An Efficient Method for Gait Recognition

Rupali Patharkar
M.Tech (Student)
Department of CSE
PIET, Nagpur

S.T. Bodkhe
Assistant Professor
Department of CSE
PIET, Nagpur

A.R. Mahajan, Ph.D
HOD & Professor
Department of CSE
PIET, Nagpur

ABSTRACT
Human identification by gait has created a great deal of interest in computer vision community due to its advantage of inconspicuous recognition at a relatively far distance. Biometric systems are becoming increasingly important, since they provide more reliable and efficient means of identity verification. Biometric gait Analysis (i.e. recognizing people from the way they walk) is one of the recent attractive topics in biometric research. It has been receiving wide attention in the area of Biometric. Human gait recognition works from the observation that an individual's walking style is unique and can be used for human identification. Recognize individuals walking characteristics is main part in this paper. In Gait biometric research there are various gait recognition approaches are available. In this paper, the gait recognition approaches such as using Ica".

General Terms
Similarity measure classifier is used.

.Keywords
Biometrics, Survey, Gait Recognition approaches, Histogram Contextualization.

1. INTRODUCTION
Gait recognition is a relatively new biometric technology which aims to identify people at a distance by the way they walk. Human gait recognition works from the observation that an individuals walking style is unique and can be used for human identification. So as to recognize individuals walking characteristics, Gait recognition includes visual cue extraction as well as classification. Gait recognition can be classified into three groups namely; motion vision based, wearable sensor based and floor sensor based [1].

Two common categories of gait recognition are appearance-based and model-based approaches. Among the two, the appearance-based approaches suffer from changes in the appearance owing to the change of the viewing or walking directions . But, model-based approaches extract the motion of the human body by means of fitting their models to the input images. Model based ones are view and scale invariant and reflect in the kinematic characteristics of walking manner [2].

Several techniques for moving object detection have been proposed in [3]-[12], among them the three representative approaches are temporal differencing, background subtraction and optical flow. Temporal differencing based on frame difference, attempts to detect moving regions by making use of the difference of consecutive frames (two or three) in a video sequence. This method is highly adaptive to dynamic environments, but of certain types of moving objects. Background subtraction is the most commonly used approach in presence of still cameras. The principle of this method is to use a model of the background and compare the current image with a reference. In this way the foreground objects present in the scene are detected. The method of statistical model based on the background subtraction is flexible and fast, but the background scene and the camera are required to be stationary when this method is applied. Optical flow is an approximation of the local image motion and specifies how much each image pixel moves between adjacent images. It can achieve success of motion detection in the presence of camera motion or background changing. According to the smoothness constraint, the corresponding points in the two successive frames should not move more than a few pixels. For an uncertain environment, this means that the camera motion or background changing should be relatively small. The method based on optical flow is complex, but it can detect the motion accurately even without knowing the background. The main idea in this paper is to integrate the advantages of these three methods.

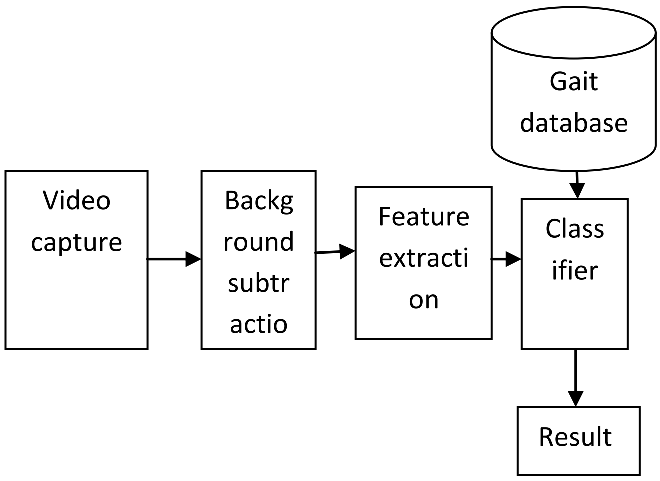

Fig 1: Block diagram of gait recognition system Scenario.

