Reflection Removal for People Detection in Video Surveillance Applications*

Dajana Conte, Pasquale Foggia, Gennaro Percannella, Francesco Tufano, and Mario Vento

Dipartimento di Ingegneria Elettronica e Ingegneria Informatica
Università di Salerno
Via Ponte don Melillo, I-84084 Fisciano (SA), Italy
{dconte,pfoggia,pergen,ftufano,mvento}@unisa.it

Abstract. In this paper we present a method removing reflection of people on shiny floors in the context of people detection for video analysis applications. The method exploits chromatic properties of the reflections and does not require a geometric model of the objects. An experimental evaluation of the proposed method, performed on a significant database containing several publicly available videos, demonstrates its effectiveness. The proposed technique also favorably compares with respect to other state of the art algorithms for reflection removal.

1 Introduction

Correct segmentation of foreground objects is important in video surveillance and other video analysis applications. In order to achieve an accurate segmentation, artifacts related to lighting issues such as shadows and reflections must be detected and properly removed. In fact, if a shadow or a reflection is mistakenly included as part of a detected foreground object, several problems may severely impact the accuracy of the subsequent phases of the application.

While many papers have been devoted to shadow removal [4,6], the problem of reflections has received comparatively much less attention; however, in some environments, reflections can be more likely than shadows, and usually they are harder to deal with. Examples are indoor scenes when the floor is smooth and shiny, or outdoor scenes in rainy weather conditions. Shadows and reflections differ under several respects; the most important differences are in position and color. The position of a shadow depends on the light sources, while reflections (assuming that the reflecting surface is a horizontal floor) are always located below the corresponding object. As regards the color, a shadow depends only on the color of the background and on the light sources (it has a darker shade of the same color of the background); on the other hand, the color of a reflection also depends on the color of the object. As a consequence of these differences, methods for shadow removal cannot be effectively applied for removing reflections.

One of the earliest work is the paper by Teschioni and Regazzoni [7], following an approach very similar to the techniques commonly used for shadow removal. In particular, a model of the color properties of a reflection is assumed; the pixels consistent with

* This research has been partially supported by A.I.Tech s.r.l., a spin-off company of the University of Salerno (www.aitech-solutions.eu).

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part I, LNCS 6978, pp. 178–186, 2011.

this model are grouped using a region growing technique, and then discarded from the foreground. The method makes the assumption that the pixels of the foreground objects are significantly different (in the RGB space) from both the ones in the background and the ones in the reflections; when this assumption is not satisfied, it is likely that parts of the objects will be mistaken as reflections, even if their position would make this unplausible.

A completely different approach is proposed by Zhao and Nevatia in [8]. Their algorithm is based on the hypothesis that the foreground object is a person, and uses a geometrical model of a person to recognize those parts of the foreground that have to be labeled as reflections. Unfortunately this method does not work if the scene includes other kinds of objects, or even people carrying large objects such as backpacks, suitcases or umbrellas.

The recent paper by Karaman et al. [5] presents a more sophisticated method that takes into account both geometric and chromatic information to remove the reflections. The method is based on the "generate and test" approach, where for each detected foreground region several hypotheses are made on the vertical position of the object baseline. For each position, the algorithm generates a synthetic reflection by combining the pixels of the background and of the part of the region that lies above the baseline, adding a blur effect to take into account the imperfect smoothness of the floor surface. Then, the baseline for which the synthetic reflection is most similar to the observed one, is selected, and all the pixels below this baseline are removed from the foreground object. This method is fairly general and robust, since it does not require an a priori knowledge of the shape of the objects. On the other hand, the "generate and test" process is computationally expensive, because for each hypothesis an image has to be generated and matched with the observed region. Furthermore, the pixel combination and blurring require parameters depending on the characteristics of the floor, implicitly assuming that the floor smoothness and reflectivity are uniform.

In this paper we propose a reflection removal technique that is similarly based on the evaluation of multiple hypotheses for the object baseline. The proposed method does not make assumptions on the characteristics of the floor surface, and so can easily work with heterogeneous floors. Furthermore, it is extremely efficient because it does not involve the actual generation of a synthetic reflection, and the test phase exploits an incremental scheme of computation to evaluate each baseline very quickly.

2 The Proposed Method

We assume that our algorithm is applied to the output of a foreground detection system based on background subtraction. It does not require a specific background subtraction technique and can be used as a postprocessing phase of any existing foreground detection module.

