

# Fast Normalized Cross Correlation for Motion Tracking using Basis Functions

A. J. H. Hii<sup>1</sup>, C. E. Hann<sup>2</sup>, J. G. Chase<sup>3</sup>, E. E. W. Van Houten<sup>4</sup>

Department of Mechanical Engineering  
University of Canterbury  
Private Bag 4800  
Christchurch  
New Zealand

Email: [Chris.Hann@canterbury.ac.nz](mailto:Chris.Hann@canterbury.ac.nz)

---

<sup>1</sup> Research Assistant, Dept. of Mech. Eng, Centre for Bio-Engineering

<sup>2</sup> Research Associate, New Zealand Science and Technology Postdoctoral Fellow, Dept. of Mech. Eng,  
Centre for Bio-Engineering

<sup>3</sup> Assoc. Prof./ Reader, Dept. of Mech. Eng, Centre for Bio-Engineering

<sup>4</sup> Senior Lecturer, Dept. of Mech Eng, Centre for Bio-Engineering

## **ABSTRACT**

Digital Image-based Elasto-tomography (DIET) is an emerging method for non-invasive breast cancer screening. Effective clinical application of the DIET system requires highly accurate motion tracking of the surface of an actuated breast with minimal computation. Normalized cross correlation (NCC) is the most robust correlation measure for determining similarity between points in two or more images providing an accurate foundation for motion tracking. However, even using fast fourier transform (FFT) methods, it is too computationally intense for rapidly managing several large images. A significantly faster method of calculating the NCC is presented that uses rectangular approximations in place of randomly placed landmark points or the natural marks on the breast. These approximations serve as an optimal set of basis functions that are automatically detected, dramatically reducing computational requirements. To prove the concept, the method is shown to be 37-150 times faster than the FFT-based NCC with the same accuracy for simulated data, a visco-elastic breast phantom experiment and human skin. Clinically, this approach enables thousands of randomly placed points to be rapidly and accurately tracked providing high resolution for the DIET system.

*Keywords*—Breast Cancer Screening, Surface Motion Tracking, Normalized Cross Correlation, Landmark points, Basis functions

## **1. Introduction**

Digital Image-based Elasto-Tomography (DIET) is an emerging technology for breast cancer screening [1]. The stiffness of breast tissue can be reconstructed from measured 3D surface motion of a sinusoidally actuated breast using multiple high resolution digital cameras. This approach is similar to full volume elastographic methods using MRI [2-4] or ultra-sound [5]. Areas of high stiffness suggest cancerous tissue as it is 3-10 times stiffer than healthy tissue types [6, 7].

To measure useful 3D surface motion requires accurate motion tracking of a large number of randomly placed landmark points on the breast, or direct tracking of the natural marks and tone on the breast. In addition, clinical application for several

hundred or thousand points will require minimal computation to ensure clinical effectiveness. Hence, the DIET concept requires highly accurate motion tracking with minimal computation to be practicable.

The standard way of tracking features between two images is by template matching. This approach involves taking a given pattern in one image and shifting a template containing the same pattern in another image until the best comparison is found. The most common and effective way of doing this task is by Normalized Cross Correlation (NCC) methods, which have a significant advantage over standard cross correlation (CC) methods, in that these methods are robust to different lighting conditions across an image and less sensitive to noise [8]. However, both methods can be computationally intense, especially for large images.

There are other methods of tracking that do not use NCC, including Gradient Descent Search (GDS) and Active Contour Matching (“Snakes”). The GDS [9] is based on a first order approximation to image motion and has the restriction that feature translation is small and the inter-frame translation must be less than the radius of the basin surrounding the minimum of the matching error surface. In the DIET system, there will be large numbers of landmark points to track, which are all close together [1]. Thus, any significant local perturbation on the surface due to a tumour [10] that causes a sudden local increase in amplitude relative to other parts of the surface could cause an error in the motion measurement predicted by GDS, since GDS only uses a first order approximation to motion. Furthermore, GDS methods only require one occasion where a particular landmark point jumps to another landmark point nearby for the whole trajectory to be corrupted. Another drawback for GDS methods is that the image gradient is required, which like any numerical derivative is sensitive to noise. Thus, to compute a reliable estimate of the gradient, the image must be smoothed, which depending on the number of smoothings applied, can distort the image introducing further potential error.

The snake method [11] tracks individual feature contours, but in a similar way to GDS, is restricted to small changes in the contour’s shape and displacement. Furthermore, the snake method is sensitive to any intrinsic fuzziness or varying lighting conditions, as the image features must have clearly defined boundaries. In the DIET system, some landmark points must be quite close together so any variation in lighting conditions could cause a contour to be placed around one point in one image and two points in another image, thus corrupting results. An improvement to the

robustness of snake tracking has been proposed [12], however the method tracks macro scale single contours like the shape of a hand. The DIET system would need thousands of contours to be tracked, which would require significantly large computation. Another feature based method has been looked at involving thin plate splines [13], however the method is very computationally heavy.

An empirical study of five template matching algorithms in the presence of various image distortions [14] found that NCC provides the best performance in all image categories. Also the choice of the correlation coefficient over alternative matching criteria, such as the sum of absolute differences, has also been justified as a maximum-likelihood estimation [15]. Thus, the NCC approach is potentially most suitable for the DIET system, which requires highly robust and accurate tissue surface motion tracking for a large number of closely located points.

Currently, a relatively efficient way of calculating the NCC is by using the Fast Fourier Transform (FFT) to compute the standard CC and then using sum-tables to perform the normalization [16]. The method of [16] has also been applied for defect detection [17, 18]. However, it was shown in [19] that if basis functions are used to approximate the template then substantial computational gains could be obtained over the FFT-based methods.

However, the method of [19] relies on choosing a suitable set of basis functions to approximate the template. The process of choosing the best set of basis functions is non-trivial with no guarantee of finding the optimal solution in terms of the best approximation with the fewest basis functions. Additionally, a threshold value that describes how close the basis functions should approximate the template image has to be chosen and impacts accuracy. In the case of tracking breast motion, there is no guarantee that this threshold value will stay constant as many complex surface motions can be produced if a tumour is present [1, 10].

The concept in this paper is to let every randomly placed landmark point on the breast correspond to a basis function. The problem is then reformulated in terms of tracking the landmarks. Thus, the basis function representation in this case is guaranteed to be the optimal and there is physical control over the number and distribution of landmark points that are placed on the breast. An automatic method of calculating the specific basis functions is also presented. The method is tested on simulated motion, as well as human skin motion to prove the concept. Note that unlike the methods of [19], the pre-calculation of sum-tables are included in the time

taken for computations. Substantial computational speed gains are obtained over the NCC method of [16] while maintaining the same high accuracy.

## 2. Methodology

### 2.1 Standard NCC Method and Motion Tracking

The NCC method is a simple template-matching method that determines the location of a desired pattern represented by a template function,  $t$ , inside a two-dimensional image function,  $f$ . The template is shifted pixel-by-pixel across the image, forming a correlation plane that provides information of where the template best matches the image. In correlation based motion tracking applications, a pair of time-dependent images  $\text{Im}_{t_0}$  and  $\text{Im}_{t_0+\Delta t}$  are compared in a pixel-by-pixel basis.

For example, consider a  $4 \times 4$  sub-image of  $\text{Im}_{t_0+\Delta t}$  denoted by  $W(x, y)$  and a  $2 \times 2$  feature template  $T(x, y)$  of  $\text{Im}_{t_0}$  which is contained in  $W(x, y)$ . The template  $T(x, y)$  is shifted into nine different positions, where at each position, intensities are multiplied and summed, producing a correlation coefficient matrix,  $\gamma_{u,v}$ , as shown in Figure 1. The pixel location  $(u, v)$  corresponding to the maximum NCC  $\gamma_{\max}$  value corresponds to the best location of the template feature in the sub-image  $W(x, y)$ . This process is continued over all sub-images  $W(x, y)$  contained in  $\text{Im}_{t_0+\Delta t}$  until  $\text{Im}_{t_0}$  is correlated to  $\text{Im}_{t_0}$  and motion tracking of features in the template is achieved [20].

Let  $f(x, y)$  be the intensity value of the  $M_x \times M_y$  image  $f$  at pixel  $(x, y)$ ,  $x \in \{0, \dots, M_x - 1\}$ ,  $y \in \{0, \dots, M_y - 1\}$ . Similarly, let  $t(x, y)$  be the intensity value of the  $N_x \times N_y$  template  $t$  at pixel  $(x, y)$  where  $N_x \leq M_x$  and  $N_y \leq M_y$ . NCC is evaluated at every point  $(u, v)$  for  $f$  and  $t$ , which has been shifted over the original image  $f(x, y)$  by  $u$ -steps in the  $x$ -direction and  $v$ -steps in the  $y$ -direction. All the NCC coefficients are stored in a correlation matrix  $\gamma_{u,v}$  defined:

$$\gamma_{u,v} = \frac{\sum_{x,y} (f(x,y) - \bar{f}_{u,v})(t(x-u, y-v) - \bar{t})}{\sqrt{\sum_{x,y} (f(x,y) - \bar{f}_{u,v})^2 \sum_{x,y} (t(x-u, y-v) - \bar{t})^2}} \quad (1)$$

where  $u \in \{0,1,2,\dots, M_x - N_x\}$  and  $v \in \{0,1,2,\dots, M_y - N_y\}$ , and  $\bar{f}_{u,v}$  denotes the mean value of  $f(x,y)$  within the area of the template  $t$  shifted by  $(u,v)$  steps and defined:

$$\bar{f}_{u,v} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x,y) \quad (2)$$

Finally,  $\bar{t}$  denotes the mean value of the template  $t$  defined in a similar way.

