Hand Gesture Extraction by Active Shape Models

Nianjun Liu, Brian C. Lovell
School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane 4072, Australia
National ICT Australia Ltd (NICTA), Canberra, Australia

ABSTRACT
The paper applied active statistical model for hand gesture extraction and recognition. After the hand contours are found out by a real-time segmenting and tracking system, a set of feature points (Landmarks) are marked out automatically and manually along the contour. A set of feature vectors will be normalized and aligned and then trained by Principle Component Analysis (PCA). Mean shape, eigenvalues and eigenvectors are computed out and composed of active shape model. When the model parameter is adjusted continually, various shape contours are generated to match the hand edges extracted from the original images. The gesture is finally recognized after well matching.

KEY WORDS
Active Shape Model, Principle Component Analysis, Morphological Operation

1 A Real-time Hand Contour Extraction

A real-time hand segmenting and tracking system [1, 2] (Figure 1) is exploited without the specialized image process hardware. The system is the first step towards replacing the mouse interface on a standard personal computer with a touch-less interface to control application software.

The hand gestures are captured by a cheap web camera and a standard Intel Pentium based personal computer. The skin color-based segmentation has been applied. When initial color images are input, the first step is to convert RGB to both the HSV and the YUV color systems. Because the H and UV values of human skin color are in the invariant ranges, the probability distribution image is discerned using a predefined lookup table for skin color hue on the base of histogram. Pre-processing and morphological operations are used to remove the noise. Then the hand regions of interest are sorted and labelled to locate the region of the hands. Camshift algorithm and Kalman filter algorithm have achieved a good performance to track the hand, which is applied to determine a region of interest (ROI) surrounding the hand in each successive frame in order to track the hand. Canny edge detection on the grayscale ROI (determined in the previous step) is used to extract the contour of the hand, and this edge map is combined with the skin-color probability distribution image to determine a reliable contour. Figure 2 is a sequence of images along the system process.

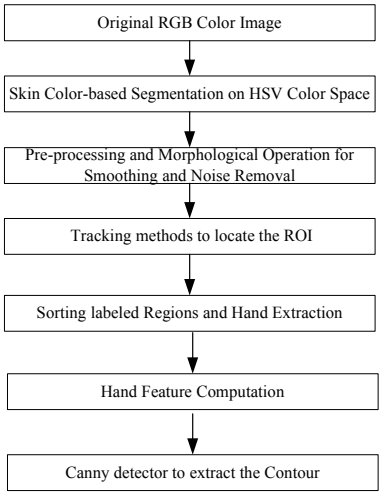


Figure 1. Hand Segmentation and Feature Extraction

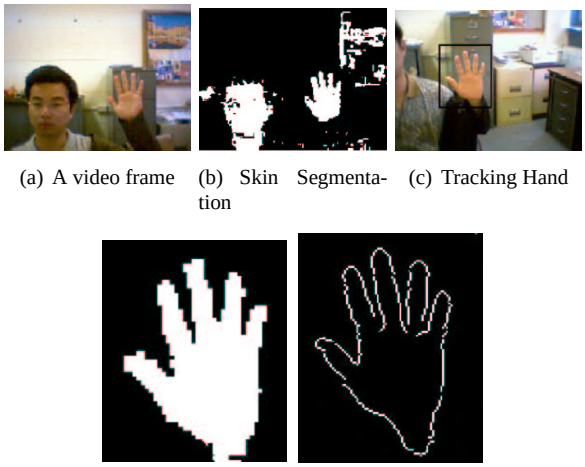
To attain the necessary processing speed, the system exploits the Multi-Media Instruction set (MMX) extensions of the Intel Pentium chip family through software including MS Visual C++, the Microsoft DirectX SDK and the Intel Image Processing and Open Source Computer Vision (OpenCV) libraries. Tracking is robust and efficient, and it can track hand motion at 30 fps.

2 Active Statistical Model

Active Shapes Models (ASM) was proposed by T.F. Cootes and C.J. Taylor [3] to deal with the deformable objects. ASM is similar to Active Contour Models presented by M. Kass, A. Witkin and D. Terzopoulos [4], it can deform variable by the ways produces from in the training set, and the models remain specific to the class of objects they are intended to represent. It could be uesd in the object recognition system, such as tracking, classification, etc.

To build the ASM models, there are two crucial properties: One is to generate any plausible example of the class of objects they represent; the other is that they also are specific, and can only generate 'legal' examples of the objects they represent.

