International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

People Counting in Transport Vehicles

Sebastien Harasse, Laurent Bonnaud, Michel Desvignes
LIS-ENSIEG, 61 rue de la Houille Blanche BP 46 38402 St. Martin d'Heres cedex France
{harasse,bonnaud,desvignes}@lis.inpg.fr

Abstract**—** Counting people from a video stream in a noisy environment is a challenging task. This project aims at developing a counting system for transport vehicles, integrated in a video surveillance product. This article presents a method for the detection and tracking of multiple faces in a video by using a model of first and second order local moments. An iterative process is used to estimate the position and shape of multiple faces in images, and to track them. the trajectories are then processed to count people entering and leaving the vehicle.

Keywords**—** face detection, tracking, counting, local statistics

I. Introduction

ESTIMATING the number of people in a noisy environment is a central task in surveillance. A real-time count can be used to enforce the occupancy limit in a building, to manage transport traffic in real time, to actively manage city services and allocate resources for public events. Our project is to develop a counting system for moving platforms such as buses, in an existing classical video recorder. Images are captured using a video camera and are analyzed to determine the number of people present. The background scene is therefore not static and vary in a large number of ways: variations in lighting levels, patterns of scene background, movements of objects that might appear or disappear in the scene. The point of view is defined by the location of the camera, in front of the people. This motivates our approach, which is to detect, track and count faces, using color information. This paper propose a method to detect and track multiple skin objects using local moments, with two different movement prediction methods. The estimated trajectories of faces are then used to count people.

II. Previous Work

Finding people in images is a difficult task [1] due to the high variability in appearance of people. Various approaches have been proposed in the past years [2], [3], including methods based on background subtraction [5], classical template matching with several patterns [8], [9], [10], [11] and statistical classifiers such as support vector machines [12], [13] or neural networks [14], [15] applied to face features vectors. However, most face detection methods use skin color information [2], [3], which is a low level and accurate information. The tracking of multiple targets in a video sequence in a cluttered environment can be done with particle filtering [16]. This paper presents a novel method for multiple targets tracking which is based on a statistical modeling of the problem, like the Condensation algorithm, but does not require sampling.

III. Statistical modeling and skin object detection

The method proposed here is based on skin color information, since it is the most robust information in cluttered environment. The main steps of our counting system are the probabilistic skin color modeling, the iterative face detection, tracking and counting.

A. Skin color model

A skin color model is needed in order to decide whether a pixel is skin colored or not. Skin chrominance is very specific, as opposed to its luminance, which has a large variability. Thus our model is defined in a chrominance color space so that skin pixels are represented in a small portion of the space, for example the normalized-rgb color space, defined from the original RGB space as:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \quad (1)$$

Since $r+g+b=1$, only two components (r,g) are used for the model. A bidimensionnal gaussian model g_{skin} is obtained to represent skin color in the rg-space. Its parameters are learned from skin pixels from the FERET faces database [19].

This model is applied to an image to obtain a skin map S_I where each value is the value of our bidimensionnal gaussian model at the corresponding pixel's color. For an image I, and skin model g_{skin}, the corresponding skin map S_I is:

$$S_I(x,y) = g_{skin}(I(x,y)) \quad (2)$$

where (x,y) is a position in the image and $I(x,y)$ is the color of I at this position, in normalized-rgb coordinates. Fig. 1(b) presents an example of skin map.

B. Statistical modeling

Our face detector is based on a statistical representation of the problem: a face is a skin region, parameterized by its position and shape. Therefore a skin object x is assumed to be a 5-dimensional vector composed of the first order moment, describing position, and the second order moment, describing shape:

$$x = (\mu_x, \sigma_x) \quad (3)$$

with

$$\mu_x = (\mu_{x1}, \mu_{x2}), \sigma_x = \begin{bmatrix} \sigma_{x11} & \sigma_{x12} \\ \sigma_{x12} & \sigma_{x22} \end{bmatrix} \quad (4)$$

Our face model can be seen as an ellipse centered in μ_x with axes defined by covariance matrix σ_x. This model has been introduced in [17] for one single face tracking using color.

The problem can be expressed as a statistical detection problem, where x is a random variable and z another random variable whose realizations are each image. We aim at detecting the local maxima of the observation density $p(z/x)$, in order to find the parameters of each skin object in the image. $p(z/x)$ is defined as proportional to the correlation between the skin map S_z and the bidimensional gaussian function g_x parameterized by x:

$$p(z/x) \propto \int S_z(t).g_x(t)dt \tag{5}$$

with t a bidimensional variable.