Direct computation of Equation (1) involves the order of  $N_x N_y (M_x - N_x)(M_y - N_y)$  calculations, which is very computationally expensive and not suitable for motion tracking of large numbers of points or features in each image. For example, to match a small  $200 \times 200$  pixel template with a  $250 \times 250$  pixel image would require a total of approximately  $10^8$  calculations.

## 2.2 FFT and Sum-Table for Denominator of NCC

A significantly more efficient way of calculating the NCC is by computing the numerator of Equation (1) via FFT. More specifically, cross-correlation in the spatial domain as in Equation (1) is equivalent to multiplication in the frequency-domain:

$$\begin{aligned} r(u,v) &= \sum_{x,y} f(x,y) \cdot t(x-u, y-v) \\ &\Rightarrow R(u,v) = F(u,v) \cdot T(u,v) \\ &\Rightarrow r(u,v) = \mathfrak{F}^{-1}(R(u,v)) \end{aligned} \quad (3)$$

Equation (3) corresponds to computing a 2D FFT on the template,  $t$ , and the region of interest (ROI) window,  $f$ , of the images followed by a complex-conjugate multiplication of the resulting Fourier coefficients. The final products are then inverse

Fourier transformed to produce the actual coefficient cross-correlation plane, as demonstrated in Figure 2. The use of the FFT to simplify the numerator calculations in Equation (1) reduces the number of NCC calculations to the order of  $M_x M_y \log_2(M_x M_y)$ . However the denominator of the NCC in Equation (1) does not have a correspondingly efficient frequency domain expression [16].

A further significant reduction to the number of computations required to compute the NCC in Equation (1) can be made using the idea of a sum-table to simplify computation of the denominator in Equation (1). The sum-table is a pre-computed data structure that acts as a lookup table, dramatically reducing the number of multiplications or additions required to evaluate a given expression. More specifically, the sum-table is a discrete version of an integral image [21, 22].

Let  $f(x, y)$  be an integrable 2-dimensional function with non-negative  $x, y$ . The integral image of  $f$  is defined:

$$I(u, v) = \int_{x=0}^u \int_{y=0}^v f(x, y) dy dx \quad (4)$$

where  $u$  and  $v$  are defined over the domain of  $f$  and  $I(u, v)$  is an integral transformation of  $f(x, y)$  into the  $u, v$  domain.

Equation (4) can be used to compute an explicit integral:

$$\int_{x=x_a}^{x_b} \int_{y=y_a}^{y_b} f(x, y) dy dx = I(x_b, y_b) + I(x_a, y_a) - I(x_a, y_b) - I(x_b, y_a) \quad (5)$$

The discrete versions of Equation (4) and (5) are defined:

$$I(x, y) = \sum_{x'=0}^x \sum_{y'=0}^y f(x', y') \quad (6)$$

$$\sum_{x=x_a}^{x_b} \sum_{y=y_a}^{y_b} f(x, y) = I(x_b, y_b) + I(x_a - 1, y_a - 1) - I(x_a - 1, y_b) - I(x_b, y_a - 1) \quad (7)$$

where Equation (6) corresponds to a rectangular numerical approximation to the analytical integral of Equation (4). For a digital image, where  $f(x, y)$  is a pixel intensity, at the  $(x, y)$  pixel, Equation (6) represents a summed or volume of intensity.

Using the sum-table notation  $s(x, y) = I(x, y)$  as defined in Equation (6), the double sums  $\sum_{x,y} f(x, y)$  and  $\sum_{x,y} f^2(x, y)$  in the denominator of Equation (1) can be rewritten in a computationally efficient form:

$$\sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) = s(u + N_x - 1, v + N_y - 1) + s(u - 1, v - 1) - s(u - 1, v + N_y - 1) - s(u + N_x - 1, v - 1) \quad (8)$$

$$\sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f^2(x, y) = s^2(u + N_x - 1, v + N_y - 1) + s^2(u - 1, v - 1) - s^2(u - 1, v + N_y - 1) - s^2(u + N_x - 1, v - 1) \quad (9)$$

where

$$s(u, v) = f(u, v) + s(u - 1, v) + s(u, v - 1) - s(u - 1, v - 1) \quad (10)$$

$$s^2(u, v) = f^2(u, v) + s^2(u - 1, v) + s^2(u, v - 1) - s^2(u - 1, v - 1) \quad (11)$$

Note that  $s(u, v) = s^2(u, v) = 0$ , if  $u, v = 0$ . The double sums in the left-hand side of Equations (8) and (9) are evaluated over the region of template  $t$  bounded by  $u < x < u + N_x - 1$  and  $v < y < v + N_y - 1$ . Equation (10) is a reformulation of Equation (6) in terms of recursive relations enabling rapid calculation of the sum-table in one global sweep over the image, and similarly for Equation (11) [16].

Once the sum-tables  $s(u, v)$  and  $s^2(u, v)$  are calculated over the whole image, they act as a pre-computed look-up table for Equations (8) and (9), significantly reducing the number of computations required to calculate the NCC at each pixel shift. To demonstrate the equivalence of Equation (10) with Equation (6), consider a  $2 \times 2$  image  $f$  and corresponding sum-table matrix  $S$  defined:



$$f = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, S = \begin{bmatrix} s(0,0) & s(0,1) \\ s(1,0) & s(1,1) \end{bmatrix} \quad (12)$$

Applying the recursive relation Equation (10) gives:

$$\begin{aligned} s(0,0) &= f(0,0) + s(-1,0) + s(0,-1) - s(-1,-1) = a + 0 + 0 - 0 = a \\ s(0,1) &= f(0,1) + s(-1,1) + s(0,0) - s(-1,0) = b + 0 + a - 0 = b + a \\ s(1,0) &= f(1,0) + s(0,0) + s(1,-1) - s(0,-1) = c + a + 0 - 0 = c + a \\ s(1,1) &= f(1,1) + s(0,1) + s(1,0) - s(0,0) = d + (b + a) + (c + a) - a = a + b + c + d \end{aligned} \quad (13)$$

Thus geometrically, for a given  $(x, y)$ , the value of the sum-table  $s(x, y)$  is the sum of all the pixels above and to the left of  $x, y$  inclusively, corresponding to the double sum over the  $x$  and  $y$  directions that is equivalent to Equation (6). This operation can be seen pictorially in Figure 3, where the sum-table value at location 1 is the sum of all the pixels in region A, and the sum table values at location 2, 3 and 4 correspond to  $A + B$ ,  $A + C$  and  $A + B + C + D$  respectively.

More importantly, the double sums in Equation (1) are calculated using the simplified double sums of Equations (8) and (9) at every  $(u, v)$ . For the case of a  $200 \times 200$  template and a  $250 \times 250$  search window, each shift of the template amounts to 3 additions/subtractions for computing the double sum in Equation (8) or (9), compared to  $50^2 = 2500$  additions for a direct computation of  $\sum_{x,y} f(x, y)$  or  $\sum_{x,y} f^2(x, y)$ .

### 2.3 Numerator of NCC with Sum-Tables and Basis Functions

In the frequency domain under FFT, the number of computations required to evaluate the numerator of the NCC in Equation (1) is still relatively high. Furthermore, there is no direct way of reformulating the numerator in terms of sum-tables to significantly reduce the number of computations, as was done for the denominator [16]. However, if the template,  $t$ , is approximated by a set of  $K$  basis functions, then an approximation to the NCC can be obtained, which enables the numerator to be written in terms of the sum-table  $s(u, v)$  given by Equation (10). Thus, further,

potentially large, computational gains could be obtained over the FFT-based methods [19].

For the case of breast tissue motion tracking considered in this paper, the set of  $K$  basis functions will correspond to a set of  $K$  landmark points on the breast. These points could be either randomly placed marker points or natural patterns on the breast skin. The problem can then be thought of in terms of matching landmark points rather than pixel templates. Thus, the basis functions will all be distinct, well defined, can be detected automatically and are guaranteed to optimally approximate the template.

The template  $t(x, y)$  is rewritten as a weighted sum of  $K$  rectangular basis functions. The 2D compact shape of each landmark point is approximated by a rectangle, which is described by indices  $x_l^i, x_u^i$  and  $y_l^i, y_u^i$  corresponding to the lower and upper bounds of the rectangular areas in the  $x$  and  $y$  direction respectively. Figure 4 shows an example of a single arbitrary-shaped landmark point in a template, which is approximated by a rectangle. In practice, the template is described by a matrix of positive integers. Each integer is a value from 0 to 255 where 0 and 255 correspond to black and white respectively, with values in between corresponding to varying intensities of grey.

A typical description of Figure 4 would be to have high numbers greater than 200 (light) describing the landmark with low numbers less than 100 everywhere else describing the background. A simple one intensity approximation to the landmark would be to make every pixel in the rectangle equal to the average intensity of all the pixels in the landmark. This approach leads to an approximation  $\tilde{t}(x, y)$  to the template function  $t(x, y)$  defined:

$$\tilde{t}(x, y) = \sum_{i=1}^K k_i t_i(x, y) \quad (14)$$

where,

$$\begin{aligned} t_i(x, y) &= 1, x_l^i \leq x \leq x_u^i \text{ and } y_l^i \leq y \leq y_u^i \\ &= 0, \text{ otherwise} \end{aligned} \quad (15)$$

and  $k_i$  is the average intensity of the  $i^{\text{th}}$  landmark point,  $i = 1, \dots, K$ . Thus, it follows from Equations (14) and (15) that for any  $x^i, y^i$  where  $x_l^i \leq x \leq x_u^i$  and  $y_l^i \leq y \leq y_u^i$  corresponding a pixel in coordinates in the rectangle surrounding the  $i^{\text{th}}$  landmark,  $\tilde{t}(x^i, y^i) = k_i$ .