We briefly recall that a foreground detection system compares the current frame to a background reference image (suitably created and updated), and finds the frame pixels whose color is significantly different from the corresponding background pixels, using some sort of thresholding technique. Such pixels are grouped into connected components called *foreground regions*. Our method assumes that each foreground region contains either a single object or a group of objects at the same distance from the camera

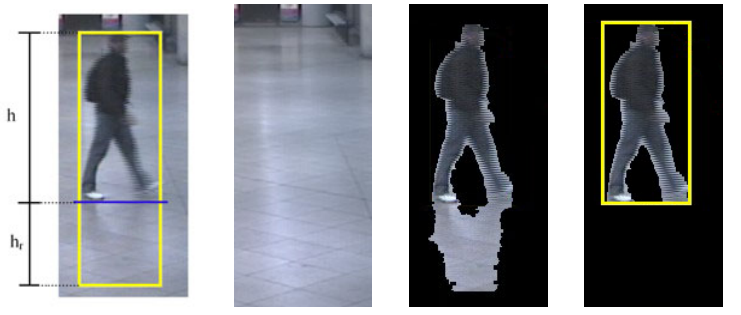

Fig. 1. a) a portion of the input image, containing a person whose height is h with its relative reflection h_r on the floor. The horizontal line represents the ideal cut separating the person from its background; b) the background reference image B(.); c) The foreground mask F(.) obtained by using a standard detection algorithm and d) the foreground mask after the removal of the reflection.

(e.g. a person with his/her luggage); hence the object can be separated from its reflection using a single horizontal line that we call the *cut line*. Note that the actual shape of the object does not need to be known in advance, so the method can be used even when the scene contains several kinds of objects. The method exploits the following property of the pixels belonging to a reflection: they are, on the average, much more similar in color to the background than the other foreground pixels are, although they are not so similar as to be considered part of the background. This happens because part of the color of the floor gets blended with the color of the reflected object to form the reflection color. Figure 1 presents an example of a person with a reflection on the floor, and the corresponding output of the foreground detection. The figure also shows the ideal cut line for this image, and the background reference image.

The proposed method, on the basis of these assumptions, determines the ideal cut line as the row of the foreground detected object that:

- minimizes the average difference in color between the detected object and the background for all the rows below it;
- on the contrary, maximizes the average difference in color between the detected object and the background for all the rows above it;

In order to quantitatively evaluate the difference in color, we introduce the following notations: $F(x, y)$ is the color of the pixel at position (x, y) in the foreground region, $B(x, y)$ the color of the corresponding pixel in the background image, and $r(k)$ the set of pixels belonging to the generic row k of the foreground region; we measure the average difference of color, along the row $r(k)$ of the detected foreground object, the following quantity:

$$d(k) = \frac{\sum_{(x,y)\in r(k)} \|F(x, y) - B(x, y)\|}{|r(k)|} \tag{1}$$

where $\|.\|$ is the Euclidean norm in the color space, and $|.|$ is the cardinality of a set.

Figure 2c reports the graph representing $d(k)$ for any row k of the detected foreground image. The actual determination of the ideal cut line is obtained on the basis of the values of $d(k)$, by considering for each candidate cut line k, the difference $\Delta(k)$ between the integral of $d(i)$ for the set of the rows above k, and the one of $d(j)$ of the set of the rows below. By denoting with $R_a(k)$ and $R_b(k)$, respectively the set of the rows above and below k, we define:

$$\Delta(k) = \frac{1}{|R_a(k)|} \cdot \sum_{i \in R_a(k)} d(i) - \frac{1}{|R_b(k)|} \cdot \sum_{j \in R_b(k)} d(j) \tag{2}$$

According to this definition, $\Delta(k)$ represents the difference between the average foreground–background dissimilarity above the candidate cut line k and the average dissimilarity below k. It is simple to verify that, if $d(i)$ is greater for the rows belonging to the object than for those of the reflection, starting from the top of the detected foreground object, $\Delta(k)$ increases, reaching its maximum in correspondence with the ideal cut line, and then it decreases as k approaches the bottom of the reflection. The ideal cut line σ can be consequently determined by searching for the relative maximum of $\Delta(k)$:

$$\sigma = k : \Delta(k) \geq \Delta(j), j \in [0, h + h_r] \tag{3}$$

Figure 2d reports the graph representing $\Delta(k)$ for any row k of the detected foreground image.