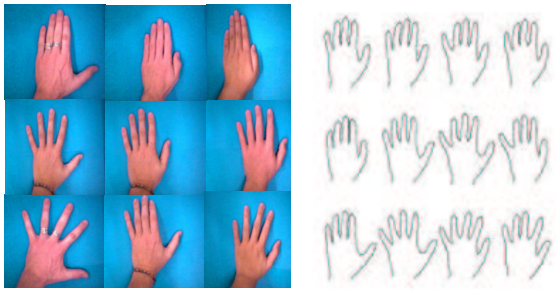
To target the object by ASM method, there are the following conditions: The object with a well-defined shape, a roughly location of the object in the image, and the enough



(a) A video frame (b) Skin Segmentation (c) Tracking Hand

(d) Hand Segmentation (e) Contour detector

Figure 2. System image sequence



(a) (b)

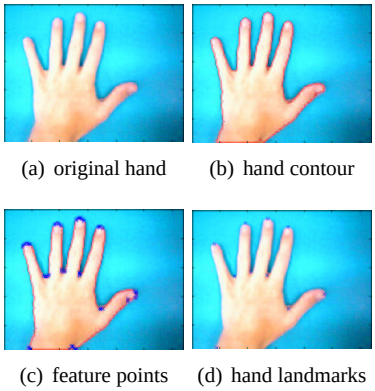
Figure 3. Images and contours from a training set

examples of that kinds of objects in order to be able to build a representative training set. If those conditions are met, ASM is appropriate to apply.

3 Training set and Landmarks computation

The first step in Active Shape Model is to build a training set from which the statistical properties of the class of objects will be learnt. Depending on the hand contour extraction on the first section, we make up the hand-contour training sets, which is composed of 60 examples of the left hand of 6 different subjects. For each subject, ten images of the hand were taken with different positions of the fingers. Each example was labelled with 35 landmarks. In order to extract the precise contour, the hand was open with no fingers crossing and the background is uniform blue. The figure3 are the example of images and contours from a training set.

In ASM, a shape is defined by a set of points called landmarks. How to label the landmark? We applied both automatical and manual ways. After the image of the hand



(a) original hand (b) hand contour

(c) feature points (d) hand landmarks

Figure 4. The labelled hand and its landmarks

t	1	...	n	n+1	n+2	...	2n+1
x	x_{i-n}	...	x_{i-1}	x_i	x_{i+1}	...	x_{i+n}
y	y_{i-n}	...	y_{i-1}	y_i	y_{i+1}	...	y_{i+n}

Figure 5. $x(t)$ and $y(t)$ table

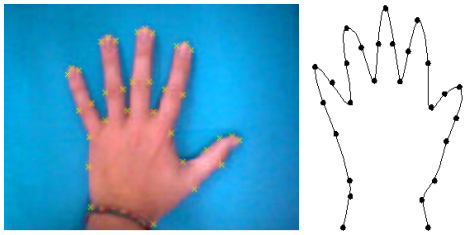
is segmented and its contour is extracted, we get the coordinates of the contour . Next, we will calculate the curvatures of the contour. For each point i of the contour, we consider the n contour points before and after it, and the parameter t is $2n+1$. We interpolate x and y in function of the parameter t by two polynomial functions(Figure 5). As a result, we obtain $x(t)$ and $y(t)$. We can then compute the curvature radius R of the parameter function defined by $(x(t), y(t))$ at the point i, see equation 1.

$$R = \frac{\left(x'^2 + y'^2\right)^{\frac{3}{2}}}{x'y'' - y'x''} \tag{1}$$

If R is below a predefined threshold, then the curvature is quite high at point i, and the point is retained (see figure 4,c). When more than a certain number of consecutive points are retained, the region of high curvature along the contour reaches, which are the fingertips and the junctions of the fingers. We then keep the median point of the consecutive high curvature points as a landmark (see figure 4,d). This method works well, and could help to find out the candidates of the landmarks, furtherly, on the base of them, we remove and add some new landmarks manually. Thereby, the landmarks vector of the image is available, which means the hand shape can be characterized by a vector s containing the coordinates of its n landmarks.

$$\mathbf{s} = \begin{pmatrix} x_1 & y_1 & x_2 & y_2 & \dots & x_n & y_n \end{pmatrix}^T \tag{2}$$