The numerator of Equation (1) can then be written:

$$N(u, v) = \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) t'(x-u, y-v) - \bar{f}_{u,v} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} t'(x-u, y-v) \quad (16)$$

where  $t'(x-u, y-v) = t(x-u, y-v) - \bar{t}$ . Since  $t'(x-u, y-v)$  has zero mean, the term  $\bar{f}_{u,v} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} t'(x-u, y-v)$  is zero [16]. Replacing  $t(x-u, y-v)$  with  $\tilde{t}(x-u, y-v)$  from Equation (14) and substituting  $t'(x-u, y-v) = \tilde{t}(x-u, y-v) - \bar{t}$  into Equation (16) gives an approximation for the numerator of the cross correlation coefficient.

$$\tilde{N}(u, v) = \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \left( \sum_{i=1}^K k_i t_i(x-u, y-v) - \bar{t} \right) \quad (17)$$

$$= \sum_{i=1}^K k_i \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) t_i(x-u, y-v) - \bar{t} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \quad (18)$$

$$= \sum_{i=1}^K k_i \left( \sum_{x=x_l^i+u}^{x_u^i+u} \sum_{y=y_l^i+v}^{y_u^i+v} f(x, y) \right) - \bar{t} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \quad (19)$$

Equation (19) follows from Equation (15) where:

$$t_i(x-u, y-v) = 1, x_l^i + u \leq x \leq x_u^i + u \text{ and } y_l^i + v \leq y \leq y_u^i + v \\ = 0, \text{ otherwise} \quad (20)$$

Equation (19) is now in a form which can utilize the sum-table formulation of Equation (8). Thus  $\tilde{N}(u, v)$  is defined:

$$\begin{aligned} \tilde{N}(u, v) = & \sum_{i=1}^K k_i (s(x_u^i + u, y_u^i + v) - s(x_u^i + u, y_l^i + v - 1) - s(x_l^i + u - 1, y_u^i + v) \\ & + s(x_l^i + u - 1, y_l^i + v - 1)) - \bar{t} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \end{aligned} \quad (21)$$

where  $\sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y)$  is given by Equation (8), which has already been evaluated during the denominator calculation of Equation (1), and can be reused.

## 2.4 Computational Efficiency Example

To demonstrate the computational efficiency of using Equation (21) to calculate NCC compared with the FFT and the traditional CC formulation, consider a  $4 \times 7$  template  $t$  with two basis functions ( $K = 2$ ) of average intensity 2 and a  $6 \times 9$  search window  $f$  matrix defined:

$$t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (22)$$

$$f = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{19} \\ f_{21} & f_{22} & \dots & f_{29} \\ f_{31} & f_{32} & \dots & f_{39} \\ f_{41} & f_{42} & \dots & f_{49} \\ f_{51} & f_{52} & \dots & f_{59} \\ f_{61} & f_{62} & \dots & f_{69} \end{bmatrix} \quad (23)$$

In this case, the NCC of the images  $t$  and  $f$  in Equation (22) and (23) will be a  $3 \times 3$  correlation matrix defined:

$$\mathcal{Y} = \begin{bmatrix} \mathcal{Y}_{0,0} & \mathcal{Y}_{0,1} & \mathcal{Y}_{0,2} \\ \mathcal{Y}_{1,0} & \mathcal{Y}_{1,1} & \mathcal{Y}_{1,2} \\ \mathcal{Y}_{2,0} & \mathcal{Y}_{2,1} & \mathcal{Y}_{2,2} \end{bmatrix} \quad (24)$$

where  $\gamma_{u,v}, u = 0, \dots, 2, v = 0, \dots, 2$  are given by Equation (1). Using Equation (21) with  $K = 2, k_1 = 2$  and  $k_2 = 2$ , the approximation  $\tilde{N}(0,0)$  to the numerator of the first entry  $\gamma_{0,0}$  in Equation (24) is given by:

$$\begin{aligned} \gamma(0,0) &= 2\left(s(x_u^1, y_u^1) - s(x_u^1, y_l^1 - 1) - s(x_l^1 - 1, y_u^1) + s(x_l^1 - 1, y_l^1 - 1)\right) \\ &\quad + 2s(x_u^2, y_u^2) - s(x_u^2, y_l^2 - 1) - s(x_l^2 - 1, y_u^2) + s(x_l^2 - 1, y_l^2 - 1) \\ &\quad - \bar{t} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \\ &= 2\left(s(2,2) - s(2,0) - s(0,2) + s(0,0)\right) \\ &\quad + 2\left(s(2,5) - s(2,3) - s(0,5) + s(0,3)\right) \\ &\quad - \bar{t} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y) \end{aligned} \quad (25)$$

The expression  $\bar{t} \sum_{x=u}^{u+N_x-1} \sum_{y=v}^{v+N_y-1} f(x, y)$ , has already been calculated in the denominator of Equation (1), thus the number of computations required to evaluate Equation (25) is 2 multiplications and 7 additions/subtractions. Similar computations are required for the numerators of the other 8 entries of Equation (24), giving a total of  $9 \times 2 = 18$  multiplications and  $9 \times 7 = 63$  additions/subtractions.

This example can be readily generalized to a  $N_x \times N_y$  template  $t$  with  $K$  basis functions and a  $M_x \times M_y$  search window  $f$  producing a  $(M_x - N_x + 1) \times (M_y - N_y + 1)$  correlation matrix requiring in total,  $K(M_x - N_x + 1) \times (M_y - N_y + 1)$  multiplications and  $(3K + 1)(M_x - N_x + 1) \times (M_y - N_y + 1)$  additions/subtractions for the  $(M_x - N_x + 1) \times (M_y - N_y + 1)$  entries of  $\tilde{N}(u, v)$ ,  $0 \leq u \leq M_x - N_x, 0 \leq v \leq M_y - N_y$  given by Equation (21). The results are summarized in Table 1 which compares the

sum-table formulation to the FFT and direct formulation of the numerator of Equation (1).

## 2.5 Basis Functions

The automatic determination of the  $K$  basis functions is equivalent to automatically identifying the marker points in the image. This task is done by initially labelling each marker point region, so that each pixel in a particular region has the same number. In practice, this task is achieved by first thresholding the template image to get a matrix of 0's and 1's, then finding all regions inside the matrix that are connected by 1's. The minimum and the maximum indices for each region are then utilized to form the best representation of a rectangular basis function.

For example, consider the case where the marker points are randomly placed circles, as shown in Figure 5 (a). These circles are first labelled and the rectangular basis functions are found as shown in Figure 5 (b). As the circles are randomly distributed in the test images, some templates may contain overlapping circles. In practice, this overlap would correspond to two marker points that are sufficiently close together that thresholding does not separate them. However, this situation would have no significant effect on accuracy, as a single rectangular basis function would cover both circles. Partially formed circles due to the template boundary will similarly have no effect on accuracy as the corresponding rectangle will just have one of its sides on the boundary of the template image. Figure 6 illustrates these two examples when automatic basis detection is applied on randomly placed circles.

## 2.6 Algorithm Summary

The fast-NCC based motion tracking algorithm presented, which uses the concepts of sum-table and basis function for efficient calculation of the NCC operates in the five steps shown in Figure 7: image acquisition, template and search window arrangement, calculation of sum-tables and basis functions, NCC calculation and motion pattern representation:

- Image acquisition involves using a high-resolution digital camera to capture a pair of images describing the given motion that is required to be tracked.
- Template and search window arrangement involves dividing the second image into smaller template images and dividing the first image into search window sub-images, which are of a sufficient size to guarantee that the corresponding template images lie within the window sub-images. The sizes of the window sub-images relative to the template images can be determined from an upper-bound on expected motion. For a more accurate motion field, the template images in the second image are made to overlap by 50 %.
- The sum-tables for the first image are calculated using the recursive formulas of Equation (10) and (11). The basis functions in each overlapping template sub-image are then calculated by thresholding the image and labelling and identifying the boundaries and centres of landmark points or natural speckle patterns on the skin.
- The NCC is calculated using the sum-table representation given by Equations (8) and (9) to match every template image in second image onto the corresponding window sub-image in first image, thus producing a motion vector for each template centre.
- The motion vectors for each template centre provide the overall motion pattern representation.

### **3. Results**

#### **3.1 Simulated Data**

Breast surface motion that occurs between two successive images of an actuation sequence is simulated by performing translational motion of up to 50 pixels on binary images of randomly placed circles. Note that allowing too much motion between images could risk missing potentially small perturbations on the surface that could arise due to a tumour [1, 10]. Also the motion between images must be sufficiently small so as to avoid any significant scaling or rotation that could affect the accuracy of the NCC, which in any form only examines average translations between

images. However, these issues can be addressed in how a DIET implementation gathers the image data.

To test the speed and accuracy of the sum-table method versus the standard FFT-based NCC method, template images of size  $150 \times 150$ ,  $180 \times 180$  and  $200 \times 200$ ,  $230 \times 230$  are matched to images of size  $200 \times 200$  and  $250 \times 250$  respectively. In each case, 5, 10, 20 and 30 circles are randomly placed in each image similar to that shown in Figure 5 (a) and Figure 6 (a). Tables 2-5 show the average CPU times from 100 random simulations of the sum-table method with and without accounting for the time for calculating the basis functions and the CPU time of the FFT-based NCC. All simulations were done on Matlab using a Pentium 4 desktop with 3.0 GHz CPU and 1.0 GMB of RAM.

The mean pixel error between the two methods over all simulations was  $\sim 0.1$  pixels with an average standard deviation of  $\sim 0.2$  pixels. Note that unlike [19], here the CPU time for pre-calculating the sum-table is included in all cases. Tables 6-9 show the comparison of CPU times for the sum-table method versus the FFT-based CC method, which only involves calculating the numerator of the NCC given by Equation 1. The results show in all cases that the sum-table method is just as accurate and at least an order of magnitude faster than both the FFT-based NCC and CC methods.

