Eliminating Target Shadows for Improved Tracking and Shape Estimation in Outdoor Monocular Diurnal Sequences

Paolo Gamba,* Massimiliano Lilla and Alessandro Mecocci†

ABSTRACT

We present an efficient method able to extract a shadow model from a scene, exploiting the HLS color components. The algorithm allows to recover target shapes in diurnal scene for improved identification. It is based on the realization of a General Bitmap Model and a more particular Strip Bitmap Model to identify shadow regions. Each pixel in the image is classified as shadow or not by a minimum distance approach to these models.

1 Introduction

In the last years many different approaches have been proposed to detect, locate and recognize targets in outdoor scenes ([1]-[4]). These techniques are mainly based on feature detection and tracking, frames difference accumulation, and optical flow computation. Typically the results are given in term of binary masks where moving regions are first labeled and then grouped according to their motion parameters.

The performances of the previous algorithms are quite good when target detection and tracking is involved. Nevertheless, they all suffer for a common problem when target shape estimation and/or classification is involved, namely: the moving regions that are detected belong to the targets and to their shadows as well. The reason is that motion information is similar for each target and its shadow. Moreover, the shadows tend to produce scene changes in proximity of the related targets. So it is very difficult to discriminate them by using motion based techniques and/or differential techniques. To improve the target tracking, shape estimation and classification, this paper proposes a processing chain for shadow removal. The algorithm works on the binary masks produced by a generic segmentation algorithm for moving target detection and on the image sequence produced by a monocular color sensor.

The efficiency of the procedure has been tested in a typical application, the oversee of a parking (fig. 1), where a typical low quality fixed camera was installed to control a large area. This choice allowed to test our approach with various kind of lighting, due to atmospheric conditions and the different hours of the day.

*Dipartimento di Elettronica, Università di Pavia, Via Ferrata, 1, I-27100 Pavia, Italy. Tel: +39-382-505923 Fax: +39-382-422583 E-mail: gamba@ipvmw2.unipv.it

†Facoltà di Ingegneria, Università di Siena, Via Roma, 77, I-53100, Siena, Italy

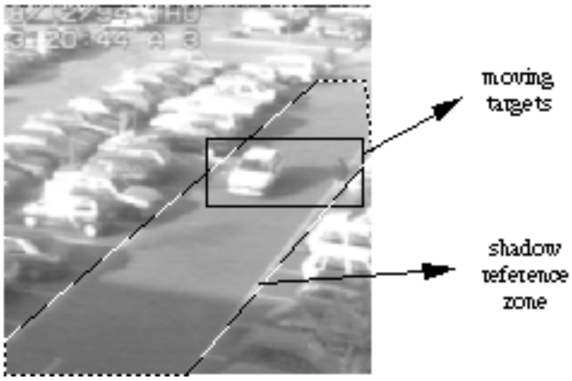

Figure 1: A typical frame of the parking sequence.

In fact, the shadows removing task requires to consider different problems. First of all, the sun can be viewed as a punctual light source (discarding the multiple-lighting problem) but, anyhow, sunlight is quite variable depending on the position in the sky, the season and the atmospheric conditions. Consequently shadows generated by the objects in the scene may be very remarkable or absent at all and a well done algorithm should be able to identify the shadow model at a given time and to quickly adapt to any lighting change. Moreover, we need to include the shadows filtering in a tracking system suitable for low-cost application and the entire system should work in real time (for instance, at 3 frames per second) on a low power machine (for our tests a PC with DSP dedicated card has been used) reserving a small time slot for the shadows analysis of each frame. This limitation implies good performances with fast, well optimized procedures to reduce the computation time as far as possible, but giving up high precisions.

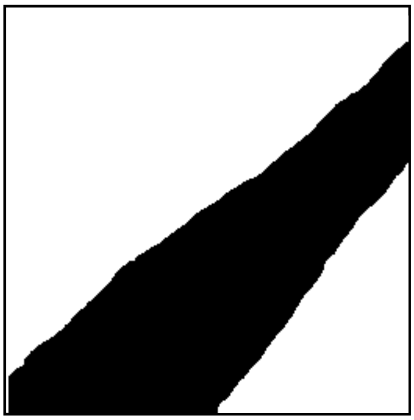

Figure 2: The reference image of fig. 1.

2 Shadow filtering

Like many colors images segmentation algorithms we do not use the RGB color space because in this space the numerical metrics do not represent color differences in a uniform scale. Therefor we chose the HLS (Hue, Luminosity, Saturation) color system.

We assume that the shadow removing system receives from a module devoted to identify moving targets an input constituted by:

- the current HLS frame;
- the binary image with the targets (blobs) identified (*blob image*);
- the list of the bounding boxes containing targets.

The output is a binary image in which those part of targets identified as shadow are removed.

