EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel, Z. Pan (Editors)

Door Access Control Using Human Face and Height

R. H. Ma[1], Z. Y. Huang[2], H. X. Zhang[2], and W. M. Huang[1]

[1] A*STAR I2R, Kent Ridge, Singapore 119613

[2] School of Computing, National University of Singapore, Singapore 117543

Abstract

Access control has much attracted research interest recently. In this paper, we propose a method using human face and height as human trait to recognize a person. We observe that eye location extracted from a human face is sta-ble to be used to compute his/her height to the ground. Using it together with face recognition can increase the ac-curacy of the access control. We have implemented the method on a PC installed with a stereo camera. The design criteria, techniques, implementation details, and performance testing are presented

Keywords: 3D reconstruction, gradient descent method, biometric fusion, application.

1. Introduction

Access control has much attracted research interest recently. A reliable and secure system requires accurate and rapid human identification. Biometrics based systems show a great potential for the purpose, where even if an unauthorized person has got the password, he/she still can not get the access. Usually, a single feature may not be reliable enough for identification. Thus, multimodal biometric features are explored.

Access control systems based on face recognition are now a commonplace. The previous version of the system presented in this paper also uses only face recognition. Particularly, we use a pair of stereo cameras to track human heads and locate the eyes robustly. We observe that eye location as a feature of a human face can be used to compute the height. Using a stereo camera, we can get two images of eyes at the same time. Applying the 3D reconstruction technique, we can compute human's height which represents the distance from people's eyes to the ground. Hence we can integrate this feature into original face recognition system such that a better performance both in accuracy and speed can be achieved.

We have implemented the method. Two calibration matrices are derived for stereo camera calibration, one for intrinsic and another one for the external parameters. This allows us to compute the 3D position of the eyes. First, it is computed using the linear least-squares method. Next, the non-linear gradient descent method is used to minimize the distance between the measured image points and their re-projection after reconstruction. As such we obtain refined 3D measurement as well as the uncertainty interval. Finally, weighted summation fusion and decision tree are used to integrate the face and height features for access control. We need to set a proper threshold to minimize the false acceptance rate (FAR). Experimental study has been conducted.

2. Related work

The biometric information is important for human identification [BJ00]. To get better performance, biometric fusion is applied [FD00]. Work includes face fused with fingerprint and hand geometry [RJ03] and speech [San02]. One research issue is to combine and evaluate the multimodal biometric identification system [FD00]. Methods include sum rule, decision trees and linear discriminate function [RJ03], majority voting, ranked list combination and post-classifier [San02].

3. Our work

In this section, first, we describe how the human height is derived. Then, we present how it can be used together with face for human identification. The height is derived from eye location [HM00] followed the process of camera calibration [TV98, Zha00, SR01] using stereo images.

3.1 Camera calibration

The first step is to calibrate the camera, from which both the intrinsic (principal point, pixel width, and lens distortion parameter) and the external (rotation and translation) parameters of the camera are estimated.

Three coordinate systems, *camera, image* and *world*, are modeled. The image coordinate system is in pixel while the other two are in *mm*. The camera coordinate system is in 3D whose center is the camera's center and the ray-axis is the Z-axis. Image coordinate system is in 2D perpendicular to the ray-axis of the camera coordinate system. Its center is located at the top left corner. The world coordinate system is in 3D. We take the ground as the XZ plane with the Y-axis upwards. The center of world coordinate is usually no more than 2 meters far away from the camera center.

Hence for a point $(X, Y, Z)^T$ in world coordinate system, its corresponding point is $(u, v)^T$ in the image coordinates system. Now we must derive some equation to determine the mapping from the world coordinate $(X, Y, Z)^T$ to the image coordinate $(u, v)^T$. The mapping is divided into 2 parts, one is the mapping from the world coordinate $(X, Y, Z)^T$ to camera coordinate $(X_c, Y_c, Z_c)^T$, the external calibration. Another one is from $(X_c, Y_c, Z_c)^T$ to $(u, v)^T$, the internal calibration.