Fig. 1. (a) original image, (b) skin map , (c) five detected objects

C. Skin objects detection

The method proposed here estimates μ_x by using a priori information about σ_x, then estimates σ_x for each detected object, using an iterative process.

1) First order moment estimation: the detection of the first order moments μ_x of objects in the image involves an a priori estimation of σ_x. σ_m is defined as the average covariance matrix representing a face. With this assumption, the observation density becomes:

$$p(z/\mu_x, \sigma_x = \sigma_m) \propto \int S_z(t).g_{\mu_x,\sigma_m}(t)dt \tag{6}$$

$$p(z/\mu_x, \sigma_x = \sigma_m) \propto \int S_z(t).g_{0,\sigma_m}(t-\mu_x)dt \tag{7}$$

with $g_{\mu,\sigma}$ denoting the gaussian function with first and second order moments μ and σ respectively.

The observation density with fixed $\sigma_x = \sigma_m$ is proportional to the 2-dimensional convolution product of S_z by a gaussian function with covariance matrix σ_m, which is an inexpensive computation. The first order moments of objects are detected by finding local maxima of the function.

2) Iterative second order moment estimation: suppose that an object x_0 is present in the image, with first order moment μ_{x_0}. Its second order moment σ_{x_0} must be estimated.

Our method is to estimate σ_{x_0} by using local moments iteratively. Let W be a 2-dimensional window defined in the same space as S_z, with $\int W(t)dt = 1$. The second order local moment [18] of S_z centered in μ_{x_0} is defined as:

$$\sigma^2_{S_z,W} = \int (t-\mu_{x_0})^2.S_z(t)W(t)dt \tag{8}$$

A sequence of local moments is defined as:

$$\begin{cases} \sigma_0 = 1 \\ \sigma^2_{n+1} = \sigma^2_{S_z,g(\mu_{x_0},\alpha\sigma_n)} \end{cases} \tag{9}$$

where $g(\mu_{x_0}, \alpha\sigma_n)$ is the bidimensional gaussian window of first and second order moments μ_{x_0} and $\alpha\sigma_n$ respectively, with α a real scalar found experimentally, so that the sequence converges: $\alpha \approx 1.3$.

Practically, the method consists in starting with a window centered in μ_{x_0} with a size smaller than the expected object size, computing the local moments of S_z in this window, then using the result multiplied by a constant α as the next window covariance matrix. This sequence converges to the second order moment of the skin object. By using local moments, the computation of σ_{x_0} is not disturbed by the other objects in the image. The detection of multiple skin objects in the image can then be achieved. Fig. 1 shows the results obtained with this method.

IV. Skin object tracking

Our method for temporal tracking of detected skin objects is tightly related to the recursive method used for the second order local moment estimation. The tracking is composed of a prediction step followed by an observation step for each object.

A. Trajectory prediction

Our tracker is designed to track several objects simultaneously. One major difficulty in multiple targets tracking is the association problem: each object detected at time t must be associated to its corresponding object at time $t+1$.

Two different prediction methods are considered: dynamic model based prediction and trajectories learning based prediction.

1) Dynamic model based prediction: the first and most common method is to define a dynamic model for the object, estimate its parameters from past observations, and predict the next state from this model. In our application, faces movement is difficult to predict accurately since framerate is low and people are close to the camera. This results in a very noisy trajectory.

Therefore, a simple but robust model is used: the tracked object is assumed to have a constant speed vector for a relatively small amount of time (about one second). The speed vector is estimated from the past positions of the object during the last second, to filter out noise. A more complex model could be used if needed by the application.

2) Trajectories learning based prediction: the second prediction method aims at predicting the next state of one object by using the estimated trajectories of past tracked objects. In our application, people are passing in front of the camera by following almost the same path every time. Thus it is possible to learn people trajectories and use this information to predict the states of future objects. A way to learn the trajectories is to store for each state, the estimated next state of tracked objects that have had this state. That is to say, for an object tracked at time t with state x^t_O, its estimated state x^{t+1}_O at time $t+1$ is stored in a table. When another tracked object state

is similar to x_O^t, its predicted state must be similar to x_O^{t+1}. For memory considerations, only the position part of the state vector is learned.

