Image Matching Based On The Co-occurrence Matrix

Hsing-Wen Hseu, Abhir Bhalerao, Roland Wilson
Department of Computer Science,
University of Warwick,
Coventry CV4 7AL

March 17, 1999

In this work, the Gray Level Co-occurrence Matrices (GLCM) and cross-correlation are compared in an image matching process. First, three images are generated in different noise environments to simulate the images being derived from different sensors. Then, a Laplacian Pyramid is used as a high-pass filter. Finally, correlation of GLCM and cross-correlation are compared. Mutual information(MI) of GLCM is also included to evaluate the framework of the GLCM.

Contents

List of Figures

1 Introduction

Image matching has been a widely studied problem in identifying the similarity of the scene or objects in two images. It plays an essential role in processes such as texture analysis, stereo vision, 3D reconstruction, object recognition, motion analysis, image classification and image registration[4]. When the image matching has been carried out, common difficulties that arise are: operating images obtained from different modalities such as CT and MR, heavy computation, detecting motion like scaling and rotation. Although cross-correlation is successfully used in a single modality, it is easily defeated when using multiple modalities. In this case, mutual information has been shown to be more effective[1]. In order to find the best match, Viola et al. [10] designed a formulation to find the maximum of mutual information between a model and the image. Bro-Nielsen[1], in his paper, has claimed that mutual information is the best general feature. There are many other people[3][9] successfully applying mutual information to image matching. In addition to mutual information, moments and entropy are also important features in gray level co-occurrence matrix(GLCM), which could be used instead. By applying one of the measures derived from the GLCM, the best match can be found between two images. The Multiresolution Fourier Transform(MFT)[11] is a linear transformation providing local frequency estimation over multiple spatial resolutions which could help analysing the motion between two images[7]. This work, therefore, will give an assessment of computation cost to the image matching by comparing GLCM and cross-correlation method. There are four major steps in this work. Firstly, a Laplacian pyramid of a pair of images is constructed by using a low-pass filter. Secondly, the cross-correlation is computed between the pair of images and the peak value will be chosen to estimate the translation between the images. Thirdly, the gray level co-occurrence matrix is also applied to calculate the correlation of the pair of images and the peak value is also chosen as the translation between the pair of images. Finally, to compare results, we calculate the squared difference between actual value and results calculated from these two methods. Future work could employ the MFT and GLCM to find the geometric transformation between two views of the same scene or between objects in two images.

2 Toward Image Matching

A digital image can be viewed as a discrete 2-D lattice pixels whose row and column indices identify a point in the image and the corresponding value of the point identifies the intensity at that point. Let $f(i,j)$ and $g(i,j)$ be two images. The cross-correlation between them can be expressed by a function of displacement (m,n) as follows[5]

$$C_{fg}(m,n) = \frac{1}{MN}\sum_i\sum_j f(i,j)g(i-m,j-n) \tag{1}$$

where M and N are the total samples of one row and one column. Eq. (1) can also be viewed as the sample average estimate of the two images, viewed as samples of a random process. If we have two random variables(RV's), X and Y, say, the correlation between them is defined as

$$R_{XY} = E[XY] = \int\int p(x,y)xydxdy \tag{2}$$

where $p(x,y)$ is the joint density. If we have a set of M samples from the joint process, X_i and Y_i, say, the sample correlation is defined as

$$C_{XY} = \frac{1}{M}\sum_{i=1}^{M} X_iY_i \tag{3}$$

Thus if Y_i represents the shifted image $\{g(i-m,j-n)\}$ and X_i represents the image $\{f(i,j)\}$, we can get a formal equivalence between Eq. (1) and Eq. (3): $C_{fg}(m,n)$ is the sample cross-correlation between f and g, viewed as samples from a random process. A histogram is a function which returns the relative frequency of a particular gray level in a given image and is useful when applying a statistical method. Now suppose we have a joint histogram $h_{XY}(k,l)$ for X and Y, based on the samples X_i and Y_i . Then Eq. (3) can be written as

$$C_{XY} = \sum_{k,l} h_{XY}(k,l)kl \tag{4}$$

Other measures of dependence between X and Y include mutual information(MI)[5].

$$I(X,Y) = E[i(x,y)] = \int\int p_{XY}(x,y)log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}dxdy \tag{5}$$

with sample estimate

$$I_{XY} = \sum_{k,l} h_{XY}(k,l)log\frac{h_{XY}(k,l)}{h_X(k)h_Y(l)} \tag{6}$$

(0,0)	(1,0)	(2,0)	$T_i = \{0,1,2\}$; $T_j = \{0,1,2\}$
(0,1)	(1,1)	(2,1)	$T_{i'} = \{i+1 \vert i+1 \in T_i\} = \{1,2\}$
(0,2)	(1,2)	(2,2)	$T_{j'} = \{j \vert j \in T_j\} = \{0,1,2\}$

$$\begin{aligned} S = \; & \{((i,j),(i',j')) \in (T_i \times T_j) \times (T_{i'} \times T_{j'}) | i' = i+1, j' = j\} \\ = \; & \{((0,0),(1,0)) \;\; ((1,0),(2,0)) \;\; ((0,1),(1,1)) \\ & ((1,1),(2,1)) \;\; ((0,2),(1,2)) \;\; ((1,2),(2,2))\} \end{aligned}$$

Table 1: The set of all pairs of corresponding indices of two 3×3 images related by a displacement (1,0)

where

$$\begin{aligned} h_X(k) &= \sum_l h_{XY}(k,l) \\ h_Y(l) &= \sum_k h_{XY}(k,l) \end{aligned}$$

are the marginal histograms for X and Y. k and l are intensities as are X and Y.