The intrinsic parameters of the camera can be represented as a simple matrix:

$$C = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where f, in *mm*, is the focal length from the center of camera to the center of image plane. (u_0, v_0), in pixel, is the principal point, which is the intersection between the optic axis and the image plane. k_u and k_v are the aspect ratios of the pixel in width and height respectively. The matrix C describes the mapping from the 3D camera coordinate system to 2D image plane:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}.$$

The external parameters of the camera can be represented as a transformation matrix E=[R|T] from the world to the camera coordinate systems:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}.$$

The basic process for camera calibration is as follows: for some points, such as corners of a chessboard [Zha00], in the world coordinate system, find their corresponding points in the image plane. The chessboard should be placed at least three different positions in space. In theory, 4 points are enough to find the calibration matrix but more object points help to improve precision. Usually a camera usually exhibits significant lens distortion. The conventional procedure is to compute the initial estimate using least-square method without considering distortion and then refine the result using a non-linear method to include the distortion estimation.

3.2 Derive the human height

The human height is defined as the Y coordinate of eye position $(X, Y, Z)^T$ in the world coordinate system computed after camera calibration. For the stereo camera we use, having the two transformation matrices, P and Q, and given the two feature points of eye centers s_1 and s_2,

$$P = \begin{bmatrix} p_{11}\ p_{12}\ p_{13}\ p_{14} \\ p_{21}\ p_{22}\ p_{23}\ p_{24} \\ p_{31}\ p_{32}\ p_{33}\ p_{34} \end{bmatrix}, \quad Q = \begin{bmatrix} q_{11}\ q_{12}\ q_{13}\ q_{14} \\ q_{21}\ q_{22}\ q_{23}\ q_{24} \\ q_{31}\ q_{32}\ q_{33}\ q_{34} \end{bmatrix},$$

$$s_1 = \begin{bmatrix} u_1 \\ v_1 \end{bmatrix}, and\ s_2 = \begin{bmatrix} u_2 \\ v_2 \end{bmatrix}.$$

We can use the following linear system to derive $(X,Y,Z)^T$:

$$\begin{bmatrix} p_{11}-u_1p_{31} & p_{12}-u_1p_{32} & p_{13}-u_1p_{33} \\ p_{21}-v_1p_{31} & p_{22}-v_1p_{32} & p_{23}-v_1p_{33} \\ q_{11}-u_2q_{31} & q_{12}-u_2q_{32} & q_{13}-u_2q_{33} \\ q_{21}-v_2q_{31} & q_{22}-v_2q_{32} & q_{23}-v_2q_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} u_1p_{34}-p_{14} \\ v_1p_{34}-p_{24} \\ u_2q_{34}-q_{14} \\ v_2q_{34}-q_{24} \end{bmatrix}.$$

Now, we can refine the result. The resulting $(X, Y, Z)^T$ is not accurate for two reasons. First we have not considered the effect of distortion. And second there should be some error in the 2D coordinates. Hence if we project $(X, Y, Z)^T$ from the world coordinate system

to the 2D image plane using camera calibration matrices:

$$\begin{bmatrix} u_1' \\ v_1' \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad and \quad \begin{bmatrix} u_2' \\ v_2' \\ 1 \end{bmatrix} = Q \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$

We can get two corresponding image coordinates $s_1'=[u_1',v_1']$ and $s_2'=[u_2',v_2']$ of two images (using the stereo camera). Compared to the original feature points s_1 and s_2, we have the distance $\Delta s_1=|s_1-s_1'|$ and $\Delta s_2=|s_2-s_2'|$. By minimizing Δs_1 and Δs_2, we can refine the eye position $(X, Y, Z)^T$. Here we apply gradient descent algorithm and Jacobian function. The major steps of this method are:

1. Add distortion coefficients to 2D image coordinates after projection from 3D world to 2D image plane.

2. Compute Δs_1 and Δs_2. These two values give us very important information. Since each pair of 2D points is associated with a pair of 3D points, i.e., Δs_1 and Δs_2 should be associated with the distance $\Delta W=|W-W'|$ in 3D, where $W=(X, Y, Z)^T$ and $W'=(X', Y', Z')^T$.

