x or y in any scale and around interest point. The SURF descriptor is computed as following steps: [11]

1) Region selection: the region is selected as a square region cantered on the interest points. The orientation is assigned orientation of this interest point. The region size is 20 * 20
2) Computing sub region features: The square region is divided by 4 x 4 sub square regions. In each sub region compute 5 x 5 sample points for the feature vector

$$v = (\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma|dy|)$$

Formula 2-13 [11]

Where, dx denotes wavelet response in x direction, dy denotes wavelet response in y direction, |dx| denotes absolute value of response from x direction, |dy| denotes absolute value of response of y direction.

3) SURF obtains 4 x 4 x 4 = 64 feature vector, which is the half in size of the SIFT feature vector.

2.3.4 Summary

SIFT and SURF are two robust algorithms, but Harris based detection is not a scale invariant feature detector. Therefore, we will consider to evaluate some open source libraries for SIFT or SIRF. In chapter 5 evaluation result is listed.

2.4 Computation of Homography matrix

Given a group corresponding interest points, how to estimate a transformation matrix is a problem. This problem can be converted to solve a system of linear or quadratic equations problem. Given different number of corresponding points the linear equations have different solution. For instance, four points are given the perspective matrix has a unique solution. More than four points are given, this a system of linear equations will meet over-determined problem. There are several solutions for over-determined liner equation. Direct Linear Transformation algorithm (DLT), gold standard algorithm and Random Sample Consensus (RANSAC) are three algorithms to estimate projective matrix under the over-determined situation. This section we will describe these three algorithms.

2.4.1 DLT (Direct Linear Transformation algorithm)

2D transformation generally can be represented by the Formula 2-14, which finally can be written by the form of A_ih=0, where A_i is 2 x 9 matrix. The derivation of this equation can be referred to the page 88-90 in the reference [6]. The problem of estimate H is that given more than required corresponding points, a system of linear equations are over-determined.

$$x_{i'} \chi H x_i = 0, \ where \ \chi \ is \ cross \ product$$

Formula 2-14

DLT algorithm aims to minimize the norm ‖ Ah ‖ to solve this problem. ‖ Ah ‖ ^{2} is the algebraic distance, which is sum of distance of the all first points project to second image and corresponding points in second image, the idea is from least squares method, which is regard as standard method to the estimate a solution of over-determined systems. The computational method for this algorithm is very simple, just applies SVD on A, then the last column vector of V is the result. Following is the details of this algorithm.

DLT Algorithm: (Given n≥ 4) [13]

i. For each correspondence x_i - x_i' compute the matrix A_i, only the first two rows need be used in general.
ii. Assemble then n 2 X 9 matricesA_i into a single 2n X 9 matrix A.

iii. Obtain the SVD of A the unit singular vector corresponding to the smallest singular value is the solution h. Specifically if A= UDV' with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then h is the last column of V.

iv. The matrix H is determined from h.

2.4.2 RANSAC (Random Sample Consensus)

RANSAC was first published by Fischler and Bolles, which is an iterative process for estimating an optimal model. Given a set of data, RANSAC assume there is an "inliers", which meet most of data in data set. On the contrary "outliers" means data do not fit the model. RANSAC starts with random selection of data to build "inliers" model and iteratively random select sample data to build this "inliers". The optimal model will be defined as the model which is able to cover most of data fit the "inliers". "Outliers" can be seen as noise. This algorithm is based on iterative computing, which can be imagined that the speed is not fast. However, it is widely used in computer graphics applications.

RANSAC Algorithm: [13]

i. Random select a sample of a data points from S and instantiate the model from this subset.

ii. Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S.

iii. If the size of S_i is greater than some threshold T, re-estimate model using all points in S_i and terminate.

iv. If the size is less than T, select a new subset and repeat the above.

After N trails the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i.

2.4.3 The gold standard algorithm

"The computational algorithm that enables this cost function to be minimized is called the Gold Standard algorithm." [13]. There are many kinds of cost functions. The algebraic distance only reflects the error in the second image, which is used in DLT.

However, the error also occurs in the first image. Therefore, the symmetric transfer error can be defined as formula 2-15

$$\sum_i d(x_i, \hat{x}_i)^2 + d\left(x_i', \widehat{x'}_i\right)^2$$

Formula 2-15 [13]

Base on the formula 2-15, the gold standards algorithm aims to minimize this symmetric transfer error. The algorithm is described in the following in details. Gold standard algorithm is regard as most accurate estimation algorithm.

Gold Standard Algorithm: [13]

i. Initialization: compute an initial estimate of $\hat{H}$ to provide a starting point the geometric minimization. For example, use the linear normalized DLT algorithm or use RANSAC to compute $\hat{H}$ from four points correspondences.
ii. Geometric minimization of – either Sampson error:
   - Minimize the Sampson approximation to the geometric error
   - The cost is minimized using the Newton algorithm or Levenberg-Marquardt algorithm over a suitable parameterization of H. For example the matrix may be parameterized by its 9 entries.

   Or Gold Standard Error

   - Compute an initial estimate of the subsidiary variables $\{\hat{x}_i\}$ using the measured point $\{x_i\}$
   - Minimize the cost

     $$\sum_i d(x_i, \hat{x}_i)^2 + d\left(x_i', \widehat{x'}_i\right)^2$$

     Over $\hat{H}$ and $\hat{x}_i, i = 1, \ldots, n$ the cost is minimized using the Levenberg-Marquardt algorithm over 2n + 9 variables: 2n for the n 2D points $\hat{x}_i$ and 9 for the homography matrix $\hat{H}$

   - If the number of points is large then sparse method of minimizing this cost function given is the recommended approach

This algorism requires an initial guess computed by DTL or RANSAC described in below, and then apply non-linear algorithm to refine the initial guess. Figure 2-8 shows an example how to implement the gold standard algorithm. Only "in liners" are used to Levenberg-Marquardt refinement. H is the initial guess computed by RANSAC, H' is refined matrix which has less back-projection error than H.

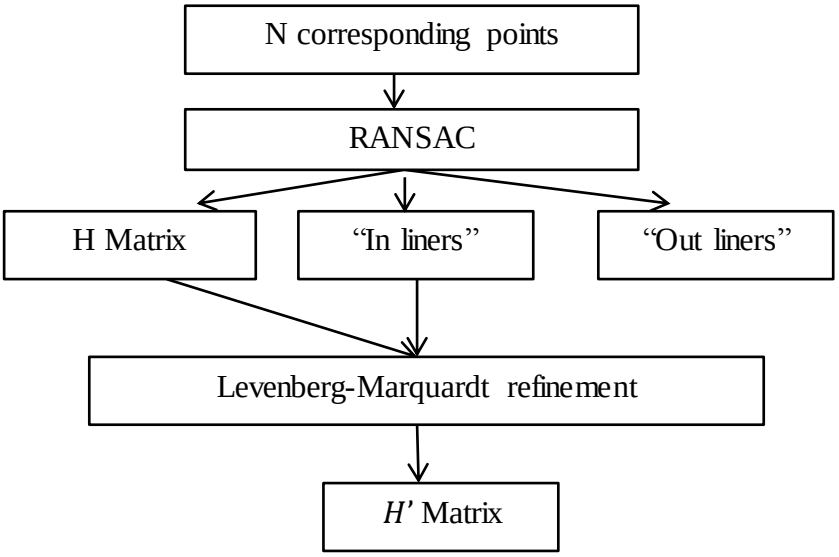

Figure 2-8 computation process of gold standard algorithm

2.4.4 Summary

Research shows that gold standard algorithm is better than other algorithms. This project will employ RANSC to estimate homography matrix to initial homography matrix then use LM refine this matrix as gold standard algorithm described above.

Chapter 3

Designation Process

This project largely depends on Open CV 2.1 which is object-based programming library. Basic data structure also depends on Open CV 2.1. Design methodology is based on prototype development model. It starts from simple implementation to complicated implementation. Modules in this project have vertical hierarchy according to the computation flow. Each module implements one algorithm. Designing of modules is based on the concept of aspect-oriented programming, which entails breaking down program logic into distinct parts. In another words, we are able to replace modules or algorithms flexibly.

The main designation flow is listed in below

1) Identify basic modules, data input and data output of modules
2) Mapping each module to a certain function
3) Implement each functions

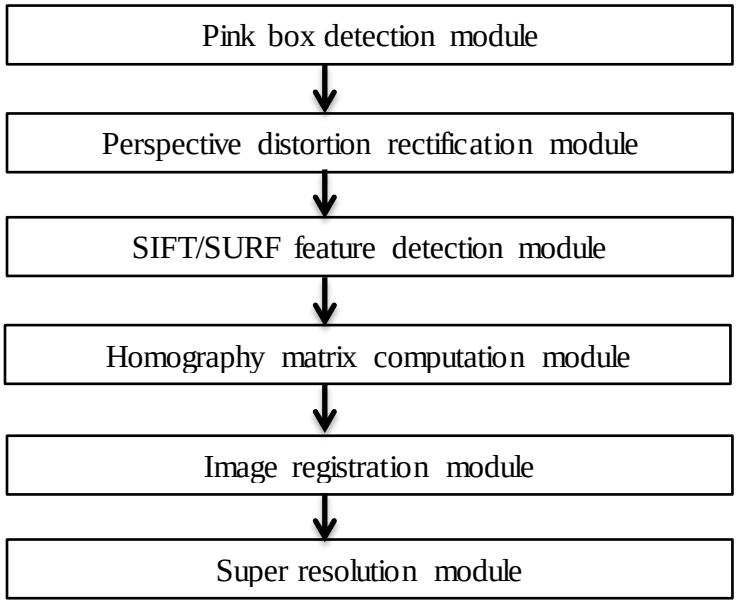

Figure 3-1 basic modules structure

We identified 6 modules: pink box detection module, perspective distortion rectification module, SIFT detection module, homography matrix computation module, image registration module and super resolution module. We also identify data input and data output for each module. Relationship of module is in vertical hierarchy figure 3-1 shows hierarchy of these six modules. The output of top layer module is the input of bottom layer module.