2.1 Getting shadow model from a steady scene

To achieve a reliable shadow filtering it is necessary to build a chromatic model of shadow to be used as reference when new targets are examined. Since few *a priori* constraints on shadow can be assumed, the model research is performed at runtime and kept up to date with continuous observations. It is realized with two different approaches: the former looks for shadows where it is spatially more probable to find them considering the scene disposition, the latter relies on the consideration that shadows when no lighting modification happens should be very similar one to the other in spite of the highly chromatic variation of the other parts of targets.

In the first approach it is essential the analysis of those parts of the scene where targets usually advance and do not stop, so that a great part of the floor can be seen (for instance, in a parking they are the passages between rows of cars). Under these conditions we can start by analyzing this part of the scene to find shadows generated by stationary objects. The procedure is as the following.

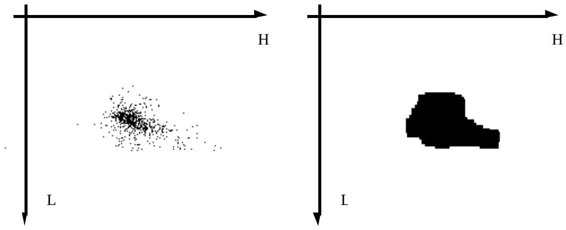

Figure 3: *GBM* for fig. 1 before and after the morphologic regularization.

1. A binary *reference image* is defined: this image, part of the total scene, is realized just once at the installation of the system, since we consider a fixed TV camera. In fig. 2 it is reported the one related to the scene of fig. 1

2. All the pixels which belong to the reference, which satisfy a uniformity criterion with respect to their eight neighbors and which are not part of blobs of the current blob image are considered.

3. These pixels are treated by a k-means clustering algorithm, working in the HLS space. The initialization uses the two pixels satisfying point 2 and having the maximum and minimum luminosity values. The k-means algorithm is implemented so that a weight can be assigned to each color component to give more emphasis to the greater informative content. In fact, the separation between shadowy and sunny parts of the floor is very definite in the L component but not in the H one where small variations in the RGB values would result in large fluctuations and would influence heavily the clustering algorithm. So we set a small weight for hue and a larger one for luminosity and saturation. The k-means algorithm reaches a good separation between clusters in less than 12 iterations.

4. The centers of the clusters are examined and the part of the reference considered shadowy is the one characterized by the lower luminosity and saturation values (in other words, the *darker* one).

5. The chromatic values of the shadow cluster are put in a two dimensional table (representing the H-L space, see fig. 3) which becomes a kind of bitmap model of shadows (called *GBM*: General Bitmap Model).

6. Morphologic erosion and dilatation are performed to make compact the region contained in the bitmap model just obtained (see again fig. 3). The analysis is iterated at a fixed rate to update the shadow model (in our application the reference is

Figure 4: The shadow map for fig. 1.

analyzed each 180 frames i.e. once a minute, at 3 frames per second).

This chain of steps could give a first idea of shadow chromatic characteristic, but it may not work if there are not steady shadows in the reference (e.g. it is cloudy day).

2.2 Masking wide shadows

Another problem arises when we deal with shadows filtering and a big part of the scene is dark due to the shadow of a building. These zones always show lower detailed definition and contrast in all the 3 HLS components, and it becomes more difficult, particularly for dark targets, to separate objects from shadows. However, we can observe that targets moving in a wide shadowy area do not generate their own shadow. Therefore, if we succeed in identifying these dark parts of the image, it is realistic to state that everything is moving inside them are real targets.

The information is stored in the shadow mask, an image having the same dimension of the image representing the scene. It must be updated at the same rate of the reference analysis and it is related to the steady shadow elements of the scene. To have a background image we report the steady elements of the current frame in the mask image iteratively. The procedures goes on till the mask is completely filled with background elements (usually less than 15 frames). Then, starting from the clusters defined in the previous section, this image is classified computing the Euclidean distance of every pixel with respect to the clusters centers. if DD (Distance from Darker) $<$ DL (Distance from Lighter), the pixel is marked as a shaded pixel in the shadow mask. Since the comparison between the two distances is not very refined and spurious parts are present, the shadow mask is thus cleared by means of a morphologic operator so that small regions are removed and wide areas are preserved (fig. 4).

2.3 Getting shadow model from targets sampling

The second way to get a shadow model refers to the similarity of the blobs. We assume that chromatic values of actual targets vary more than their shadows and introduce a matching algorithm to build a shadow model able to refine the previous one. Due to the perspective of the scene, shadow chromatic values depends also from the distance from the TV camera (shadows on the top of the image, far from TV camera, are lighter than that on the bottom). So, as we need a good model, the image is divided in a number n_s of stripes to adapt to this local property of shadows. For each strip a bitmap model is obtained. The steps of the procedure are as follows:

1. *Blob sampling.* To choose the blob sampling rate we must take into account that we do not know the tracking history of each target and we should avoid to sample many time the same blob, that may be slowly moving and be present in the scene for a long time. On the contrary we would like to get rapidly a series of significant samples for the matching algorithm so that the system can rapidly adapt to possibly lighting changes. Our tests show that taking 1 blob every 20 is a good compromise. The chosen blob must not coincide with the eventual wide shadow of the shadow mask to become a sample.