Gait an integration of temporal differencing method, optical flow method and double background filtering method with morphological processing is represented. The main goal of this algorithm is to separate the background interference and foreground information effectively and detect the moving object accurately. Firstly, temporal differencing method is used to detect the coarse motion object area for the optical flow calculation. Secondly, the DBF method is used to obtain and keep a stable background image to address variations on environmental changing conditions and is used to eliminate the background interference and separate the moving object from it. The morphological processing methods are used and combined with DBF to gain the better results. Different from the paper [13], a new improved strategy is proposed which not only improves the capability of detecting the object in motion, but also reduces the computation demands.

2. TECHNIQUES USED

The method is depicted in the flow chart of Fig.1. As can be seen, the whole algorithm is comprised of four steps: (1)Temporal differencing method, which is used to detect the initial coarse object motion area; (2) Optical flow detection, which is based on the result of (1) to calculate optical flow for each frame; (3) Double background filtering method with morphological processing, which is used to eliminate the background interference and keep the foreground moving information; (4) Motion area detection, which is used to detect the moving object and (5) Apply Fast ICA; (6) The nearest neighbor classifier is used for measure similarity between gait database and test database.

3. TEMPORAL DIFFERENCING DETECTION METHOD

Temporal differencing is based on frame difference which attempts to detect moving regions by making use of the difference of consecutive frames (two or three) in a video sequence. This method is highly adaptive to static environment. So temporal differencing is good at providing initial coarse motion areas. Generally does a poor job of extracting the complete shapes. In this paper, the two subsequent 256 level gray images at time t and t+1, I(x,y,t),I(x,y,t+1), are selected and the difference between images is calculated by setting the adaptive threshold to get the region of changes. The adaptive threshold T can be derived from image statistics. In order to detect cases of slow motion or temporally stopped objects, a weighted d coefficient with a fixed weight for the new observation is used to compute the temporal difference image I(x,y,t) as shown in following equations:

$$Id(x,y,t+1) = \begin{cases} 255 & \text{if}(Ia(x,y,t+1) > Td) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

And

$$Ia(x,y,t+1) = (1-w)\ Ia\ (x,y,t) + w|I(x,y,t+1)-I(x.y.t)| \quad (2)$$

And

$$Td = 3 * \text{mean}\ (Ia\ (\ x,y,t+1)) \quad (3)$$

where w is a real number between 0 and 1 which describes the temporal range for difference images.I (x,y,t-1) is initialized to an empty image. In our method, we set a T as the three times of mean value I (x,y,t+1).

4. OPTICAL FLOW METHOD

The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t + \Delta t$ at every voxel position. These methods are called differential since they are based on local Taylor series approximations of the image signal; that is, they use partial derivatives with respect to the spatial and temporal coordinates. For a 2D+t dimensional case (3D or n-D cases are similar) a voxel at location $(x, y, t)$with intensity$I(x, y, t)$ will have moved by Δx , Δy and Δt between the two image frames, and the following image constraint equation can be given:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (4)$$

Assuming the movement to be small, the image constraint at $I(x, y, t)$with Taylor series can be developed to get:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T \quad (5)$$

From these equations it follows that:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \quad (6)$$

Or

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0 \quad (7)$$

which results in

$$\frac{\partial I}{\partial x} Vx + \frac{\partial I}{\partial y} Vy + \frac{\partial I}{\partial t} = 0 \quad (8)$$

where Vx, Vy are the x and ycomponents of the velocity or optical flow of I(x,y,t) and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$and $\frac{\partial I}{\partial t}$ are the derivatives of the image at (x, y, t)in the corresponding directions. Ix, Iy and It can be written for the derivatives in the following.Thus:

$$IxVx + Ix\ Vy = -It \quad (9)$$

Or

$$\nabla I^T . \vec{V} = -I\ t \quad (10)$$

This is an equation in two unknowns and cannot be solved as such. This is known as the aperture problem of the optical flow algorithms. To find the optical flow another set of equations is needed, given by some additional constraint. All optical flow methods introduce additional conditions for estimating the actual flow.