Pink box detection module

- Main description: This module converts each frame from RGB three channels to GRAY scale one channel and detects region of interesting for the pink box.
- Data Input: Raw frame from video
- Data Output: Region of interesting (pink box region)

Perspective distortion rectification module

- Main description: This module removes perspective distortion for the region of interesting and interpolates rectified text to original frame.
- Data Input: Region of interesting
- Data Output: rectified frame

SIFT/SURF feature points detection module

- Main description: This module is required to compute corresponding points for two frames. SIFT algorithm is applied to detect feature points.
- Data Input: rectified two adjacent frames
- Data Output: pairs of corresponding points

Homography matrix computation module

- Main description: This module computes homography matrix given by pairs of corresponding points and warps one frame to the other by the homography matrix.
- Data Input: pairs of corresponding points
- Data Output: two similar images

Image registration module

- Main description:This module will align two similar images
- Data Input: two similar images
- Data Output: two aligned images

Super resolution module

- Main description: This module creates a high resolution image from two aligned images and updates this high-resolution image to the current frame.

- Input Data: two aligned images
- Output Data: one high-resolution image merged two images

The Methodology of development is prototype, which starts from a simple prototype and continue to improve this prototype in order to obtain a better result. This method is also rapid because prototype is often built in a quick way. First we implement each module in a simple way, latter analyse this module and then improve it.

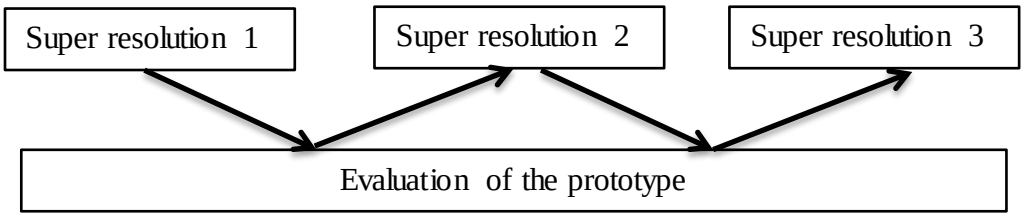

Figure 3-2 an example of prototype development in this project

Figure 3-2 illustrates one part of development process of this project. First we implement super resolution algorithm 1 then evaluate its result. Then we implement super resolution algorithm 2 which is more complicated than the first and evaluate its result. Again, we implement super resolution algorithm 3 which is more complicated than super resolution 2 and evaluate its result which is expected better than former algorithms.

Chapter 4

Development Process

This chapter describes implementation details step by step including computation process, temporary computation results.

Main computation flowchart of this project is described in figure 4-1, which is more specific than figure 3-1. The similar computation flow is used by Capel [5], which aims mosaicking for a sequence of images acquired by a camera rotating about its centre. Capel's techniques are also based on homography matrix computation and super resolution construction. The difference of Capel's computation flow and this project is that Capel employs Harris corners detection finding feature points and maps corresponding points in local windows. This project employs SIFT feature detection algorithm and maps corresponding points with kd-tree algorithm. Another difference is about eliminating perspective distortion. Capel eliminates perspective distortion by projecting the images onto a cylinder centred on the camera centre, and aligned with the dominant axis of rotation. This method requires the knowledge of internal camera parameters. This project removes perspective distortion based on the Hough transformation and background boundaries detection, which doesn't need the knowledge of any camera parameters.

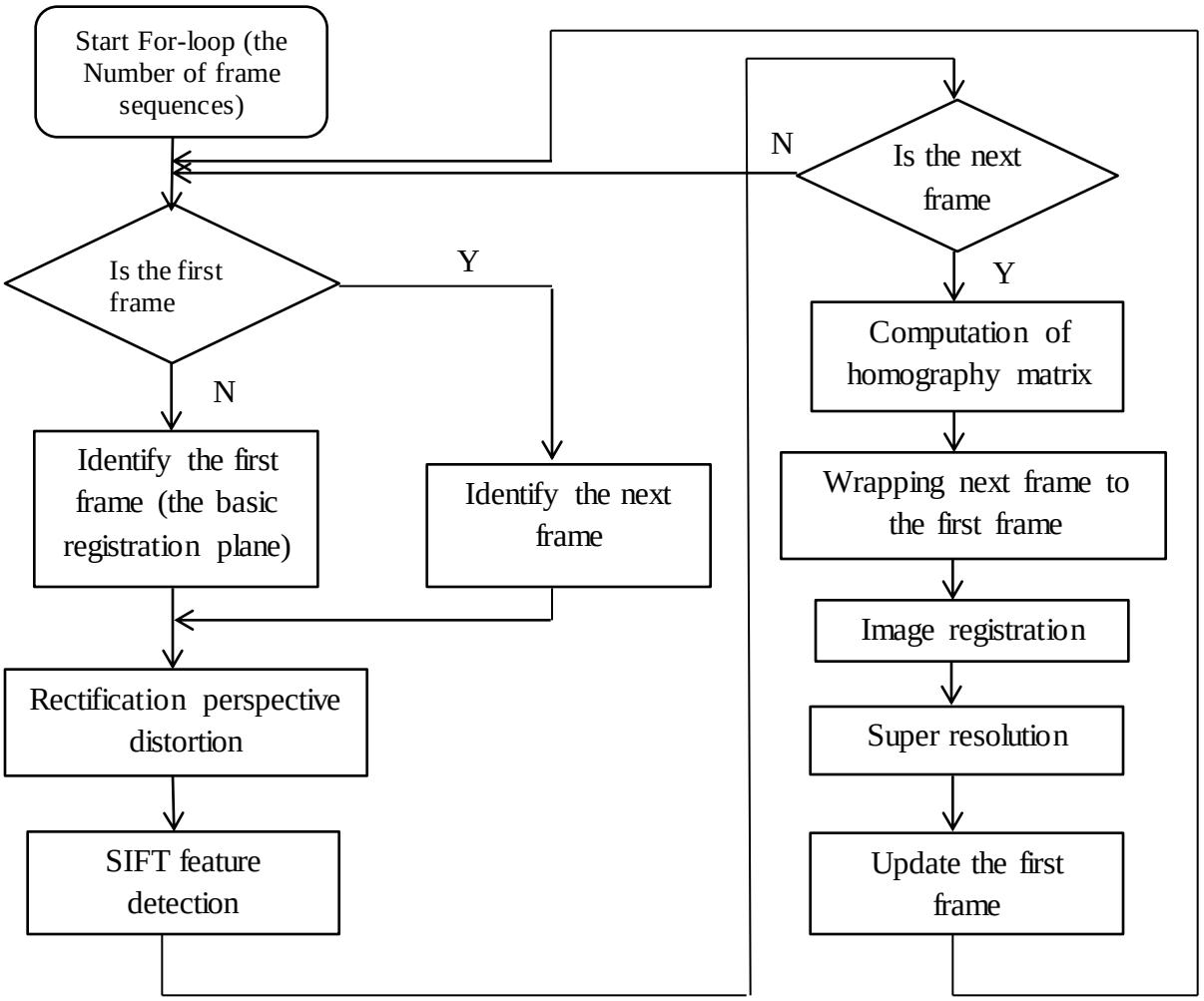

Figure 4-1 main computation flowchart

4.1 Data description

General information is that videos are recorded at 15 fps and with width in 640 dimension pixel and height in 480 dimension pixel. Range of highlighted text area is from 112 x 28 to 130 x 128. Video frames are in 8-bit RGB three channels format. Average of video sequences is around 300, which means a piece of video contains around 300 frames. Following we list main descriptions of each video. Text contents of each video are in capital letter referred as name of each video, main characters are also listed below.

- BORDERS: with smooth perspective change. (outdoor)
- UOB: including an occlusion in a highly textured scene background. (outdoor)
- ST. MICHAEL'S HOSPITAL: with significant perspective change. (outdoor)
- LORRY: undergoing viewpoint changes. (outdoor)

Each video is a series of video frame sequences, in which two adjacent frame sequences can be described by $I(x,y,t)$ and $I(x,y,t+1)$ respectively. The relationship between $I(x,y,t)$ and $I(x,y,t+1)$ is described by equation $I(x+p(x,y,t),y+q(x,y,t),t+1)=I\ (x,y,t)$, where $p(x,y,t)\ and\ q(x,y,t)$ is an arbitrary function which means there is a certain kind of transformation from t^{th} and $t+1^{th}$ frame and t^{th} frame can be computed from $t+1^{th}$ frame. Figure 1-1 shows one frame of our data source. Each frame in our data source contains a text scene which has already been tracked and highlighted by a pink box.