2.1 Co-occurrence matrices

Gray Level Co-occurrence Matrices (GLCM) is a method for estimating spatial gray level dependence[6]. Given a displacement between two images, a sample space XY is defined and the total number of pairs of pixels within it having the same pair of gray levels, (r,s), are accumulated to generate a joint histogram, $h_{XY}(r,s)$. Table 1 illustrates the set of all pairs of corresponding indices of two 3×3 images related by a displacement (1,0). The joint histogram of the example can be expressed as,

$$h_{XY}(r,s) = \frac{N_{r,s}}{N_S} \tag{7}$$

where $N_{r,s}$ is the total number of pairs of corresponding indices in S (as given in Table 1)where X gives a gray level value r to the first pair of indices and Y gives a gray level value s to the second pair of indices. N_S is the total number of pairs of pixels in S. Since the joint histogram can be represented in a matrix form, it is called a co-occurrence matrix. Thus GLCM can be used to determine the correspondence between a stack of images or registration of a pair of images. The joint histogram estimated from GLCM can also be used to generate many measures such as correlation, and mutual information[6].

2.2 Pyramid Structures in Mutiresolution methods

Multiresolution methods are widely used in image processing and analysis[8]. The structures, in general, represent the image across a number of different resolutions: from coarse-to-fine. There are links between consecutive levels of the structure providing the possibility of reducing computational cost of various image operations using a divide-and-conquer methodology[8]. The Gaussian pyramid is an important example of multiresolution structure. It consists of the input image at the bottom of the stack and each level obtained by applying a lowpass-filter to the one beneath. The Laplacian pyramid is obtained by taking the difference between consecutive levels of a Gaussian pyramid. The difference-of-Gaussians resemble the Laplacian operators commonly used in image processing to enhance image features such as edges. A low-pass filter is a mask used to smooth the intensities of an image. Thus the differences between the intensities of an image and those of the smoothed image can be calculated. For a Laplacian image the average difference should be zero. Thus the main reasons to use Laplacian pyramids in this work are as follows

- detecting a displacement at a level which is a fraction of original images.
- zero mean intensity at every level.
- robust in the presence of noise.

These give a chance to optimise the computations.

3 Experiments

We select the fourth level of the Laplacian Pyramid as an initial matching level in this work. At this level, 2 pixels shift is equivalent to 32 pixels shift in the original image. We use an image of size 256x256, Lena, to create a set of images, $L = \{L_0, L_1, L_2, L_3\}$, in which each one of them is 32 pixels shift of the other one shown in Fig 1. A 24 pixel shift case is also included in this work. We wish to examine

1. The comparison between cross-correlation and GLCM-based methods.
2. The effects of noise on the performance of the methods.

Figure 1: 32 pixels horizontal or vertical shift images

In order to carry out the experiment five other sets of images are needed as follows

1. a set of images, $I = \{I_0, I_1, I_2, I_3\}$, by adding Gaussian noise with variance: 80.6381(PSNR: 10db) to the original image and applying 32 pixels shift.

2. a set of images, $J = \{J_0, J_1, J_2, J_3\}$, by adding Gaussian noise with variance: 25.5000(PSNR: 20db) to the original image and applying 32 pixels shift.

3. a set of images, $K = \{K_0, K_1, K_2, K_3\}$, by applying 24 pixels shift.

4. a set of images, $M = \{M_0, M_1, M_2, M_3\}$, by adding Gaussian noise with variance: 80.6381(PSNR: 10db) to the original image and applying 24 pixels shift.

5. a set of images, $N = \{N_0, N_1, N_2, N_3\}$, by adding Gaussian noise with variance: 25.5000(PSNR: 20db) to the original image and applying 24 pixels shift.

3.1 The Laplacian Pyramid

A pyramid of an image is the structure where each layer or level is half the resolution of the one below. For an N level Gaussian Pyramid, the upward function, REDUCE,

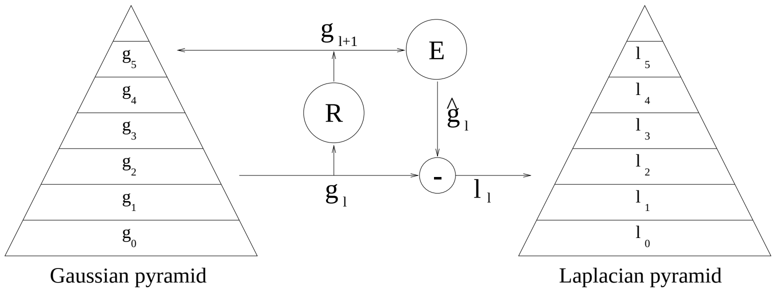


Figure 2: Relationship between Gaussian and Laplacian Pyramid

of a level can be expressed as

$$g_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n)\, g_{l-1}(2i+m, 2j+n) \tag{8}$$

