Online 3D Signature Verification by using Stereo Camera & Tablet

Jay Dave
Indian Institute of Technology
Dept. of Electrical Engineering
India (208016) , Kanpur
jay.dave.1988@gmail.com

KS Venkatesh
Indian Institute of Technology
Dept. of Electrical Engineering
India (208016) , Kanpur
venkats@iitk.ac.in

Garima Jain
Indian Institute of Technology
Dept. of Electrical Engineering
India (208016) , Kanpur
gari1217@gmail.com

Abstract

The signature of a person is an important biometric attribute which can be used to authenticate human identity. Conventional online approaches to signature verification only use either a single camera to track the pen tip position or a tablet to extract the dynamic features of the signature, hence the signature has only two spatial dimensions. In this paper we combine data inputs from a pressure sensitive device (tablet digitizer) and stereo vision to record signatures in 3D. Stereo vision from a pair of low cost SONY Eyecam cameras is used to track the pen tip position in x, y, & in z when the pen is off the surface as well as the pen angle with respect to the surface at all times. The digitizing tablet on the other hand, tracks x, y as well as pressure magnitude (which we denote as $-z$) when the pen contacts the surface. In all, we record the following parameters as functions of time through the duration of the signature: x, y, z, θ, ϕ, where all the linear paramaters are bipolar, with the particular case of z representing motion with positive values and pressure level with negative values. The angular values are two dimensional. The distance between the input signature's features recorded as a 5-variate parameter time sequence and the template signature's features whichwere collected during the training phase is computed using Dynamic Time Warping (DTW), and is thresholded to take a decision. While better learning techniques and more intensive experimentation will help suggest improvements, even as of the present, we have a fully working prototype of the system.

Keywords

stereo triangulation; feature vector; dynamic programming matching; stereo camera; pressure digitizing

1 INTRODUCTION

Signature verification is one of the behavioural biometrics which is commonly used to identify human beings. Signatures are very useful for identification purpose as they are very unique, especially if we consider the dynamic features of the signature in addition to the its static features.

Online signature verification techniques can be classified into two methods: function based and parameter based [yasuda2010]. In the function based approach, the features of the signatures are extracted as a function of time. For example, x position with respect to time $x(t)$, y position with respect to time $y(t)$, pressure with respect to time $p(t)$, etc. In the parameter based approach, the signature is represented as a vector of elements, each one carrying the value of the feature. The parameter based approach is further classified into global parameters (total signature time duration, number of pen ups/downs, etc.) and local parameters (speed at certain bending points, the pen direction when the signature finishes, etc.). In general, the function based approach results in better verification performance compared to the parameter based approach, but is more time consuming due to the rigorous matching procedure.

Nalwa [nalwa1997] proposed an algorithm based on the shape of the signature rather depending on the pen dynamics. The author proposed that pen dynamics have very high intra-class variability which makes the use of the features, extracted from the pen, impractical. He used global features such as aspect ratio of the signature, jitter and local features such as spatial torque, coordinates relative to the center of mass, etc. Pippin [pippin2004] proposed a new method by applying separate filters to the global features and the local features. Feng [feng2003] proposed a new warping method for verification. He used the functional approach and used extreme points warping (EPW) for verification. EPW warps only a set of important points, and hence the time complexity is less as compared to Dynamic Time Warping (DTW).

Munich [munich2003] proposed a visual system for signature verification. An ordinary camera tracking the spatial position of the pen tip in each frame was used as an input device to the system. Dynamic Time Warping (DTW) and Continuous Dynamic Time Warping (CDTW) techniques were used to match the signature.

Kumiko Yasuda [yasuda2010] also proposed a visual system for signature verification. He used seven webcams as input devices to the system and a sequential Monte Carlo (SMC) method to track the pen tip in each frame.

Nidal S. Kamel [kamel2008] proposed a glove based signature verification method. He used the glove as an input device to the system. He proposed to use Singular Value Decomposition (SVD) as a numerical tool for matching signatures.

In this paper, we demonstrate a novel approach for signature verification, in which we use a stereo camera setup along with a pressure digitizer to make and verify signatures in 3D. A stereo camera setup gives the 3D trajectory of the pen tip position, i.e., $X, Y, +Z$ in each frame. The pressure digitizer gives pressure information when the pen is in contact with the surface during which time the pen tip has insignificant Z motion. We consider pressure information as the $-Z$ component of the 3D trajectory. By combining both, we get integrated representation for the entire signature. This is called a feature level fusion method. In this method we have also included the pen's inclination in terms of θ and ϕ to make our system more discriminative and robust.