Each example in the training set then has to be labelled in a similar way. As a result, if the training set is composed of m images labelled with n landmarks, a set of



(a) labelled hand (b) hand landmarks

Figure 6. The labelled hand and its landmarks

m $2n$-length vectors is then available.

$$\mathbf{s_1} = \begin{pmatrix} x_1 & y_{11} & \dots & x_{1n} & y_{1n} \end{pmatrix}^T \quad (3)$$

$$\mathbf{s_2} = \begin{pmatrix} x_{21} & y_{21} & \dots & x_{2n} & y_{2n} \end{pmatrix}^T \quad (4)$$

$$\vdots \quad (5)$$

$$\mathbf{s_m} = \begin{pmatrix} x_{m1} & y_{m1} & \dots & x_{mn} & y_{mn} \end{pmatrix}^T \quad (6)$$

We define the mean shape of the training set:

$$\mathbf{s} = \begin{pmatrix} x_1 & y_1 & x_2 & y_2 & \dots & x_n & y_n \end{pmatrix}^T \quad (7)$$

$$\bar{x}_i = \frac{1}{m}\sum_{j=1}^{m} x_{ji} \quad (8)$$

$$\bar{y}_i = \frac{1}{m}\sum_{j=1}^{m} y_{ji} \quad (9)$$

4 Align the shape

The position, scale and rotation is different between the extracted hand shapes, so we will align the shape sets. The flowing chart is in the figure 7.

Shape alignment is to apply a transformation to each shape such that it becomes as similar as possible to the others (in the sense of the minimization of a least-square function). The transformation used is a similarity (translation, scaling and rotation are allowed). The method to find the parameters of the similarity that best aligns a shape with another. The equation is presented in the following.

If we want to align $s2$ with $s1$, this can be done by finding the similarity transformation T which minimize:

$$E = |s_1 - T(s_2)|^2 \quad (10)$$

A two dimensional similarity is defined by a scaling factor k, a rotation by an angle θ and a translation by a vector (t_x, t_y). Thus,

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} k\cos\theta & -k\sin\theta \\ k\sin\theta & k\cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (11)$$

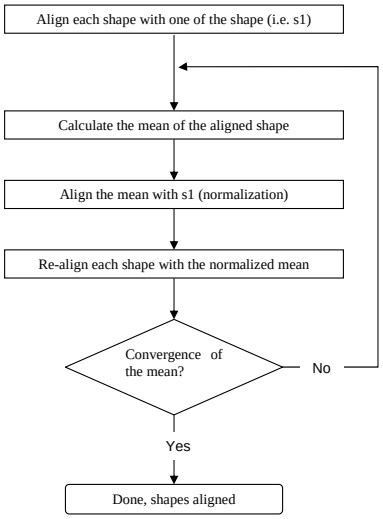


Figure 7. Flowing chart of aligning the shapes

If the shapes are preliminary translated in order to have their centroid on the origin, the problem is then reduced to a scaling and a rotation. We assume the following is in this case.

Let $a = k\cos\theta$ and $b = k\sin\theta$, so $k^2 = a^2 + b^2$ and $\theta = \arctan\left(\frac{b}{a}\right)$, then

$$\begin{pmatrix} x_{1i} \\ y_{1i} \end{pmatrix} - T\begin{pmatrix} x_{2i} \\ y_{2i} \end{pmatrix} = \begin{pmatrix} x_{1i} \\ y_{1i} \end{pmatrix} + \begin{pmatrix} a & -b \\ b & a \end{pmatrix}\begin{pmatrix} x_{2i} \\ y_{2i} \end{pmatrix} = \begin{pmatrix} x_{1i} - ax_{2i} + by_{2i} \\ y_{1i} - bx_{2i} + ay_{2i} \end{pmatrix} \quad (12)$$

Let

$$E_i = \left| \begin{pmatrix} x_{1i} \\ y_{1i} \end{pmatrix} - T\begin{pmatrix} x_{2i} \\ y_{2i} \end{pmatrix} \right|^2 \quad (13)$$

$$E_i = (x_{1i} - ax_{2i} + by_{2i})^2 + (y_{1i} - bx_{2i} + ay_{2i})^2 \quad (14)$$

$$\frac{\partial E_i}{\partial a} = 2a(x_{2i}^2 + y_{2i}^2) - 2(x_{1i}x_{2i} + y_{1i}y_{2i}) \quad (15)$$

$$\frac{\partial E_i}{\partial b} = 2a(x_{2i}^2 + y_{2i}^2) - 2(y_{1i}x_{2i} + x_{1i}y_{2i}) \quad (16)$$