The method was then applied on 1M pixel images, where the first image is shown in Figure 8. Each image contains 500 circles so that on average a  $250 \times 250$  image would contain  $\sim 30$  circles. A non-uniform motion field with a maximum displacement of 25 pixels was simulated, as shown in Figure 9, where the motion vectors are scaled for ease of viewing. Note that it is assumed here that the maximum potential displacement between two images during a sinusoidal actuation is known in advance. In practice, this maximum displacement would be an upper bound on expected motion that could be fixed at a chosen value as there is physical control on the time taken between images in a sequence.

The first image ( $I_1$ ) and the second image ( $I_2$ ) are both broken into 64 smaller 50% overlapping  $250 \times 250$  sub-images. The  $250 \times 250$  sub-images in  $I_1$  are then each represented by a  $200 \times 200$  template image by truncating each image by 25 pixels on all sides. Since a maximum displacement of 25 pixels is assumed, each  $200 \times 200$  template in  $I_1$  is guaranteed to lie somewhere inside the corresponding  $250 \times 250$  image in  $I_2$ . Larger maximum displacements could be handled by either decreasing



the template sizes in  $I_1$  or increasing the sub-images sizes in  $I_2$ . Each  $200 \times 200$  template image in  $I_1$  was then matched to the corresponding  $250 \times 250$  images in  $I_2$  to calculate the motion using NCC. The difference in motion between the FFT-based and sum-table methods was on average 0.25 pixels with a standard deviation of 0.43 pixels. However, the total CPU time for the FFT-based NCC was 37.67 s while the sum-table took 2.84 s including basis function calculation. Without including the basis function calculation, the total CPU time was 1.01 s. In this case, the sum-table method is about 13-37 times faster than the FFT method.

### **3.2 Visco-Elastic Breast Phantom**

The sum-table and basis functions method presented is applied on two 1M pixel images of two different deformations of a visco-elastic breast phantom with randomly placed markings, as shown in Figure 10. For this example, 75 pixels were used as an upper-bound on the maximum motion between the images. Image 1 was broken into 50% overlapping templates of size  $250 \times 250$  with corresponding  $400 \times 400$  search windows in the second image to account for up to 75 pixel movement in either the  $x$  or  $y$  directions of the  $250 \times 250$  templates. Figure 11 shows an example of the calculation of basis functions for a template after thresholding and Figure 12 shows the basis functions in Figure 11 registered onto the corresponding search window after finding the best template match.

The overall motion field of the first image in Figure 10 is shown in Figure 13. The total mean error is 0.12 pixels for the FFT-based NCC motion field with a standard deviation of 0.11 pixels. This result shows that approximating the template with basis functions for computational efficiency does not affect accuracy.

### **3.3 Human Skin**

Figures 14 and 15 show images of two different deformations of human skin. The first image in Figure 14 is the reference image and is of size  $250 \times 250$ . The second image in Figure 15 is the template and is of size  $230 \times 230$ . In this case, the

natural patterns on the skin will be used as basis functions in the sum-table method. To calculate the basis functions to represent the template, the image is first thresholded to detect the brighter pixels corresponding to speckles, as shown in Figure 16. All regions connected by 1 pixel are then labelled and features that have less than 10 pixels are removed. This produces 30 basis functions as shown in Figure 17.

The sum-table method is then used to approximate the NCC and match the template of Figure 15 onto Figure 14. Figure 18 shows a close up of the registration of speckles on Figure 16 to speckles on Figure 14, which shows a close match. The motion vectors for the FFT-based NCC and sum-table based NCC were the same with the vector value (12, 19), thus reaffirming that approximating the template with basis functions does not cost accuracy.

## 4. Discussion and Conclusions

The motion calculated using the sum-table based NCC was always within 1 pixel of the FFT-based NCC for both simulated motion and human skin motion. This shows that there is no compromise on accuracy when using the sum-table method. However, the sum-table method is at least an order of magnitude faster than the FFT for both NCC and CC in all cases. Depending on the size of the images, and for 20+ circles, the sum-table method is ~11-28 (9-20) times faster than FFT for NCC (CC) including basis function calculation and ~37-150 (187-271) times faster than FFT for NCC (CC) not including basis function calculation.

For the case of breast surface motion tracking, given the high accuracy required to detect small perturbations on the surface, every landmark point must be identified between images for accurate interpolated motion and camera calibration. Thus, both the FFT and sum-table based methods require the known position of the landmarks in every image. The process of finding the position of a landmark point is effectively the same as calculating basis functions. Thus, the sum-table based NCC is effectively 37-150 times faster than the FFT-based NCC depending on the sizes of the images considered.

Note that the computational saving does not utilize the fact that the motion vector between two 50% overlapping sub-images does not change significantly. To

find the motion vector for the current template centre, the closest motion vector of a previously computed template centre could be used as an approximation. Since the motion fields of the previous and current template centres may not differ significantly, this approach would dramatically reduce the number of shifts required to match the current template to the first image and thus reduce  $M_x - N_x$  and  $M_y - N_y$  in Table 1. This bootstrapping idea could be applied over the whole image further significantly reducing computational requirements. Note that the FFT approach is not able to take advantage of such bootstrapping as the computational time is largely dependent on the size of the template, not the number of the shifts required to match the template to the image. Such a large computational saving for calculating the NCC is important for real time clinical application of the DIET system in breast cancer screening, or any other application tracking large numbers of points.

## References

- [1] A. Peters, A. Milsant, J. Rouze, L. Ray, J. G. Chase, and E. E. W. Houten, "Digital image-based elasto-tomography: Proof of concept studies for surface based mechanical property reconstruction.," *Japanese Society of Mechanical Engineers (JSME) International Journal*, vol. 47(4), pp. 1117-1123, 2005.
- [2] R. Sinkus, J. Lorenzen, D. Schrader, M. Lorenzen, M. Dargatz, and D. Holz, "High-resolution tensor MR elastography for breast tumor detection," *Phys. Med. Biol.*, vol. 45, pp. 1649-1664, 2000.
- [3] E. E. W. V. Houten, M. M. Doyley, F. E. Kennedy, J. B. Weaver, and K. D. Paulsen, "Initial in-vivo experience with steady-state subzone-based MR elastography of the human breast.," *Journal of Magnetic Resonance Imaging*, vol. 17, pp. 72-85, 2003.
- [4] R. Muthupillai, P. J. Rossman, D. J. Lomas, J. F. Greenleaf, S. J. Riederer, and R. L. Ehman, "Magnetic resonance elastography by direct visualization of propagating acoustic strain waves," *Science*, vol. 269, pp. 1854-1857, 1995.
- [5] M. M. Doyley, P. M. Meaney, and J. Bamber, "Evaluation of an iterative reconstruction method for quantitative elastography," *Phys. Med. Biol.*, vol. 45, pp. 1521-1540, 2000.
- [6] T. A. Krouskop, T. M. Wheeler, F. Kallel, B. S. Garra, and T. Hall, "Elastic moduli of breast and prostate tissues under compression," *Ultra-sonic Imaging*, vol. 20, pp. 260-274, 1998.
- [7] A. Samani, J. Bishop, C. Luginbuhl, and D. B. Plewes, "Measuring the elastic modulus of ex-vivo small tissue samples," *Phys. Med. Biol.*, vol. 48, pp. 2183-2198, 2003.
- [8] M. Raffel, C. Willert, and J. Kompenhans, *Particle Image Velocimetry: A Practical Guide*. Berlin: Springer-Verlag, 1998.
- [9] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," presented at International Joint Conference on Artificial Intelligence, Vancouver, 1981.
- [10] S. Kinoshita, "Tissue Actuation Studies for a Digital-Image Elasto-Tomography (DIET) Breast Cancer Screening System," Master thesis, Department of Mechanical Engineering, University of Canterbury, 2005.
- [11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, vol. 1(4), pp. 321-331, 1988.
- [12] N. Peterfreud, "Robust tracking of position and velocity with kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21(6), pp. 564-569, 1999.

- [13] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, pp. 114-141, 2003.
- [14] P. J. Burt, C. Yen, and X. Xu, "Local correlation measures for motion analysis: a comparative study," presented at IEEE Conference in Pattern Recognition Image Processing, 1982.
- [15] T. W. Ryan, "The Prediction of Cross-Correlation Accuracy in Digital Stereo-Pair Images," PhD thesis, University of Arizona, 1981.
- [16] J. P. Lewis, "Fast Normalized Cross-Correlation," *Vision Interface*, pp. 120-123, 1995.
- [17] D. M. Tsai and C. T. Lin, "Fast normalized cross-correlation for defect detection," *Pattern Recognition Letters*, vol. 24, pp. 2625-2631, 2003.
- [18] D. M. Tsai, C. T. Lin, and J. F. Chen, "The evaluation of normalized cross-correlations for defect detection," *Pattern Recognition Letters*, vol. 24, pp. 2525-2535, 2003.
- [19] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross-correlation," presented at Proceedings of SPIE, Aero-Sense Symposium, Vol. 4387, Orlando, Florida, 2001.
- [20] J. C. Russ, "Template Matching and Correlation," in *The Image Processing Handbook*, 2 ed. Raleigh: CRC Press, 1994, pp. 341-346.
- [21] H. Schweitzer, J. W. Bell, and F. Wu, "Very Fast Template Matching," presented at 7th European Conference on Computer Vision, Copenhagen, Denmark, 2002.
- [22] P. Viola and M. J. Jones, "Robust Real-time Object Detection," Cambridge Research Laboratory, Massachusetts 2001.

