Shadow detection: A survey and comparative evaluation of recent methods

Andres Sanin [a,b], Conrad Sanderson [a,b,*], Brian C. Lovell [a,b]

[a] NICTA, PO Box 6020, St Lucia, QLD 4067, Australia
[b] The University of Queensland, School of ITEE, QLD 4072, Australia

ARTICLE INFO

ABSTRACT

This paper presents a survey and a comparative evaluation of recent techniques for moving cast shadow detection. We identify shadow removal as a critical step for improving object detection and tracking. The survey covers methods published during the last decade, and places them in a feature-based taxonomy comprised of four categories: chromacity, physical, geometry and textures. A selection of prominent methods across the categories is compared in terms of quantitative performance measures (shadow detection and discrimination rates, colour desaturation) as well as qualitative observations. Furthermore, we propose the use of tracking performance as an unbiased approach for determining the practical usefulness of shadow detection methods.

The evaluation indicates that all shadow detection approaches make different contributions and all have individual strength and weaknesses. Out of the selected methods, the geometry-based technique has strict assumptions and is not generalisable to various environments, but it is a straightforward choice when the objects of interest are easy to model and their shadows have different orientation. The chromacity based method is the fastest to implement and run, but it is sensitive to noise and less effective in low saturated scenes. The physical method improves upon the accuracy of the chromacity method by adapting to local shadow models, but fails when the spectral properties of the objects are similar to that of the background. The small-region texture based method is especially robust for pixels whose neighbourhood is textured, but may take longer to implement and is the most computationally expensive. The large-region texture based method produces the most accurate results, but has a significant computational load due to its multiple processing steps.

1. Introduction

Many computer vision applications dealing with video require detecting and tracking moving objects. When the objects of interest have a well defined shape, template matching or more sophisticated classifiers can be used to directly segment the objects from the image. These techniques work well for well defined objects such as vehicles but are difficult to implement for non-rigid objects such as human bodies. A more common approach for detecting people in a video sequence is to detect foreground pixels, for example via Gaussian mixture models [35,40]. However, current techniques typically have one major disadvantage: shadows tend to be classified as part of the foreground. This happens because shadows share the same movement patterns and have a similar magnitude of intensity change as that of the foreground objects [24].

Since cast shadows can be as big as the actual objects, their incorrect classification as foreground results in inaccurate detection

and decreases tracking performance. Example scenarios where detection and tracking performance are affected include: (i) several people are merged together because of their cast shadows, (ii) the inclusion of shadow pixels decreases the reliability of the appearance model for each person, increasing the likelihood of tracking loss. Both scenarios are illustrated in Fig. 1. As such, removing shadows has become an unavoidable step in the implementation of robust tracking systems [22].

The last review of shadow detection and removal techniques was done in 2003 by Prati et al. [31]. The review categorised shadow detection methods in an algorithm-based taxonomy. From each class, the authors selected one algorithm to do a comparative evaluation. The main conclusion was that only the simplest methods were suitable for generalisation, but in almost every particular scenario the results could be significantly improved by adding assumptions. As a consequence, there was no single robust shadow detection technique and it was better for each particular application to develop its own technique according to the nature of the scene.

Since the review by Prati et al. [31], many new methods have been proposed. In this paper, we present an updated survey and an extensive comparative evaluation. In our survey, we categorise the cast shadow detection methods published during the last
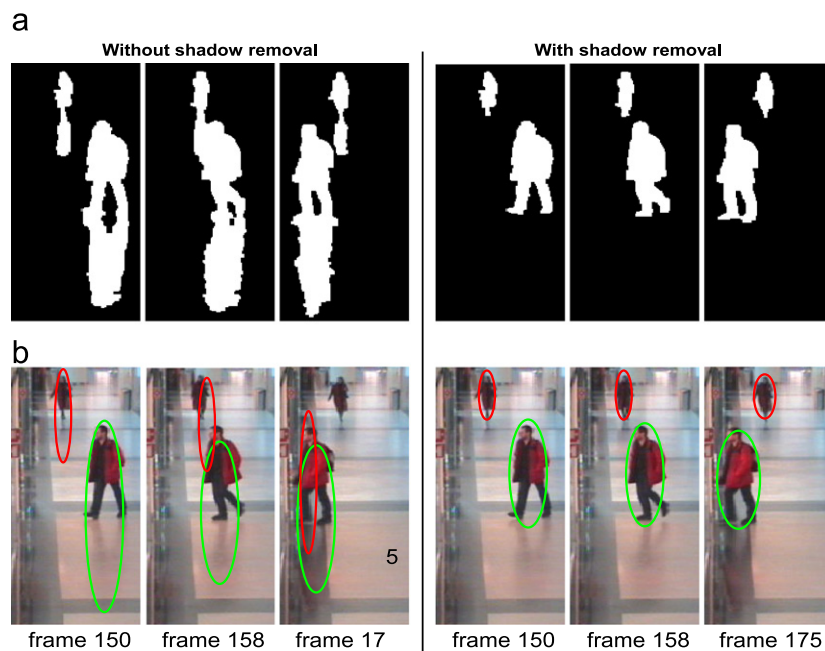
Fig. 1. A case where the correct tracking trajectory can only be obtained when shadows are removed. (a) Foreground masks without shadow removal (left group) and with shadow removal (right group). (b) Tracking results, with the boundary of each object represented by an ellipse with a unique colour. Without shadow removal the boundaries are unreliable, resulting in the loss of tracking of one of the objects. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