Since $E = \sum_{i=1}^{n} E_i$ Let $\frac{\partial E_i}{\partial a} = 0$ $\frac{\partial E_i}{\partial b} = 0$

$$a = \frac{\sum_{i=1}^{n} x_{1i}x_{2i} + y_{1i}y_{2i}}{\sum_{i=1}^{n} x_{2i}^2 + y_{2i}^2} = \frac{s_1 \cdot s_2}{|s_2|^2} \quad (17)$$

$$b = \frac{\sum_{i=1}^{n} y_{1i}x_{2i} + x_{1i}y_{2i}}{\sum_{i=1}^{n} x_{2i}^2 + y_{2i}^2} = \frac{\sum_{i=1}^{n} y_{1i}x_{2i} + x_{1i}y_{2i}}{|s_2|^2} \quad (18)$$

Convergence is established if the estimate of the mean does not change more than a pre-defined threshold after iteration. The step of normalization of the mean is necessary otherwise the convergence could never occur.

5 PCA and Generate the new shape

5.1 Applying Principal Component Analysis

After the above alignment, a set of m aligned vectors is available. Since the length of each vector is $2n$, a shape can be represented by a point in a 2n-dimensional space. Thus, the whole training set can be fully characterized by a distribution of m points in that space. We then used a Principal Components Analysis (PCA) to extract the directions of independent variation in the cloud of points.

Principal Components Analysis is a technique of multivariate analysis, which was first introduced by Pearson in 1901. As defined in [5], the central idea of PCA is to reduce the dimensionality of a data set which consists of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of uncorrelated variables, the principal components, which are ranked so that the first few ones retain most of the variation presenting in all the original variables.

All the vectors can be gathered in a $(2n, m)$ matrix S, which representing the training set.

$$S = (s_1 - \bar{s}|s_2 - \bar{s}| \dots |s_m - \bar{s}) \tag{19}$$

To perform a PCA, the first step is to compute the covariance matrix C of the data.

$$C = \frac{1}{m-1} SS^T \tag{20}$$

Let p_i be a unit eigenvector associated with the eigenvalue λ_i of C.

$$Cp_i = \lambda_i p_i \tag{21}$$

$$\lambda_i = p_i^T C p_i \tag{22}$$

$$\lambda_i = \frac{1}{m-1} \sum_{k=1}^{m} \left((s_k - \bar{s})^T p_i \right)^2 \tag{23}$$

$$\lambda_i = \frac{1}{m-1} \sum_{k=1}^{m} (s_k^T p_i - mean(s_k^T p_i))^2 \tag{24}$$

$$\lambda_i = var(S^T p_i) \tag{25}$$

$S^T p_i$ being the vector representing the projections of the shapes of the training set on the eigenvector p_i, then λ_i accounts for the variance of the data set in the direction of p_i.

Since a covariance matrix can be diagonalized by an orthogonal basis of eigenvectors. These eigenvectors can be chosen to be unit vectors. Let us sort the eigenvalues such that $\lambda_1 > \lambda_2 > \lambda_3 > \dots$ and be a unit eigenvector associated with the eigenvalue λ_i.

Let $P = (p_1|p_2| \dots |p_{2n}|)$

Any shape s from the training set can then be described with the equation:

$$s = \bar{s} + Pb \quad where \quad b = P^T(s - \bar{s}) \tag{26}$$

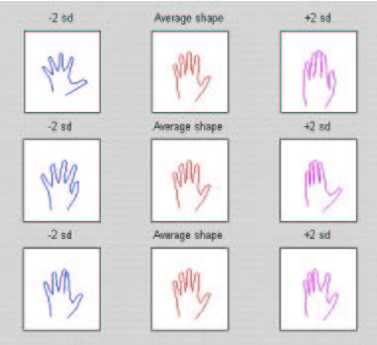


Figure 8. First three modes of a data set

It is proved that the eigenvalue corresponding to an eigenvector represents the variance of the data set in the direction of that eigenvector. Therefore, the eigenvalues of the covariance matrix being sorted in decreasing order, the shape s can be approximated without losing too much information by:

$$s \approx \bar{s} + P_t b_t \tag{27}$$

with $P_t = (p_1|p_2| \dots |p_t|)$ and $b_t = (b_1 \quad b_2 \quad \dots \quad b_t)$ where $t \ll 2n$, t is chosen to explain a given proportion(i.e. 95%) of the variance exhibited in the training set. As the sum of all the eigenvalues accounts for the total variance in this training set, t is then defined such that:

$$\frac{\sum_{i=1}^{t} \lambda_i}{\sum_{i=1}^{2n} \lambda_i} \geq \eta \tag{28}$$

The vector b_t can be seen as a set of parameters that can be used to deform a model and generate a new plausible shape.