The paper is organized as follows: Section 2 describes the system used by us for online signature verification. It includes the description of the hardware, i.e., a stereo camera setup and a pressure digitizing tablet as well as the technique of stereo calibration. Section 3 illustrates the algorithms for feature vector generation. Section 4 deals with the dynamic time warping algorithm and decision making. The experimental results are presented in Section 5. Section 6 concludes the paper with the discussion on future scope.

2 SYSTEM DESCRIPTION

The whole signature verification process is divided into two phases:

1. Database collection phase: In this phase, the database of all the users is created. We obtain multiple signatures' patterns of each user by taking their signatures at different times, since any user cannot exactly replicate his signature without variations.
2. Verification phase: In this phase, the user's signature is compared to the database. The distance is calculated between the input and the stored template signature vectors. A threshold is used to make a decision.

2.1 Stereo Camera & Calibration

A stereo camera system works on the concept of stereo vision to get the depth information of a scene. It has two or more lenses with separate image sensors for each lens. Stereo camera setup can be made by using two ordinary cameras. Both the cameras which are located at two different places, take individual images of the same scene. The 3D view of the scene is created by combining these two images. Our stereo camera setup, using two PS3 Eye cameras is shown in Fig. 1. These cameras are fixed on the aluminium sheet and the positions of their lenses are secured with an aluminium plate.

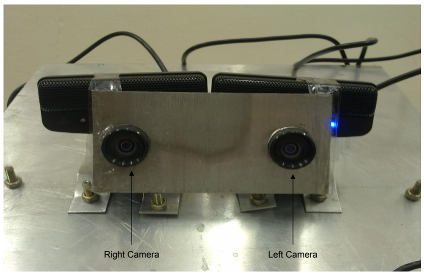


Figure 1: *Stereo Camera setup used in the Experiment*

The stereo camera calibration is the backbone of this project, as the calculation of the 3D world coordinates of the scene is on the basis of calibration. For the final product design, given sufficient standardization and precision, calibration can be restricted to the production stage. Calibration is used to obtain the "Projection matrix" and the "Distortion parameters" of the camera.

We have used the "Four-step Camera Calibration Procedure" as proposed by Heikkila & Silven [heikkila1997] for camera calibration. This method was implemented using the Camera Calibration Toolbox developed for MATLAB [matlabcameratoolbox] which is based on the OpenCV implementation.

2.2 Calibration Result

Intrinsic parameters of the left and right cameras are given in Table 1. Here focal lengths (f_x, f_y) and principle points (C_x, C_y) are given in terms of pixel units. (k_1, k_2) shows the Radial Distortions and (p_1, p_2) shows the Tangential Distortion. Extrinsic parameters, i.e., position of the right camera with respect to the left camera are: rotation vector $(\theta_x, \theta_y, \theta_z) = (-0.01405, 0.19919, -0.11265)$ and translation vector $T = (T_x, T_y, T_z) = (-77.55565, 7.00082, 16.28213))$ (in mm units). Rotation matrix R can be found by using rotation vector.

 ISBN 978-80-86943-65-7

Parameters	Left Camera	Right Camera
(f_x, f_y)	$(755.2, 754.33)$	$(765, 765)$
(C_x, C_y)	$(313, 267)$	$(320, 236)$
(k_1, k_2)	$(-0.081, 0.278)$	$(-0.079, 0.222)$
(p_1, p_2)	$(0.006, 0.001)$	$(-0.003, -0.001)$

Table 1: Intrinsic parameters of Left and Right Cameras

2.3 Pressure Digitizing Tablet

The tablet [geniustablet] used as a pressure digitizing device is shown in Fig. 2. The tablet measures the pressure of the pen tip on the scale of 10 bits, i.e., the maximum pressure level is 1023. The tablet also measures the trajectory of the pen tip while writing on it.

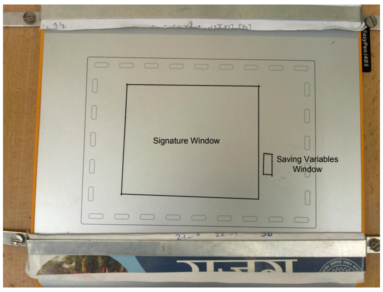


Figure 2: *Pressure Digitizing Tablet*

We have made two windows on the tablet surface for user's ease.

1. Signature window: Users have to sign in this window only. This window size is application dependent.
2. Saving Variables window: The user has to press this window once the signing is complete, to save all the features provided by the tablet.