decade into four feature-based categories. From each category, we select and implement one or more prominent methods. We compare the selected methods in a detailed evaluation, involving qualitative, quantitative and applied experiments, to show the strengths and weaknesses of each method.

We continue the paper as follows. In Section 2, we describe the main features that are used for detecting moving cast shadows and use them to categorise recent shadow detection methods. In Section 3, a subset of the methods is chosen for implementation, with each selected method explained in detail. In Section 4 we perform an extensive quantitative and qualitative comparison of the selected methods. In Section 5 we present our concluding remarks.

2. Detecting moving cast shadows

Prati et al. [31] classified shadow detection methods in an algorithm-based taxonomy. As a secondary classification, they mentioned the types of features used by each method among three broad classes: spectral, spatial and temporal features. We have observed that the choice of features has greater impact on shadow detection results compared to the choice of algorithms. Therefore, we present a feature-based taxonomy with a secondary mention of the types of algorithms. Furthermore, we divide spectral features into intensity, chromacity and physical properties. We divide spatial features into geometry and textures.

In Section 2.1, we explain how each type of feature can be used to detect shadows. In Section 2.2, we categorise recent shadow detection methods into a feature-based taxonomy.

2.1. Useful features for shadow detection

Most of the following features are more useful for detecting shadows when the frame, which contains objects and their shadows, can be compared with an estimation of the background,

which has no objects or moving cast shadows. This review focuses in detecting shadows produced by moving objects in video sequences, where it is reasonable to assume that a clear view of the background can be obtained, or that the background can be estimated even in the presence of foreground objects [34,43].

2.1.1. Intensity

The simplest assumption that can be used to detect cast shadows is that regions under shadow become darker as they are blocked from the illumination source. Furthermore, since there is also ambient illumination, there is a limit on how much darker they can become. These assumptions can be used to predict the range of intensity reduction of a region under shadow, which is often used as a first stage to reject non-shadow regions [12,13,17,42,48]. However, there are no methods which rely primarily on intensity information for discriminating between shadows and objects.

2.1.2. Chromacity

Most shadow detection methods based on spectral features use colour information. They use the assumption that regions under shadow become darker but retain their chromacity. Chromacity is a measure of colour that is independent of intensity. For instance, after a green pixel is covered by shadow it becomes dark-green, which is darker than green but has the same chromacity. This colour transition model where the intensity is reduced but the chromacity remains the same is normally referred to as colour constancy [11] or linear attenuation [13]. Methods that use this model for detecting shadows often choose a colour space with better separation between chromacity and intensity than the RGB colour space (e.g. HSV [7], $c1c2c3$ [36], YUV [6], normalised RGB [4]), or a combination of them [41]. Most of these methods are simple to implement and computationally inexpensive. However, because they make comparisons at the pixel-level, they are susceptible to noise [31]. Furthermore, they

Table 1

Taxonomy of recently published shadow detection methods. Methods are sorted by year in the first column and grouped by category. The additional columns show secondary classifications within each category. Chromacity-based methods are subdivided according their colour space, level of granularity and additional spatial or temporal verification. Physical methods are subdivided according to their physical shadow model, learning algorithm and additional spatial or temporal cues. Geometry-based methods are subdivided according to their supported object type, whether they support multiple objects per blob, their main geometrical cue and additional cues. Texture-based methods are subdivided according to their weak shadow detector, texture correlation method and the size of the regions used in the correlation. The highlighted methods were chosen for the comparative evaluation.