Therefore, Applying PCA in the training set, we set up the model, which includes the mean shape, some eigenvalues and eigenvectors,etc.

5.2 Generating new plausible shapes

If we assume for model that the different parameters b_i are independent, gaussian distributed with zero mean, then 95% of the distribution of one parameter is covered in the range $[-2sd, 2sd]$.

Consequently, since λ_i is the variance of the i^{th} parameter, by constraining this parameter to $[-2\sqrt{\lambda_i}, 2\sqrt{\lambda_i}]$, we ensure that the shape generated is similar to those in the training set. t defines the number of 'modes' necessary to capture the proportion of the total variance. The i_{th} mode of variation is to let b_i vary, while the other parameters are set to zero. The figure 8 shows the first three modes of variation obtained from a training set.

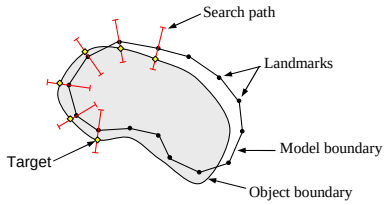


Figure 9. The search of a new set of points

6 Adjust the model to fit a shape in the image

In last section a statistical shape model of the hand has been built. It consists a mean shape $\bar{s}$, parameter vector B_t, eigenvector. By varying the parameters under certain constraint, the new plausible shapes can be generated.

6.1 Fitting the target points in the image

The algorithm described in the above section is used to fit the model to a new set of points. But how to find the target set of points? If we assume, as it is most likely, that the shape model represents by the strong edges of an object, the following approach can be considered: for each model point, a target point is looked for along a profile of a certain length normal to the model boundary. The nearest point lying on a strong edge of the image is then selected as a target point, as shown on the figure 9.

It means that we have an initial knowledge of the position of the target object in an image, so that to approximately place the model on it.

This method allows finding a target set of points before applying the algorithm introduced in the section before. After each deformation of the model, the search for a target shape is repeated in order to refine the process.

6.2 Adjust the model to fit a shape

Given a new set of points s (a "target" shape), the problem is to translate, rotate, scale and deform our model in order that it best fits s. This can be done by applying a similarity transformation to find the best pose, and compute some new shape parameters to find the best deformation.

The first step is merely applying the process of two-shapes alignment described in the previous section. Given the mean shape $\bar{s}$ and the set of target points s (which are now aligned), the shape parameters can be obtained. Compute the vector with the following formula:

$$b_t = P_t^T(s - \bar{s}) \tag{29}$$

Apply constraints on b_t (in order the shape generated remains a plausible one):

$$b_t = (b_1 \quad b_2 \quad \dots \quad b_t) \tag{30}$$

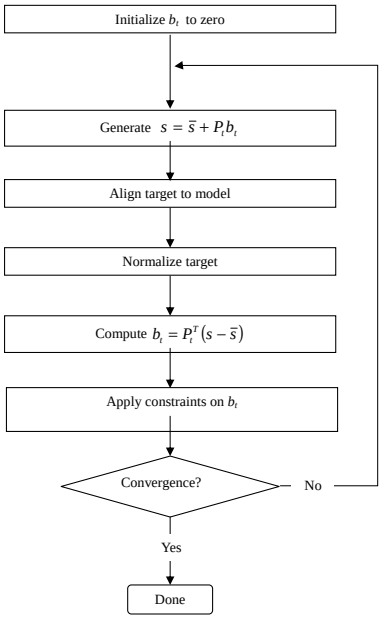


Figure 10. Algorithm to fit the model to a new set of points

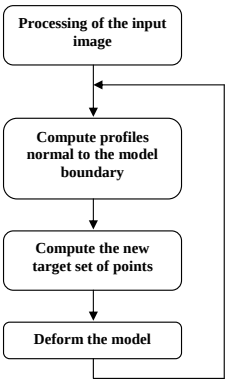


Figure 11. Fit the model to a new set of points

$$if \quad b_i > 2\sqrt{\lambda_i} \quad then \quad b_i = 2\sqrt{\lambda_i} \quad 1 \le i \le t \tag{31}$$

$$if \quad b_i < -2\sqrt{\lambda_i} \quad then \quad b_i = -2\sqrt{\lambda_i} \quad 1 \le i \le t \tag{32}$$

The figure 10 introduces the algorithm used to fit a model to a new set of points. Convergence is established if the estimate of the deformed model does not change more than a pre-defined threshold after iteration.