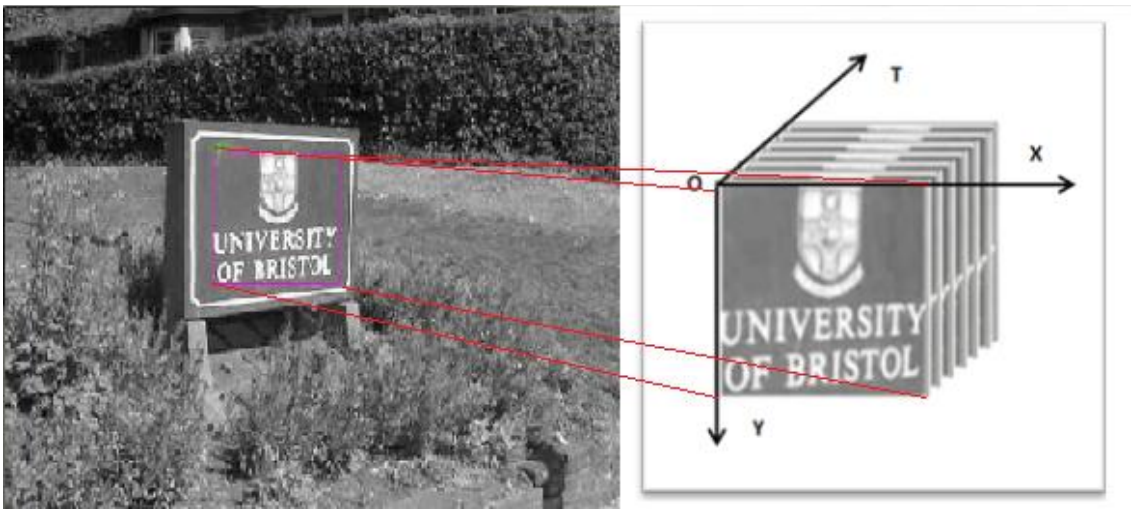

Figure 4-2 data source description Left: original frame with detected text scene in a pink box. Right: a model of a series of video frame sequences.

Figure 4-2 right shows the model of a series of frame sequences. Frame sequences are mapping to three dimension coordinate system, x indicates horizontal pixel, y indicates vertical pixel, t indicates time sequences. More specifically, different frames such as $I(x,y,t)$ and $I(x,y,t+1)$ have a certain relationship of 2D linear projective transformation, this is known as Homography. A mathematical module of describing our data source can be built from supposing the point x_i is in the frame $I(x,y,t)$ and supposing the corresponding point x_i' is in the frame $I(x,y,t+1)$. The relationship between these two points is our basic module which described by the formula $x_i' = Hx_i$, where H is a 3 x 3 homography matrix. We assume x'= $(x',y')^T$ and x=$(x,y)^T$, the corresponding points in two frames $I(x,y,t+1)$ and $I(x,y,t)$, then

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

, where $h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}$ are the 9 entries of homography matrix.

$$H=\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}.$$

We also collect some statistic information for 4 piece videos.

	Total frames	Available frames (with pink box text scene)	Average intensity(available frames)
1	327	48	35.24
2	325	107	129.87
3	133	65	86.12
4	144	80	125.34

Table 4-1 basic statistic information of our data source

4.2 Data pre-processing

Our source data is a piece of video, which text scene has already been detected. The boundaries of text scene have been highlighted by a pink box. The task of data pre-processing is to detect those pink boundaries to prepare raw input image data. The highlighted pink box region is set to Region of Interesting (ROI). Three channels of RGB image is converted to GRAY scale one channel image which is ready to the next step of computation.

4.3 Perspective distortion rectification

This project employs the Hough transformation based method to rectify perspective distortion. Moreover, it also relies on the boundaries of the text area from the real scene. However, in the case of video 4, there are no boundaries for text area. We also employ the Hough transformation algorithm to detect main direction of the text area such as top line and bottom line, which only rectifies horizontal perspective distortion. It is only able to obtain a roughly rectified image regarded as a complement algorithm for the special case. The implementation of this subtask depends on Open CV's cvCanny, cvHoughLines2, cvSolve, cvGetPerspectiveTransform and cvWarpPerspective function.

1) Canny edge detection

In order to detect high contrast block and low contrast block, we have to tune parameters of canny edge detection. For low contrast block we choose lower thresholds and for high contrast block we choose higher thresholds. Figure 4-2 shows different

canny edge detection result by using different thresholds. The original frame has low contrast between the board and the background.

Figure 4-3 Left: original frame Centre: canny edge detection result with two higher canny thresholds (150, 200) Right: canny edge detection result with two lower thresholds (20, 20).

In figure 4-3, the original frame is a low contrast block image. It is unable to detect whole text board boundaries with higher canny thresholds shown in the centre of figure 4-3. Changing canny edge detection parameters is necessary for different video pieces.

2) The Hough transformation line detection

We apply the Hough transformation for the output of canny edge detection and receive numbers of line segments. The line's direction is various which relies on the content of images.

3) Boundaries detection

We compute tangent of each line segments and classify them into four categories, each category contains all possible lines for a boundary such as left boundary, right boundary, etc. For each category, we find the most central line segment as boundary of text scene, because we have known text area always in the centre of text scene. Indicated by figure 4-4, lines are detected by Hough transformation including red lines and yellow lines. Only yellow lines are the result of boundaries detection. We also compute count of lines for each category in order to find main direction for each frame, which is an extra information in the case of there is no boundaries.

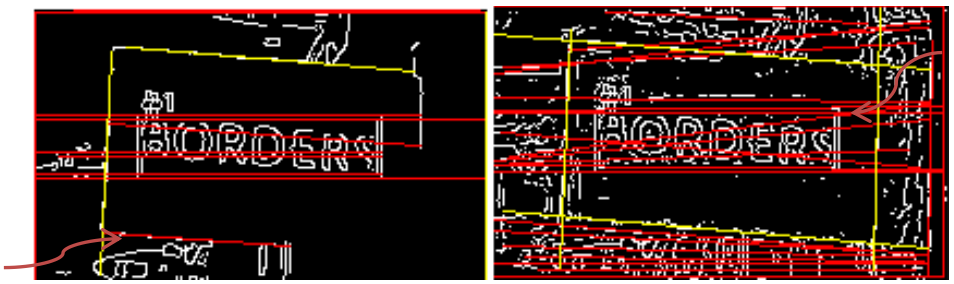

Figure 4-4 Left: boundaries detection result with bad canny output image. Right: boundaries detection result with good canny output image. Yellow lines indicate the boundaries. Red lines indicate the result of the Hough transformation.

4) Computation of intersection of four boundaries as quadrilateral points, mapping quadrilateral points to rectangle points.

By solving four linear equations, we obtain four quadrilateral points. Map these four quadrilateral points to rectangle points with approximately same area. We assume we have four corner of quadrilateral $x_1 = (x_1, y_1)^T$, $x_2 = (x_2, y_2)^T$, $x_3 = (x_3, y_3)^T$, and $x_4 = (x_4, y_4)^T$ and four corner of rectangle $x_1' = (x_1', y_1')^T$, $x_2' = (x_2', y_2')^T$, $x_3' = (x_3', y_3')^T$ and $x_4' = (x_4', y_4')^T$ .Then $h = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^T$ can be computed by solving a set of linear equations, which is the same problem of solving Ah=b. Formula 4-1 shows how to construct the matrix given four corresponding points, where A is a 9 X 8 matrix.

$$\begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1y_1' & y_1y_1' & y_1' \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' & x_1' \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2y_2' & y_2y_2' & y_2' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2' & -y_2x_2' & x_2' \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3y_3' & y_3y_3' & y_3' \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x_3' & -y_3x_3' & x_3' \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4y_4' & y_4y_4' & y_4' \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x_4' & -y_4x_4' & x_4' \end{bmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = 0$$

Formula 4-1

5) Correct rectangle points by measured same area of quadrilateral and rectangle.
6) Computation of perspective matrix from four pairs of points: quadrilateral points and rectangle points.
7) Perspective warping raw image to rectified image by perspective matrix

Figure 4-5 Left: original image. Centre: Canny edge detection result (white line), the Hough line detection result (red line) and boundaries detection (yellow line) Left: result after rectification of perspective distortion

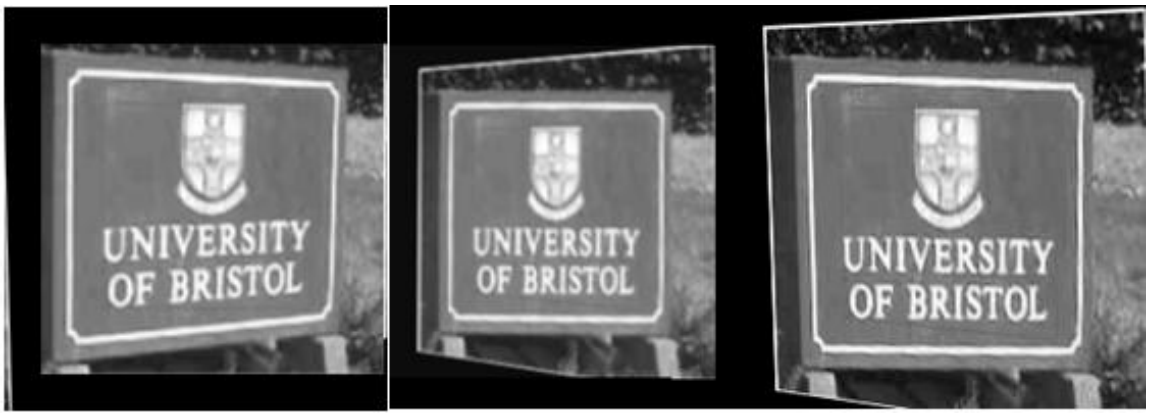

Figure 4-6 Left: original image. Centre: result after rectification of perspective distortion Left: Rectification with the correct relative scaling using a same area of quadrilateral and rectangle.

A complementary algorithm is described below to cope with no boundaries text scene. This algorithm is not good as the first algorithm but it is able to roughly handle no boundaries real world text scene. The difference of this algorithm and the first algorithm is that this algorithm employs two to four main direction lines to remove perspective distortion rather than four boundaries.