4.1 Lukas kanade method

Computer vision, the Lucas–Kanade method is a widely used differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade. It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in those neighborhoods, by the least squares criterion.

By combining information from several nearby pixels, the Lucas-Kanade method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise than point-wise methods. On the other hand, since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image. The Lucas-Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point p under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at p. Namely, the local image flow (velocity) vector (x, y) must satisfy

$$Ix(q1)Vx + Iy(q1) = - It(q1) \quad (11)$$

$$Ix(q2)Vx + Iy(q2) = - It(q2)$$

.

.

$$Ix(qn)Vx + Iy(qn) = - It(qn)$$

where q1, q2, ….. qn are the pixels inside the window, and $Ix(qi), Iy(qi), It(qi)$ are the partial derivatives of the image I with respect to position *x*, *y* and time *t*, evaluated at the point qi and at the current time.

These equations can be written in matrix form AV=B, where

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \quad (12)$$

This system has more equations than unknowns and thus it is usually over-determined. The Lucas-Kanade method obtains a compromise solution by the least squares principle. Namely, it solves the 2×2 system

$$A^T A_{V = A^T b} \quad (13)$$

$$V = (A^T A)^{-1} A^T \mathrm{b} \quad (14)$$

where A^T is the transpose of matrix A . That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) I_y(q_i) \\ \sum_i I_x(q_i) I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i) I_t(q_i) \\ -\sum_i I_y(q_i) I_t(i) \end{bmatrix} \quad (15)$$

with the sums running from *i*=1 to *n*.

The matrix $A^T A$is often called the structure tensor of the image at the point *p*.

5. DOUBLE BACKGROUND FILTEERING

Everything explained in previous sections works fine while the background is adequately updated. In this paper we show a novel approach to update the background. This approach is based on a double background. In one of them (Long-term background) we will save the information which has happened in a long time, and in the other one (Short-term background) we will consider recent changes.

This couple of background images are modified to update adequately the background image (BKG) and to detect and correct abnormal con- ditions as we will see in Section 3. Moreover, the BKG image is the background image used in the motion detection algorithm depicted in Fig. 1 (‘‘A’’ subsystem).

5.1 Long-term background

This background image is used to maintain the information of the sequence during a long time. It must be a noise free image and it helps to under- stand the scene changes from the last updating to the current instant. The Long-term background image is built by taking median pixel value of a long sequence of frames for each pixel (Orrite et al., 2001). This process is carried out by the Temporal Median Filter as depicted in Fig. 1 inside the area C.

Let S_i be a 3-D array of the n last saved back- ground images (controlled by the switch T3), in which the two first dimensions are the ðx; yÞ posi- tion of the pixels in the image reference, and the third dimension indicates the image index into the array. Therefore, S_i ðx; y; lÞ is the value of gray- level of the pixel located in the ðx; yÞ position of the image number l.

5.2 Short-term background

As we have seen in Section 2.1, some pixel groups over static zones are considered as moving zones (B_{st} image). This is happening because the background image is obsolete and must be up- dated as soon as possible.

Using this image we can build the Short-term background image. So, the pixels of STB that are active in B_{st} are updated with values of the current image. Nevertheless, some short duration spurious changes could corrupt this background. Therefore, we must detect only those pixel groups of B_{st} that keep active during a short period of time (Con- sistent Static Zones).

6. MOTION AREA DETECTION

After applying the step of DBF method with morphological processing, the optical flow information of the background interference should be eliminated and only the optical flow information of real moving object is left. During the experimental test, we find that the appearance of a moving object can make a big influence on the instantaneous rate of change between the foreground motion information and the accumulative background optical flow information. In this paper, we use the result of DBF method with morphological processing as the foreground motion information *FM*. Because the result of DBF method with morphological processing comes from the last three frames accumulative optical flow.

