MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Automatic Visual Control of a Pedestrian Traffic Light

Massimiliano Loria Alberto Machí

Istituto di Fisica Cosmica ed Applicazioni dell'Informatica
Via Mariano Stabile, 172 90139 Palermo - Italia
E-mail: *machi@ifcaii.pa.cnr.it*

Abstract

This paper presents an algorithm using scene analysis methods to detect, track and label moving objects in order to optimize the active time of a traffic light performing, on demand, car flow control at a pedestrian road crossing. The algorithm detects blobs moving on the scene, try to label them as pedestrians or vehicles and tracks them along the scene.

The decision logic turns on the traffic light after pedestrian's request and switches it off as far as no pedestrians nor unidentified blobs are anymore present in a predefined region of interest.

In the first assessment study, the described algorithm has shown almost full reliability in detecting absence of meaningful moving components, in various lighting conditions. It was also able to optimize about fifty percent of sequences, correctly classifying well behaving pedestrians and vehicles.

Work is ongoing to improve its efficiency in recognizing and correctly tracking objects moving on convergent trajectories.

1 Introduction

Image processing methodologies have been extensively used in the last decade to perform automatic evaluation of features for traffic scene analysis.[1- 5].

This paper presents an algorithm using similar techniques but tuned to optimize the active time of a traffic light which performs, on demand, car flow control at a pedestrian road crossing.

The problem consists in monitoring a section of a road near a zebra crossing, in detecting movement of pedestrians and in mantaining the traffic light into the stop status (for vehicles) for just the time required by pedestrians to cross the road.

Because severe outdoor condititions are expected and pedestrian safety is of main concern, algorithm robustness is to be preferred to efficiency in the minimization of cars stop time.

The choice of a fixed point of view on a relevant position minimizes occlusions due to perspective effects. Pedestrians come into the scene from sidewalks and generally cross the lane orthogonally to the direction of incoming and outcoming vehicles.

Typical events on the scene last from hundreds of milliseconds (lighting variations from clouds or reflexions) to seconds (movement of vehicles) to tens of seconds (movements of slow pedestrians).

2 Algorithm Outline

The tracking algorithm uses a reconstructed background picture as a reference image to detect moving blobs in subsequent image frames, acquired at regular time intervals. Background subtraction [6] is preferred to other instantaneous difference measures like optical flow [7] because the latter give noisy fields, while a stable reference image may be easily computed, owing to the fixed point of view on the scene.

The background image is continuously updated to take care of changes in luminosity due to varying weather conditions, reflections, car parking and other slow time varying events.

For each image frame is created a list containing a symbolic description of each moving blob detected.

The new obtained list is compared with the one inherited from the previous frame and correspondences among new and old blobs are evaluated using a mixed parametric and fuzzy logic approach.

Best pairwise associations are selected and a confidence index is assigned to each one: a strong index gives evidence of a successfully tracked blob.

Tracked blobs are then labeled as pedestrians or cars according to the magnitude and direction of their speeds; new appearing blobs and old blobs associated to new ones with a poor confidence index are labeled as *unidentified traversing objects (UTOs).*

The algorithm tries to recover UTO's association for one more frame, than it discards their descriptors and reinitializes tracking (see fig. 1).

The procedure stops tracking when neither tracked pedestrians, nor UTOs are present in the ROI .

3 Video Sequence Acquisition

The target scene for the case study consists in a part of a one-way road near a pedestrian traffic light, including a section of the lane, a zebra crossing and side-walks. The Region of Interest (ROI) includes the zebra crossing and a limited neighborhood on the lane sides. The scene is observed from an height of about 15 meters to obtain a quite vertical view and a size of about 15 by 15 meters.

Frames are taken two per second, so that cars moving at 40 Km/h take at least three frame time intervals to traverse the scene, while pedestrians remain on the ROI for a few tens of frames. Subsampling to 128x128 pixel 8 bit gray scale resolution allows optimization of the processing time.