3 FEATURE VECTOR GENERATION

3.1 Pen tip Detection and Tracking Algorithm

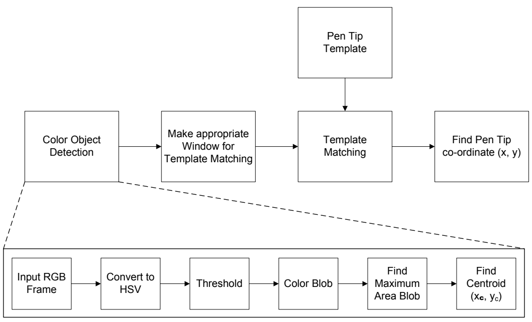


Figure 3: *Block Diagram of Pen tip Detection algorithm*

Fig. 3 shows the block diagram of the pen tip detection algorithm. We have used the template matching method for detecting exact pen tip location followed by color object detection as shown in Fig. 4. The centroid of the blob is found using [bradski1998].

$$x_c = \frac{M_{10}}{M_{00}}; \quad y_c = \frac{M_{01}}{M_{00}}$$

where $M_{00} = \sum_x \sum_y I(x,y)$ is the zeroth moment. $M_{10} = \sum_x \sum_y xI(x,y)$ and $M_{01} = \sum_x \sum_y yI(x,y)$ are the first moment.

3.2 Stereo Triangulation

The 3D coordinates of the object are calculated by using stereo triangulation. This is also known as 3D recovery [hillman2005]. In order to get the 3D coordinates of the object, we have to back project the line of the pixel in the left camera and the right camera as shown in Fig. 5. In this way we apply the inverse projection matrix to get from the 2D image point to the 3D line. These lines are the 3D back projection lines which usually meet at exactly one point.

For simplicity, we have taken left camera coordinate system as the world coordinate system as shown in Fig. 5. Hence, the left camera center becomes the world's origin $(0,0,0)$ and all the three axes of the left camera becomes the axes of world coordinate system respectively. Now the relation between the right camera coordinate system and the left camera coordinate system is:

$$C_r = RC_l + T$$

where $C_l = (X_l, Y_l, Z_l)$ and $C_r = (X_r, Y_r, Z_r)$ are object point location with respect to the left and the right camera respectively, R = Rotation Matrix which shows the rotation between the right camera coordinate system and the left camera coordinate system, and $T = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}'$ = Translation Matrix which shows the translation between the left camera coordinate system and the right camera coordinate system.

We can write the above equation as:

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (1)$$

Now from the Inverse Perspective Transformation:

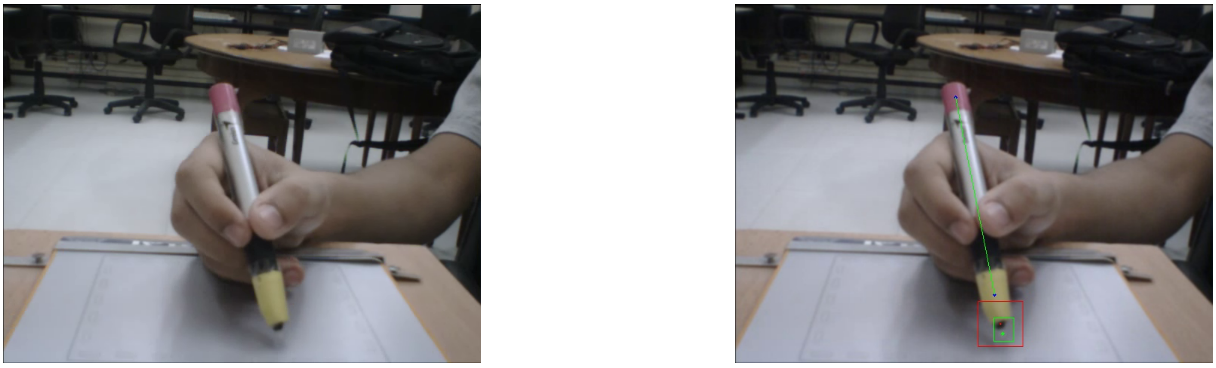
$$X_l = \frac{x'_l * Z_l}{f_{x_l}}, \quad Y_l = \frac{y'_l * Z_l}{f_{y_l}} \quad (2)$$

where $x'_l = x_l - C_{x_l}, \quad y'_l = y_l - C_{y_l}$

$$\text{and} \quad X_r = \frac{x'_r * Z_r}{f_{x_r}}, \quad Y_r = \frac{y'_r * Z_2}{f_{y_r}} \quad (3)$$