3.Consider the pair of 3D points the pair of 2D points are associated with a continue function $S=f(W)$ differentiation we got below equation:

$$\frac{\Delta S}{\Delta W} = \frac{S_1 - S_0}{W_1 - W_0} = f' \Rightarrow \Delta S = \Delta W \times f',$$

$$where\ \Delta S = \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \\ \Delta x_2 \\ \Delta y_2 \end{bmatrix} and\ \Delta W = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}.$$

4. Solve the above function and get a distance value ΔW, then we can optimize W by subtracting ΔW from it, then repeat step 1 using new W'. This loop ends when current ΔS is smaller than previous one, which means it gets the lowest point of the gradient function. W is our optimized 3D point.

3.3 Fusion of the face and height

Our biometric fusion system operates in two modes:

1. **Registration mode:** In the registration mode a user's biometric data (face and height) is acquired using our system as reader and stored in a database. The stored template is labeled with a user identity (name) to facilitate authentication.

2. **Authentication mode:** In the authentication mode, a user's biometric data is once again acquired and the system identifies who the user is. Identification involves comparing the acquired biometric information against templates corresponding to all users in the database.

Our system has three modules:

1. **Feature extractions module** in which facial features are extracted and the height is computed.

2. **Matching module** in which the face and height features are compared against those in the database.

3. **Decision-making module** in which the user's identity is established and the claimed identity is either accepted or rejected based on the matching scores generated in the matching module.

The performance of our system is measured by its false acceptance rate *FAR* and false rejection rate *FRR* at various thresholds.

In our system, the face and height features are generated and matched separately in the extraction module and matching module. The main concern is on the decision making module. Here we want to combine the two scores together to get better performance. Here we set some higher thresholds to decrease the FAR.

As we know, people's height may change from time to time. Shoes, standing position and head position may "change" the height. So we need to add some constraint to users. It requires that people should stand near to the camera (*0.5m to 2m*) and look at the camera lens. As we measure the distance from the ground to eyes, the result will not be affected by hair. The only affected factor the result is shoes.

It is clear that height alone cannot identify a person. So we cannot use it as the identifier but the classifier. It can be used to reduce the range of searching in database. Hence it can increase both speed and accuracy of our system. Because of the constraint, one of the better ways is combining the decision tree method and weighted summation together: The layout of the decision is shown below:

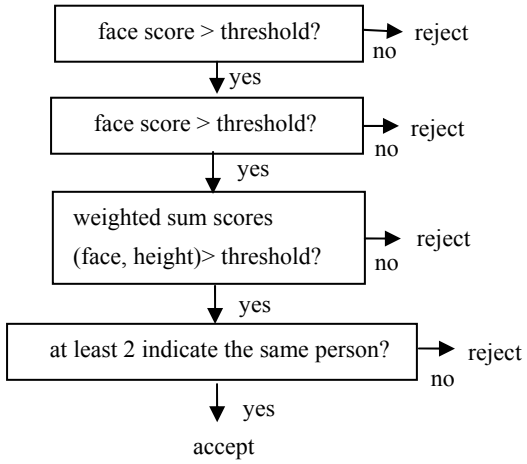


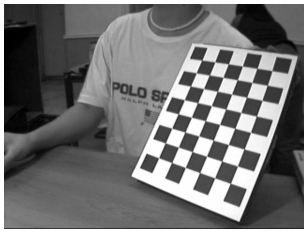
Figure 1: *The cascade identification process*

4. Implementation and results

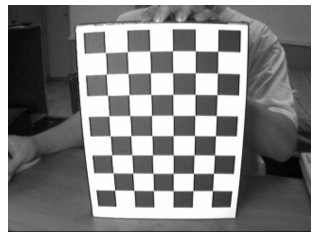
The system developed uses Digiclops and Triclops SDK library (http://www.ptgrey.com/products/digiclops), and Microsoft Visual C++ 6.0. It includes camera internal calibration using chessboard and external one using a T-shape pole. We will describe how to set up the threshold to accept and reject candidates in face feature recognition, and how to assign weight to each feature in weighted summation. We will also give the performance of our whole system.