1) Canny edge detection
2) Hough transformation line detection
3) Detection of two lines in dominant direction
4) Rectification of two or four lines in main direction
5) Computation of perspective matrix from four pairs of points: quadrilateral points and rectangle points
6) Warping one frame to the another

Figure 4-7 Up-Left: original image. Bottom-Left: Canny edge detection (white line), Hough line detection (red line) and two dominant detection lines (yellow line) Right: roughly rectified frame

4.4 Selection of frames

Selection of appropriate candidate frames to create a single image is necessary, because some noise frames should be. We categories noise frames and sort them out from computation.

1. Frames with text scene but do not highlighted by a pink box, we have to ignore it.
2. Adjacent frames should have similar size of highlighted pink box. If the size difference of two adjacent frames greater than a threshold, we ignore the second frame. Figure 4-8 are two adjacent frames but with different size of pink box, we ignore the second frame.
3. Frame with feature points less than 10, we ignore this frame, because we have to find at least over than 4 pairs of corresponding points to compute homography matrix.
4. Two adjacent frames have less than 4 corresponding points, we ignore the second frame, which means SIFT matching algorithm can't find enough points pair for information merging.
5. Residual error over than a threshold we ignore this frame.

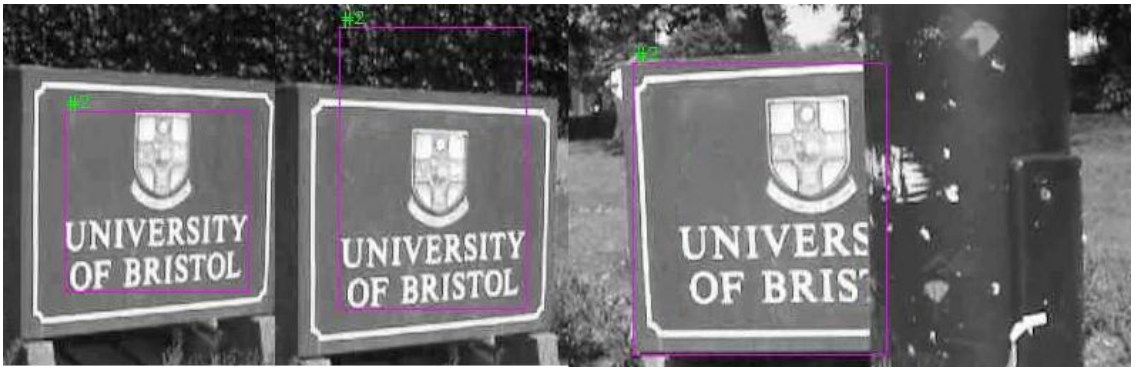

Figure 4-8 Left: the first frame Centre: the second frame Right: a noise frame. The first frame and the second frame are two adjacent frames. The pink box in the second has different size of the first frame. It is regarded as a noise frame.

Total	**325**	**Ignored frame caused by**
1	202	No pink box
2	8	Different size of pink box
3	0	Feature points less than 10
4	1	Matching points less than 4 pairs
5	20	Residual error over the threshold but it is various for different supper resolution algorithms

Table 4-2 statistic number of ignored frames for different reasons

4.5 Computation for homography matrix and Levenberg-Marquardt refinement

The computation of homography matrix is implemented by Open CV. We employ function cvFindHomography from Open CV to compute this matrix which is able to mark "inliners" and "outliners". We also integrated lm.lib to help the further refinement. Figure 4-9 shows the process of computation a good homography matrix. RANSAC is described in chapter 2.4.3 which is applied to initial homography H. "inliners" corresponding points remains and "outliners" corresponding pints are removed. Then, we only use "inliners" to compute homography H again. *H'* is computed out by Levenberg-Marquardt non-liner algorithm described in chapter 2.4.2. *H'* has less re-projection error than H.

Figure 4-9 are two residual error value images computed by different homography matrix. Left error image is computed by the initial homography matrix H. Right error image is computed by the refined homgraphy matrix *H'*. (White dots are with higher value than black dots) It can be seen that left error image is lighter than right error image, which means Levenberg-Marquardt reduces the residual error. Figure 4-10 shows residual error of each frames, pink line in the top indicates residual error without Levenberg-Marquardt refinement. Green line indicates residudal error with Levenberg-Marquardt refinement. It is clear that pink line always lies above than green line. It is able to reduce approximately 10% ~ 20% residual error.

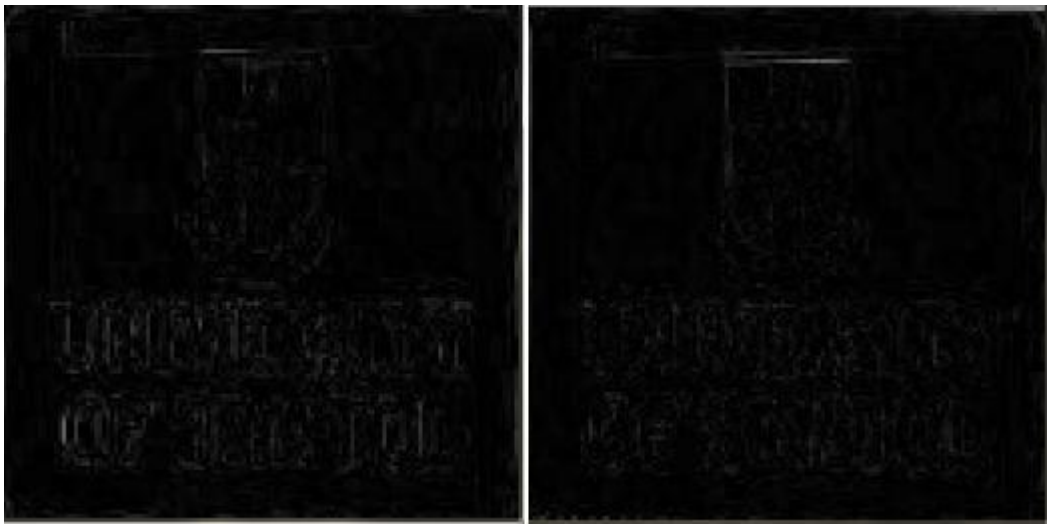

Figure 9 Left: residual error value image is computed by the initial homography matrix H. Right: residual error image is computed by the refined homgraphy matrix *H'*. White dots indicate the error value. More white dots mean more error points. These two error images show the distribution of residual error.

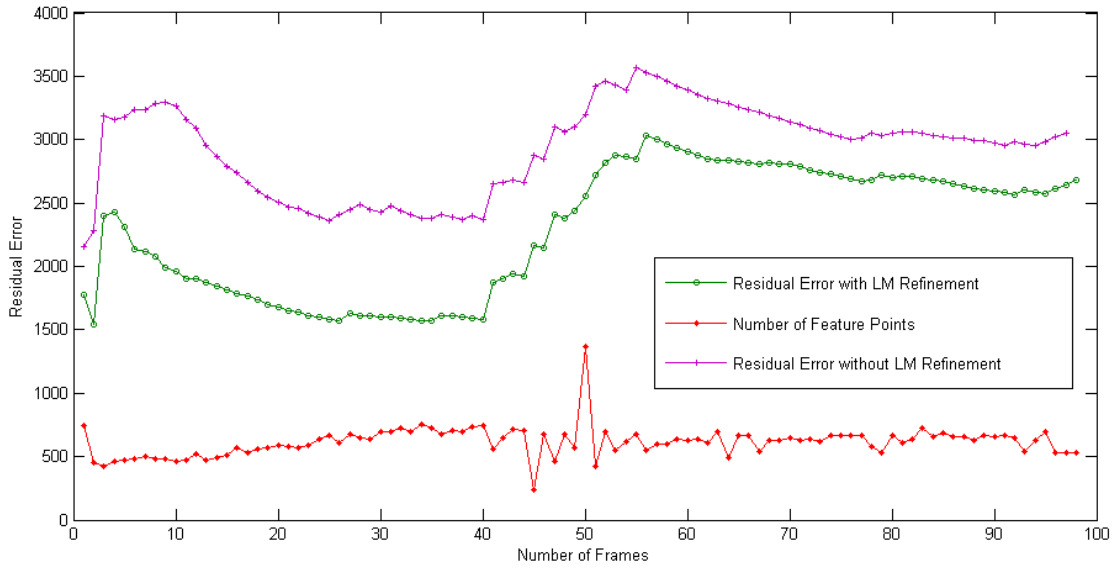

Figure 4-10 Residual error of each frames.(pink line indicates residual error without Levenberg-Marquardt refinement, green line indicates residudal error with Levenberg-Marquardt refinement, red line is the number of feature points in each frames)

4.6 Image registration

Image registration is widely used in remote sensing, medical imaging, computer vision etc. The purpose of image registration is to align two or several images in order to analysis multi-views. Normalized Cross-Correlation (NCC) is the typical algorithm for the image registration. We plot NCC in figure 4-11. Generated image called template searching in original frame and ready for fusion of information. In figure 4-11 right image is our plot for NCC registration. The circle in the plot indicates a white point, which is the point with the maximum likelihood position of the template in the original frame. By using image registration we are able to align two similar images, and then we are able to merge the information carried by the two similar images.

A more efficient image registration manner is based on histograms. This project dose not implements this algorithm which can be an item of future work.