Where l is the number of a level in the pyramid range $0 < l < N-1$ and $g(i,j)$ is the intensity of position (i,j) at level l of the pyramid. $w(m,n)$ is a 2D weighting kernel generated by an 1D Gaussian function $w(m) * w(n)$. By using the same weighting kernel the downward function, EXPAND, of two consecutive levels can be expressed as follows

$$\hat{g}_l(i,j) \;=\; 4 \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n)\, g_{l+1}\left(\frac{i+m}{2} \,,\, \frac{j+n}{2} \right) \tag{9}$$

where g_l gives a value only when $i+m$ and $j+n$ are even numbers. The relationship between these two functions are described as Figure 2. Because a Gaussian-like kernel is a low-pass filter, the difference of the two functions can be viewed as a high-pass image at each corresponding level. A Laplacian pyramid is a sequence of the high-pass images which can be expressed as

$$l_l(i,j) \;=\; g_l(i,j) \;-\; \hat{g}_l(i,j) \tag{10}$$

In this work, we use $w(.)$ =(0.005, 0.25, 0.4, 0.25, 0.005) as the Gaussian-like kernel[2]. The Gaussian and Laplacian pyramid of the image Lena is shown in Figure 3. For each prepared image set such as L, we can generate a corresponding Laplacian image set, $L^4 = \{L_0^4, L_1^4, L_2^4, L_3^4\}$.

3.2 Correlation

The experiment can be arranged as in Fig 4. In the 32 pixels case, X_0 is L_0 and Y is one of the Laplacian image sets L, I, or J. In the 24 pixel shift case, X_0 is K_0 and Y is one of the Laplacian image sets K, M, or N.

(a) The Gaussian pyramid

(b) The Laplacian pyramid

Figure 3: Gaussian and Laplacian Pyramid of Lena

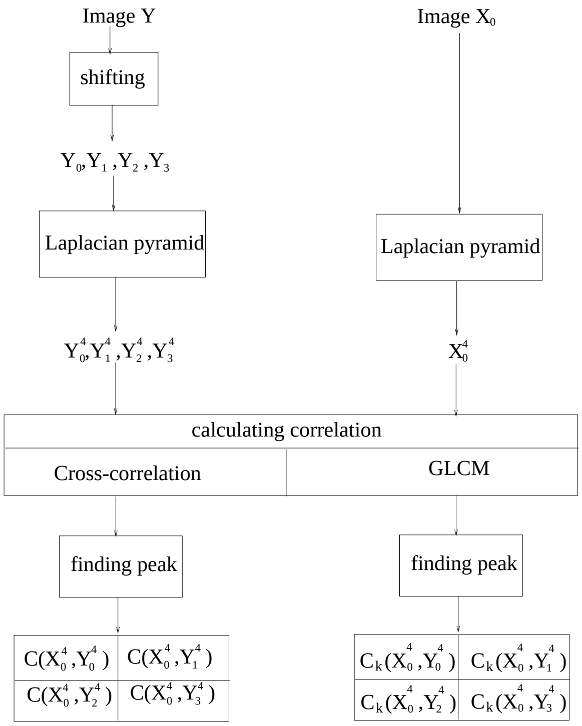


Figure 4: Arrangement of experiments

$(0,0)$	$(2,0)$
$(0,2)$	$(2,2)$

(a)

$(0,0)$	$(1.5,0)$
$(0,1.5)$	$(1.5,1.5)$

(b)

$(0,0)$	$(2,0)$
$(0,2)$	$(2,2)$

(c)

$(0,0)$	$(2,0)$
$(0,2)$	$(2,2)$

(d)

$(0,0)$	$(2,0)$
$(0,2)$	$(2,1)$

(e)

$(2,1)$	$(4,1)$
$(2,3)$	$(4,3)$

(f)

Table 2: (a) Actual displacement of 32 pixels. (b) Actual displacement of 24 pixels. (c) Actual displacement of reverse intensities and 32 pixels. (d) Cross-correlation result of L^4 , I^4 and J^4. (e) Cross-correlation result of K^4 , M^4 and N^4. (f) cross-correlation result of reverse intensities and 32 pixels.

$r(0,0)$	$r(0,1)$
$r(1,0)$	$r(1,1)$

$C_k(X_0^4, Y_0^4)$	$C_k(X_0^4, Y_1^4)$
$C_k(X_1^4, Y_0^4)$	$C_k(X_1^4, Y_1^4)$

$$square\ \ difference(k) = \sqrt{\sum_{i=0}^{i=1}\sum_{j=0}^{j=1}(r(i,j) - C_k(X_i^4, Y_j^4))^2}$$

Table 3: square differences of actual displacement and correlation of GLCM at a bin width k

In GLCM, the bin width is the range of intensities in an image and the maximum correlation can be generated for every bin width. Thus we will have maximum correlation in each bin width.