4.1 Camera calibration

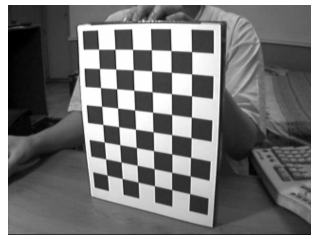
First, we show the results to extract the camera's intrinsic parameters. Here we use Intel OpenCV library (http://www.intel.com/research/mrl/research/opencv/index.htm) for calibration. A 9x7 chessboard of 48 corners is used as a modal plane (Figure 2). The size of each square is 35cm x 35cm. We track the chessboard and detect its corners. If all 48 corners are detected on an input image, this image will be considered successful. However, for accuracy, we need at least five such successful images. Those images in Figure 2 are successful (all from left camera).



(a)



(b)



(c)

Figure 2: *Using a chessboard for camera calibration*

For a group of 5 images, we have one internal calibration matrix and ten external calibration matrices (five for left camera and five for right camera). An example is shown as follows:

Left camera	Right camera
Distortion coefficients -0.28048 9.04166 0.00831 -0.01362	Distortion coefficients -0.28960 22.55838 0.00175 -0.04806
Internal calibration matrix 1688.73718 0.00000 293.77576 0.00000 1683.38306 282.47647 0.00000 0.00000 1.00000	Internal calibration matrix 1647.68567 0.00000 234.86148 0.00000 1635.55139 235.63890 0.00000 0.00000 1.00000

Table 1: *An example of camera intrinsic parameters from calibration*

To extract external parameters, we use a "T" shape pole as model (Figure 3), which consists of a horizontal pole (90 cm in length) and a vertical pole (190 cm in height). Each end of the horizontal pole connects to a line whose end connects to a mass. And there are a few marks on these 2 lines. The distance between each mark is 10 cm. This design makes us easy to define the world coordinates system: the bottom of this vertical pole is the origin, the vertical pole is the Y-axis, and the horizontal pole is parallel to the Z-axis.

Figure3: *T shape pole for extracting the external parameters of the camera*

4.2 Eye position

Our 3D reconstruction makes use of the gradient descent and the Jacobian function as described in subsection 3.2. We limit the iteration of gradient descent to 30 and use 4 distortion coefficients. In the experiments, usually after 5 or 6 iterations, we can find a satisfactory result.

Now we show the detailed experiments to test the performance of the gradient descent optimization algorithm. The experiments can be done in the same way that we obtain the internal matrix:

1. We select a group of 6 successful chessboard images from left camera and another group of 6 successful chessboard images from right camera. Without loss of generality, we take all images on different time and place the board at different position.

2. For each image, we record down all the corners' original 3D coordinates $(X_0, Y_0, Z_0)^T$ and obtain their corresponding 2D coordinates.

3. Do calibration using 3D and 2D coordinates, and we will find 2 intrinsic calibration matrices for two groups and 12 external calibration matrices for all images.

4. Then we use all corners' 3D coordinates $(X_0, Y_0, Z_0)^T$ to measure their distances to the real position $(X_1, Y_1, Z_1)^T$. For more flexible testing, we use different image combinations to do 3D reconstruction. Here, we give 3 different combinations: (1) both images come from left, (2) both images come from right, (3) one comes from left and another one comes from right. The formulas we use to measure distance are $d = \sqrt{(X_1 - X_0)^2 + (Y_1 - Y_0)^2 + (Z_1 - Z_0)^2}$ and $\Delta Y = |Y_1 - Y_0|$, because we only care about the Y value.

From experiments, we found that error of 3D reconstruction algorithm is quite small, around 1mm in average. To clearly show the performance of our 3D reconstruction algorithm, we examine every steps of the gradient descent method. From the figures below, we can find that it can get to the point of the minimum error.