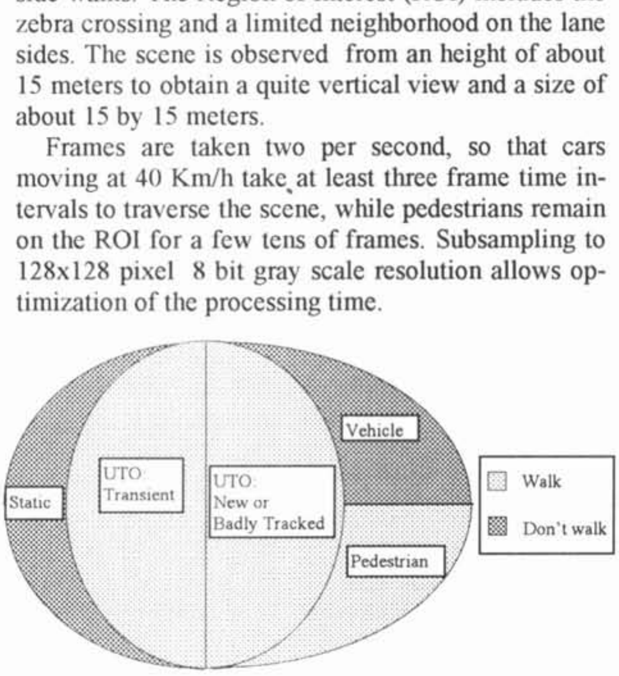

Figure 1 Component classification

4 Background Evaluation

Background is an image representation of the scene including just static or very slow moving components (objects). A parking car joins the background while a car stopping at the traffic light doesn't. We update the reference image continuously while no pedestrian is crossing and cars move freely on the lane.

To eliminate fast moving components we follow the method described by Inoue and Seo in [6]; we acquire a sequence of 30 frames (sampled one per sec.) and process it in groups of three. We collect stable pixels to obtain a first evaluation of the background, affected just by components varying their intensity with time constant greather than one second.

To statistically lower contributes from components moving on the scene with time constant shorter than 10s we get a new sequence of 10 frames. For each frame we evaluate its difference from the reference and add to the background each pixel varying less than a heuristic low threshold. A histogram collects the distribution of intensities of stable pixels. At the end of the procedure we obtain both a refined background image and a (local) measure of variance of pixels intensities.

5 Moving Component Segmentation

When a pedestrian issues a request to cross the road, it activates the car stop light and starts the algorithm tracking step. The frequency of scene sampling is set to half a second, then each acquired frame is subtracted from the background image and thresholded with the local value of background variance. Each blob (component) in the obtained binary image is supposed to represent one or more objects on the scene moving or suddenly changing their luminance. The image is filtered and blobs with meaningfully large areas are labelled.

A feature extractor operator then creates a list with a symbolic description of the labelled blobs; the description includes figures for the position of the blob centroid, area, coordinates of the minimum including rectagle, its mean gray level and elongation.

6 Component Tracking

Aim of this step is to identify continuously moving components, and to distinguish them from components generated just by noise, reflections or other sudden variations in luminosity due to car lights, clouds or movement of trees.

The procedure searches for the best pairwise association among the components descriptors in the most recent frame and in the previous one; the values of old components are modified before comparison to take in account their expected. The algorithm makes use both parametric and fuzzy logic criteria to compute the distance in features space and to assign to each couple an affinity index.

Firstly an association matrix is computed by measuring the distance in the features space between each component in the *old* list and each one in the *new* list.

To each feature $\boldsymbol{k}$ is assigned a relative weight $\boldsymbol{w_k}$, (heuristically determined on the base of the relative discriminating power of the feature in the application) [8], and it also associates a fuzzy function $\boldsymbol{F_{\sigma k}}$ which defines the distance metric to be used in the feature subspace (i.e. the meaningfulness of the difference between differing values of the features). The total distance $\boldsymbol{A(n,o)}$ between two components $\boldsymbol{n}$ and $\boldsymbol{o}$ consists in the weighted sum of distances in the various features subspaces $\boldsymbol{k}$. The effect of the fuzzy function $\boldsymbol{F_{\sigma k}}$ is to enhance similarities and to bias negatively the association index in case components strongly differ in some features.