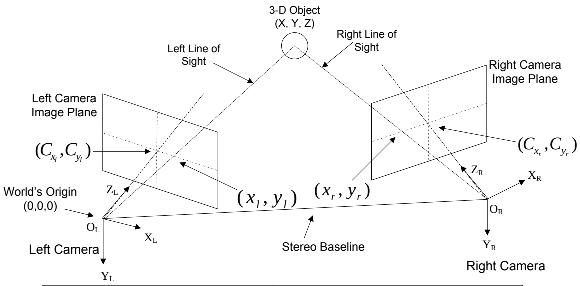
where $x'_r = x_r - C_{x_r}$, $y'_r = y_r - C_{y_r}$, (f_{x_l}, f_{y_l}) and (f_{x_r}, f_{y_r}) are the focal lengths of the left and right camera in the x and y direction respectively.

 ISBN 978-80-86943-65-7



(i) *Input BGR Frame* **(ii)** *Detected Pen tip point as a green point*

Figure 4: *Pen tip detection*



(X_L, Y_L, Z_L) = Left Camera Axes	(X_R, Y_R, Z_R) = Right Camera Axes
(C_{x_l}, C_{y_l}) = Principle Point of Left Camera	(C_{x_r}, C_{y_r}) = Principle Point of Right Camera
(x_l, y_l) = Image Point in Left Camera	(x_r, y_r) = Image Point in Right Camera
(X, Y, Z) = World Coordinate Axes = (X_L, Y_L, Z_L)	

Figure 5: *Reconstruction of 3D Point in Space*

3.3 Tablet Feature Vector

The tablet gives the pressure information p along with the x and y position of pen tip at the rate of 100 Sa/s. We include time-stamp with these features. We use the system time for the time-stamp. Hence, the system records $\{x(i), y(i), p(i)\}$ feature values approximately at each time $t = 10ms$, and the tablet feature vector becomes $F_{tab} = \{t_{tab}(i), x(i), y(i), p(i)\}$.

The 2D reconstruction of the signature using tablet features is shown in Fig. 6.

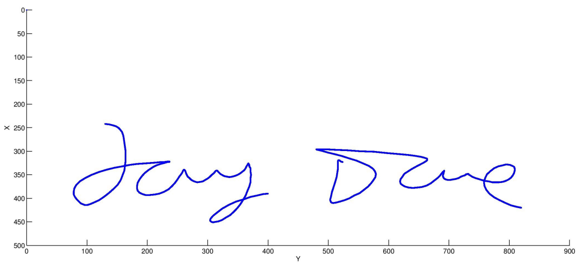


Figure 6: *2D reconstruction of signature using tablet features* $\{x(i), y(i)\}$

3.4 Camera Feature Vector

The stereo camera setup gives the continuous 3D trajectory of the pen tip location by applying stereo triangulation. We have added two trackers to get two 3D positions, where the first corresponds to the pen tip position $\{X_t, Y_t, Z_t\}$ and the second corresponds to the pink marker $\{X_h, Y_h, Z_h\}$ which sticks on the head of the pen. Hence we get the two end points of the pen. We can find the orientation of the pen by using these two 3D points with respect to the tablet surface. Pen orientation can be calculated as:

$$\theta = cos^{-1}\left(\frac{Z_h - Z_t}{r}\right); \qquad \phi = tan^{-1}\left(\frac{Y_h - Y_t}{X_h - X_t}\right)$$

where $r = \sqrt{(X_h - X_t)^2 + (Y_h - Y_t)^2 + (Z_h - Z_t)^2}$

We have also added the time stamp along with these features. The PS3 Eye camera works at 60 fps. Hence the system records camera feature values at approximately every $t = 17$ ms. The final camera feature vector becomes $F_{cam} = \{t_{cam}(i), X_t(i), Y_t(i), Z_t(i), \theta(i), \phi(i)\}$.

The 3D reconstruction of the signature using camera features is shown in Fig. 7.