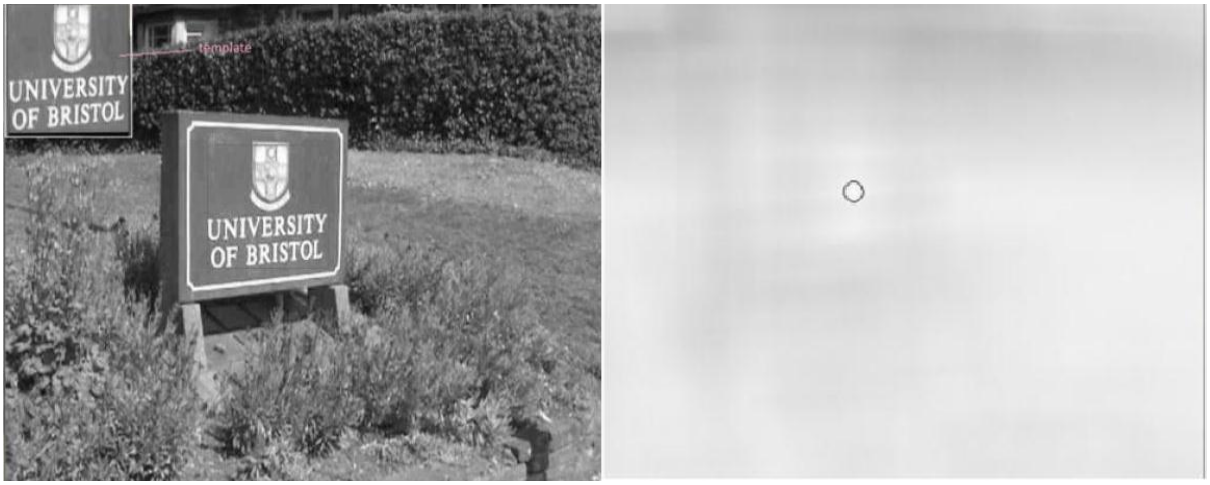

Figure 4-11 Left: image registration, a template matching in the big frame Right: the result of matching, the circle indicates the maximum likelihood position of the template in the big frame.

4.7 Image quality improvement by super resolution

After alignment, choosing which surface to reproject forming the single image is freely chosen. The simplest method is to choose the first frame as repoject surface plane and warp all frames to the first frame. Computation benefit of this is clear, it does not need loop for all candidate frames and select one as the reprojection surface plane. In order to eliminate accumulator error, we compute the homography matrix between a new frame and the frame we have interpolated by the super resolution algorithm.

4.7.1 Super resolution 1 - Average

Average of all low resolution images to one single image is one of easiest way to merge information. One reason of implementation this method is to set this result as a baseline to compare with other sophisticated measures. Moreover, Irani and Peleg's method needs average of all low-resolution pixel values as initial guess of high-resolution values. Figure 4-12 is the result of average super resolution algorithm, which looks like a blurred image, but some noises are disappear. We will use this image as Irani and Peleg's algorithm's initial guess value.

Figure 4-12 the result of super resolution 1(video 2) – average

4.7.2 Super resolution 2 - Simple super resolution

We also implement another simple super resolution by bi-cubic interpolation and fusing two low frames. Bi-cubic interpolation is often used to super resolution, which is an extension of cubic interpolation. The interpolation by bi-cubic method is smoother than bilinear interpolation or nearest-neighbour, but speed of computation is slower than others.

Figure 4-13 illustrates our simple super resolution algorithm. After the image registration, we up sampling one frame by bi-cubic interpolation. One pixel from the first frame becomes four pixels after up sampling. We fill the second frame pixel to the diagonal pixel in the first image. Finally, we down sampling by bi-cubic interpolation obtaining the result of fused information image. This method is simple but it is effectively merging information and removing noising.

Figure 4-14 is the result of this simple super resolution algorithm. We create a double resolution image displayed in top-right of figure 4-14. Comparison by original frame top-left of figure 4-14, this high-resolution image contains more pixels. When the time it is enlarged, high- resolution image is much clearer than low resolution image referred to bottom of figure 4-14.

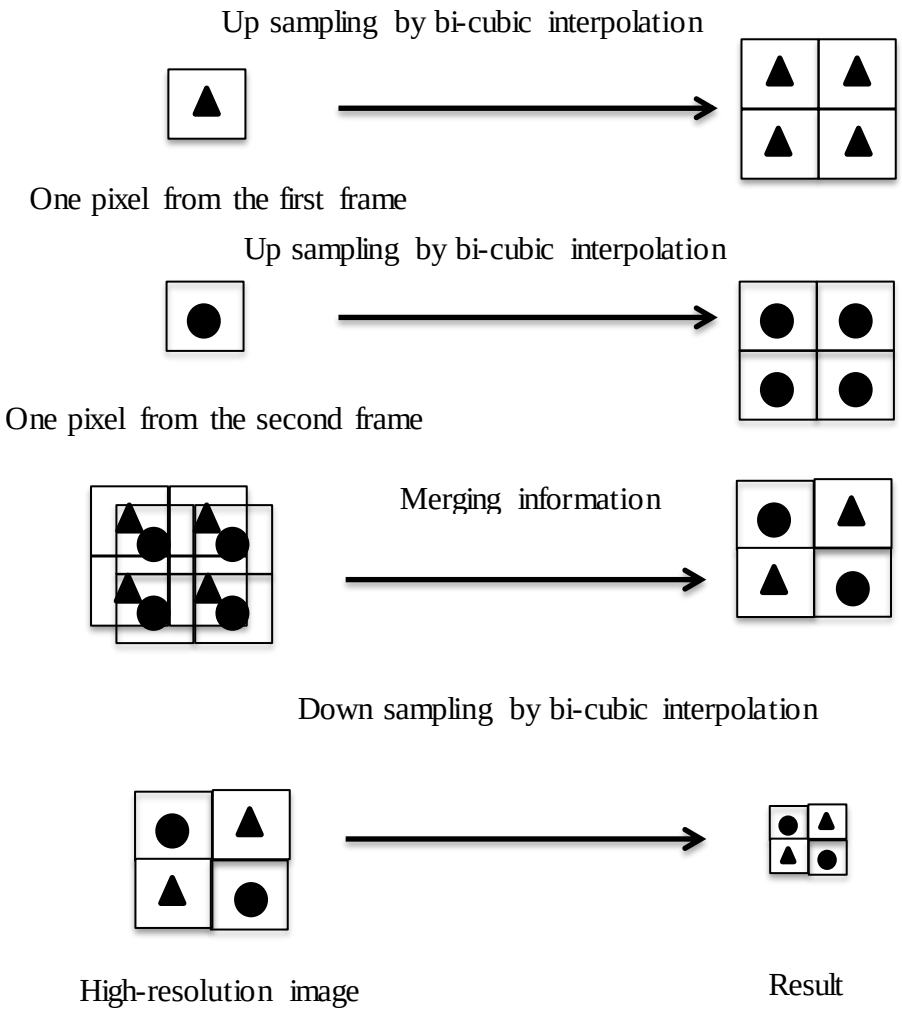

Figure 4-13 an illustration of our second super resolution algorithm

Figure 4-14 the result of the second super resolution algorithm – simple super resolution. Up-left: a part of original frame. Up-right: the same area in the result frame. Bottom-left: a sample enlarged area from original frame Bottom-right: a sample enlarged area from the result frame.

4.7.3 Super resolution 3 - Irani and Peleg's super resolution

Due to the constraint of hardware limitation we choose Irani and Peleg's algorithm to estimate high-resolution image. This subtask starts with creation of a kernel matrix. This matrix is an essential element of Irani and Peleg's algorithm, which reflects the relationship between low-resolution image and high-resolution image. Rows indicate high-resolution pixels, columns indicate low-resolution pixels. Each entry reflects 2D location distance from high resolution pixel to low resolution pixel. The i^{th} row values reflect the relationship of i^{th} pixel in the low resolution image and every pixel in high resolution image. Therefore, this matrix is a sparse matrix, because each low resolution pixel only can influence a window (a small area) of high resolution image pixels. We set a window size of this area from 1 to 5, which is an area in high resolution image can be influenced by one pixel in low resolution image.

Window size 1 means one pixel in low resolution image is influenced by 9 pixels in high resolution image. Window size 3 means one pixel in low resolution image is influenced by 9 * 9 = 81 pixels in high resolution image. Window size 5 means one pixel in low resolution image is influenced by 9 * 25 = 225 pixels in high-resolution image. Figure 4-15 shows the different window size's high-resolution image results. It could be compared by the initial guess image figure 4-12 that initial guess looks like a blurred image but our Irani and Peleg's result 's edge is more clear than the initial guess.

Larger window size result requires higher computation cost and it is more precise than smaller window size result. In order to balance the computation cost and accuracy, we choose window size equals 3 as our solution. It can be seen from figure 4-15 the result from window size 3 and window size 5 are much more similar than the result from window size 1. Therefor window size equal 3 is enough to create a high-resolution image.

Figure 4-15 Left: our Irani and Peleg's result of window size 1 x1 Centre: our Irani and Peleg's result of window size 3x3, Right: our Irani and Peleg's result of window size 5 x 5.

By using formula 2-3 described in chapter 2, we iterate update initial guess computed from the first super resolution algorithm. Finally, each available frame's information is added to the initial guess. Figure 4-16 right side is our enlarged final result of Irani and Peleg's algorithm. Compared by the left side original frame from figure 4-16, it demonstrates our result is a de-blurred high-resolution image. Table 4-3 and table 4-4 give a simple example of how to construct a kernel matrix and how simulate a high-resolution image by this kernel matrix.

256	0.0680	0.0968	0.0680	0.0235	0.0040	0.0003	0.0000	0.0000	...
256	0.0003	0.0036	0.0213	0.0614	0.0875	0.0614	0.0213	0.0036	...
256	0.0000	0.0000	0.0000	0.0003	0.0036	0.0213	0.0614	0.0875	...
...	...	...	...	...	...	...	...	...	