A tracked object has only a small probability to be estimated at the exact same state as another object. It is therefore necessary to predict the next state of an object O from the learned trajectories of other objects that presented a state close to the O object's current state. All memorized state predictions close to the object O current state are taken into account.

An a priori probability density is defined for object O's next state, from the memorized trajectories, as:

$$p(x_O^{t+1}/x_O^t) \propto \sum_{k=1}^{N} f(\|\mu_{x_k} - \mu_{x_O}\|).g_{P(x_k)} \qquad (10)$$

with N the number of entries in the trajectories table, x_O^t the current state of object O, x_O^{t+1} its predicted state, x_k the $k-th$ memorized state, $P(x_k)$ the learned predicted state for state x_k, and $g_{P(x_k)}$ the bidimensional gaussian function parameterized by x_k. f is a positive decreasing function.

Since only the predicted position of skin objects are memorized, the second order moment part of state x_k is considered equal to the second order moment σ_O^t of object O at current time t.

This prediction is integrated in our tracking algorithm by using this probability density as the initial window to estimate the local moments for the object in the next image.

B. Observation step

The observation step corrects the predicted position and shape of the object with respect to the observed image. The gaussian function parameterized with the predicted state defines the window in which the first and second order local moments of the object are computed. This step is iterated by using the last computed local moments as the parameters of the gaussian window:

$$\begin{cases} \mu_0 = \mu_{predicted} \\ \sigma_0 = \sigma_{predicted} \\ \mu_{n+1} = \mu_{S_z, g(\mu_n, \alpha\sigma_n)} \\ \sigma_{n+1}^2 = \sigma_{S_z, g(\mu_n, \alpha\sigma_n)}^2 \end{cases} \qquad (11)$$

with $\mu_{S_z, g(\mu_n, \alpha\sigma_n)}$ the first order local moment of S_z in the window $g(\mu_n, \alpha\sigma_n)$, defined by:

$$\mu_{S_z, g(\mu_n, \alpha.\sigma_n)} = \int t.S_z(t).g(\mu_n, \alpha\sigma_n)dt \qquad (12)$$

In this sequence, the σ update step is the same as in (9). This sequence converges to the first and second order moments of each object for the current image. figure 2 shows an example of the tracking of two faces (red and violet ellipses). One arm is also detected in the middle image (white ellipse).

C. Targets occlusions

Our system must be robust to temporal targets occlusions, that can appear because of a scene object or another target crossing the first one. A target is considered lost from one video frame to the next when there is not enough information in the second frame to estimate the state of the target. The decision is made by computing the ratio of skin pixels by the area of the ellipse parameterized by the estimated second order moment:

Fig. 2. Tracking example, two people passing each other

$$A = \frac{\int z(t) W_{lim}(t)}{area(W_{lim})} \qquad (13)$$

with W_{lim} the gaussian window parameterized by the limit of sequence $\sigma_{(n)}$. A is compared to a reference ratio A_{ref}.

When a target is lost, the predicted state is assumed to be the estimated state. If the target is lost for too much time, it is considered definitely lost.

V. People counting

The counting of people is done in a simple way, by counting the tracked objects crossing a segment defined in the image space. The segment is defined manually so that the faces cross it when people enter the vehicle. The counting of a target tracked from position P_1 at a frame to position P_2 at the next frame, is done by checking if P_1P_2 crosses the counting segment C_1C_2, with dotproduct and crossproduct tests:

$$\begin{cases} \overrightarrow{C_1P_1}.\overrightarrow{C_1.C_2} > 0 \\ \overrightarrow{C_2P_1}.\overrightarrow{C_2C_1} > 0 \\ \overrightarrow{C_1P_2}.\overrightarrow{C_1.C_2} > 0 \\ \overrightarrow{C_2P_2}.\overrightarrow{C_2C_1} > 0 \\ \overrightarrow{C_1P_1} \wedge \overrightarrow{C_1C_2} < 0 \\ \overrightarrow{C_1P_2} \wedge \overrightarrow{C_1C_2} > 0 \end{cases} \qquad (14)$$

This counts people passing from left to right, as illustrated in figure 3. To count people passing from right to left, the two last inequalities are reversed:

$$\begin{cases} \overrightarrow{C_1P_1} \wedge \overrightarrow{C_1C_2} > 0 \\ \overrightarrow{C_1P_2} \wedge \overrightarrow{C_1C_2} < 0 \end{cases} \qquad (15)$$

The main advantage of this method is that if the tracking fails before or after the counting segment but succeeds at the counting segment, the face will be counted.