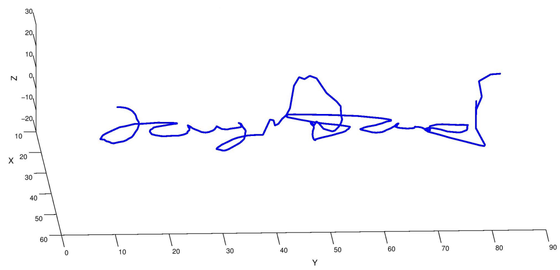


Figure 7: *3D reconstruction of signature using the camera features*

There are significant differences between feature values of the two feature vectors on account of the differences between the respective sensors' ranges and resolutions. This necessitates feature normalization. The main objective of a feature normalization process is to

 ISBN 978-80-86943-65-7

modify the mean and variance of the feature values by applying suitable transformation functions (max, min-max, median, z-Score, etc.). The normalization process maps the feature values of different feature vectors into a common domain. In our algorithm, we use Min-Max normalization for normalizing signature features from the two sensors. which is defined as:

$$a_i' = \frac{a_i - \text{Min}}{\text{Max} - \text{Min}}$$

where, a_i and a_i' denote the i^{th} feature value before and after normalization process respectively, $\text{Max} = \max_i\{a_i\}$, finds the maximum value, and $\text{Min} = \min_i\{a_i\}$, finds the minimum value.

3.5 Synchronization

In our experiment, we obtain the tablet feature vector at 100 Sa/s and stereo camera feature vector at 60 fps. To combine these feature vectors, we normalize each of them by using Min-Max normalization technique, and then use linear interpolation for the purpose of synchronization. After the synchronization, time stamps for both the feature vectors will become identical, i.e., $t(i) = t_{tab}(i) = t_{cam}(i)$.

After normalization and synchronization, we can combine the two different feature vectors. This combining process is known as feature level fusion. We have used feature level fusion for combining F_{cam} and F_{tab}. We have made one common feature vector which contains $\{t(i), x(i), y(i), p(i), \theta(i), \phi(i)\}$ when the pen tip touches the tablet surface and $\{t(i), X_l(i), Y_l(i), Z_l(i), \theta(i), \phi(i)\}$ when the pen tip is poised above the surface without contact. Hence, our final feature vector after feature level fusion becomes $F_{Final} = \{t(i), X(i), Y(i), Z(i), \theta(i), \phi(i)\}$.

The 3D reconstruction of the features generated from the feature fusion method is shown in Fig. 8, where blue is used under the pen down condition and red color under pen up. All the features are shown as a function of time in Fig. 9.

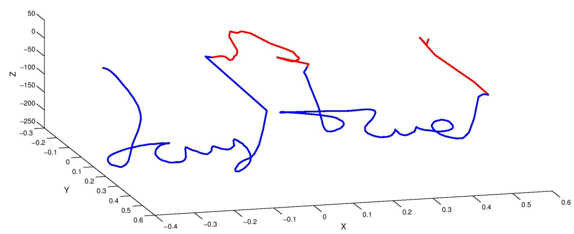


Figure 8: *Reconstruction of 3D signature made by Feature Fusion method*

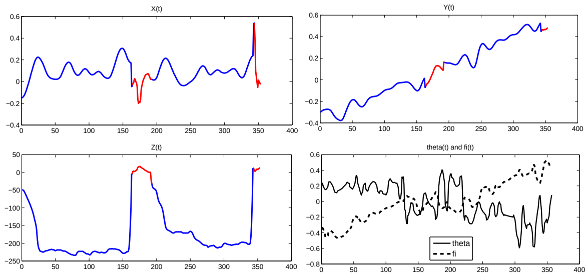


Figure 9: *Final features made by feature fusion as a function of time*

4 SIGNATURE VERIFICATION ALGORITHM

4.1 Dynamic Programming Matching (DPM)

We can summarize DPM by following way:

1. Initialization :

$$dist(1,1) = 0; \qquad \xi(1,1) = (1,1)$$

where, $dist(x,y)$ = total distance upto (x,y) point and $\xi((n_x,n_y),(n_{x'},n_{y'}))$ shows the warping path between nodes (n_x,n_y) and $(n_{x'},n_{y'})$.

2. Recursion : for $1 \le i \le N_a, 1 \le j \le N_b$, such that i and j must follow the monotonicity constraint,

$$dist(i,j) = min \begin{cases} dist(i-1,j) + d((i-1,j),(i,j)) \\ dist(i-1,j-1) + \\ \quad d((i-1,j-1),(i,j)) \\ dist(i,j-1) + d((i,j-1),(i,j)) \end{cases}$$

$$\xi(i,j) = argmin \begin{cases} dist(i-1,j) + d((i-1,j),(i,j)) \\ dist(i-1,j-1) + \\ \quad d((i-1,j-1),(i,j)) \\ dist(i,j-1) + d((i,j-1),(i,j)) \end{cases}$$

where, $d(,)$ is the Euclidean norm function.

3. Termination :

$$Dist(S_1,S_2) = dist(N_a,N_b); \qquad \theta_1 = (N_a,N_b)$$

Here $Dist(S_1,S_2)$ represents the final distance between two signals.