After the correlation has been computed, the square difference can be defined as shown in Table 3 where the left table is the actual displacement and the right table is generated by the cross-correlation or correlation of GLCM at a bin width k. The square difference is the tool with which to analyse the results of this experiment.

By applying a DFT, the cross-correlation can be computed efficiently in frequency domain and transfered back to spatial domain to find the best match as follows[5]

$$C_{fg}(m,n) = \sum_i \sum_j f(i,j)g(i-m, j-n) <=> F(u,v)G^*(u,v) \tag{11}$$

where $F(u,v)$ is the Fourier transform of $f(i,j)$ and $G^*(u,v)$ is the complex conjugate of Fourier transform of $g(i,j)$. The actual displacements and results of applying the cross-correlation method is shown in Table 2(a)(b)(d)(e). The result show us the noise(PSNR=10db and 20db) has no effect on the method and it can find the best displacement at this level.

The bin width is an odd number in this experiment so that we can take the mean value to be the centre. We take an observation range of twice the standard deviation of the original image, σ, from the mean value of the original image, μ, on each side $[\mu-2\sigma, \mu+2\sigma]$. For example, given a bin width 3, we can build a histogram from the gray level value 0 in which the intensity ranges from $[\mu-1,\mu+1]$. On the left the gray level values decrease by 1 and on the right the gray level values increase by 1. Once the relationship between the gray level values and intensities are built, we assign a gray level value to every pixel in both images. Let $\hat{f}(i,j)$ and $\hat{g}(i,j)$ be the two new images and can be viewed as two RV's $\hat{X}$ and $\hat{Y}$. If the two RV's are related by a displacement (m,n), the joint histogram can be calculated by introducing a valid region function $V_{\hat{X}\hat{Y}}(i,j)$. The valid region function can be defined as

$$V_{\hat{X}\hat{Y}}(i,j) = \begin{cases} 1 & if\ (0,0) \le (i+m, j+n) \le (15,15). \\ 0 & else\ where. \end{cases}$$

For example, given a displacement $(-4,-4)$, $V_{\hat{X}\hat{Y}}(i,j)$ maps a valid region range from $(4,4)$ to $(16,16)$ in the first image to the region range from $(0,0)$ to $(11,11)$ in the second image. Thus the valid region size can be expressed as

$$\begin{aligned} R_{\hat{X}\hat{Y}} &= \textstyle\sum_{i=0}^{16}\sum_{j=0}^{16} V_{\hat{X}\hat{Y}}(i,j) \\ &= (16-|m|)(16-|n|) \end{aligned} \tag{12}$$

After a valid region size is defined, we can count the frequency of a gray level value in one image to a gray level value in another image. The counter function can be defined as

$$N_{\hat{X}\hat{Y}}(r,s) = \sum_{i=0}^{16}\sum_{j=0}^{16} U_{\hat{X}\hat{Y}}(r,s|V_{\hat{X}\hat{Y}}(i,j)) \tag{13}$$

where

$$U_{\hat{X}\hat{Y}}(r,s|V_{\hat{X}\hat{Y}}(i,j)) = \begin{cases} 1 & if\ V_{\hat{X}\hat{Y}}(i,j) = 1 \\ & \quad and\ \hat{f}(i,j) = r \\ & \quad and\ \hat{g}(i+m, j+n) = s. \\ 0 & else\ where. \end{cases}$$

Then the joint histogram matrices can be defined as

$$h_{\hat{X}\hat{Y}}(r,s) = \frac{N_{\hat{X}\hat{Y}}(r,s)}{R_{\hat{X}\hat{Y}}} \tag{14}$$

Because the displacements is ranged from (-4,-4) to (4,4) and each displacement gives a joint histogram matrix, we will have 81 matrices in this work. The correlation of a displacement (m, n) can be calculated as follows

$$C_{\hat{X}\hat{Y}} = \sum_r \sum_s r\ s\ h_{\hat{X}\hat{Y}}(r, s) \tag{15}$$

The first experiment is to compare the correlation results in different noise environment with the actual displacement. The results are described as

1. The square differences of actual displacement and correlation of GLCM in 32 pixels shift case are all zeros in every bin width except 195 where only one bin is used to measure the correlation.

2. Fig 5 is 24 pixels shift case shown us that the results are all the pixels at the fourth level which are the closest to the actual result at every bin width except the ranges $[141 \sim 171, 177 \sim 185, 195]$.