The formula to calculate the association matrix is:

$$A(i,j) = \mathrm{Max}\{0, \sum_{k=1}^{k=N} F_{\sigma_k}(C_i^k, C_j^k) w_k\}$$

where:

N: number of features;

w_k: weight of feature $\boldsymbol{k}$ ($\sum_{k=1}^{k=N} w_k = 1$);

σ_k: variance of feature $\boldsymbol{k}$;

$$F_\sigma(a,b) = 2 \cdot \mathrm{Exp}\left[-\left(\frac{a-b}{\sigma}\right)^2\right] - 1 \quad \text{: fuzzy function}$$

Two relative indexes, namely the relative association index NRAI(n,o) for the *new components* ones and ORAI(n,o) for the *old* ones, are derived from the association matrix A(n,o) by normalizing it by rows and by columns. They express the relative preference of each component for the components in the complementary list and are used in the subsequent step to make associations:

$$\mathrm{NRAI}(n,o) = \frac{A(n,o)}{\sum_{k=1}^{k=NO} A(n,k)}$$

$$\mathrm{ORAI}(n,o) = \frac{A(n,o)}{\sum_{k=1}^{k=NN} A(k,o)}$$

where NO and NN are respectively the number of old and new components, *n* and *o* are the elements of the new and old lists.

The two input lists are then scanned iteratively looking for the best pairwise associations. A minimum threshold value is selected for the relative association index. If two components reveal maximum reciprocal association, over the confidence level, the new component is labeled as *updated* and moved to the output list, the old one is discarded and the list scan is iterated. New components remaining unmatched, if in ROI, are labelled as *unidentified traversing objects* UTOs and moved to the output list; old ones are discarded if all new components have been updated, else they are moved to the output list and used in the next step to recover UTOs.

UTOs are generally related to components moving on colliding paths occluding or touching each an other one, due to sudden change of luminosity, reflections, clouds or car light illumination. If an UTOs has been revealed inside the ROI, the sequence tracking is reinitialized.

7 Pedestrian Labelling

If all components have been successfully associated a search for well behaving components is performed. Any blob whose speed component parallel to the zebra crossing is preminent over the orthogonal one, (at least twice greather) and whose speed magnitude is appropriate, is labelled as a pedestrian.

If no pedestrian are present on the ROI, analysis is halted and the status of the traffic light is toggled.

8 Experimental Results

To qualitatively assess the performance of the described algorithm, one hundred crossing sequences were recorded and processed in real time on Sun 5/10 workstations.

Sequences were acquired from two different scenes with different lighting conditions, at different times during day and night hours.

After heuristically tuned the various thresholds, the algorithm was always able to reliably stop sequences as soon as no intensity changing components were present in the ROI, and in about fifty percent of cases it was also able to correctly track and label well behaved moving components.

Tracking failed and labelled as UTO components originated by sudden variations in lighting subsequent to background evaluation, or by incoherent motion of trees.

It also failed to correctly track a number of blobs with intersecting trajectories. This effect is dependent from the naive rules used in the pedestrian labelling procedure.

At present, labelling of objects as vehicles or pedestrians rely on the expectation that pedestrians cross the road on a path almost parallel to the zebra crossing and that the direction of motion can be inferred from the apparent motion of the center of mass of the corresponding blob.

In practice, noisy changes in the blob shape induce variations of the coordinates of the center of mass which, in case of small blobs, may affect the apparent direction of movement of the blob. Furthermore, if two blobs merge, because they move on colliding paths, the shape of the merged blob is highly changing in the subsequent frames and the center of mass moves in an apparently incoherent fashion.

This is a potentially dangerous case because it may hide the presence of pedestrians, as shown in fig.3.

Some effort actually was done to segment merged components using techniques based on optical flow algorithms (both gradient and feature-based) [7]; the results were unsatisfactory for three main reasons:

a) moving blobs have often a statistically too small area to apply a gradient-based algorithm (a few pixel tickness is typical for pedestrians and cycles);
b) component features as corners or edges change on a time scale smaller than one half of a second;
c) components with highly different velocities may be contemporary present on the scene and frame sampling should be set accordingly to satisfy the smooth movement condition.

Work is presently progressing in two directions: firstly exploiting measures of texture to improve segmentation of components while still merged, and secondly extending the memory of the track procedure so that merging can be foreseen and blob recognition tried after split.