7. INTRODUCTION TO ICA

ICA will be rigorously defined as a statistical ‘latent variable' model. Assume that we observe *n* linear mixtures *x*1, ..., *x* of *n* independent components *n*

$$x_j = a_{j1} s_1 + a_{j2} s_2 + \ldots\ldots + a_{jn} s_n$$

$$\text{For all j.} \quad (16)$$

we have now dropped the time index *t*; in the ICA model, we assume that each mixture *x* as well as each independent component *s* is a random variable, instead of a proper time signal. The observed values xj*k*(*t*), e.g. the microphone signals in the cocktail party problem, are then a sample of this random variable. Without loss of generality, we can assume that both the mixture variables and the independent components have zeromean. If this is not true, then the observable variables *x* can always be centered by subtracting the sample mean, which makes the model zero-mean [13].

It is convenient to use vector-matrix notation instead of the sums like in the previous equation. Let us denote by **x** the random vector whose elements are the mixtures *xi*, ...,*xn.* And likewise by **s** the random vector with elements *s*1 ..., *s*. Let us denote by **A** the matrix with elements *aijn*.Generally, bold lower case letters indicate vectors and bold upper-case letters denote matrices. All vectors are understood as column vectors; thus **x** or the transpose of **x**, is a row vector. Using this vector-matrix notation, the above mixing model is written as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{17}$$

times we need the columns of matrix **A**; denoting them by **a** the model can alSomeso be written as,

$$\mathbf{x} = n\sum = i\ 1\ \mathbf{ai}\mathrm{si}oj \tag{18}$$

The statistical model in equation.(2) is called independent component analysis or ICA model[11]. The ICA model is a generative model which means that it describes how the observed data are generated by a process of mixing the components *s*. The independent components are latent variables, meaning that they cannot be directly observed. Also the mixing matrix is assumed to be unknown. All we observe is the random vector **x**, and we must estimate both **A** and **s** using it. This must be done under as general assumptions as possible.

7.1 The FastICA Algorithm

$$X^* = XM = ASM = AS^* , \tag{19}$$

In the preceding sections, we introduced different measures of non gaussianity, i.e. objective functions for ICA estimation. In practice, one also needs an algorithm for maximizing the contrast function, for example the one in (25). In this section, we introduce a very efficient method of maximization suited for this task. It is here assumed that the data is preprocessed by centering and whitening as discussed in the preceding section.

7.1.1 FastICA for one unit

To begin with, we shall show the one-unit version of FastICA. By a "unit" we refer to a computational unit, eventually an artificial neuron, having a weight vector w that the neuron is able to update by a learning rule. The FastICA learning rule finds a direction, i.e. a unit vector w such that the projection w T x maximizesnon gaussianity.

Nongaussianity is here measured by the approximation of negentropy J(wTx) given in (25). Recall that the variance of wTx must here be constrained to unity; for whitened data this is equivalent to constraining the norm of w to be unity.

The FastICA is based on a fixed-point iteration scheme for finding a maximum of the nongaussianity of wx,as measured in (25),

The basic form of the FastICA algorithm is as follows:

1. Choose an initial (e.g. random) weight vector w.

2. Let w + = E{xg(w Tx)} -E{g 0(wx)}w

3. Let w = w+/kw+k

4. If not converged, go back to 2.

Note that convergence means that the old and new values of w point in the same direction, i.e. their dot-product is (almost) equal to 1. It is not necessary that the vector converges to a single point, since w and -w define the same direction. This is again because the independent components can be defined only up to a multiplicative sign. Note also that it is here assumed that the data is prewhitened.[13]

The derivation of FastICA is as follows. First note that the maxima of the approximation of the negentropy of wTx are obtained at certain optima of E{G(wTx)}. According to the Kuhn-Tucker conditions (Luenberger, 1969), the optima of E{G(wTx)} under the constraint E{(wTx)2 }

In practice, the expectations in FastICA must be replaced by their estimates. The natural estimates are of course the corresponding sample means. Ideally, all the data available should be used, but this is often not a good idea because the computations may become too demanding. Then the averages can be estimated using a smaller sample, whose size may have a considerable effect on the accuracy of the final estimates. The sample points should be chosen separately at every iteration. If the convergence is not satisfactory, one may then increase the sample size.