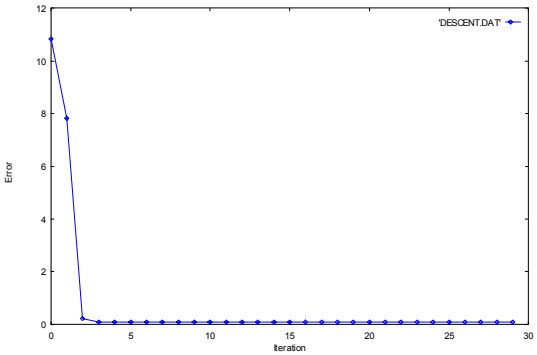


Figure 4: *Convergence test results*

Now we discuss the threshold for height comparison. The key point here is that there is uncertainty (or error) in the height estimation and it changes with the distance from center of eye to the center of camera. It is well known that the farther the person from the camera, the bigger the error. Therefore when setting the threshold for height comparison, we must know the range of reconstruction error. We compute the error for distance varies from 1m to 2m. We randomly choose 20 pairs of chessboard images in this region and 40 corner points from each image. Then in total we have 800 corner points in this region. For every pair of images (one from left, another from right, and both are taken at the same time; their coordinates are represented in the local coordinate system W_1 of the chessboard):

1. Using the T-shape modal to set up the world coordinate system W_0, and compute the best external calibration matrix E_1 and E_2 for left and right cameras respectively.

2. Record 40 corners' local coordinates $(X, Y, Z)^T$ in W_1, and image coordinates $(u, v)^T$ on both left and right images, 40 coordinates for left images and 40 for right ones.

3. Use the 3D and 2D coordinates recorded in step 2 to do calibration, and find the external calibration matrix E_1' and E_2' for each image plane in W_1.

4. Convert each corner's 3D coordinates from the local coordinate system W_1 to the world coordinate system W_0. For simplicity, we only consider the left image, hence we use E_1' and inverse of E_1. Assume that the Y-value got in this step is Y.

5. For each corner, use 2 corresponding 2D coordinates to do 3D reconstruction and get another *Y*-value, *Y'*.

6. For each corner, use $(X, Y, Z)^T$ and E_1' to get the 3D coordinate in the camera coordinate system, say $(X_c, Y_c, Z_c)^T$.

7. For each corner, construct a mapping from $d = \sqrt{X_c^2 + Y_c^2 + Z_c^2}$ to $\Delta Y = |Y\text{-}Y'|$.

The graph below shows the relation between ΔY and *d*: (x-axis is *d*, the range of *d* is [1000mm, 2000mm]).

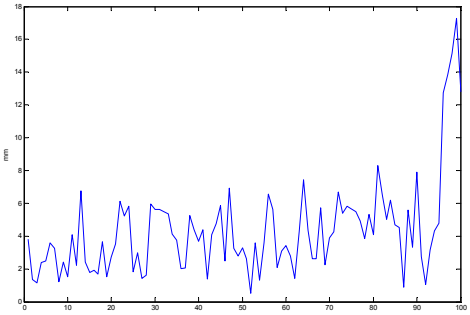


Figure 5: *The threshold determination. From the figure, we can see that most of time, the error is smaller than 1 cm. If we require the visitor stand no more than 2 meters from the camera, we can take 1 cm as the threshold for height comparison*

4.3 Biometric fusion

Before the system can perform authentication, we need to register member's biometric feature data. In the implementation, 5 groups of biometric feature data are acquired for each person. In each group of data, we record the left and right eyes' positions, image data, height and face feature data. Each person is identified with his/her name and ID. These data are stored into database as templates.

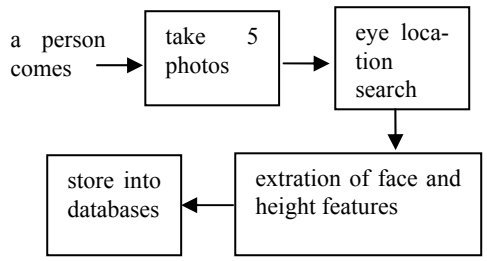


Figure 6: *Registration mode process*

In the authentication mode, we have 4 modules to perform the whole work. We use the graph below to show the details:

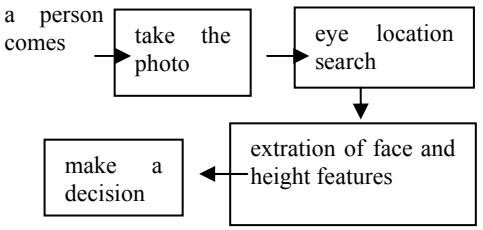


Figure 7: *Authentication mode process*

First, start tracking image. If an image has a head, it will be sent with its disparity image to find the eyes' position. Next, use the eye's position, we reconstruct the 3D points of eyes and get the height. At the same time, we extract the face feature for later comparison. At the last stage we will make the final decision. As described in subsection 3.3, the main idea is using decision tree and weighted summation.