4.2 Signature Verification using DPM

It is practically impossible for a user to produce exactly the same signature every time. Hence we can not compare two signatures and find the distance between them by using a simple Euclidean distance formula. DPM (Dynamic Programming Matching) [DPBellman] is a method that finds correspondences between two signatures. It takes each sample in the 1^{st} signature and finds

ISBN 978-80-86943-65-7

the closest sample in the 2^{nd} signature using a predefined metric. Given this similarity, it is possible to estimate a distance between the two signatures.

The warping function for two signatures is not linear. Here the signature varies with time, hence DPM is called Dynamic Time Warping (DTW).

Fig. 10 shows one example of signature verification process. Two different signatures of the same user is shown in Fig. 10i. 1^{st} signature is shown by red line and 2^{nd} signature is shown by blue line. Only min-max normalized x position $x_i(t)$ and min-max normalized y position $y_i(t)$ of the signatures are considered here as a feature vector (see Fig. 10ii and Fig. 10iii). Then DTW algorithm is applied to calculate the distance between $x_1(t)$ & $x_2(t)$ and $y_1(t)$ & $y_2(t)$. After applying DTW, we get warped $x(t)$ and $y(t)$ sequences as shown in Fig. 10iv and Fig. 10v respectively. The optimal correspondence paths for both the sequences are also shown in Fig. 10vi and Fig. 10vii. The distance between two x sequences is $Dist_x = 0.0437$ and between two y sequences is $Dist_y = 0.1834$. And the overall distance is $Dist_{overall} = 0.1885$ which is calculated by using equation (4). In general if we have N different sequences, the overall distance is calculated by:

$$Dist_{overall} = \sqrt{\sum_{i=1}^{N} Dist_i^2} \qquad (4)$$

4.3 Threshold Selection

Threshold value selection is a very critical task in the verification process as it affects the classification accuracy. A high threshold value increases the FAR (False Acceptance Rate) of the system and a low threshold value increases the FRR (False Rejection Rate). As explained in [jain2002], we can choose either a global threshold for all the users or a user dependent threshold.

We have used a global threshold based method to find the threshold value in our algorithm. The global threshold can be calculated as :

$$\text{Threshold} = \frac{\sum_{i \neq j, i<j} \text{dist}(S_r^i, S_r^j)}{\frac{N(N-1)}{2}} \times \gamma$$

where, γ is the adjustment factor, N is the total number of reference signatures, S_r^i is the i^{th} reference signature, $Dist(:,:)$ function finds the distance between two signatures.

In general, if we have N different sequences, the overall distance is calculated by:

$$dist_{overall} = \sqrt{\sum_{i=1}^{N} dist_i^2}$$

4.4 Decision Making

In this step we identify the user as fraud or genuine. The distance is calculated between user's unknown signature S_u and all the reference signatures S_r^i of that user. If the calculated distance is lower than the threshold value, then the signature is a genuine otherwise it is a forged one. Alternate formulations of the decision criterion are possible.

$$\begin{cases} S_u\text{= genuine if} & \frac{1}{N}\sum_{i=1}^{N} dist(S_r^i, S_u) < \text{Threshold} \\ S_u\text{= fraud if} & \frac{1}{N}\sum_{i=1}^{N} dist(S_r^i, S_u) \geq \text{Threshold} \end{cases}$$

5 EXPERIMENTAL RESULTS

We have used the setup shown in Fig. 11. The stereo camera setup is mounted on an aluminium sheet, which is fixed on the wooden plank at about 45° angle, to enable the cameras to see the tablet surface and the head of the pen. The stereo camera and the tablet are fixed rigidly to maintain calibration.

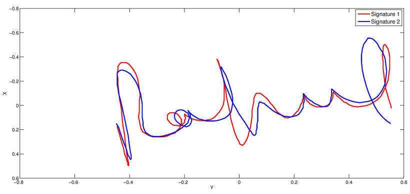
We have collected signatures from 8 users. Table 2 shows the number of signatures used for database collection, verification and forgery.

We compute Equal Error Rate (EER) of each feature individually, separately for the stereo camera setup as well as the pressure digitizing tablet, to evaluate the accuracy of each feature. We also compute EER of each feature individually of the feature level fusion method. The value of γ is set to be 1.5.