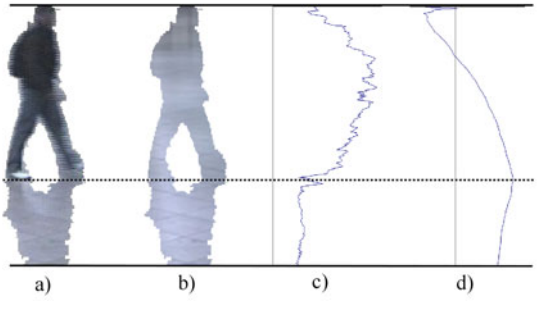

Fig. 2. a) a portion of an image and b) the corresponding background; c) the function $d(k)$ for each row of the image (k is on the vertical axis, the value of $d(k)$ on the horizontal one); d) the function $\Delta(k)$ for each row of the image (k is on the vertical axis, the value of $\Delta(k)$ on the horizontal one)

In real cases, the $\Delta(k)$ function is not as well-behaved as in the ideal case, showing a few spurious maxima in addition to the one corresponding to the ideal cut line. These spurious maxima are due to the effect of noise and to dishomogeneity in the color of the foreground object, which may be locally very similar to the background color. To filter out the spurious maxima, we have introduced the following criteria, based on geometrical and physical considerations:

a. the maximum is discarded if it is too isolated, i.e. the average value of $\Delta(k)$ in a neighborhood of the maximum differs from the maximum by more than a threshold (both the width of the neighborhood and the threshold are parameters of the algorithm); the rationale of this criterion is that an isolated maximum is more likely due to noise than to the underlying trend of the function;
b. the maximum is discarded if its position is below the middle of the detected foreground region; in fact, it is geometrically unlikely that a reflection is larger than the actual object, if the floor surface is horizontal and the object is not significantly inclined with respect to the vertical;
c. the maximum is discarded, if its value is negative; a negative value of $\Delta(k)$ would mean that the object is more similar to the background that its reflection, and this is incompatible with the assumptions of the method.

On the basis of these considerations, the method operates according to Algorithm 1.

From a computational complexity point of view, a naive computation of $\Delta(k)$ would require for each value of k the scanning of the whole detected region, in order to compute the average difference from the background above and below row k. Since this process would have to be repeated for each row, the resulting complexity would be $O(w \cdot h^2)$ (w and h are respectively the width and the height of the region).

The method described by Algorithm 1 computes the function much more efficiently, using two improvements with respect to a naive implementation:

– the algorithm keeps the values of $d(i)$ in a data structure, so that each $d(i)$ is only computed once; this would reduce the computational complexity to $O(w \cdot h + h^2)$, where the first term is due to the computation of $d(i)$ and the second term to the computation of $\Delta(k)$ given the $d(i)$ values;
– while iterating over the rows for computing $\Delta(k)$, the algorithm keeps in two variables the sum of the $d(i)$ above and below row k; these variables can be updated in $O(1)$ at each step, and avoid the need to iterate over $d(i)$ for computing the two sums of equation 2; hence the overall complexity is reduced to $O(w \cdot h + h) = O(w \cdot h)$.

Thus the proposed algorithm is very efficient even on large foreground regions, requiring a time that is negligible with respect to the overall processing of a frame.

3 Experimental Evaluation

Experiments were carried out using the object detection algorithm described in [3], characterized by a good trade-off between the detection performance and the computational complexity. The dataset used for the tests is composed by four real-world videos. All refer to indoor scenarios with reflecting floorings. The first and the second video sequences (hereinafter referred to as V1 and V2), belonging to the PETS2006 dataset [1], were taken in the hall of a railway station. Both videos show the same scene but from different view angles. The third video (hereinafter referred to as V3), was acquired by the authors nearby a subway platform. Finally, also the last video refers to a subway platform, and it belongs to the AVSS2007 dataset [2]. In all videos the reflections are

Algorithm 1. The pseudo-code of the algorithm.