Table 4-3 an example of the construction of kernel matrix. The first column (black in background) is the low resolution pixels. The right side matrix is the kernel matrix. This kernel matrix reflects the relationship of high-resolution image and low-resolution image. High-resolution image can be obtained by low-resolution image multiply this matrix. Low-resolution image can be simulated by this matrix multiply a high-resolution image.

17.4097	24.7935	17.4097	6.0277	1.0290	0.0866	0.0036	0.0001	...
0.0782	0.9296	5.4451	15.7269	22.3970	15.7269	5.4451	0.9296	...
0.0000	0.0001	0.0032	0.0782	0.9295	5.4448	15.7261	22.3959	...
...	...	...	...	...	...	...	...	...

Table 4-4 an example of computation of high-resolution values. These values are computed from the above matrix. One pixel in low-resolution is split to several pixels in high-resolution image. The number of influenced pixels in high-resolution image is decided by the window size. We suppose value 256 is black and value 0 is white in this table

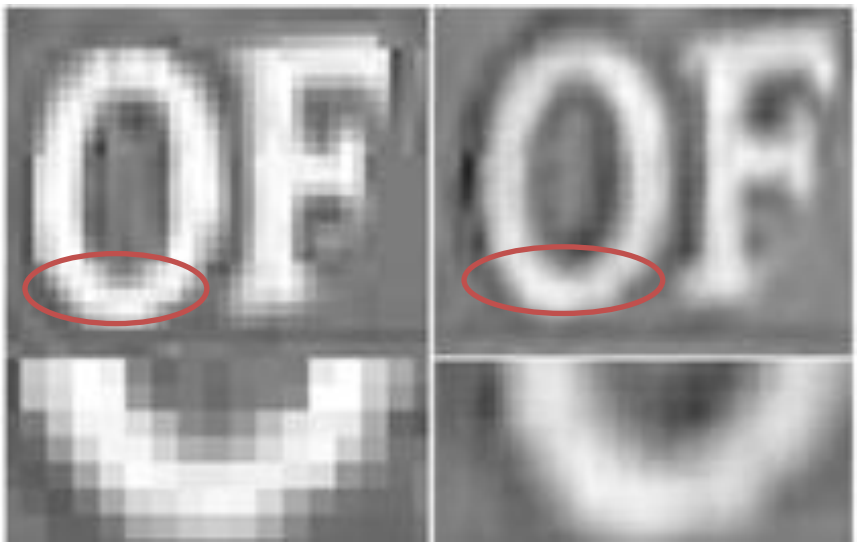

Figure 4-16 Left: part frame from original frame. Right: same area of our Irani and Peleg's result frame.

4.8 Development Environment description

The development environment is window 7 by develop tools visual studio 2008. Our programming language is C++. This software depends on several open source compute vision and image processing libraries.

Figure 4-17 shows libraries depended by this project's architecture. Open CV 2.1 is our main depended library. Rob Hess SIFT library is used to obtain feature points described above. Levmar is used to enhance homegraphy matrix's accuracy which relies on the library CLAPACK.

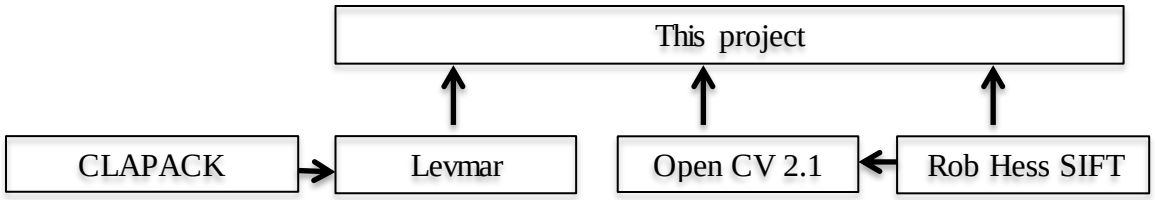

Figure 4 -17 dependence libraries architecture

- OpenCV 2.1: Intel® Open Source Computer Vision Library, which has BSD-like licence including cv210.lib, cvaux210.lib, highgui210.lib, cxcore210.lib,ml210.lib
- Rob Hess SIFT: a SIFT tools relies on OpenCV.
- Levmar: An implementation of Levenberg-Marquardt optimization algorithm, GNU General Public License including Levmar.lib
- CLAPACK: A freely-available software package. The package levmar relies on this package including BLAS.lib, BLAS_nowrap.lib, cblaswrap.lib, cblaswrapd.lib, clapack.lib, clapack_nowrap.lib, clapackd.lib, clapackd_nowrap.lib, f77blaswrap.lib, f77blaswrapd.lib, libf2c.lib, libf2cd.lib, tmglib.lib, tmglib_nowrap.lib, tmglibd.lib, tmglibd_nowrap.lib

Chapter 5

Experimental Results

This chapter describes experimental results including the result of SIFT/SURF tools evaluation, Rob Hess SIFT parameters tuning and OCR result analysis. In chapter 5.1 SIFT and SURF algorithms are evaluated based on criteria of performance and detected numbers of feature points. In chapter 5.2 the experimental results of tuning Rob Hess SIFT will be shown, and will be compared to our result parameters with Low's papers suggestion. In chapter 5.3 OCR results are demonstrated, which are the final results of this project. A comparison of three super resolution algorithm's OCR accuracy and performance will be explained at the end of this chapter.

5.1 SIFT/SURF tools evaluation

The focus is on Open CV SURF, vlfeat SIFT, Rob Hess SIFT, because they are open source feature detection tools and are very popular in machine vision applications. This sub task is to evaluate three of them and decided on an appropriate one for integration into the project.

5.5.1 Evaluation of SIFT and SURF

Due to the consideration of performance, the SIFT or SURF algorithm is applied only on highlighted text areas. The computation performance is compared for both algorithms SIFT and SUFT and is regarded as one of the criteria for selecting an algorithm. The founded number of key points is another consideration for the selection of algorithms. We choose two different sizes of text which is extracted from the data source as sample data for evaluation. Table 5-1 shows the evaluation results of Rob Hess' SIFT and Open CV SURF algorithms. It can be seen that the SURF algorithm has a higher performance than SIFT, but detects a fewer number of feature points than SIFT. For example, SURF only detects 3 feature points in the image size of 111 x 28, which is not enough to compute homography matrix. However, Rob Hess' SIFT is able to detect 81 feature points in the same size of image, which is enough for further computation. Therefore, for smaller sizes of image SIFT rather than SURF must be

used, but for the larger image both can be chosen. Figure 5-1 and figure 5-2 show the feature points detected by SIFT or SURF in the two data sets.

	Number of points	Performance (s)	Image size
Rob Hess SIFT	81	0.49	111 x 28
OpenCv SURF	3	0.01	111 x 28
Rob Hess SIFT	476	0.48	130 x 128
OpenCv SURF	154	0.088	130 x 128

Table 5-1 Comparison of Rob Hess SIFT and Open CV SURF

Figure 5-1 Left: original text scene (111 x 28) Centre: same text scene with feature points detected by SURF Right: same text scene with feature points detected by SIFT

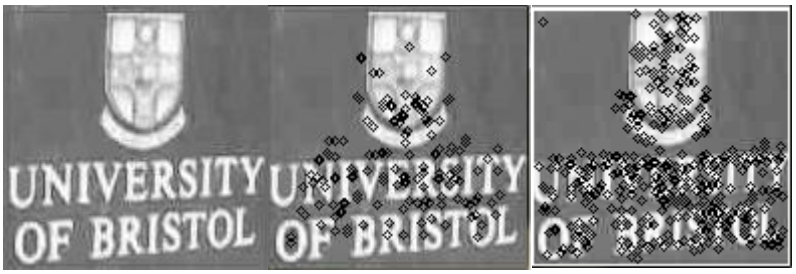

Figure 5-2 Left: original text scene (130 x 128) Centre: same text scene with feature points detected by SURF Right: same text scene with feature points detected by SIFT

Vlfeat and Rob Hess are two open source tools for SIFT. Rob Hess' SIFT tool was chosen over vlfeat SIFT, because Rob Hess' SIFT is based on Open CV, but vlfeat SIFT does not depend on OpenCV, it is a pure C library which is harder to integrate into this project. Another reason for choosing Rob Hess is that the API of Vlfeat's is more complicated than Rob Hess and there is no matching implementation in vlfeat to be integrated. Rob Hess' SIFT API is much easier than vlfeat SIFT and Rob Hess implements the kd-tree algorithm to match corresponding points. Therefore, the best choice is Rob Hess SIFT for this project.

5.5.2 Tuning Rob Hess SIFT parameters

SIFT is a complicated algorithm, so tuning SIFT input parameters is necessary. Residual error of computing homography matrix was employed as judgement criteria to tune the parameters of SIFT, because the purpose of feature finding is to obtain a higher accuracy in the computing homography matrix. According to Low's paper, three parameters are mentioned which affects the result of feature detection. We plan to tune these there parameters in order to achieve a higher accuracy. The three parameters are listed below.