8. SIMILARITY METRICS USING CLASSIFIERS

As we are using the ICA algorithm the feature extraction time required is more as compared to other methods. For the algorithm developed in this paper required around few seconds time as extraction time. The query image will be more similar to the database images if the distance is smaller. For similarity measurement we used the Euclidean distance classifier and Mahalanobis distance classifier [14], for calculating the minimum distance between the query image and images to be matched from the database. If x and y are the two d-dimensional feature vectors of the database image and query image respectively, then these distance metrics are defined as follows. Euclidean distance to determine closeness reduces the problem to computing the distance measures:

If the distance is small, we say the images are similar and we can decide which the most similar images in the database are. Another distance metrics for comparison of the retrieval of images used is Mahalanobis metric:

The results of these classifiers are very much close to each other. In Mahalanobis metrics the time required for similarity measure is more due to involvement of the covariance matrix.

9. REFERENCES

[1] G. Venkata Narasimhulu1, Dr. S. A. K. Jilani2, " Fuzzy Principal Component Analysis based Gait Recognition" , Ijcsit,vol3,No.3,pp.4015-4020, 2012.

[2] Hayder Ali, Jamal Dargham, Chekima Ali, Ervin Gobin Moung, "Gait Recognition using principle Component Analysis" , ICMVpp. 539-543, 2011.

[3] Vili Kellokumpu ,Guoying Zhao, Stan Z.Li and Matti Pietikainen , " Dynamic Texture Gait Recognition" , LNCS 5558,pp. 1000-1009, 2009.

[4] M. Pushpa Rani and G.Arumugam, "An Efficient Gait Recognition System For Human Identification Using Modified ICA" IJCSIT,Vol.2,No.1,pp.55-67, 2010.

[5] C.Sasi varnan, A. Jagan, Jaspreet Kaur, Divya Jyoti, Dr.D.S.Rao, "Gait Recognition Using Extracted Feature Vectors", IJCST Vol. 2, Issue 3, September 2011.

[6] Jiwen Lu a, Erhu Zhang, "Gait recognition for human identification based on ICA and fuzzy SVM through

multiple views fusion" , Pattern Recognition Letters, Vol. 28, pp. 2401–2411, 2007.

[7] Yi Huang, Dong Xu,Member IEEE, And Tat-Jen Cham,"Face and Human Gait Recognition Using Image-to-Class Distance" , IEEE , pp no1051-8215, 2010 .

[8] Shuai Zheng, Junge Zhang, Kaiqi Huang, Ran He, Tieniu Tan, "Robust View Transformation Model For Gait Recognition",IEEE 18th conference, 2011.

[9] Muzhir Sh. Al-Ani, Isra H. Al-Ani,"Gait Recognition Based Improved Histogram", VOL. 2, NO. 12, December 2011.

[10] Jyoti Bharti , M.K Gupta, "Human Gait Recognition using All Pair Shortest Path**"** **,** IPCSIT vol.9 , (2011).

[11] Kailash J.Karande & Sanjay N Talbar **"**Independent Component Analysis of Edge Informationfor Face Recognition**"** **,** International Journal of Image Processing (IJIP) Volume (3) : Issue (3).

[12] Pratibha Mishra, Shweta Ezra, "Human Gait Recognition Using Bezier Curves" , (IJCSE) Vol. 3 No. 2 Feb 2011.

[13] Aapo Hyvärinen and Erkki Oja **, "**Independent Component Analysis: Algorithms and Applications", Neural Networks Research Centre Helsinki University of Technology, Finland Neural Networks, 13(4-5):411-430, 2000**.**

[14] Nitin Bhatia, Vandana "Survey of Nearest Neighbor Techniques", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 2, 2010.

[15]