In order to observe the effect of the noise, we replace the actual displacements in Table 3 with the correlation of GLCM. Then we compute the square differences between the correlation of GLCM where PSNR=10db and PSNR=20db. The results of the second experiment are

1. The noise has no effect on 32 pixels shift case.

2. The effect on 24 pixels shift case is at bin width $[71, 97, 119, 141, 157, 159, 171, 183]$ shown as Fig 6.

3.3 Mutual Information

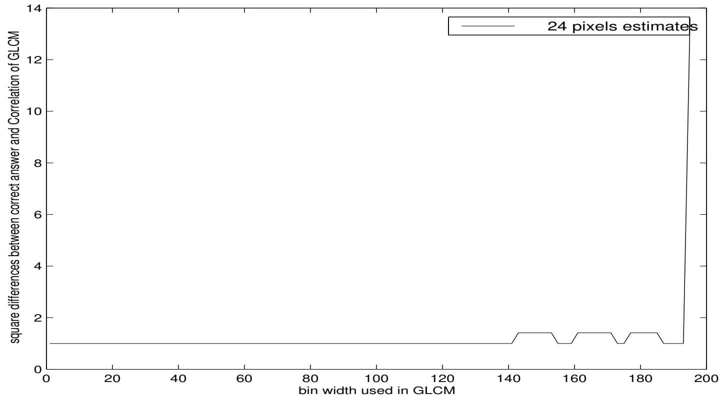
MI of GLCM is defined by the probability matrices obtained from the previous section. The equation can be expressed as

$$I(m, n) = \sum_r \sum_s h_{\hat{X}\hat{Y}}(r, s)\ log\ \frac{h_{\hat{X}\hat{Y}}(r, s)}{h_{\hat{X}}(r)\ h_{\hat{Y}}(s)} \tag{16}$$

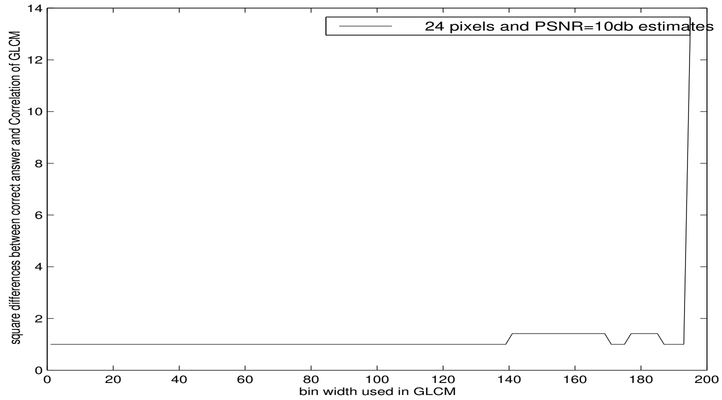
We can do an experiment as in the previous section by replacing the correlation of GLCM with the MI of GLCM. Results of the first experiment are described as

1. In 32 pixels shift case, the results are as good as those of correlation except the bin width $[1, 3]$.

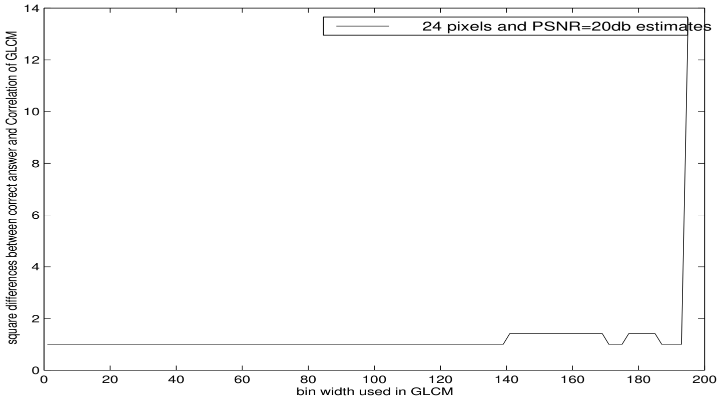
2. The results shown in Fig 7 are also as good as those of correlation except the bin width $[1 \sim 13, 137]$. Fig 8 show us that the effect happened at the bin width $[1 \sim 17, 43, 55, 71, 97, 101, 103, 111, 119, 137, 195]$.



(a) Square difference results of K_0^4 and K^4 (PSNR=0db) at different bin widths

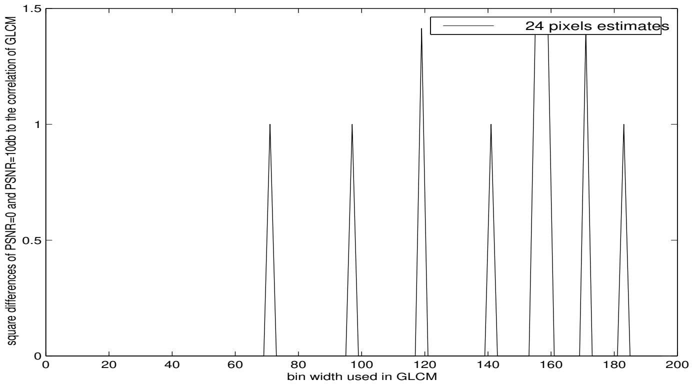


(b) Square difference results of K_0^4 and M^4 (PSNR=10db) at different bin widths

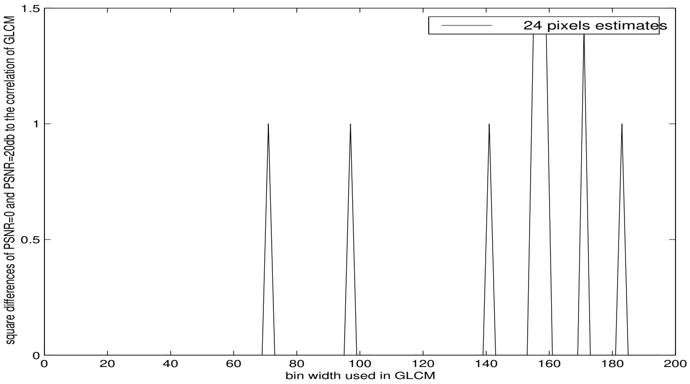


(c) Square difference results of K_0^4 and N^4 (PSNR=20db) at different bin widths