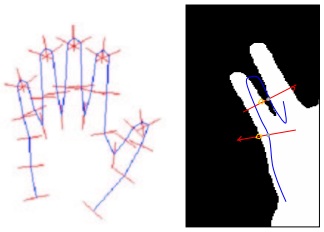
7 Experiment

The flowing chart of the process is show on figure 11.

The processing of search is relying on the searching for strongest edges along the profiles on the binary image. After segment the original image, the binary image of the hand and the clear contour is the output(Figure 12.

Figure 12. Binary image obtained from the skin color detection

The figure 13 shows an instance of the model with the profiles generated. Along each profile, the search is done from the 'interior' of the hand to the 'exterior', and any jump from white to black in the binary image is then searched for. The closest point from the landmark on the profile where that change occurs is then set as the target point. If no such change happened along the whole profile, the landmark itself is set as the target point. After it does along all the profiles, a target set of points is then available.



(a) (b)

Figure 13. search paths from model to target

The algorithm in section 6 is used to perform the deformation of the model. The process is repeated until no significant change happens on the model after iteration. The figure 14 shows one example of the result obtained. The convergence is reached after 12 iterations. The result is well, and the global shape of the hand is retrieved.

The model does not fit exactly to the hand contour. This can be explained by several reasons. Only the first five principal components have been used to deform the model in that example, so little variations have not been taken in account. The model is built from only 35 landmarks, which is not enough to represent precisely the shape of the hand (for example, only one landmark has been used to model each junction of fingers). Some variations are not enough exhibited in the training set (the bent of the thumb, for example). The success or failure of the search is quite dependent on the choice of the initial parameters and especially the start position of the model.

8 Conclusions

We present a method to train active shape model(ASM) through the data set from real-time hand contour extraction

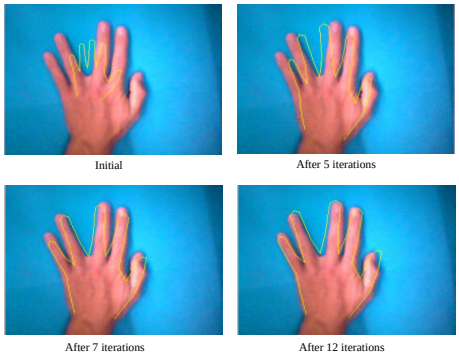


Figure 14. Fitting an ASM to a hand

and adjust the ASM parameters to generate a new shape contour to match hand edges in the original images consecutively. When the new generating shape contour matches the hand edges quite well, the hand gesture is recognized and described by a set of parameters so that the gesture could be defined concisely.

Acknowledgement

The authors would like to thank Etienne Guy from University of Nantes for his contributions on providing the images and assisting the experiments.Financial support for this project from the Australian Research Council is gratefully acknowledged. National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs.

References

[1] Nianjun Liu and Brian C. Lovell, "Real-Time Two Hands Tracking System" Proceedings of The International Technical Conference on Circuits/Systems, Computers and Communications, Phuket, Thailand, pp 1491-1494,July 16-19, 2002

[2] Nianjun Liu and Brian C. Lovell, "MMX-Accelerated Real-Time Hand Tracking System," Proceedings of Image and Vision Computing New Zealand 2001, pp 381-385, Dunedin , 26-28 November, 2001.

[3] T.F.Cootes, G.J.Edwards, and C.J.Taylor. Active appearance models. In H.Burkhardt and B.Neumann,editors,5^{th} European Conference on Computer Vision, volume 2,pages 484-498. Springer,Berlin,1998.

[4] M.Kass, A.Witkin, and D. Terzopoulos. Snakes:Active contour models. In 1^{st} international Conference on Computer Vision, pages 259-268, London, June 1987.

[5] I.T. Jolliffe. "Principal Component Analysis" New Yor;Berlin;Tokyo: Springer-Verlag, 1986.