Here we describe a method to define the threshold for face recognition. In our database, we have 69 people; each has 5 groups of face feature data from 5 image data (one face feature data is a 40x21 matrix).

1. For every people, we compute the mean value of face feature data. Next, compute the distance between the mean and the each data and for each distance we assign a score to it. This score is called genuine score. So we will have 5x69 scores.

2. For every two group, we compare the distance between their mean values. So we have score, which are called imposter scores.

3. We normalize those genuine scores and imposter scores to [0, 100] (Bigger is better).

4. Compute FAR and FRR for different threshold and compute the proper threshold.

The graph below is the PDF graph for imposter scores and genius scores:

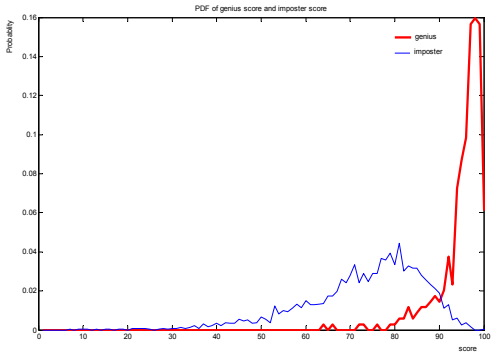


Figure 8: *It shows the PDF of imposter and genius scores. From the graph, we can see they are quite close*

Now we need to measure FAR and FRR. For different scores as thresholds, we will get different FAR and FRR. We need to define 2 thresholds, one for rejection and another one for acceptance. For the one used as rejection threshold, it must have a small FRR, but for acceptance threshold, it must minimize the FAR due to the requirement of a secure system.

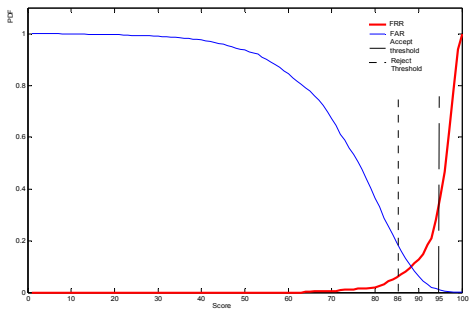


Figure 9: *The experiments to decide the acceptance and rejection thresholds*

From the above graph, we can define acceptance and rejection thresholds:

	Score	FRR	FAR
Rejection threshold	86	7.0%	16.1%
Acceptance threshold	95	28.1%	1%

Table 2: *Acceptance and rejection thresholds*

The rejection threshold is quite ok, since FRR is not quite big and is considerable. However, the acceptance threshold makes FRR too big since we need a quite small FAR. To decrease FRR, we can combine the height measurement with the face feature.

Now we give an experiment to show the performance of reducing the search space in the database using this threshold. Our database has 69*5 of height values. We take each one to compare with rest 4 height values in the same group, using the threshold 10mm and compute the FRR. For each height value, all 4 tests return as acceptance. We add the genius score. If any is rejected, we increase the imposter score. Below is the final result:

Genuine Score	685
Imposter Score	5
FRR	0.72%

Table 3. *Testing on the height threshold 10mm*

We use the same way to test performance of rejection using threshold, which means we examine the FRR for this threshold. Here we consider 2 cases. One uses the whole database; another one uses the cleared database by height threshold.

	Whole database	Cleared database
Genuine Score (>86)	679	81
Imposter Score (≤86)	11	1
FRR	1.59%	1.22%

Table 4. *Testing on the face rejection threshold: 86*

Now, we define and test w_1, w_2 and acceptance threshold T. In our linear weight summation fusion, w_1, w_2 and acceptance threshold are trained at the same time. We use a function to represent the whole scenario: For groups of genius scores (o_1, o_2) (o_1 is the face score vector, o_2 is the height score vector) and groups of combined imposter scores (o_1', o_2'), we use $F(o_1, o_2) = w_1o_1 + w_2o_2 - T$ to represent the acceptance/rejection scenario: if $F(o_1, o_2)>0$, we accept the visitor; otherwise, reject the visitor. Then we can compute FRR using $F(o_1, o_2)$ and FAR using $F(o_1', o_2')$.