Figure 5: Square Differences of actual displacement and correlation of GLCM at different bin widths for 24 pixels shift



(a) Square difference results of K_0^4 and K^4 (PSNR=0db) and of K_0^4 and M^4(PSNR=10db)



(b) Square difference results of K_0^4 and M^4(PSNR=10db) and of K_0^4 and N^4 (PSNR=20db)

Figure 6: Noise effect on the correlation of GLCM in 24 pixels shift

	32 pixels shift	*24 pixels shift*
correlation of GLCM	1-193	1-141, 173, 175,187-193
MI of GLCM	5-193	17-135, 139,141, 169-175,187-193

Table 4: bin widths with the best square differences at the fourth level

Actual Displacement	*Cross-correlation*	*Correlation of GLCM*	*MI of GLCM*
(0,0)	(0,0)	(0,0)	(0,0)
(0,2)	(0,2)	(0,2)	(0,2)
(2,0)	(2,0)	(2,0)	(2,0)
(2,2)	(2,2)	(2,2)	(2,2)
(0,0)	(0,0)	(0,0)	(0,0)
(0,1.5)	(0,2)	(0,2)	(0,2)
(1.5,0)	(2,0)	(2,0)	(2,0)
(1.5,1.5)	(2,1)	(2,1)	(1,2)

Table 5: The best results of the three methods

Results of the second experiment are described as

1. The noise has no effect on 32 pixels shift case.

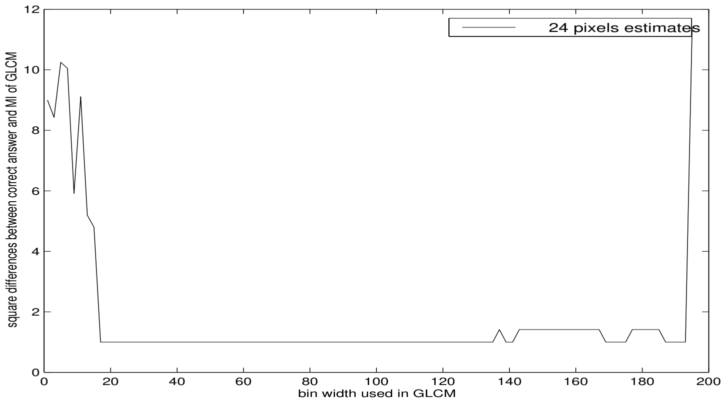
2. The effect on 24 pixels shift case is at bin width $[1 \sim 17, 43, 55, 71, 97, 101, 103, 111, 119, 137, 195]$ shown as Fig 8.

The reversed Laplacian images are also used in this section. The cross-correlation gives an incorrect result shown in Table 2(e). However, the MI of GLCM works well in this case shown in Fig 9.

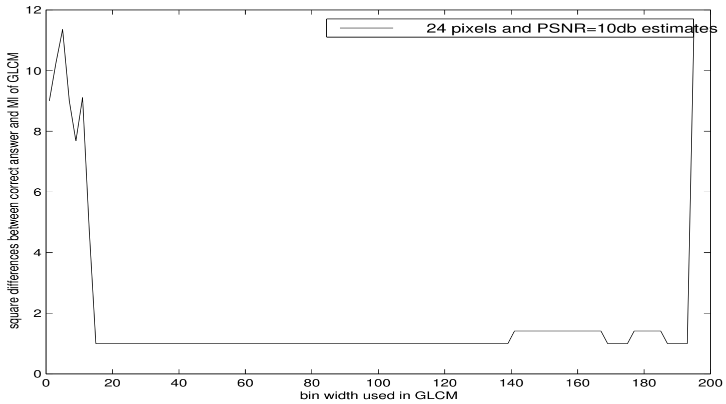
3.4 Computation

From the results in section 3.1 ,3.2 and 3.3, we can collect the bin widths with zero square differences for 32 pixels and the square difference value lower than or equal to one for 24 pixels. Acceptable bin widths for the Lena image is shown in Table 5. This means that the acceptable bin width is quite large and the results of the three methods shown in Table 4 are the closest pixels to the actual displacement. The bigger the bin width we select, the more computation time we can save, so we select the bin width range $[187 \sim 193]$.