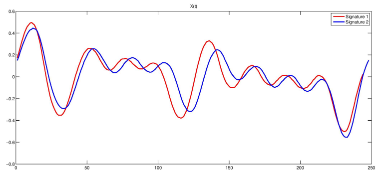
- Camera Features : Table 3 shows the FAR, FRR and EER values of the camera features. Fig. 12 shows the error trade off curve of camera features.
- Tablet Features : Table 5 shows the FAR, FRR and EER values of the tablet features. Fig. 13 shows the error trade off curve of tablet features.
- Feature level Fusion method : Table 4 shows the FAR, FRR and EER values of the all the features calculated using the feature level fusion method. Fig. 14 shows the FAR and the FRR curves and Fig. 15 shows the error trade off curve for feature level fusion method.

6 CONCLUSIONS AND FUTURE WORKS

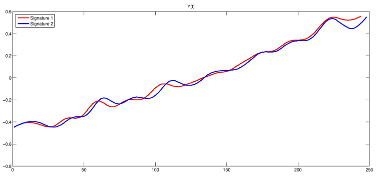
We have presented a novel low-cost approach to online signature verification method, built with off the shelf components such as a pressure digitizing tablet and a stereo camera pair. The tablet gives the pressure information as well as the pen tip position. The stereo camera setup gives the 3D trajectories of the pen tip.



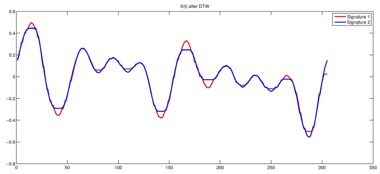
(i) *Two different signatures of the same user*



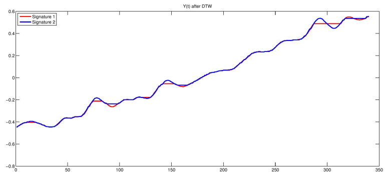
(ii) *$x(t)$ of two signatures*



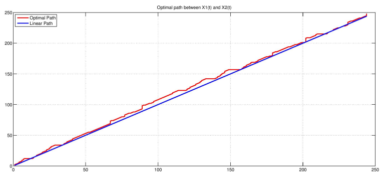
(iii) *$y(t)$ of two signatures*



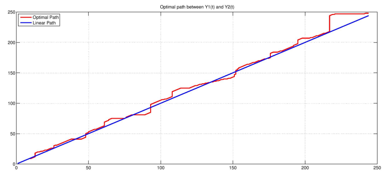
(iv) *$x(t)$ of two signatures after applying DTW*



(v) *$y(t)$ of two signatures after applying DTW*



(vi) *Optimal correspondence between $x_1(t)$ and $x_2(t)$*



(vii) *Optimal correspondence between $y_1(t)$ and $y_2(t)$*

Figure 10: *Matching of signatures parameters using Dynamic Time Warping (DTW)*

	Total Signatures per User	Total Signatures
For Database collection	10	80
For Verification purpose	15	120
Fraud Signature	6	48

Table 2: Total number of signatures

Features	X	Y	Z	θ	ϕ	*Overall*
FRR	25	20.83	20.83	12.5	20	14.17
FAR	50	41.67	52	43.75	31.25	31.25
EER	36.27	33.45	40.93	28.62	22.39	20.54

Table 3: Evaluation of Camera Features

Features	X	Y	Z	θ	ϕ	*Overall*
FRR	20	20	14.17	12.5	20	8.33
FAR	14.58	18.75	20.83	43.75	31.25	8.33
EER	14.58	19.58	16.55	28.62	22.39	8.33

Table 4: Evaluation of Features of feature level fusion method

Features	X	Y	P	*Overall*
FRR	21.67	30.83	30.83	21.67
FAR	6.25	6.25	16.67	4.17
EER	15.41	16.46	20.83	12.5

Table 5: Evaluation of Tablet Features

In our analysis, we found EER = 20.54% for the camera features and EER = 12.5% for the tablet features. For the proposed feature level fusion method, FAR = 8.33%, FRR = 8.33% and EER = 8.33%. These figures are likely to improve considerably with a better finished prototype.

A larger database can significantly decrease the FAR error rate. Pen tip tracking was done by color blob detection followed by template matching. Color blob detection is affected by the background color, and hence we should try and make the pen tip tracking independent of color blob detection. Accurate techniques like Kalman filter for pen tip detection and tracking can increase the precision and accuracy of the system. We considered only local features of the signatures for verification and