<b>Method</b>	<b>Number of Multiplications</b>	<b>Number of additions/subtractions</b>
Sum-Table + basis functions	$K(M_x - N_x + 1)(M_y - N_y + 1)$	$(3K + 1)(M_x - N_x + 1)(M_y - N_y + 1)$
FFT	$M_x M_y \log_2(M_x M_y)$	$M_x M_y \log_2(M_x M_y)$
Direct	$N_x N_y (M_x - N_x + 1)(M_y - N_y + 1)$	$N_x N_y (M_x - N_x + 1)(M_y - N_y + 1)$

Table 1: Number computations required for calculating numerator of the NCC given by Equation (1).

<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.5892	0.5892	0.5892	0.5892
Sum-Table	0.0111	0.0092	0.0103	0.0104
Sum-Table (with basis)	0.0207	0.0226	0.0271	0.0298

Table 2: Performance time measured for template of size  $150 \times 150$  inside a  $200 \times 200$  search region.

<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.4538	0.4538	0.4538	0.4538
Sum-Table	0.0107	0.0113	0.0101	0.0094
Sum-Table (with basis)	0.0256	0.0315	0.0399	0.0466

Table 3: Performance time measured for template of size  $180 \times 180$  inside a  $200 \times 200$  search region.

<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.6286	0.6286	0.6286	0.6286
Sum-Table	0.0147	0.0163	0.0152	0.0149
Sum-Table (with basis)	0.0390	0.0395	0.0507	0.0573

Table 4: Performance time measured for template of size  $200 \times 200$  inside a  $250 \times 250$  search region.

<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	1.7589	1.7589	1.7589	1.7589
Sum-Table	0.0136	0.0110	0.0116	0.0126
Sum-Table (with basis)	0.0370	0.0455	0.0635	0.0747

Table 5: Performance time measured for template of size  $230 \times 230$  inside a  $250 \times 250$  search region.

<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.3747	0.3747	0.3747	0.3747
Sum-Table	0.0005	0.0014	0.0020	0.0019
Sum-Table (with basis)	0.0100	0.0130	0.0192	0.0230

Table 6: Performance time measured for template of size  $150 \times 150$  inside a  $200 \times 200$  search region.

<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.3729	0.3729	0.3729	0.3729
Sum-Table	0.0008	0.0008	0.0016	0.0017
Sum-Table (with basis)	0.0164	0.0255	0.0306	0.0383

Table 7: Performance time measured for template of size  $180 \times 180$  inside a  $200 \times 200$  search region.

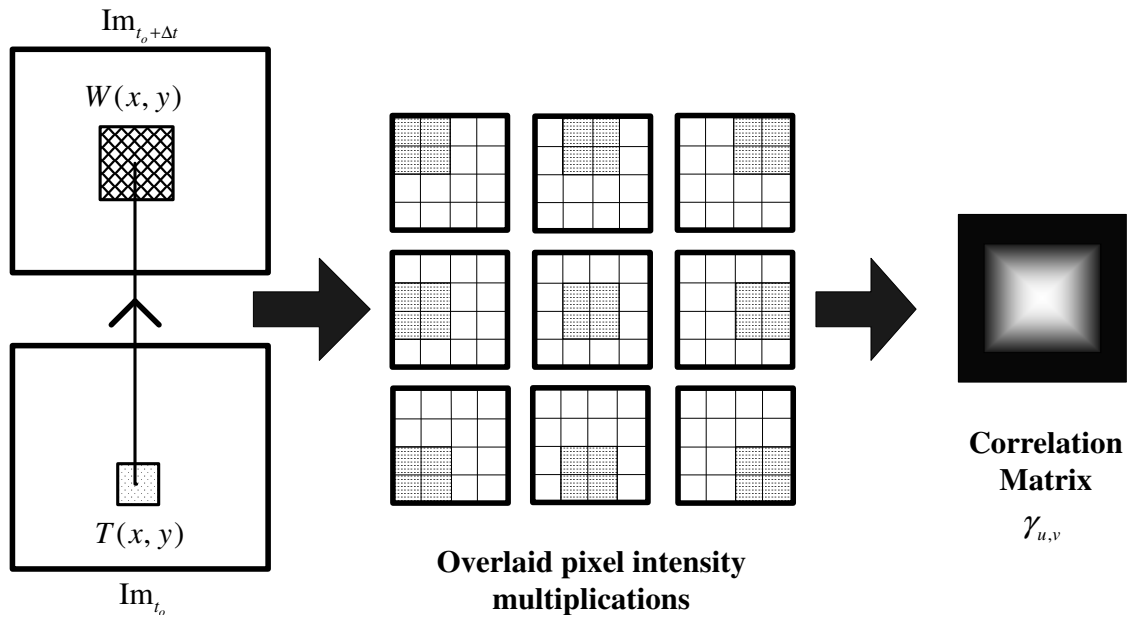
<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.4285	0.4285	0.4285	0.4285
Sum-Table	0.0027	0.0019	0.0016	0.0042
Sum-Table (with basis)	0.0171	0.0252	0.0375	0.0434

Table 8: Performance time measured for template of size  $200 \times 200$  inside a  $250 \times 250$  search region.

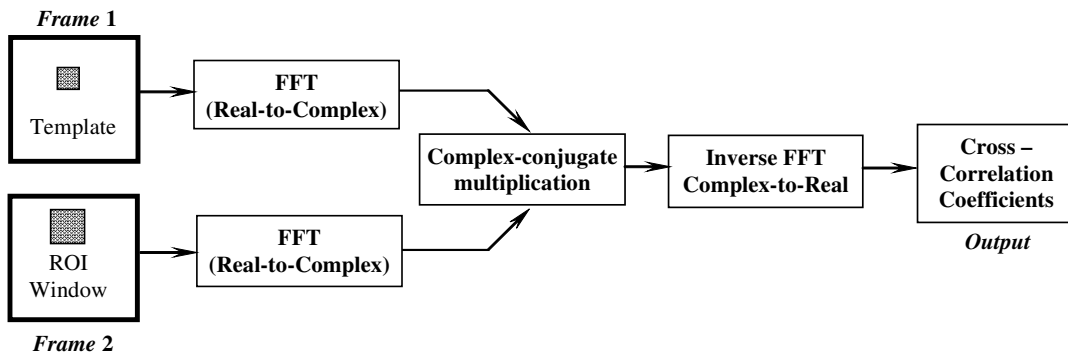
<b>CPU Time (NCC) (s)</b>				
<b>Density of Circles</b>	5	10	20	30
FFT	0.4332	0.4332	0.4332	0.4332
Sum-Table	0.0004	0.0007	0.0020	0.0016
Sum-Table (with basis)	0.0210	0.0325	0.0494	0.0613

Table 9: Performance time measured for template of size  $230 \times 230$  inside a  $250 \times 250$  search region.

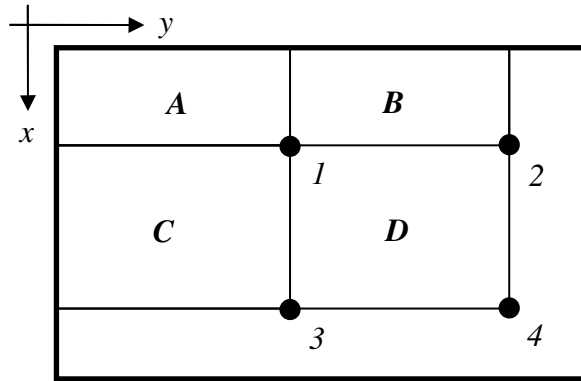




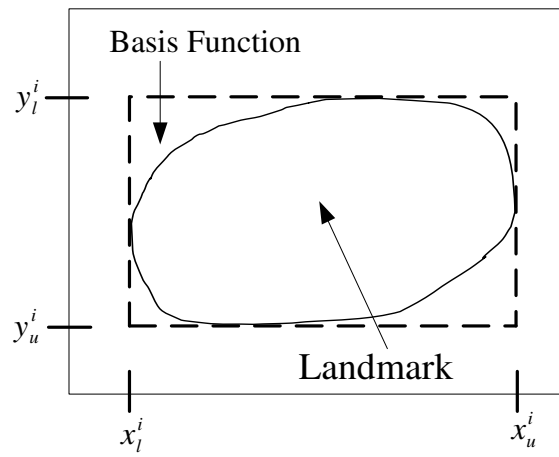
**Figure 1:** Formation of a correlation plane by directly cross-correlating a  $2 \times 2$  pixel template  $T(x, y)$  with a  $4 \times 4$  pixel search window  $W(x, y)$ , resulting in a  $3 \times 3$  pixel correlation plane, where the brightest pixel, at the (2,2) location in this example, indicates  $\gamma_{\max}$  corresponding to the best match.



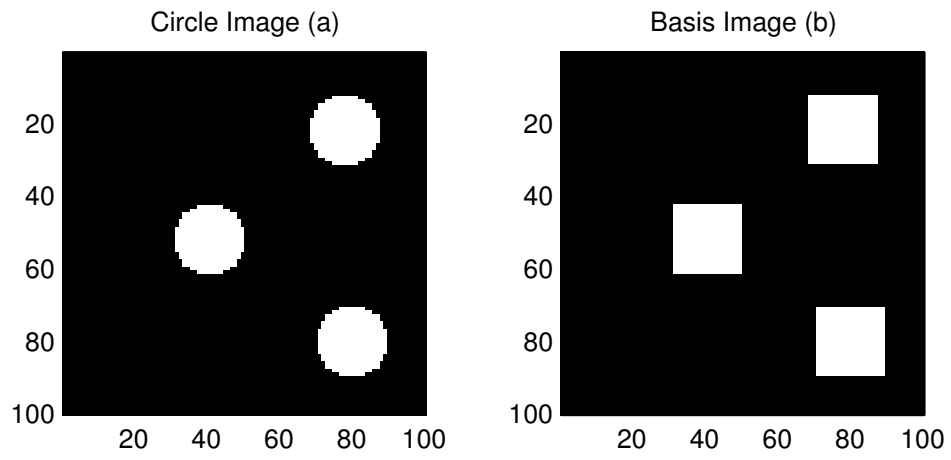
**Figure 2:** Implementation of the numerator of NCC by using FFT algorithm.