{ Compute $d(.)$ and its sum }
$Sum \leftarrow 0$
for $i = 0$ to $height - 1$ **do**
  $d(i) \leftarrow \sum_{(x,y)\in r(i)} \|F(x,y) - B(x,y)\|/|r(i)|$
  $Sum \leftarrow Sum + d(i)$
end for

{ Compute $\Delta(.)$ }
$SumAbove \leftarrow d(0)$
$SumBelow \leftarrow Sum - d(0)$
for $row = 1$ to $height - 1$ **do**
  $\Delta(row) \leftarrow \frac{SumAbove}{row} - \frac{SumBelow}{height - row}$
  $SumAbove \leftarrow SumAbove + d(row)$
  $SumBelow \leftarrow SumBelow - d(row)$
end for

{Compute the best local maximum among the ones satisfying the criteria of feasibility}
$BestMax \leftarrow -1$
$BestCut \leftarrow -1$
for $row = height/2$ to $height - 1$ **do**
  if $\Delta(row)$ is a local maximum AND $\Delta(row) > 0$ AND $\Delta(row)$ is not isolated **then**
    if $\Delta(row) > BestMax$ **then**
      $BestMax \leftarrow \Delta(row)$
      $BestCut \leftarrow row$
    end if
  end if
end for
if $BestMax > 0$ **then**
  RETURN $BestCut$
else
  RETURN Nothing
end if

mainly generated by persons in the scene. For each video, a ground truth has been produced by inspecting the objects detected at each frame and choosing by hand the most appropriate cutting line, on the basis of the visual appearance. Of course, the objects missed by the used detection algorithm, as well as the wrongly detected ones (those corresponding to partial detections of the persons) have been discarded: the method cannot recover from such errors due to the previous detection phase. Table 1 reports the main characteristics of the used video sequences, and for each of them the total number of considered objects. For experimental purposes, the detected objects have been classified into two classes, *reflected objects* and *unreflected objects*: we considered an object as affected by reflection when $h_r/(h_r + h) \geq 0.15$, i.e. when its reflection is at least 15% of its apparent height.

Table 1. Dataset main characteristics. All videos were acquired at 4CIF resolution and 25 fps.

ID	Dataset / Video sequence	Number of frames	Type of objects	Total objects
V1	PETS2006 / S1-T1-C (view 1)	3021	unreflected	117
			reflected	1528
V2	PETS2006 / S1-T1-C (view 3)	3021	unreflected	2375
			reflected	556
V3	sequence acquired by the authors	880	unreflected	214
			reflected	955
V4	AVSS2007 / AB_Easy	5474	unreflected	1206
			reflected	1326

In order to measure the effectiveness of the proposed system, we have to consider that there can be two kinds of errors:

- the algorithm fails to remove completely the reflection of a detected object;
- the algorithm remove completely the reflection, but also cuts away part of the object; we call this situation an *overcut*.

The following indices have been defined to provide a quantitative evaluation of the two errors for a single object i:

$$RH(i) = \frac{h_r(i)}{h_r(i) + h(i)}, \quad OE(i) = \frac{h_o(i)}{h(i)}$$

where $h_o(i)$ is the height of the portion of the object that is erroneously removed by the algorithm in case of an overcut error.

$RH(i)$ is the height of the reflection normalized on the total height of the i-th bounding box; if the algorithm manages to completely remove the reflection, $RH(i)$ should become 0. More generally, it is expected that the value of $RH(i)$ is reduced by the application of the algorithm. We call $RH(i)$ the *reflection error*.

$OE(i)$ is a measure of the overcut relative to the true height of the object; in the ideal case, if the algorithm does not cut away a part of the object, the value of $OE(i)$ after the removal is 0. We call $OE(i)$ the *overcut error*. Notice that, before the application of the algorithm, $OE(i) = 0$, since no part of the detected region has been removed.

It is evident that only one of the two indices can be greater than 0; in fact, RH is meaningful when part of the reflection still remains after the cut, while OE must be considered when the cut removes the whole reflection and (possibly) part of the actual object. If we denote with N and M the cardinality of the two disjoint sets of boxes on which the proposed cut line is respectively below or above the ideal cut line, then the performance of the system over all the objects can be expressed in terms of the following two indices:

$$MRH = \frac{1}{N} \cdot \sum_{i:RH(i)>0} RH(i)$$

$$MOE = \frac{1}{M} \cdot \sum_{i:OE(i)>0} OE(i)$$

which are the average reflection and overcut errors.