- SIFT_INTVLS: default number of sampled intervals per octave
- SIFT_SIGMA: default sigma for initial Gaussian smoothing
- SIFT_CONTR_THR: default threshold on key points contrast

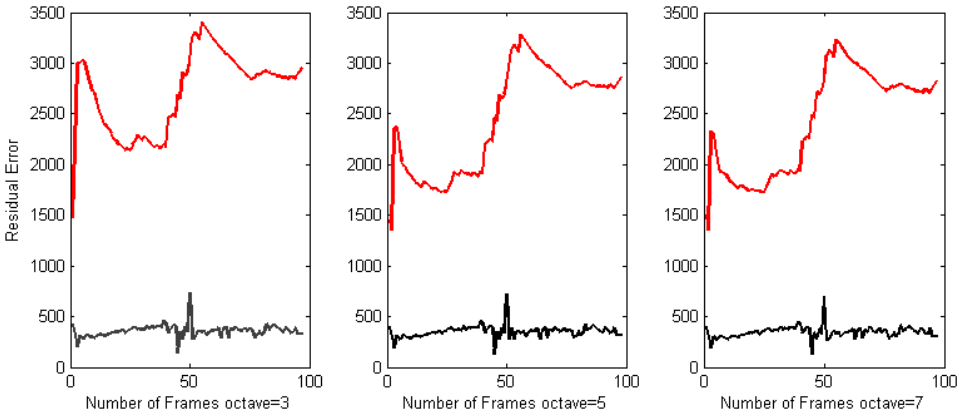

Figure 5-3 average residual for computing homography matrix with different octave values. The red line is the average of residual error, the black line is the number of key points detected by SIFT.

SIFT_INTVLS	7	5	3
Average of residual error	2822.98	2858.37	2951.63

Table 5-2 the final frame residual error for different octave value

Table 5-2 shows the final frame residual error which means the final result. We choose octave equals 5, because the difference of final residual error from octave 7 and octave 5 is not too much but the octave equals 7 takes more time than octave 5. In order to balance the performance and accuracy we choose octave equals 5.

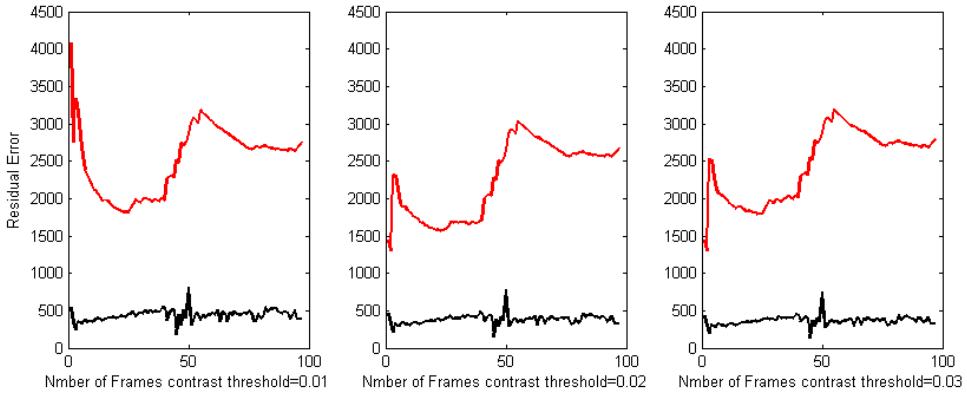

Figure 5-4 Average residual for computing homography matrix with different contrast threshold value. Red line is the average of residual error the black line is the number of key points detected by SIFT. Obviously, contrast threshold value equals 0.02 is best.

SIFT_CONTR_THR	0.01	0.02	0.03
Average of residual error	2750.87	2673.80	2794.72

Table 5-3 final frame residual error for different contrast threshold value

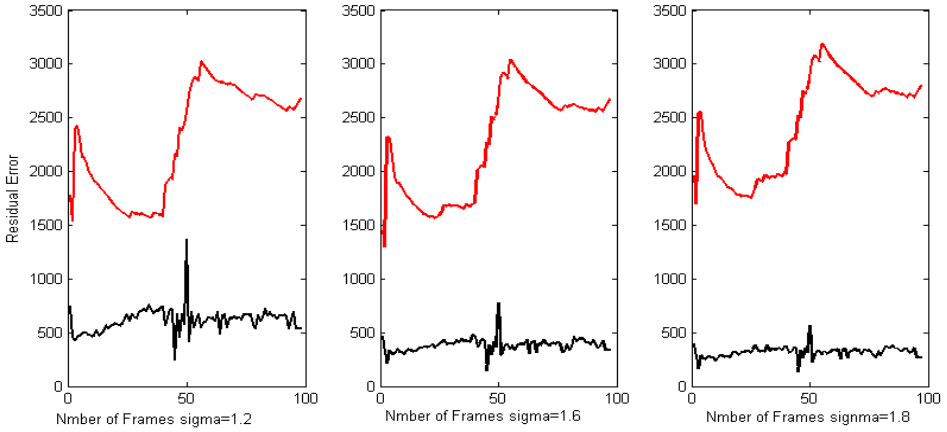

Figure 5-5 average residual for homography matrix with different sigma value. Red line is the average of residual error the black line is the number of key points detected by SIFT

SIFT_SIGMA	1.6	1.2	1.8
Average of residual error	2673.80	2681.80	2801.76

Table 5-4 final frame residual error for different sigma value

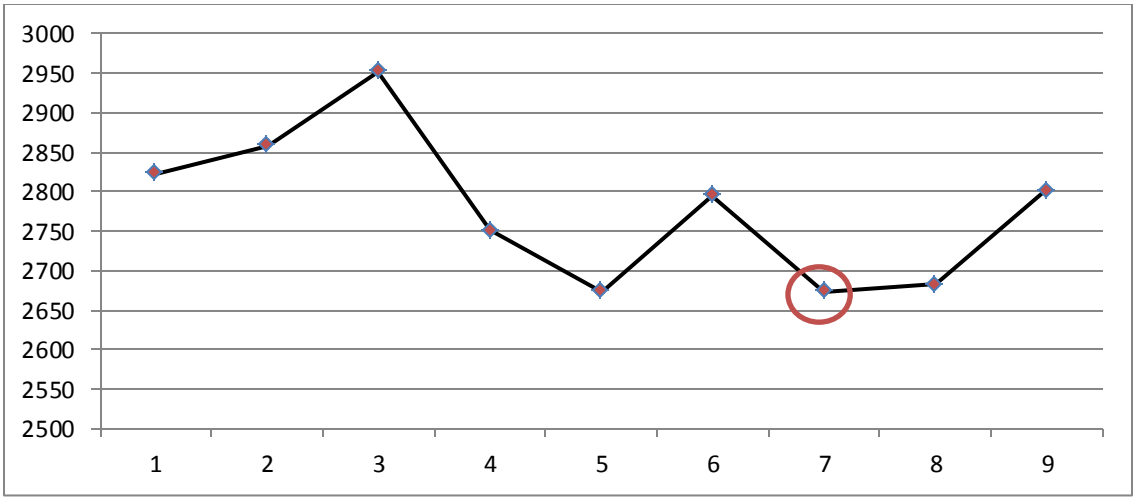

Figure 5-6 Residual Error with different SIFT parameters

	SIFT_INTVLS	SIFT_CONTR_THR	SIFT_SIGMA	RESIDUAL_ERROR
1	7	0.04	1.6	2822.98
2	5	0.04	1.6	2858.37
3	3	0.04	1.6	2951.63
4	5	0.01	1.6	2750.87
5	5	0.02	1.6	2673.80
6	5	0.03	1.6	2794.72
7	5	0.02	1.6	2673.80
8	5	0.02	1.2	2681.80
9	5	0.02	1.8	2801.76

Table 5-5 Summary of residual error with different SIFT parameters

	Low's suggestion	Experimental Result
SIFT_INTVLS	3	5
SIFT_CONTR_THR	< 0.03	0.02
SIFT_SIGMA	1.6	1.6

Table 5-6 Comparison of three parameters with low's suggestion

We compare our experimental result with Low's. Basically, our best experimental results are with similar value of Low's suggestion. It demonstrates that Rob Hess SIFT is a reliable library. Finally, we apply our best experimental result parameters in other three videos.

5.2 OCR accuracy analysis

We employ the Tesseract OCR engine to test our results, which was created by the HP lab but is now in Google.

The raw data was sent to this OCR engine to obtain a number of accuracy as a baseline in order to compare the results processed by our algorithm. The raw data was extracted from the video without any process, which is often with the perspective dissertation and noises. Table 5-7 shows that the OCR engine manages to recognize the 55.7% text scene without any processing. After rectification of perspective rectification, the OCR engine is able to recognize the 77.6% text scene. It demonstrates that the perspective distortion affects the accuracy of OCR severely.

Perspective rectification raw data for OCR

Raw Data	OCR accuracy
Without Perspective distortion rectification	55.7%
With perspective distortion rectification	77.6%

Table 5-7 OCR accuracy for raw data

Table 5-8 shows the OCR accuracy for the quality improved data. Three bunch of quality improved data to OCR was sent and obtained from three super resolution methods mentioned above. Super resolution 1 – average super resolution has an accuracy of 66.6%. It is less than 77.6 %, which means it is unable to improve OCR accuracy. The average low-resolution image leads to the result image blurring, therefore it might be the reason of that OCR hardly recognizes it. Super resolution 2 – simple resolution obtains higher accuracy than the super resolution algorithm. It has 91.6%, which is a good result. This algorithm is able to remove some noises. Figure 4-14 shows an image final result of this simple super resolution algorithm. The result image is smoother than the original one as some white dot noises have disappeared. Super resolution 3 – Irani Peleg's super resolution algorithm has the highest accuracy of OCR. It reaches and accuracy of 96.4%.