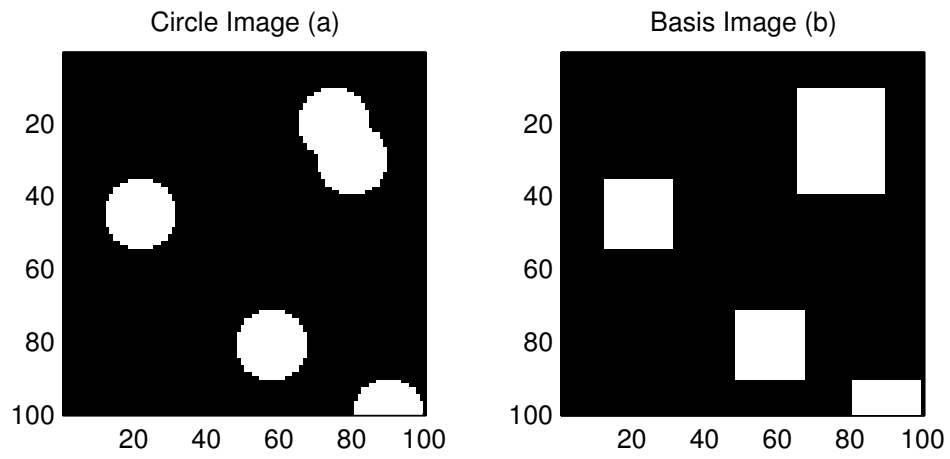
**Figure 3:** Pictorial representation of calculating  $s(x, y)$ .



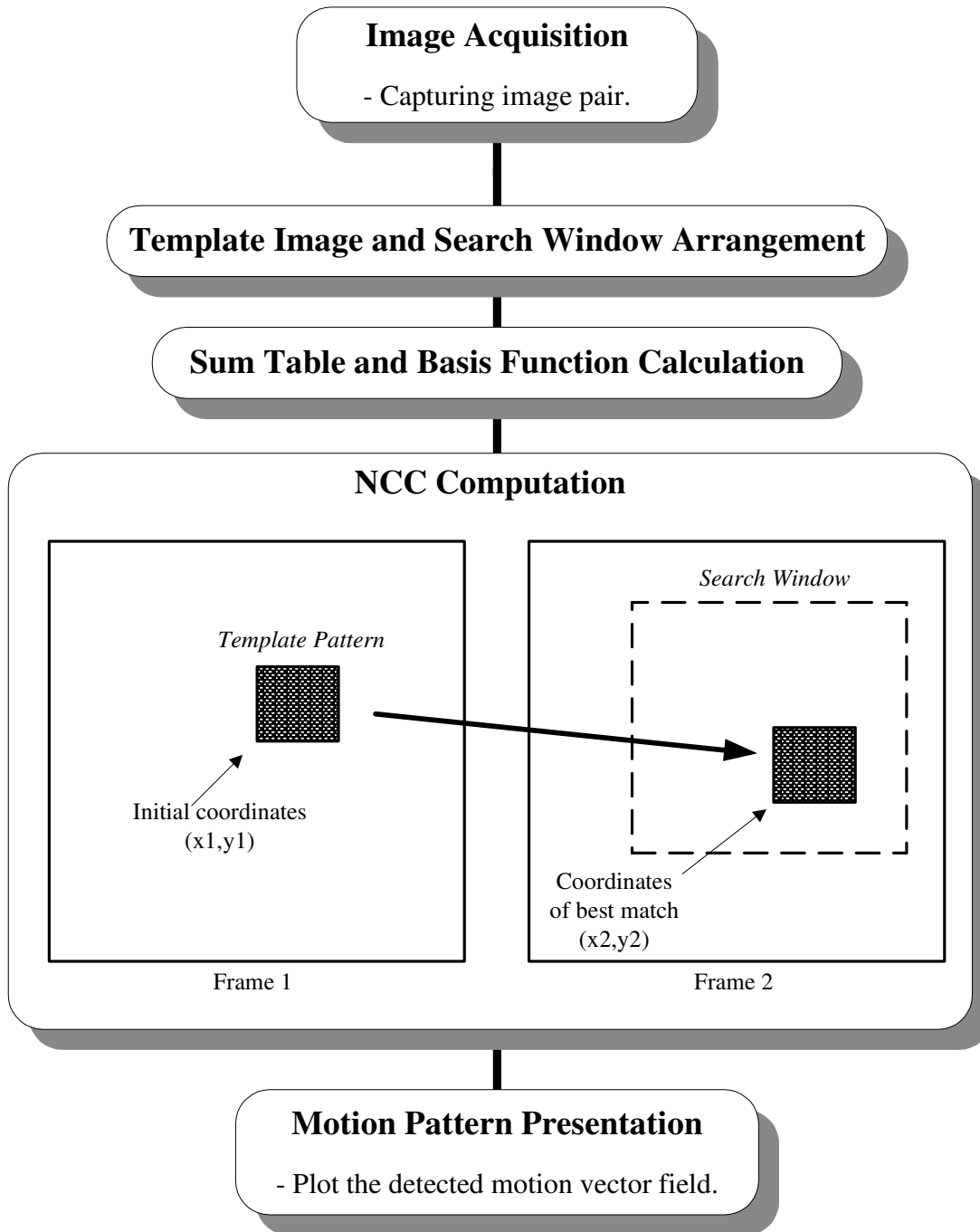
**Figure 4:** Approximating a simple landmark point in a template by a rectangle.



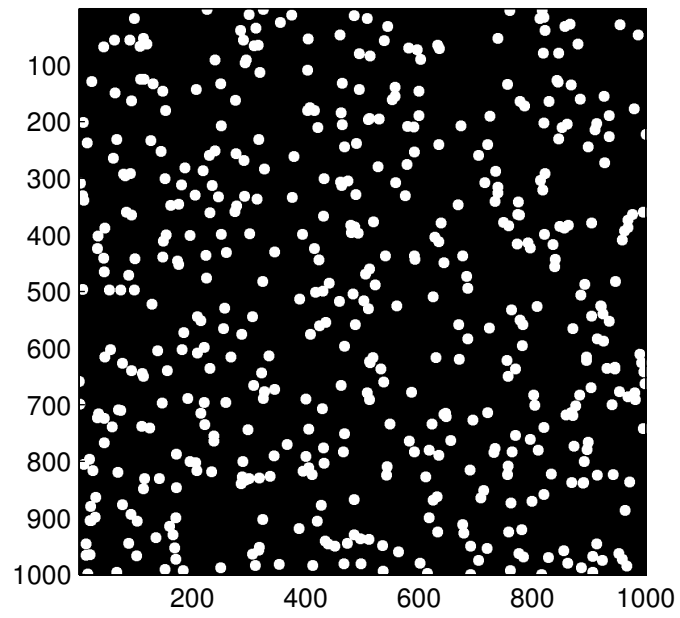
**Figure 5:** Template with circles converted to basis functions.



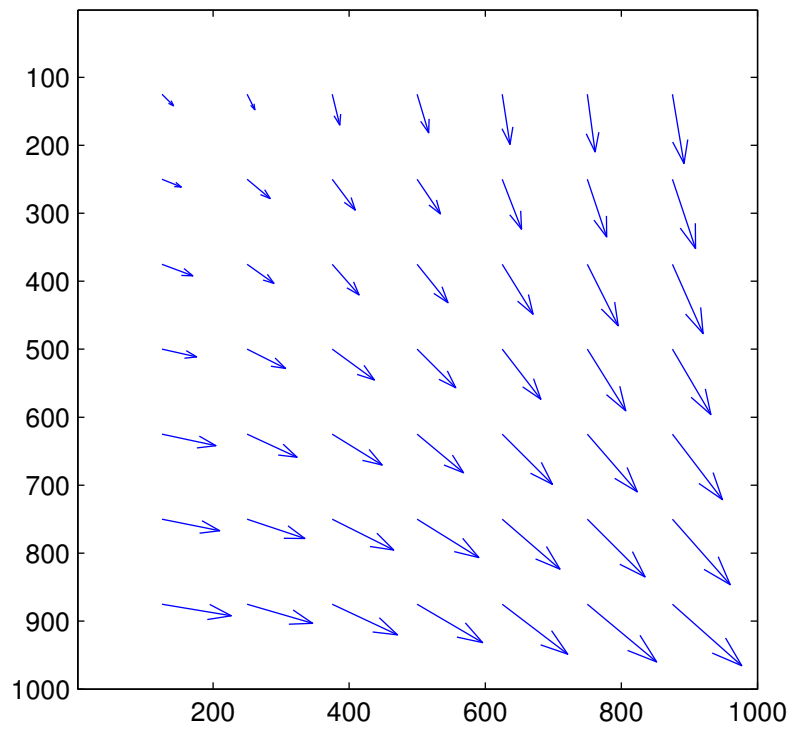
**Figure 6:** Template with overlapped and partial circles converted to basis functions.