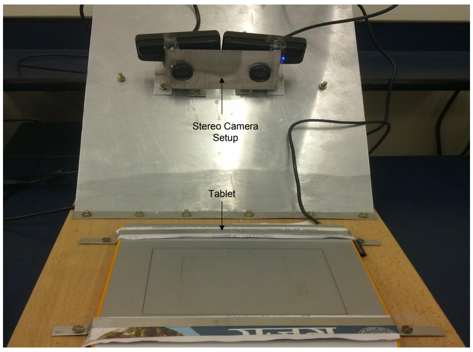


Figure 11: *Experimental setup*

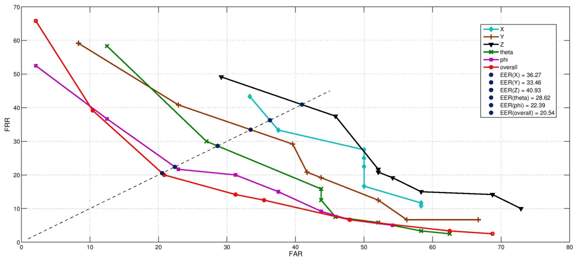


Figure 12: *Error Trade off Curve of Camera features*

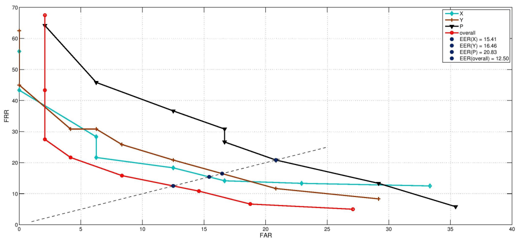


Figure 13: *Error Trade off Curve of Tablet features*

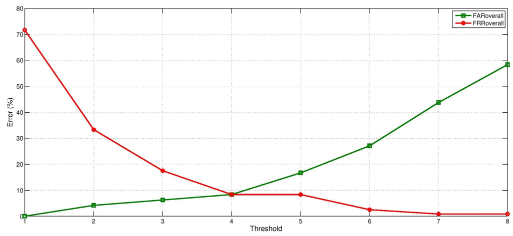


Figure 14: *FAR and FRR curves of the Feature Fusion method*

evaluation. We can additionally use global features like velocity, acceleration, etc., to make the system more reliable and robust. The use of a user dependent threshold can also result in lower error rates.

7 REFERENCES

[yasuda2010] Yasuda, K, Daigo M, Satoshi S, and Takashi M. Visual-based online signature verification using features extracted from video. Journal of Network and Computer Applications 33, no. 3 (2010): 333-341.

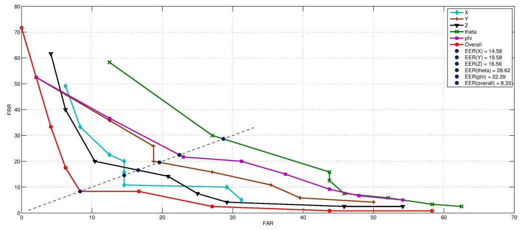


Figure 15: *Error Trade off Curve of the feature fusion method*

[nalwa1997] Nalwa, Vishvjit S. Automatic on-line signature verification. Proceedings of the IEEE 85, no. 2 (1997): 215-239.

[pippin2004] Pippin C.E., Dynamic signature verification using local and global features, *Georgia Inst. Inform. Technol, Atlanta, Technical report*, 2004.

[feng2003] Feng, Hao, and Chan Choong Wah. Online signature verification using a new extreme points warping technique. Pattern Recognition Letters 24, no. 16 (2003): 2943-2951.

[munich2003] Munich, Mario E., and Pietro Perona. Visual identification by signature tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on 25, no. 2 (2003): 200-217.

[kamel2008] Kamel, Nidal S., et al. Glove-based approach to online signature verification. Pattern Analysis and Machine Intelligence, IEEE Transactions on 30, no. 6 (2008): 1109-1113.

[heikkila1997] Heikkila, Janne, and Olli S. A four-step camera calibration procedure with implicit image correction. In Computer Vision and Pattern Recognition. Proceedings., IEEE Computer Society Conference on, pp. 1106-1112. IEEE, 1997.

[matlabcameratoolbox] J. Y. Bouguet, Camera calibration toolbox for Matlab, 2008.

[geniustablet] http://www.geniusnet.com/ Genius tablet i405

[DPBellman] R. Bellman, Dynamic Programming, Princeton University Press, 1957.

[jain2002] Jain, Anil K., Friederike D. Griess, and Scott D. Connell. On-line signature verification. Pattern recognition 35, no. 12 (2002): 2963-2972.

[hillman2005] Hillman, Peter. White paper: Camera calibration and stereo vision. Lochrin Terrace, Edinburgh EH3 9QL, Tech. Rep (2005).

[bradski1998] Bradski, Gary R. Computer vision face tracking for use in a perceptual user interface. (1998).