VI. Results and conclusion

The counting method has been tested under controlled conditions, in an indoor office, as well as under real conditions, on video streams from a transport vehicle. Using an appropriate skin model, the detection and tracking of skin objects is efficient, with a few tracking loss because of illumination conditions changes. By using an adaptive color skin model, it would be possible to achieve better tracking. A 85% counting success rate is achieved compared to the real count, while most

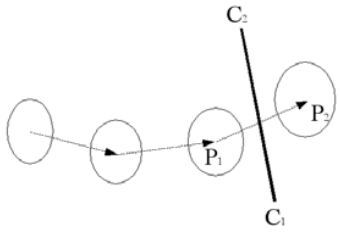

Fig. 3. Skin object crossing the counting segment

non detection were caused by faces not passing through the counting segment. False positives were caused by some arms being counted.

The main features of our approach are the iterative local moments estimation, the absence of threshold for the detection of skin pixels and objects, and the trajectory prediction based on learning of past trajectories. We are currently working on improving the skin color model to achieve a better detection of skin pixels.

REFERENCES

[1] S. Ioffe, D. A. Forsyth, "Probabilistic Methods for Finding People". International Journal of Computer Vision 43(1), pp 45-68, 2001.
[2] M.H. Yang, D. Kriegman, and N. Ahuja. "Detecting face in images: a survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1), pp 34-58, 2002.
[3] Erik Hjelmas "Face Detection: A Survey", Computer Vision and Image Understanding, 83(3), pp. 236-274, 2001.
[4] C. Wren, A. Azarbayejani, T. Darell, A. Pentland, "Pfinder: Real-time tracking of human body", IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7), pp. 780-785, 1997.
[5] I. Haritaoglu, D. Harwood, and L. Davis, "W4: A real-time system for detection and tracking of people and monitoring their activities", IEEE Pattern Analysis and Machine Intelligence, 22(8), pp. 809-830, 2000.
[6] Collins, Lipton, Kanade, Fujiyoshi, Duggins, Tsin, Tolliver, Enomoto, and Hasegawa, "A System for Video Surveillance and Monitoring: VSAM Final Report," Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May, 2000.
[7] G. Yang, T.S. Huang, "Human face detection in complex background", Pattern recognition,27(1):53, 1994.
[8] Y.H. Kwon and N. da Vitoria Lobo, "Face Detection Using Templates",International Conference on Pattern Recognition, pp. 764-767, 1994.
[9] H. Nanda and L. Davis, "Probabilistic template based pedestrian detection in infrared videos". IEEE Intelligent Vehicles, 2002, Versailles, France, pp 15-20, 2002,
[10] M. Bertozzi et al, "Pedestrian detection in infrared images," IEEE Intelligent Vehicles Symposium 2003, Columbus, USA, pp662-667, 2003
[11] C. Stauffer and E. Grimson, "Similarity templates for detection and recognition", Computer Vision and Pattern Recognition, pp. 221-228, Kauai, HI,. 2001.
[12] P. Campadelli, R. Lanzarotti, G. Lipori, "Face detection in color images of generic scenes", International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS), 2004.
[13] F. Xu, X. Liu, and K. Fujimura, "Pedestrian Detection and Tracking with Night Vision", IEEE Transactions on Intelligent Transportation Systems, 5(4), 2004
[14] L. Zhao and C. Thorpe, "Stereo- and neural network based pedestrian detection", IEEE Int. Conf. on Intelligent Transportation Systems, Tokyo, Japan, pp 148-154, 2000.
[15] H. Rowley, S. Baluja, T. Kanade, "Neural Network-Based Face Detection," IEEE Trans. Pattern Analysis and Machine Intelligence,20(1), pp.23-38, 1998.
[16] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking", International Journal of Computer Vision 29(1), pp. 5–28, 1998.
[17] K. Schwerdt and J. L. Crowley, "Robust face tracking using color", in Proc. of 4th International Conference on Automatic Face and Gesture Recognition, Grenoble, France, 2000, pp. 90–95.
[18] M-K. Hu, "Visual pattern recognition by moment invariants", IRE Trans. on Information Theory, IT-8:pp. 179-187, 1962.
[19] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, "The FERET evaluation methodology for face recognition algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, October 2000.