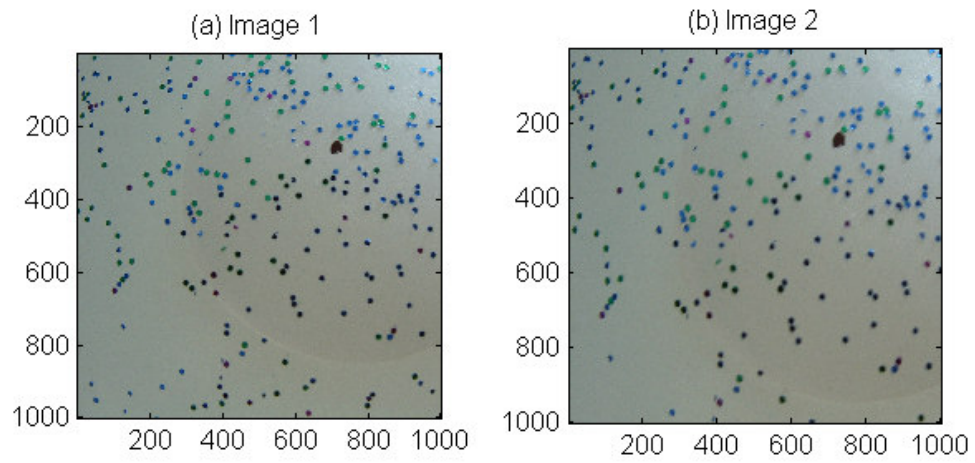
**Figure 7:** Motion tracking procedure.



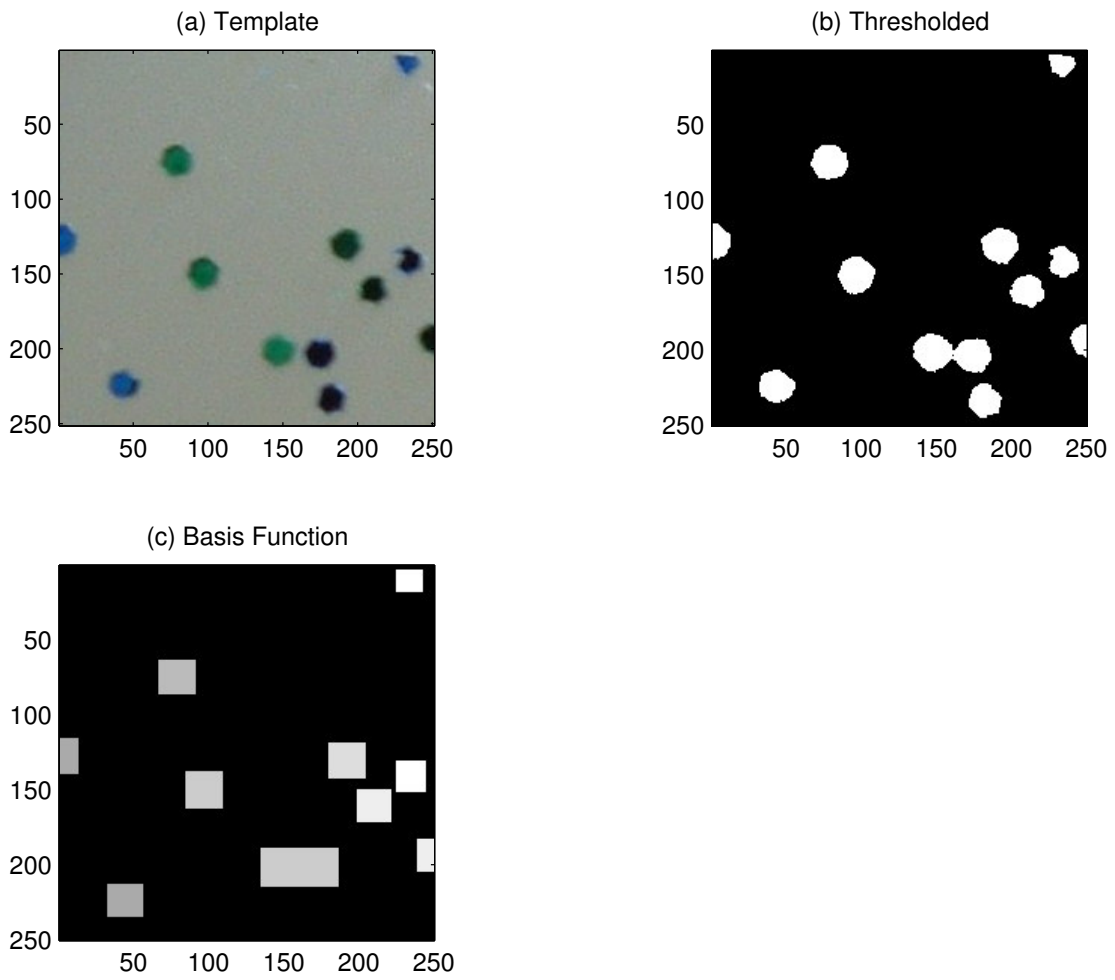
**Figure 8:** 1M pixel image with randomly placed circles.



**Figure 9:** Non-uniform motion field, scaled for ease of viewing.

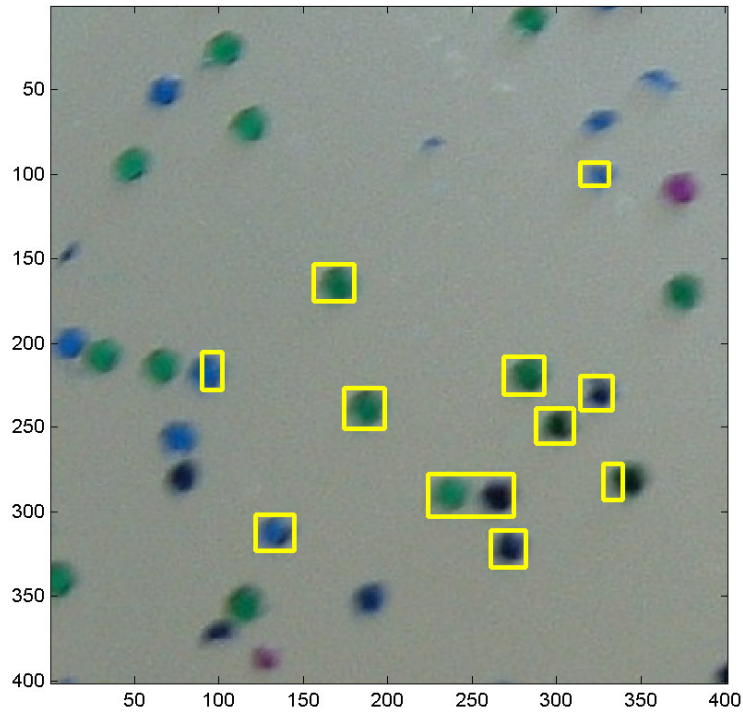


**Figure 10:** Two 1M pixel images of two different deformations of a visco-elastic breast phantom with randomly placed markings.