2. *Clustering the samples.* The sampled blob is clustered by means of k-means algorithm in the HLS space with the same weights adopted before to make less important hue information. The k-means algorithm works on k starting vectors randomly initialized basing on k points belonging to the blob. The algorithm is stopped after 10 iterations and the clusters centers and the points composing each cluster are stored in a data structure. The data structures are in the same number of the strips (n_s), and each data structure contains a number N of samples, where N is sufficiently large to realize a good matching. The sampled blob is assigned to the right data structure verifying its bounding box center position.

3. *Matching among clusters.* As soon as N samples are available the matching procedure starts for each strip:

   - For $i = 1$ to N the Euclidean distance of each cluster center of the i-th sample with all the cluster center of other samples is computed. The two lowest distance values for each sample are kept. along with their references to the corresponding points of the cluster.
   - A N x k vote matrix is filled in the following way. Suppose that the exam of th i-th samples gives two minimum distance values d_1 and d_2. If d_1 is the distance between the j-th cluster of the i-th sample and the l-th cluster of the k-th sample then two votes are added in the matrix in position i, j and l, k; the same happens for d_2 and for the minimum distances of other samples.

Figure 5: The moving targets of fig. 1 before and after the shadow removing process.

- The vote matrix is examined looking for the highest value, the corresponding cluster is called C_{max}.

4. *Generating the bitmap model.* Each data structure contains a two dimensional table which represents the HL space (called SBM_i : Strip Bitmap Model referred to strip i). The points belonging to C_{max} are inserted in this bitmap model so that for each instance of a couple h_i, l_i the corresponding item in the bitmap of coordinates (h_i, l_i) is incremented of a fixed value a. Then the sample list is cleared to be filled again with new samples.

5. *Aging the bitmap model.* As the shadow filter must follow lighting changes an aging mechanism is introduced to keep up to date the bitmap model. When a samples series is ready, the bitmap model of the corresponding strip is scanned and each value $v(i, j)$ is decreased of a fixed value b.

Setting b with respect to a will result in varying the algorithm memory and thus the decaying rate of old models with respect to the new ones.

2.4 Classifying shadows

Once shadow models are available the algorithm is working fully and blobs filtering can be done (fig. 5). When a bounding box is available:

1. Only blob pixels which do not coincide with shadow mask are considered.

2. A temporary copy of the bounding box is done, HL blob pixel values are compared with the bitmap model *GBM* and those couples of values which match the model are marked with a defined value (g_s). The same is done with the strip bitmap model SBM_i marking pixels with a different value (s_s).

3. All pixels marked with s_s are classified immediately as shadow, while not marked pixels are classified as actual target. Finally, pixels marked g_s are evaluated depending on their position: if they are neighbors of s_s pixels they are classified as shadow, otherwise they are target.

4. Morphological filtering is done to regularize the results.

We note that the saturation value S is taken into account by controlling that shadow pixels' values remain into the variance of the defined models.

3 Results

The algorithm has been tested on many sequences acquired from videotapes recorded by a TV camera placed on the top of a building in front of a parking. To emulate the real conditions we use the same low cost and very aged TV camera placed there by the security service. Image sequences thus show high noise in chromatic restitution which affects above all the colors similar to the grey, showing hue components that vary remarkably in the same frame in apparently uniform regions. In spite of these difficulties the tested sequences, which include a large set of situations with different weathers and diurnal lighting, have been satisfactory filtered. Trying to give a mere quantitative description of simulation results may be not so meaningful so we decided to give a less formal but more evident report. Each filtered blob is judged on the base of its qualitative appearance considering that it should be sent to a shape classification module.

The chosen classification classes are: *optimum* (shadows correctly removed and target not corrupted); *good* (most part of shadow removed and/or target almost complete); *scarce* (parts of shadow not removed and/or less than 50% of the target corrupted); *worst* (shadows not removed and/or more than 50% of the target corrupted). About 500 blobs were tested, giving 14% optimum, 46% good, 32% scarce and 8% worst results .

References

[1] J.R. Bergen, P.J. Burt, R. Hingorani and S. Peleg, "A three-frame algorithm for estimating two-component image motion", *IEEE Trans. Patt. Anal. Machine Intell.*, PAMI-14, No. 9, pp. 886-896, Sept. 1992.

[2] Z. Zhang and O.D. Faugeras, "Three dimensional motion computation and object segmentation in a long sequence of stereo frames", *International Journal of Computer Vision*, Vol. 7, pp. 211-241, 1992.

[3] F. Leymarie and M.D. Levine, "Tracking deformable objects in the plane using an active contour model", *IEEE Trans. Patt. Anal. Machine Intell.* , Vol.PAMI-15, No. 6, pp. 617-634, June 1992.

[4] D.G. Lowe, "Robust model-based motion tracking through the integration of search and estimation", *International Journal of Computer Vision*, Vol. 8, pp. 113-122, 1992.