FRR=(number of $F(o_1,o_2)<0$) / (number of genius score)

FAR=(number of $F(o_1',o_2')\geq 0$) / (number of imposter score)

TE = FRR + FAR.

As the requirement of the system, we need to minimize FAR and TE. We tried different w_1, w_2, T and compute all possible situations. At last we obtain the best combination at $w_1 = 0.9895$, $w_2 = 0.0105$, $T = 93$. The figure below gives the FAR and FRR for each score.

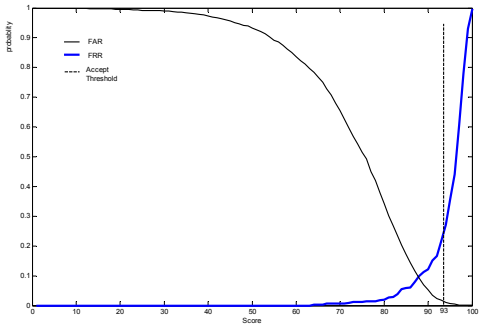


Figure 10: *The FRR and FRA values for each score*

Finally, the table below gives the measurement of the acceptance threshold.

	Threshold	FRR	FRA
W_1= 0.9895	93	16.37%	1%
W_1= 1	94	17.43%	1%

Table 5: *The measurement of the acceptance threshold. As we can see, the weight assigned to height is very small. We can consider discarding it. The performance of whole system will not be affected after the database reducing.*

Now we give the overall performance of our fusion algorithm and compare it with the performance of single feature identification. To make the comparison more standard, we fix the FRA to 1%.

	FRR	FRA
Single feature	28.1%	1%
Combined feature	11.2%	1%

Table 6: *The overall performance using face and height. From the table, we can see that, without change the FRA, we reduce the FRR so the performance of the system is improved.*

5. Conclusion and future work

In this paper, we proposed a human recognition method using human face and height. One of the key points is 3D reconstruction since it is related to the accuracy of the whole system. Only solving a linear computation, the accuracy is not high. We employ the gradient descent method to find the optimized value. At the same time, we use the Jacobian function to find the error interval of 3D reconstruction. For biometric fusion, we use decision tree together with the weighted summation. From the experiment, we find the previous system, which only uses face feature for recognition gives larger FRR when it minimizes FAR. After we integrate height value into the system, we can see from experiment result that the FRR is reduced rapidly while FAR does not change.

Since our database is small (we have not got bigger number of people to register), future work is to increase the database for more experimental study.

References

[TV98] E. Trucco and A. Verri, Introductory to Techniques for 3D Computer Vision, Prentice Hall, 1998.

[Zha00] Z. Zhang, A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330-1334, 2000.

[SR01] Z. Sun and C. Rayburn, Camera Calibration, Internship with UNR & Ford Motor Company (2001).

[HM00] W. Huang and R. Mariani, Face Detection and Precise Eyes Location, *ICPR'00*, Volume 4, September 03-08, 2000 Barcelona, Spain.

[PMWP00] P. J. Phillips, A. Martin, C. L. Wilson and M. Przybocki, An Introduction to Evaluating Biometric Systems, *IEEE Computer*, 33(2), February, 2000, pp. 56-63.

[BJ00] P. Bolle and A Jain, Biometrics: The Future of Identification, *IEEE Computer*, 46-49, February, 2000.

[FD00] R. Frischholz and U. Diechmann, BioID: A Multimodal Biometric Identification System, *IEEE Computer*, 33(2), pp. 64-68, February 2000.

[RJ03] A. Ross and A. Jain, Information fusion in biometrics, *Pattern Recognition Letters*, 24(13), pp. 2115-2125, September 2003.

[San02] C. Sanderson, Automatic person verification using speech and face information, PhD Thesis, Griffith University, Australia, August 2002