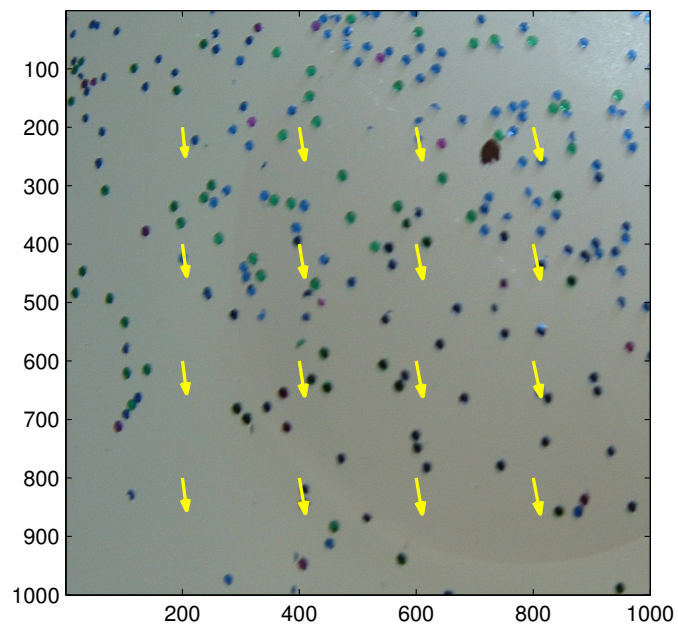


**Figure 11:** Template pattern from first image of Figure 10 converted into basis functions after being thresholding.

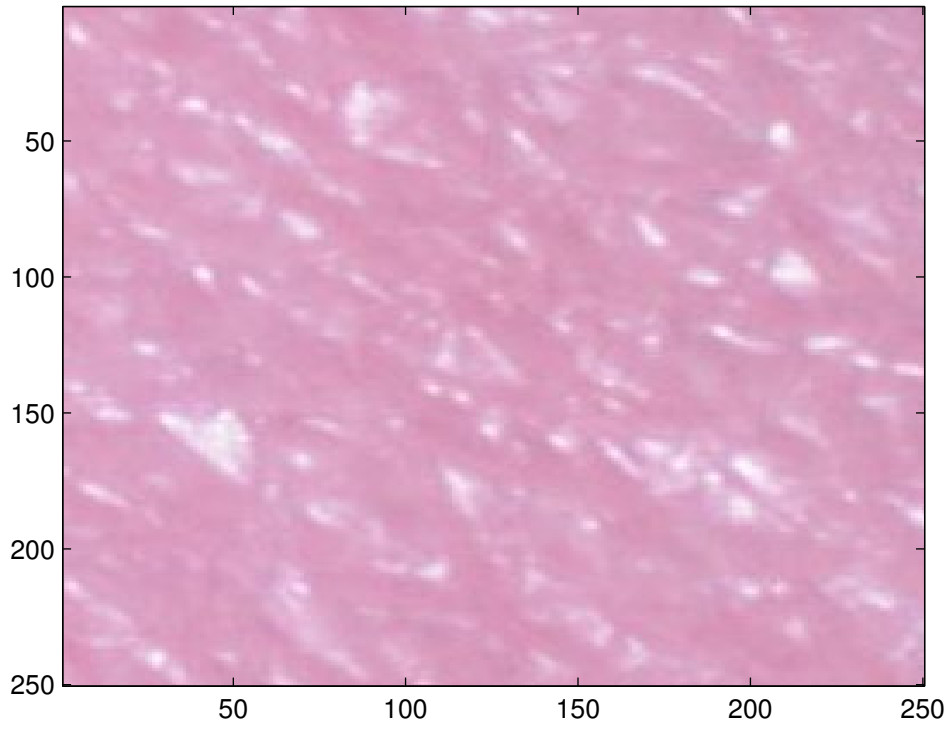




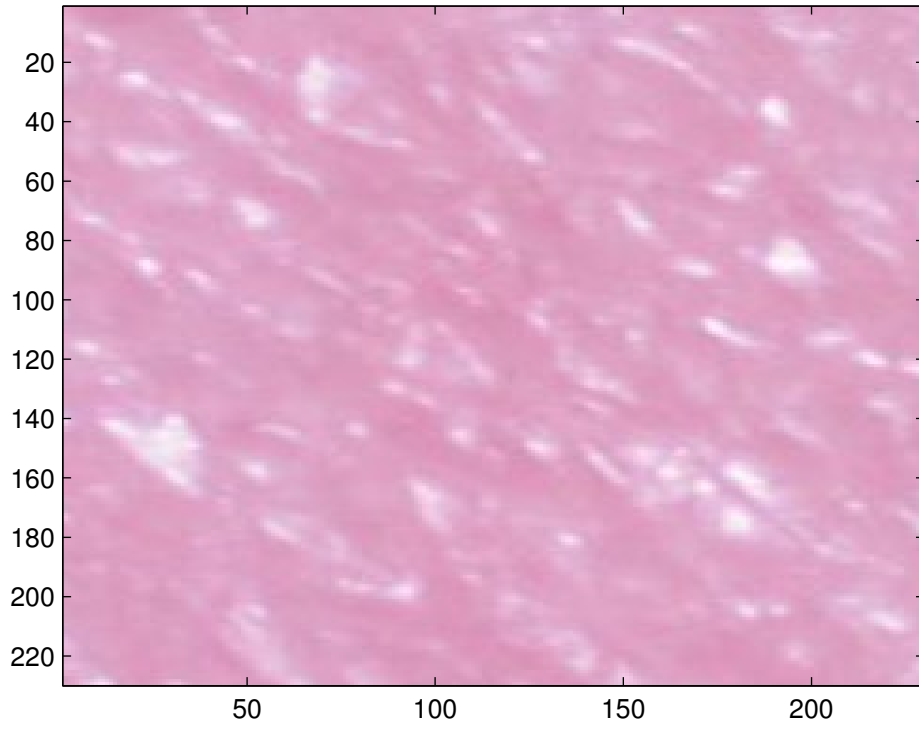
**Figure 12:** Template basis function pattern registered in the corresponding search window image.



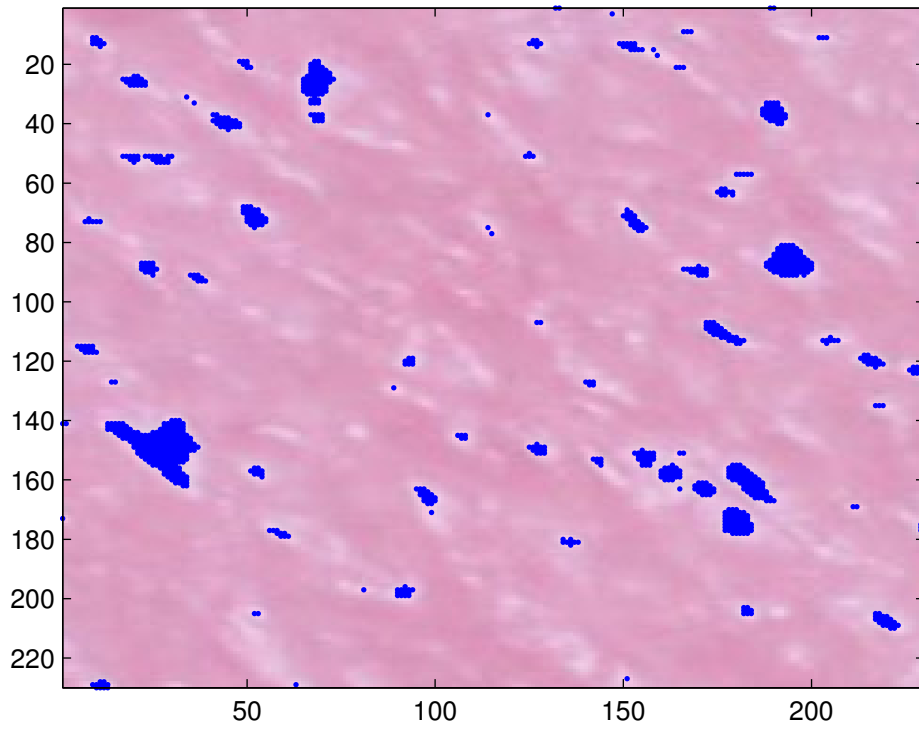
**Figure 13:** Motion field representation of the first image in Figure 10.



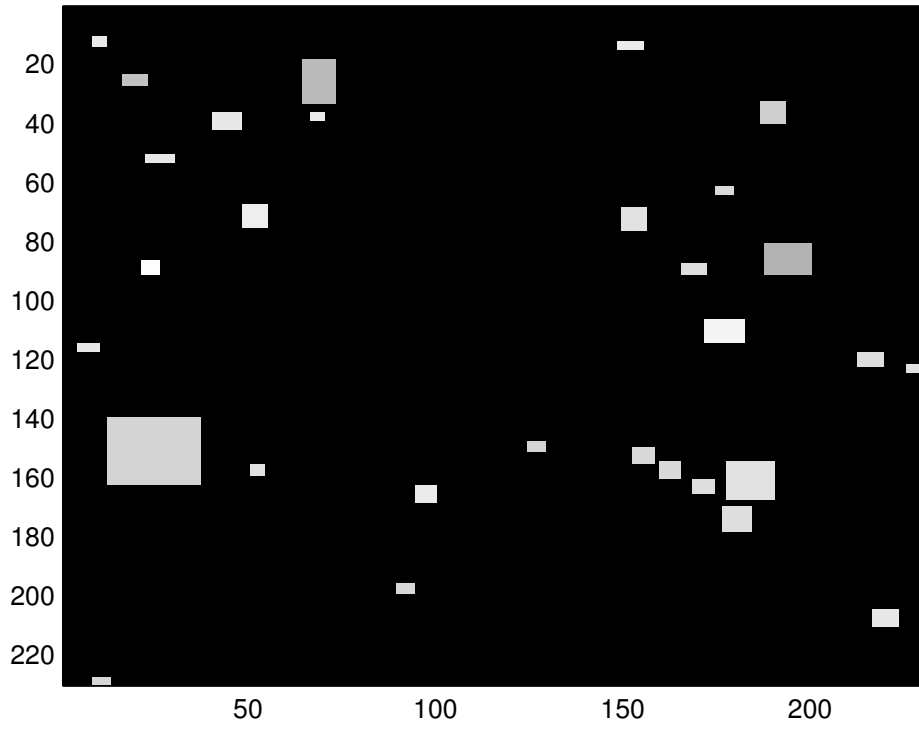
**Figure 14:** One deformation of human skin.



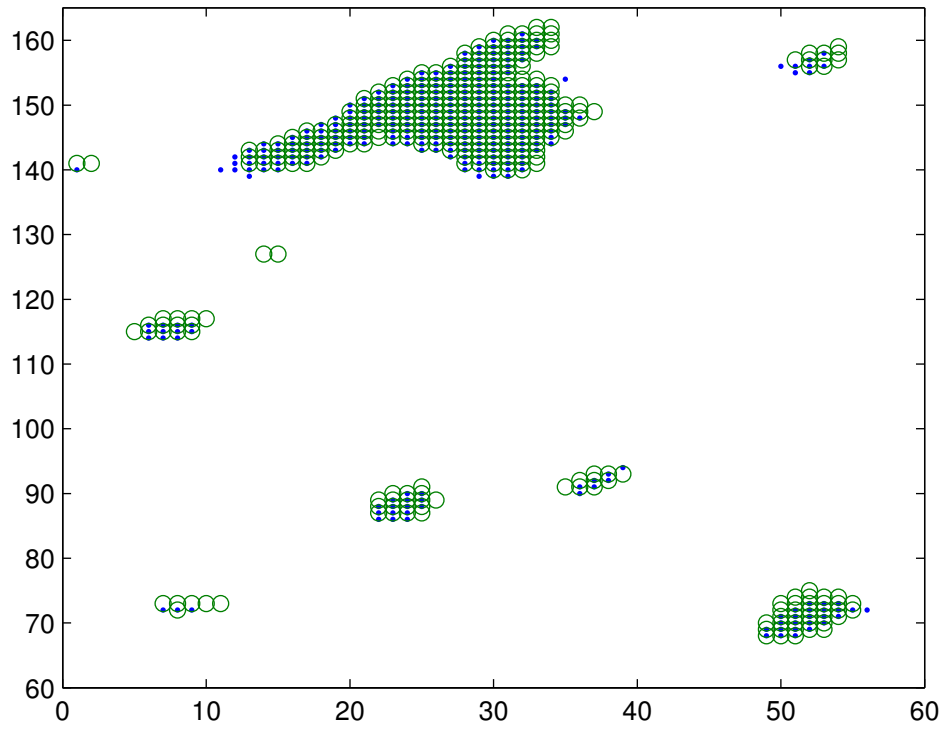
**Figure 15:**  $230 \times 230$  template corresponding to Figure 14.



**Figure 16:** Thresholding template to obtain brighter pixels corresponding to speckles.



**Figure 17:** Representing natural patterns on the skin as rectangular basis functions.



**Figure 18:** Registration of speckles on Figure 14 to speckles on Figure 12.