Chromacity-based methods

Paper	Colour space	Level	Spatial verification	Temporal verification
Cucchiara et al.(2003) [7]	HSV	Pixel	–	–
Salvador et al. (2004) [36]	c1c2c3	Window	No internal shadows	–
Cavallaro et al. (2005) [4]	Normalised RGB	Pixel	No internal shadows	Tracking life-span
Chen et al. (2010) [6]	YUV	Pixel	Morphology	–
Sun and Li (2010) [41]	HSI and c1c2c3	Pixel	Morphology	–

Physical methods

Paper	Model	Learning	Spatial or temporal cues
Nadimi and Bhanu (2004) [24]	Dichromatic	–	Spatio-temporal test
Porikli and Thornton (2005) [30]	–	Shadow flow	–
Liu et al. (2007) [18]	–	Gaussian mixture model	Markov random fields and tracking
Martel-Brisson and Zaccarin (2007) [20]	–	Gaussian mixture model	–
Martel-Brisson and Zaccarin (2008) [21]	General	Kernel based	Gradients (direction)
Joshi and Papanikolopoulos (2008) [15]	–	Semisupervised (SVM)	Edges
Huang and Chen (2009) [13]	General	Gaussian mixture model	Gradients (attenuation)

Geometry-based methods

Paper	Objects	Blob segmentation	Main cue	Other cues
Hsieh et al. (2003) [12]	People	Via head detection	Orientation	Intensity and location
Yoneyama et al. (2003) [47]	Vehicles	–	2D models	Vanishing point
Nicolas and Pinel (2006) [27]	Any	–	Light source	Temporal filter
Fang et al. (2008) [8]	Vehicles	–	Wave transform	Spectral
Chen and Aggarwal (2010) [5]	People	–	Log-polar coordinates	Colour and oriented gradients

Texture-based methods

Paper	Weak detector	Texture correlation	Correlation level
Javed and Shah (2002) [14]	Colour segmentation	Gradient direction correlation	Medium region
Xu et al. (2005) [46]	–	Static edge correlation	Pixel
Tian et al. (2005) [42]	Intensity range	Normalised cross-correlation	Pixel
Wang et al. (2006) [45]	–	Intensity and edge DCRF filter	Small region
Zhang et al. (2006) [48]	–	Orthogonal transforms	Small region
Leone and Distante (2007) [17]	Photometric gain	Gabor filter	Small region
Zhang et al. (2007) [49]	Intensity constraint	Ratio edge test	Small region
Nghiem et al. (2008) [26]	Chromacity based	Intensity reduction ratio	Small region
Shoaib et al. (2009) [39]	–	Gradient background subtraction	Pixel
Pei and Wang (2009) [29]	–	PCA based	Small region
Sanin et al. (2010) [37]	Chromacity based	Gradient direction correlation	Large region
Nakagami and Nishitani (2010) [25]	–	Walsh transform domain	Small region
Panicker and Wilscy (2010) [28]	–	Foreground edge detection	Pixel
Qin et al. (2010) [32]	Shadow colour model	Local ternary pattern MRF	Small region

are sensitive to strong illumination changes and fail with strong shadows [24].

2.1.3. Physical properties

The linear attenuation model assumes that the illumination source produces pure white light [24], which is often not the case. In outdoors environments, the two major illumination sources are the sun (white light) and the light reflected from the sky (blue light). Normally, the white light from the sun dominates any other light source. When the sun's light is blocked, the effect of sky illumination increases, shifting the chromacity of the region under shadow towards the blue component. Nadimi and Bhanu [24] proposed a dichromatic model which takes into account both illumination sources to better predict the colour change of shadowed regions. Further work has been done to create more general non-linear attenuation models accounting for various illumination conditions in both indoor and outdoor scenarios [13,21]. Alternatively, some methods address the non-linear attenuation problem by learning the appearance that every pixel has under shadow without explicitly proposing an attenuation model [15,18,20,30]. These methods that try to model or learn the specific appearance of shadow pixels are typically referred to as physical approaches. By learning or modelling particular scenarios, these methods tend to be more accurate than chromacity methods (direct comparisons are reported in [15,18]). However, since they are still limited to spectral properties, their main disadvantage involves dealing with objects having similar chromacity to that of the background [13].