Quality improved data for OCR (without perspective distortion)

	Super resolution 1(Average)	Super resolution 2(Simple resolution)	Super resolution 3(Irani and Peleg)
Accuracy	66.6%	91.66%	96.4%

Table 5-8 OCR accuracy for quality improved data (Residual error threshold 3000)

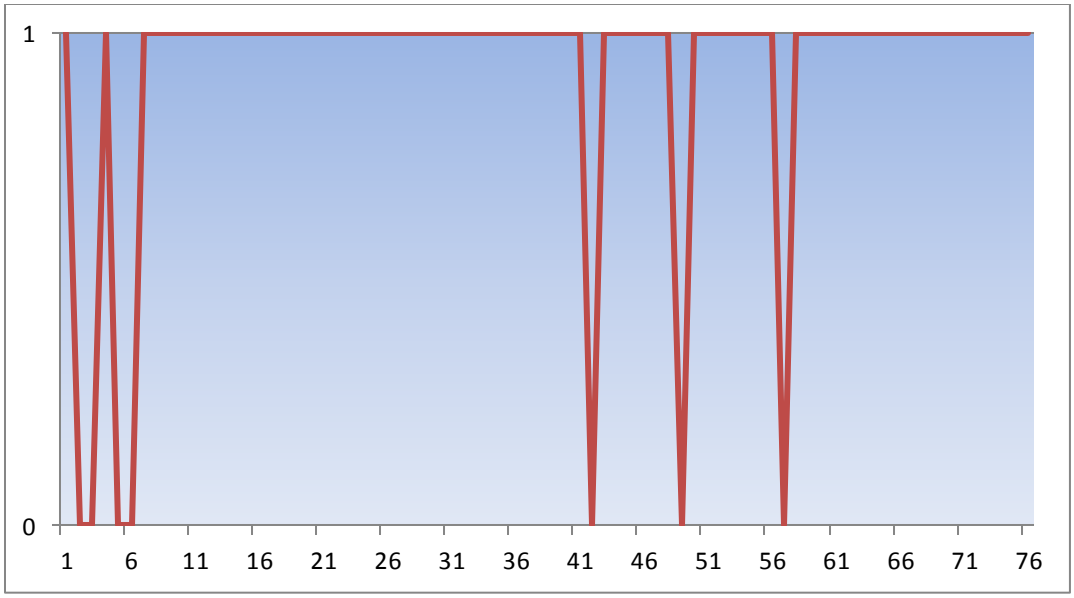

Figure 5-7 OCR accuracy of super resolution 2 - simple super resolution. Horizontal axis indicates frame sequences. Vertical axis indicates success and failure of OCR's recognition. Value 1 indicates success and value 0 indicates failure. From this figure, it can be seen that sequences at the beginning have a high possibility of failure. However, it becomes stable latter and able to recognize the right characters.

5.3 Performance analysis

Three super resolution algorithms take different time in a wide range. Obviously, the third Irani and Peleg's super resolution algorithm takes more time than the first and second super resolution algorithm. This experimental result is carried on the computer with 2.67 GHZ double processor.

Performance analysis

	Super resolution 1(Average)	Super resolution 2(Simple resolution)
Times (s)	0.001	0.001

Table 5-9 performance evaluation two basic super resolution algorithms only on merging information computation.

	Super resolution 3(Irani and Peleg) window size 1	Super resolution 3(Irani and Peleg) Window size 3	Super resolution 3(Irani and Peleg) Window size 5
Times(s)	0.09	5.8	13.2

Table 5-10 performance evaluation for Irani and Peleg algorithms in different window size only on merging information computation.

5.4 Summary

According to OCR accuracy result analysis and performance analysis, our second super resolution algorithm obtains better OCR accuracy 91.66% and with the best performance result. Although the third super resolution algorithm is able to achieve higher OCR accuracy, the cost of computation is also much higher than the first and the second super resolution algorithm. It takes more than a second to merge information when the window size is over than 1. In the consideration of balance the performance and accuracy we suggest the second super resolution algorithm if the accuracy of it is acceptable.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This project is able to figure out an appropriate computation flow to merge a series of frame sequences for the purpose of achieving two main aims rectification perspective distortion and improvement of image quality.

A basic perspective distortion rectification method is implemented which is based on the Hough transformation line detection and extra information of text boundaries. Three super resolution algorithms are also implemented to improve the quality of our result. We also evaluate the SIFT and SURF feature detection tools and tune out a set of suitable SIFT parameters specify to this project. SIFT feature detection is used to finding corresponding points which helps to compute homography matrix. We employ gold standard algorithm to compute this matrix and employ the residual error to filter out irrelevant frames which often contains more noise. Finally we obtain a perspective rectified and quality improved image to send to OCR.

6.2 Future work

Although we have achieved the goals of this project, improvement of it is still needed in the future.

- In chapter 4, we have described a complement algorithm to rectify no boundaries text image. But, that is a roughly algorithm rectifying the perspective dissertation with low accuracy. We need to find another way to deal with this kind of text scene more precise.
- We also need to investigate perspective distortion rectification method in deep, because our rectification method is largely depends on the boundaries of text scene, which is a limitation of this project.
- Performance improvement is another future work. We have mentioned in chapter 4.6 image registration. In image registration section we use Normalized Cross-Correlation (NCC) algorithm to align two images. But there is another histograms based algorithm which has better performance.

- Irani and Peleg's iterated method is not a sophisticated method, other machine learning algorithm like maximum likelihood algorithm might be better than the iterated method at least in the computation performance aspect. We need to investigate other machine learning algorithms. But most of them are based on kernel matrix and require solving large matrix construction problem.

6.3 Summary

The final section discussed main achievement and possible future works.

Bibliography

[1] A. Zramdini , R. Ingold, "Optical font recognition from projection profiles", Electronic Publishing, VOL. 6(3), 249–260, 1993

[2] B. Yu, A.K. Jain, A robust and fast skew detection algorithm for generic documents, Pattern Recognition 29 (10) (1996) 1599–1629.

[3] C. Harris, M. Stephens, "A combined corner and edge detector" Proceedings of the Alvey Vision Conference, pp. 147–151, 1988.

[4] C. Loop, Z. Zhang, "Computing Rectifying Homographies for Stereo Vision" Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Vol.1,pp 125-131,1999.

[5] D. Capel , A. Zisserman, "Automated Mosaicking with Super-resolution Zoom", In Proc. CVPR, pages 885-891,1998

[6] D. Lowe , "Distinctive Image Features from Scale-Invariant Key points", Int'l J. Computer Vision, vol. 2, no. 60, pp. 91-110, 2004.

[7] D.Liebowitz,A Zisserman "Metric Rectification for Perspective Images of Planes", In Proc. CVPR, 1998

[8] DA Forsyth, J Ponce, *Computer vision: a modern approach*, 2002.

[9] D. Kong, M. Han, W. Xu , H Tao, Y. Gong "Video Super-resolution with Scene-specific Priors", in BMVC, 2006

[10] E. Kavallieratou, N. Fakotakis, G. Kokkinakis, "Skew angle estimation for printed and handwritten documents using the Wigner Ville distribution", Image and Vision Computing 20 pages 813–824, 2002

[11] H. Bay, T. Tuytelaars, and L.V. Gool, "Surf: Speeded Up Robust Features," Proc. Ninth European Conf. Computer Vision, 2006.

[12] H.K. Kwag, S.H. Kim, S.H. Jeong, G.S. Lee, "Efficient skew estimation and correction algorithm for document images", Image and Vision Computing 20 (2002) 25–35.

[13] R Hartley, A Zisserman, Multiple view geometry, 2002.

[14] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors", in: CVPR, vol. 2, pp. 257–263, June 2003.

[15] K. Mikolajczyk, C. Schmid, " Scale and affine invariant interest point detectors", IJCV 60 (1), pp 63–86, 2004.

[16] K. Mikolajczyk, C. Schmid, "Indexing based on scale invariant interest points", Proc. ICCV, 2001.

[17] K. Kanatani, "High Accuracy Fundamental Matrix Computation and Its Performance Evaluation".

[18] Lyndsey C. Pickup and David P. Capel and Stephen J. Roberts Andrew Zisserman, "Bayesian image super-resolution, continued", Advances in Neural Information Processing Systems,pages 1089-1096, 2006

[19] L.G. Brown "A Survey of Image Registration Techniques", ACM Computing Surveys,VOL.24, pages 325-376,1992

[20] L. Juan, O. Gwon, "A comparison of SIFT, PCA-SIFT and SURF".

[21] L. Li, C. Lim Tan, "Recognizing Planar Symbols with Severe Perspective Deformation"

[22] M. Irani, S. Peleg, "Super Resolution from Image Sequences", 1 990

[23] M. Sonka, V. Hlavac and R. Boyle,*Image Processing, Analysis, and Machine Vision*,1999.

[24] P. Clark and M. Mirmehdi ,"Estimating the Orientation and Recovery of Text Planes in a Single Image", In Proc. 12th Conf. on British Machine Vision ,pages 421-430, 2001

[25] Schaffalitzky, F., Zisserman, "A.: Multi-view matching for unordered image sets", or"How do I organize my holiday snaps?"In: ECCV. Vol.1, pp 414-431, 2002.

[26] S. Lu, Ben M. Chen, C. C. Ko, "Perspective rectification of document images using fuzzy set and morphological operations", Image and Vision Computing,VOL. 23, 541-553,2005

[27] T. Tuytelaars Luc, L. Van Gool, M. Proesmans , T. Moons, "cascaded Hough transform as an aid in aerial image interpretation", in Proc. ICCV,pages 67-72, 1998

[28] V. Cantoni, L. Lombardi, M. Porta, N. Sicard, "Vanishing point detection:representation analysis and new approaches", Proc. Conf. International Conference on Image Analysis and Processing, 2001

[29] Y. Ke and R. Sukthankar, PCA-SIFT: *"A More Distinctive Representation for Local Image* Descriptors", Proc. Conf. Computer Vision and Pattern Recognition, pp. 511-517, 2004.