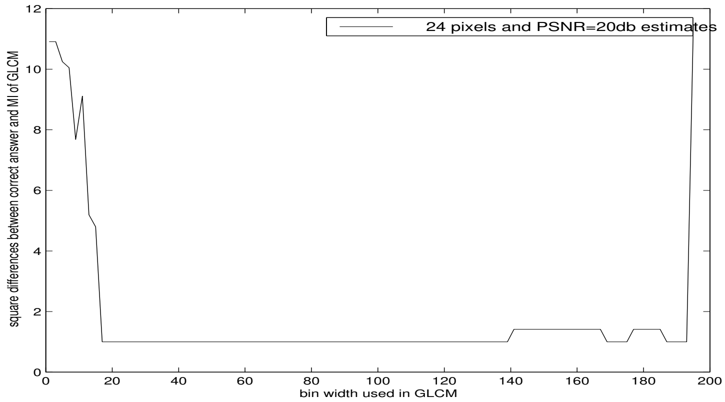
The computation times of the two methods are shown as Table 6. In this work,



(a) Square difference results of K_0^4 and K^4 (PSNR=0db) at different bin widths

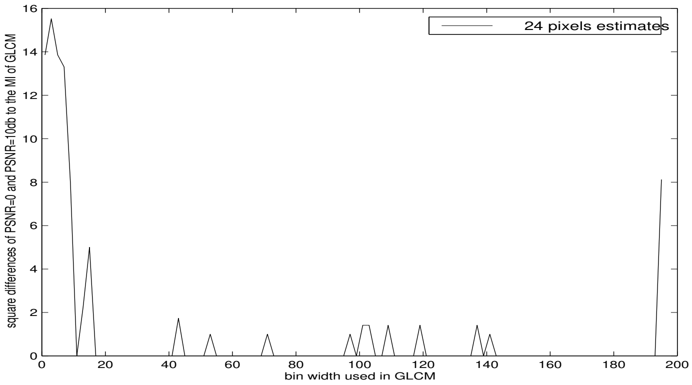


(b) Square difference results of K_0^4 and M^4 (PSNR=10db) at different bin widths

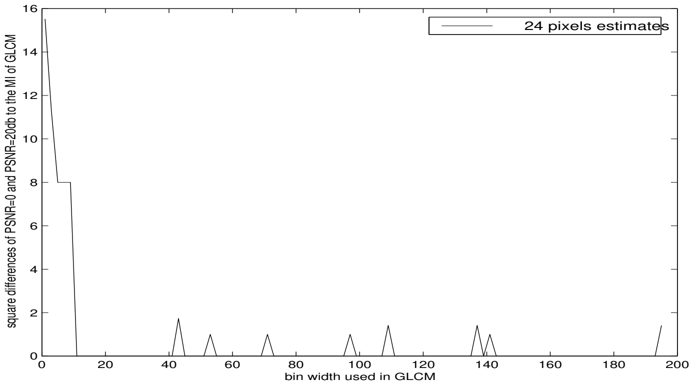


(c) Square difference results of K_0^4 and N^4 (PSNR=20db) at different bin widths

Figure 7: Square Differences of cross-correlation and MI of GLCM at different bin widths in 24 pixels shift case



(a) Square difference results of K_0^4 and K^4 (PSNR=0db) and of K_0^4 and M^4(PSNR=10db)



(b) Square difference results of K_0^4 and M^4(PSNR=10db) and of K_0^4 and N^4(PSNR=20db)

Figure 8: Noise effect on the MI of GLCM in 24 pixels shift

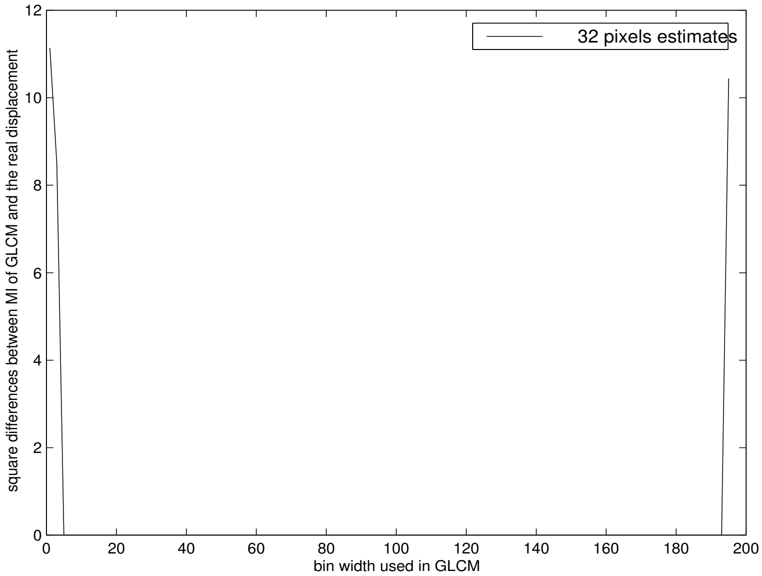


Figure 9: Square difference between actual displacement and MI of GLCM in negative and 32 pixels shift case

Item	*Comparison*	*Add/Subtract*	*Multiply/Divide*
FFT		$2(\frac{N}{2^k}log_2\frac{N}{2^k})$	$2(\frac{1}{2}\frac{N}{2^k}log_2\frac{N}{2^k})$
Correlation		$2(2\frac{N}{2^k}\frac{N}{2^k})$	
IFT		$2(\frac{N}{2^k}log_2\frac{N}{2^k})$	$2(\frac{1}{2}\frac{N}{2^k}log_2\frac{N}{2^k})$
Peak Finding	$\frac{N}{2^k}\frac{N}{2^k}$		