2.1.4. Geometry

In theory, the orientation, size and even shape of the shadows can be predicted with proper knowledge of the illumination source, object shape and the ground plane. Some methods use this information to split shadows from objects [5,8,12,27,47]. The main advantage of geometry features is that they work directly in the input frame; therefore, they do not rely on an accurate estimation of the background reference. However, methods that use geometry features impose scene limitations such as: specific object types, typically pedestrians (ie. standing people) [5,12] or vehicles [8,47]; requiring objects and shadows to have different orientation [5,12]; and assuming a unique light source [27] or a flat background surface [8]. Additionally, current geometry-based methods are not designed to deal with objects having multiple shadows or (except for [12]) multiple objects detected as a single foreground blob.

2.1.5. Textures

Some methods exploit the fact that regions under shadow retain most of their texture. Texture-based shadow detection methods typically follow two steps: (1) selection of candidate shadow pixels or regions, and (2) classification of the candidate pixels or regions as either foreground or shadow based on texture correlation. Selection of the shadow candidates is done with a weak shadow detector, usually based on spectral features. Then, each shadow candidate is classified as either object or shadow by correlating the texture in the frame with the texture in the background reference. If a candidate's texture is similar in both the frame and the background, it is classified as shadow. Various methods perform this correlation with various techniques (e.g. normalised cross-correlation [42], gradient or edge correlation [14,37,46], orthogonal transforms [48], Markov or conditional random fields [32,45], Gabor filtering [17]). Texture correlation is a potentially powerful method for detecting shadows as textures are highly distinctive, do not depend on colours, and are robust to illumination changes. However, texture-based shadow detection methods tend to be slow as they often have to compute one or several neighbourhood comparisons for each pixel.

2.1.6. Temporal features

Finally, since moving cast shadows share the same movement pattern as the objects that produce them, the same temporal consistency filters that have been applied to the objects can be applied to the shadows [4,18,24,27]. This filtering usually enhances the detection results by keeping only the pixels that are consistent in time. However, as with the intensity features, there are no methods which rely primarily on temporal features for shadow detection.

2.2. Taxonomy of recent shadow detection methods

We categorised the shadow detection methods published during the last decade according to their feature choice. Although some of the methods use more than one feature, we take into account the feature that makes the dominant contribution to the detection results or the novelty of the paper. As mentioned in Section 2.1, intensity features are used mainly as a first step for detecting shadows, and temporal features are mainly used for filtering the detection results. Therefore, all the reviewed methods were classified into one of four categories: (i) chromacity-based methods, (ii) physical methods, (iii) geometry-based methods, (iv) texture-based methods. Our taxonomy is detailed in Table 1. The highlighted methods were chosen for the comparative evaluation and are explained in Section 3.

3. Methods selected for implementation

For practical reasons, only a subset of the reviewed methods was implemented. In this section, we detail a selection of one or more prominent methods from each category. These methods were implemented and used for the comparative evaluation in Section 4.

3.1. Chromacity-based method

Among the chromacity methods, the most important factor is to choose a colour space with a separation of intensity and chromacity. Several colour spaces such as HSV [7], c1c2c3 [36] and normalised RGB [4] have proved to be robust for shadow detection [38]. We chose the HSV approach proposed by Cucchiara et al. [7], since that colour space provides a natural separation between chromacity and luminosity. This shadow detection method has been widely used in surveillance applications (e.g. [9,19]). Since the value (V) is a direct measure of intensity, pixels in the shadow should have a lower value than pixels in the background. Following the chromacity cues, a shadow cast on background does not change its hue (H) and the authors noted that shadows often lower the saturation (S) of the points. Therefore, a pixel p is considered to be part of a shadow if the following three conditions are satisfied:

$$\beta_1 \leq (F_p^V/B_p^V) \leq \beta_2 \tag{1}$$

$$(F_p^S - B_p^S) \leq \tau_S \tag{2}$$

$$|F_p^H - B_p^H| \leq \tau_H \tag{3}$$

where F_p^C and B_p^C represent the component values, C, of HSV for the pixel position p in the frame (F) and in the background reference image (B), respectively. β_1, β_2, τ_S and τ_H represent thresholds that are optimised empirically. Working with alternative colour spaces

may produce different but not necessarily better results [38]. However, extending the pixel-level based analysis to an observation window improves results by countering pixel-level noise [36]. In our implementation of the HSV method we used a 5-by-5 observation window rather than treating each pixel separately.