Table 2. Performance of the proposed reflection removal method

Video ID	Method	Type of objects	Reflection Error			Overcut Error
			MRH (before)	MRH (after)	$\Delta\%$	MOE
V1	proposed method	unreflected	0.082	0.057	31.0%	0.031
		reflected	0.436	0.391	10.3%	0.030
	Karaman	unreflected	0.080	0.068	15.4%	0.071
		reflected	0.417	0.304	27.2%	0.027
V2	proposed method	unreflected	0.095	0.037	61.1%	0.026
		reflected	0.206	0.070	66.0%	0.057
	Karaman	unreflected	0.089	0.053	40.4%	0.026
		reflected	0.241	0.133	44.8%	0.106
V3	proposed method	unreflected	0.127	0.068	46.6%	0.064
		reflected	0.294	0.075	74.4%	0.039
	Karaman	unreflected	0.135	0.035	74.3%	0.063
		reflected	0.320	0.217	32.0%	0.220
V4	proposed method	unreflected	0.057	0.023	59.6%	0.040
		reflected	0.202	0.065	67.8%	0.031
	Karaman	unreflected	0.060	0.042	30.0%	0.103
		reflected	0.203	0.095	53.2%	0.063

The experimental results are reported in Table 2. For comparison, the table also reports the performance of another recent reflection removal algorithm by Karaman et al. [5] for which we have provided our own implementation. The motivation behind the choice of Karaman's algorithm is twofold: first, it is a very recent approach presented in the literature that was tested also on standard datasets, where it has shown a very interesting performance; second, our proposed approach is similar to Karaman's one as they are both based on the evaluation of multiple hypotheses for the object baseline, even if the two methods make different assumptions about the properties of the reflections. Table 2 reports in the fourth, fifth and sixth columns the reflection removal performance expressed in terms of the MRH index. This analysis was done considering only the objects with no overcut, comparing the reflection error before and after the reflection removal. The sixth column shows the relative improvement in the MRH index. Finally, the rightmost column of the table shows the MOE index.

Considering the non-overcut objects, the results show a consistent reduction of the reflection error, which is decreased in most cases by more than 50%. Notice that the error is reduced also for unreflected objects. As evident from the last column in Table 2, the algorithm does not introduce appreciable overcut errors: less than 5% in the average on both the unreflected and the reflected objects. It should be also considered that on reflected objects, an overcut means that the whole reflection has been eliminated; thus for the reflected samples reported in this table, the algorithm yields a significant improvement in the height estimation. In comparison with the algorithm by Karaman, the proposed method performs better on some video sequences (V2 and V4), and has similar results on the others (V1 and V3).

Finally, it is important to highlight that, as already anticipated in a previous section, the proposed reflection removal method has a negligible impact on the overall processing time. In fact, we have experimentally verified that the adoption of the reflection removal procedure produces an increase of 1.5% of the processing time with respect to the original foreground detection algorithm.

4 Conclusions

In this paper we have presented a novel algorithm for reflection removal, based on fairly general assumptions and computationally efficient.

The algorithm has been experimentally validated on a significant database of real videos, using quantitative measurements to assess its effectiveness. The experiments have shown that the proposed algorithm significantly reduce the error in the estimation of the actual height for objects with a reflection, while unreflected objects are left substantially unchanged. The method has been also experimentally compared with the algorithm by another recent approach for reflection removal by Karaman et al, showing in almost cases significant performance improvements.

As a future work, a more extensive experimentation will be performed, adding other algorithms to the comparison and enlarging the video database to provide a better characterization of the advantages of the proposed approach.

References

1. Dataset for PETS 2006, `http://www.cvg.rdg.ac.uk/PETS2006/`
2. i-Lids dataset for AVSS 2007, `http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`
3. Conte, D., Foggia, P., Petretta, M., Tufano, F., Vento, M.: Evaluation and improvements of a real-time background subtraction method. In: Kamel, M.S., Campilho, A.C. (eds.) ICIAR 2005. LNCS, vol. 3656, pp. 1234–1241. Springer, Heidelberg (2005)
4. Horprasert, T., Harwood, D., Davis, L.: A statistical approach for real-time robust background subtraction and shadow detection (1999)
5. Karaman, M., Goldmann, L., Sikora, T.: Improving object segmentation by reflection detection and removal. In: Proc. of SPIE-IS&T Electronic Imaging (2009)
6. Shen, J.: Motion detection in color image sequence and shadow elimination. Visual Communications and Image Processing 5308, 731–740 (2004)
7. Teschioni, A., Regazzoni, C.S.: A robust method for reflection analysis in color image sequences. In: IX European Signal Processing Conference, Eusipco 1998 (1998)
8. Zhao, T., Nevatia, R.: Tracking multiple humans in complex situations. IEEE Trans. on Pattern Analysis and Machine Intelligence 26, 1208–1221 (2004)