(a) Computations of cross-correlation

Item	*Comparison*	*Add/Subtract*	*Multiply/Divide*
standard deviation (σ)		$2(\frac{N}{2^k}\frac{N}{2^k}-1)$	$2(\frac{N}{2^k}\frac{N}{2^k}+1)$
gray Level Assignment	$\frac{N}{2^k}\frac{N}{2^k}\lceil\frac{2\sigma}{w}\rceil$	$\lceil\frac{2\sigma}{w}\rceil$	2(3)
Co-occurrence matrices	$2(2)(2d+1)^2$	$2(4)(2d+1)^2+2(3 \sum_{d_x}\sum_{d_y}(\frac{N}{2^k}-\lvert d_x\rvert)(\frac{N}{2^k}-\lvert d_x\rvert)+\sum_{d_x}(\frac{N}{2^k}-\lvert d_x\rvert))$	
Correlation		$2[(2\lceil\frac{2\sigma}{w}\rceil-1)(2\lceil\frac{2\sigma}{w}\rceil-1)-1](2d+1)^2$	$2(2\lceil\frac{2\sigma}{w}\rceil-1)(2\lceil\frac{2\sigma}{w}\rceil-1)2(2d+1)^2$
Peak Finding	$(2\lceil\frac{2\sigma}{w}\rceil-1)(2\lceil\frac{2\sigma}{w}\rceil-1)(2d+1)^2$		

(b) Computations of correlation of GLCM

w is the selected bin width.
σ is the standard deviation of the original image.
d is the total number of possible displacement.
d_x and d_y are the displacement in x axis and y axis.

Table 6: Computation comparison of two methods

Item	*Comparison*	*Add/Subtract*	*Multiply/Divide*	*Total*
FFT		128	64	192
Correlation		1024		1024
IFT		128	64	192
Peak Finding	256			256
Total	256	1280	128	1664

(a) Computations of cross-correlation

Item	*Comparison*	*Add/Subtract*	*Multiply/Divide*	*Total*
standard deviation (σ)		510	514	1024
gray Level Assign	512	2	6	520
Co-occurrence matrices	324	47024		47348
Correlation		1296	2916	4212
Peak Finding	729			729
Total	1565	48832	3436	52833

(b) Computations of correlation of GLCM

Table 7: Computation comparison of two methods in this experiment for image size 256x256

we take the image, Lena, as the original image and take the Laplacian pyramid at the fourth level, $k = 4$. At this level we take the ratio of 2 times standard deviation to bin width, $\lceil 2\sigma/w \rceil = 1$. In this case $2\sigma = 98$ and bin width is 193 which we found above and the displacement range is from -4 to 4. We can then complete the comparison shown as Table 7. The computational complexity of correlation of GLCM is almost 32 times that of cross-correlation. Note also that the correlation of GLCM can only detect the best match range from (-4, -4) to (4, 4).

4 Conclusions and further work

From the experiments we know the image matching can be done by using these two different methods. Because the Laplacian image is robust in the presence of noise, the cross-correlation is not affected by the noise and GLCM is also not affected by the noise. The comparison of computational complexity shows that the GLCM are not as efficient as cross-correlation and the matching is restricted by the displacements set in this experiment. However, the cross-correlation cannot be applied to image matching where one image is negative to the other one while the MI of GLCM is a good choice to solve the problem. GLCM is a robust framework for solving this problem, but the performance needs to be improved. Further work is to optimise the computation and apply GLCM to detect a rotation and scaling.

References

[1] Morten Bro-Nielsen. Rigid Registration of CT, MR and Cryosection Images Using a GLCM Framework. *Proc. CVRMed-MRCAS'97*, vol. 1205:171–180, 1997.

[2] P. J Burt and E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communication*, COM-31:532–540, 1983.

[3] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated Multi-modality Image Registration Based on Information Theory. *Proc. Information Processing in Medical Imaging*, pages 263–274, 1995.

[4] Sébastien Gilles. Description and Experimentation of Image Matching Using Mutual Information. Technical report, Department of Engineering Science, Oxford University, UK, 1996.

[5] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley Publishing Co., 1992.

[6] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision.* Addison-Wesley Publishing Co., 1992.

[7] T.I. Hsu. *Texture Analysis and Synthesis Using the Multiresolution Fourier Transform.* PhD thesis, Department of Computer Science, The University of Warwick, UK, 1994.

[8] A. Rosenfeld(ed.). *Multiresolution Image Processing and Analysis.* Springer-Verlag, 1984.

[9] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Automated 3-D Registration of MR and CT Images of the Head. *Medical Image Analysis*, vol. 1(2):163–175, 1996.

[10] Paul Viola and Willam M. Wells III. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[11] R. Wilson, Calway A, and E.R.S. Pearson. A Generalised Wavelet Transform for Fourier Analysis: The Multiresolution Fourier Transform and Its Application to Image and Audio Signal Analysis. *IEEE Transaction on Information Theory*, 38(2):674–690, 1992.