3.2. Physical method

Research in physical models for cast shadow removal has been done incrementally. The more recent papers are extensions of previous physical models, typically removing some assumptions and improving on previous results. We chose a recent approach by Huang and Chen [13] which does not make prior assumptions about the light sources and ambient illumination, and reports better results than similar methods. For a pixel p, given the vector from shadow to background value denoted as $v(p)$, the colour change is modelled using the 3D colour feature $x(p) = [\alpha(p), \theta(p), \phi(p)]$. Here, $\alpha(p)$ represents the illumination attenuation, while $\theta(p)$ and $\phi(p)$ indicate the direction of $v(p)$ in spherical coordinates

$$\alpha(p) = \|v(p)\| / \|BG(p)\| \tag{4}$$

$$\theta(p) = \arctan(v^G(p) / v^R(p)) \tag{5}$$

$$\phi(p) = \arccos(v^B(p) / \|v(p)\|) \tag{6}$$

where $BG(p)$ is the background value at the pixel p, and the superscripts R, G, B indicate the components in the RGB colour space. This colour feature describes the appearance variation induced by the blocked light sources on shaded regions. The model is learned in an unsupervised way. First, a weak shadow detector identifies pixels in the foreground that have reduced luminance and different saturation from that of the background. Then, the attenuation of these candidate shadow pixels is used to update a Gaussian mixture model of the 3D colour features, penalising the learning rate of pixels with larger gradient intensities than the background, which are more likely to be foreground objects. Finally, posterior probabilities of the model are used to classify each pixel in the foreground as object or shadow.

3.3. Geometry-based method

Most geometry methods assume that each foreground blob contains a single object and shadow, which is not guaranteed in many computer vision applications. For this reason, we chose the method proposed by Hsieh et al. [12], which separates the blobs into individual objects before doing the geometric analysis. As in most geometry-based methods, their work assumes that the objects of interest are persons and that their shadows have a different orientation. First, they analyse the vertical peaks on each blob to detect potential heads, and then use this information to split the blobs into person-shadow pairs. Given a person-shadow region R, its centre of gravity $(\overline{x}, \overline{y})$ and orientation θ are found as follows:

$$(\overline{x}, \overline{y}) = \left(\frac{1}{|R|} \sum_{(x,y) \in R} x, \frac{1}{|R|} \sum_{(x,y) \in R} y \right) \tag{7}$$

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right) \tag{8}$$

where $|R|$ is the area of the region in pixels, and $\mu_{p,q}$ are the correspondent central moments. The point below the centre of gravity with the maximum vertical change is considered to be the pixel where the shadow begins, and a segment oriented according to θ is used to roughly split a candidate shadow region R_2. Then,

the following Gaussian model is built from the pixels in R_2:

$$G(s,t,g) = \exp\left[-\left(\frac{w_s s^2}{\sigma_s^2} + \frac{w_t t^2}{\sigma_t^2} + \frac{w_g (g - \mu_g)^2}{\sigma_g^2} \right) \right] \tag{9}$$

where s and t are the elliptical coordinates of each pixel and g its intensity (i.e. $g = I(s,t)$), and w and σ^2 are the weight and variance of each component in the Gaussian. This model summarises the intensity of the shadow pixels and includes the coordinates where the shadow is expected to be. Once the model is built, every pixel in the original region R is classified as object or shadow, according to whether it agrees with the Gaussian model or not.

3.4. Small region (SR) texture-based method

Texture-based methods present the greatest diversity among the various categories. As a representative of methods which use small regions (or neighbourhoods) to correlate textures, we chose the method proposed by Leone and Distante [17], as it correlates textures using Gabor functions. Region-level correlation is more robust than pixel-level correlation and Gabor functions can provide optimal joint localisation in the spatial/frequency domains [33]. As in the majority of the texture-based shadow detection methods, the method first creates a mask with the potential shadow pixels in the foreground. Then, if the textures of small region centered at each pixel are correlated to the background reference, the pixels are classified as shadow. In this case, the potential shadow pixels are found using a photometric gain measure which gives higher probability to pixels whose intensity is lower than the background. The texture analysis is performed by projecting a neighbourhood of pixels onto a set of Gabor functions with various bandwidths, orientations and phases, and the matching between frame and background is found using Euclidean distance. Since a full Gabor dictionary may be expensive to compute, a sub-dictionary with the most representative kernels can be first found using the matching pursuit strategy [1].

3.5. Large region (LR) texture-based method

The problem of using small regions is that they are not guaranteed to contain significant textures. Sanin et al. [37] proposed using colour features to first create large candidate shadow regions (ideally containing whole shadow areas), which are then discriminated from objects using gradient-based texture correlation.