References

[1] J.M. Blosseville et al "TITAN: a Traffic Measurement System using Image Processing techniques", IEEE Second Intern. Confrernce on Road Traffic Monitoring, pp. 84-88 London 1989

[2] Neil Hoose "Computer Image Processing in Traffic Engineering" J. Wiley and Sons Traffic Engineering Series

[3] G.L. Foresti, V. Murino, "Moving Object Recognition from an Image Sequence for Autonomous Vehicle Driving" in V. Cappellini ed.:Time Varying Image Processing and Moving Object Recognition 3". Elsevier 1994 pp.383-390

[4] G. Nicchiotti, E. Ottaviani "Automatic vehicle counting from Image Sequences" in V. Cappellini ed. "Time Varying Image Processing and Moving Object Recognition 3". Elsevier 1994 pp.410-417

[5] S.M. Smith and J.M. Brady "ASSET-2: Real-Time Motion Segmentation and Shape Tracking" IEEE Trans. on PAMI Vol.17, No 8, August 1995

[6] Katsunori Inoue, Wonchan Seo "Background Image Generation by Cooperative Parallel Processing Under Severe Outdoor Conditions" in Proceedings of IAPR Int. Workshop on Machine Vision Appplications MVA92, Tokyo 1992

[7] J.K. Aggarwal, N. Nandhakumar, "On the computation of Motion from Sequences of Images - A review" Proc. of IEEE Vol. 76, No 8, Aug. 1988

[8] A. M. Darwish, A. K. Jain "A Rule Based Approach for Visual Pattern Inspection" IEEE Trans. on PAMI, Vol 10,No 1, Jan 1988, pp 56-68

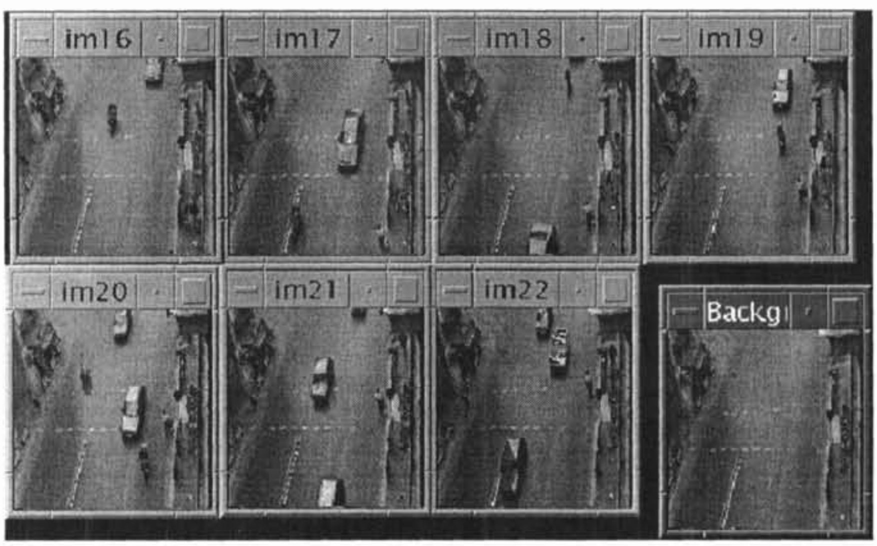

Figure 2 Reconstructed background

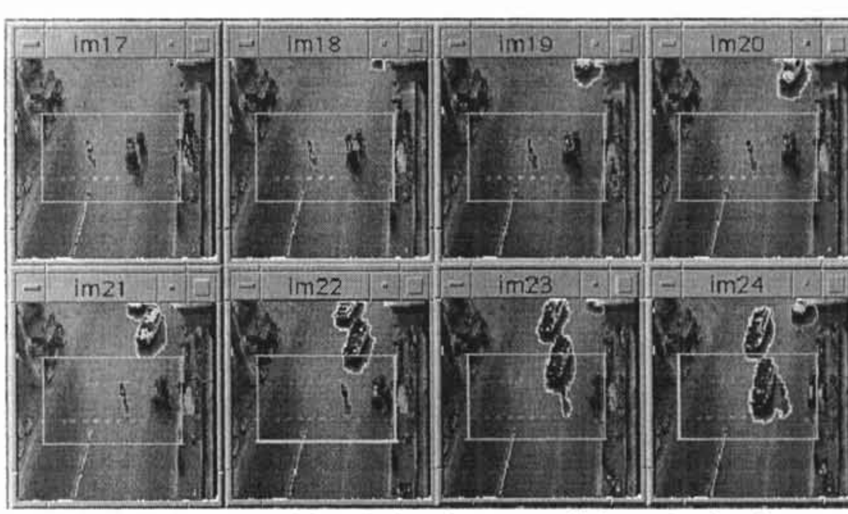

Figure 3 Sequence with occlusions between vehicles and pedestrians: a misinterpreting of the pedestrian speed direction occurs