The purpose of the first step is to create regions of pixels as large as possible which contain shadow pixels or object pixels, but not both. Candidate shadow pixels are found using intensity and chromacity features in the HSV colour space as in [7], but adapting the thresholds to ensure high detection accuracy. Classifying all shadow pixels as shadows is important in this step because it will impose the upper bound of the final detection accuracy. Misclassifying object pixels as shadows is not an issue since they are later discriminated using texture features. The connected components of shadow pixels are used as candidate shadow regions. Some candidate shadow regions may contain both shadow and object pixels. To avoid this scenario, we implemented the optional step described in [37] where regions are split using edges that occur in the foreground but not in the background reference.

In the second step, the texture for each candidate region is correlated between the frame and the background reference. Since shadows tend to preserve the underlying textures, shadow regions should have a high texture correlation. First, for each candidate region, the gradient magnitude $|\nabla_p|$ and gradient

direction θ_p at each pixel $p = (x,y)$ are calculated using

$$|\nabla_p| = \sqrt{\nabla_x^2 + \nabla_y^2} \tag{10}$$

$$\theta_p = \arctan2(\nabla_y/\nabla_x) \tag{11}$$

where ∇_y is the vertical gradient (difference in intensity between the pixel and the pixel in the next row), while ∇_x is the horizontal gradient. The function $\arctan2(\cdot)$ is a variant of $\arctan(\cdot)$ that returns an angle in the full angular range $[-\pi,\pi]$, allowing the gradient direction to be treated as a true circular variable. Only the pixels with $|\nabla_p|$ greater than a certain threshold τ_m are taken into account to avoid the effects of noise, which is stronger in the smooth regions of the frame.

Since the gradient direction is a circular variable, the difference has to be calculated as an angular distance. For each pixel $p = (x,y)$ that was selected due to significant magnitude, the difference in gradient direction between the frame F and the background reference B is calculated using

$$\Delta\theta_p = \arccos \frac{\nabla_x^F \nabla_x^B + \nabla_y^F \nabla_y^B}{\sqrt{(\nabla_x^F 2 + \nabla_y^F 2)(\nabla_x^B 2 + \nabla_y^B 2)}} \tag{12}$$

The gradient direction correlation between the frame and the background is estimated using

$$c = \frac{\sum_{p=1}^{n} H(\tau_a - \Delta\theta_p)}{n} \tag{13}$$

where n is the number of pixels selected in the candidate shadow region and $H(\cdot)$ is the unit step function which, in this case, evaluates to 1 if the angular difference is less than or equal to the threshold τ_a, and 0 otherwise. In essence, c is the fraction of pixels in the region whose gradient direction is similar in both the frame and the background. If c is greater than threshold τ_c, the candidate region is considered a shadow region and it is removed from the foreground mask.

4. Comparative evaluation

In this section, we present two sets of experiments to compare the performance of the five methods[1] selected in Section 2. The first experiment is a direct measure of shadow detection performance. We show both quantitative and qualitative results. The second experiment is an applied empirical study that shows the improvement in tracking performance after using each of the five compared methods. Below, we describe the sequences used for the experiments and the steps for creating the ground truth frames. In the following subsections, we explain each experiment in detail and present the results.

The sequences used in our experiments are summarised in Table 2. We used a wide range of scenes with variations in the type and size of objects and shadows. The first six sequences were introduced in [31][2] and [21][3] and have been widely used for testing shadow detection performance. The last entry summarises 25 sequences from the CAVIAR dataset.[4] Each sequence presents a different challenge for the shadow detection methods to tests their robustness. The *Campus* sequence is a particularly noisy outdoor sequence where some of the shadows are extremely long. The *Hallway* sequence has a textured background and the size of the people changes significantly according to their distance to the

camera. The *Highway* 1 and *Highway* 3 sequences show a traffic environment with two different lighting conditions and vehicle sizes. In particular, the *Highway* 3 has very small vehicles which could be misclassified as shadows. The *Lab* and *Room* indoor sequences show two laboratory rooms in two different perspectives and lighting conditions. From the CAVIAR dataset, we selected 25 sequences that show people walking in various patterns inside a shopping mall.

Manually labelling shadow pixels in a frame is much more difficult than labelling object pixels. For instance, it may be easy to determine which pixels belong to the people in the frame in Fig. 2 (a), but it is difficult to tell which pixels correspond to their cast shadows. For this reason, we first used the standard Gaussian mixture model (GMM) foreground extraction method from OpenCV 2.0 [3] to extract a foreground mask on each frame. The resulting mask should contain only object and shadow pixels. By superimposing the original frame on the mask, labelling pixels in the mask as object or cast shadow becomes straightforward. The labelling process is summarised in Fig. 2. The labelled masks, along with the original frames and the backgrounds estimated with the GMM method, are a further contribution of this paper.[5] The original sequences can be used for learning-based methods which require all the intermediate frames.

4.1. Shadow detection performance

First, we measured the shadow detection performance of each method in every test sequence. We then gradually decreased the colour information of each sequence to test the dependency of each method on colour features. Finally, we show qualitative results for each sequence and summarise their observed behaviour. All the methods were faithfully implemented as described in Section 2. The thresholds for all methods were selected to obtain the best overall performance on the test sequences, using one setting for all sequences (i.e. no specific tuning for each sequence).

4.1.1. Quantitative results

To test the shadow detection performance of the five methods we used the two metrics proposed by Prati et al. [31], namely shadow detection rate (η) and shadow discrimination rate (ξ)

$$\eta = \frac{TP_S}{TP_S + FN_S} \tag{14}$$

$$\xi = \frac{TP_F}{TP_F + FN_F} \tag{15}$$

where TP and FN stand for true positive and false negative pixels with respect to either shadows (S) or foreground objects (F). The shadow detection rate is concerned with labelling the maximum number of cast shadow pixels as shadows. The shadow discrimination rate is concerned with maintaining the pixels that belong to the moving object as foreground. In this paper we often use the average of the two rates as a single performance measure.

Fig. 3 shows the average shadow detection and discrimination rates on each test sequence. Each bar represents the average of the detection and discrimination rates on each sequence, while the individual detection and discrimination rates are indicated by a square and a circle, respectively. In all cases, the large region texture-based method performs considerably better than all the others, obtaining high values for both the detection and discrimination rates in all sequences. We discuss these sequence-related results more deeply in the qualitative results section.

The average processing time per frame of each shadow detection method is shown in Table 3. Times are shown in milliseconds,

---

[1] C++ source code for the implemented methods can be obtained from http://arma.sourceforge.net/shadows/

[2] http://cvrr.ucsd.edu/aton/shadow/

[3] http://vision.gel.ulaval.ca/~CastShadows/

[4] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
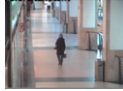
[5] The ground truth masks can be obtained from http://arma.sourceforge.net/shadows/

Table 2

Image sequences used in the comparative evaluation. Sequences are described in terms of: number and size of frames and number of examples with manually labelled shadows; scene type, surface and noise; type, size (relative to the frame size) and speed (in pixels/frame) of the objects; and size, strength and direction of the shadows. The main features or challenges presented by each sequence are highlighted using bold text. Note that the last column summarises the properties of 25 sequences from the CAVIAR dataset, all of which are different recordings of the same scene.

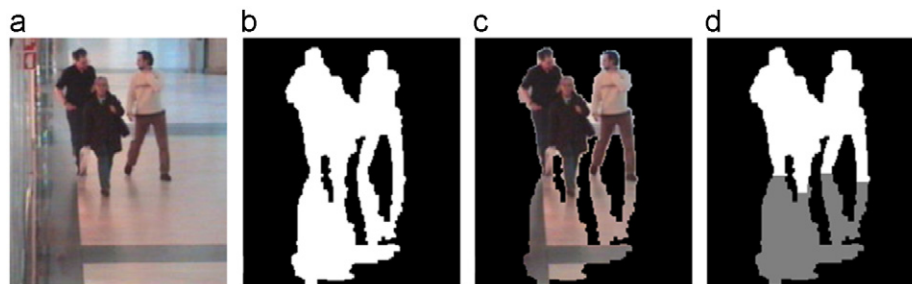|	Campus	Hallway	Highway 1	Highway 3	Lab	Room	CAVIAR
Frames							
Number	1179	1800	440	2227	887	300	1388 ($\times$ 25)
Labelled	53	13	8	7	14	22	45 ($\times$ 25)
Size	352 $\times$ 288	320 $\times$ 240	320 $\times$ 240	320 $\times$ 240	320 $\times$ 240	320 $\times$ 240	384 $\times$ 288
Scene							
Type	Outdoor	Indoor	Outdoor	Outdoor	Indoor	Indoor	Indoor
Surface	Asphalt	**Textured**	Asphalt	Asphalt	**White**	Carpet	**Reflective**
Noise	**High**	Medium	Medium	Medium	Low	Medium	Medium
Objects							
Type	Vehicles/people	People	Vehicles	Vehicles	People/other	People	People
Size	Medium	**Variable**	Large	**Small**	Medium	Medium	Medium
Speed	5–10	5–15	**30–35**	20–25	10–15	2–5	1–4
Shadows							
Size	**Very large**	Medium	Large	Small	Medium	**Large**	Medium
Strength	Weak	Weak	**Strong**	**Very strong**	Weak	Medium	Medium
Direction	Horizontal	Multiple	Horizontal	Horizontal	**Multiple**	**Multiple**	**Vertical**

Fig. 2. Ground truth creation steps: (a) original frame; (b) foreground mask obtained with GMM; (c) frame superimposed on foreground mask; (c) labelled mask with object pixels in white and shadow pixels in grey.
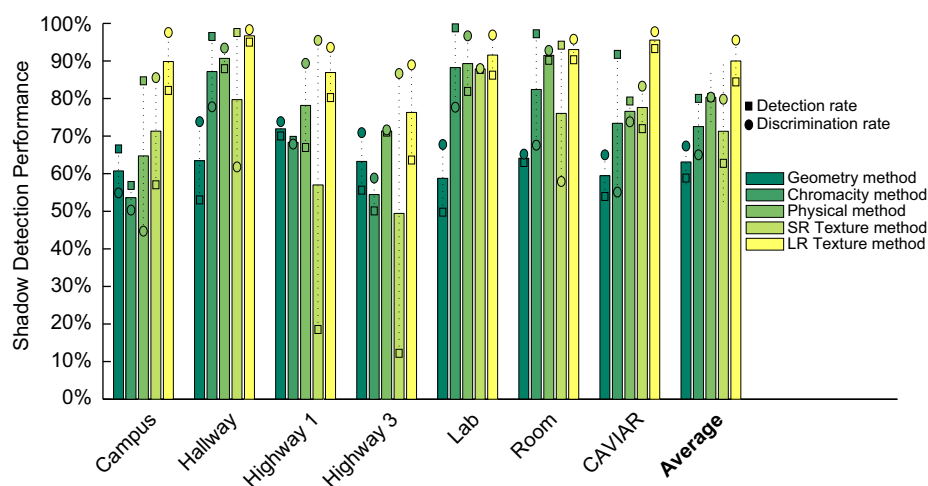

Fig. 3. Comparison of shadow detection results by sequence. The performance measure is the average between the shadow detection and discrimination rates. Each bar represents the average performance with a vertical segment whose length indicates the average difference between the detection rate (marked with a square) and the discrimination rate (marked with a circle).

obtained on a 32-bit Intel CPU running at 2.6 GHz. Four of the methods, excluding the small region texture-based, have the same asymptotic complexity of constant operations per pixel. The chromacity-based method was the quickest to implement and run. The geometry-based and physical methods need more operations for calculating the central moments and updating the shadow models,

Table 3
Average frame processing time (in milliseconds) per sequence for various shadow detection methods. The first value shown for the small region (SR) texture-based method corresponds to the processing time using 48 Gabor kernels. This configuration was used in the previous experiments for maximum accuracy. The second value in brackets represents the time when using a reduced set of 16 Gabor kernels (for faster processing).

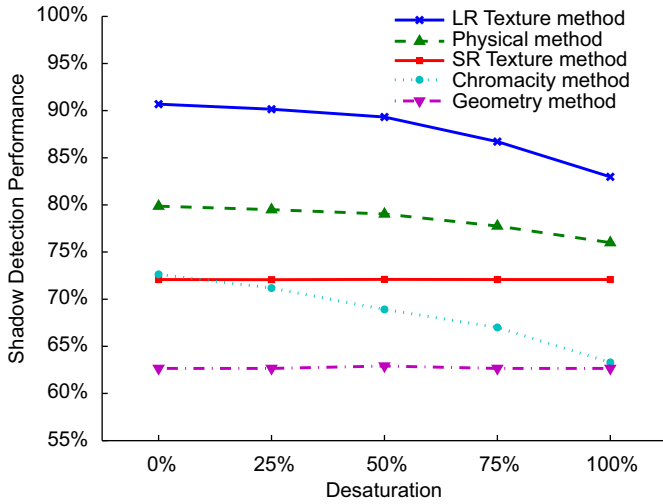|	Chromacity	Geometry	Physical	SR textures	LR textures
Campus	8.72	9.44	10.00	156.46 (48.81)	20.76
Hallway	11.28	8.91	12.81	223.64 (77.37)	21.59
*Highway*1	10.73	24.75	16.93	341.32 (116.28)	34.71
*Highway*3	6.82	6.49	7.15	120.36 (37.07)	11.75
Lab	8.95	17.68	15.34	253.82 (82.48)	22.73
Room	7.14	8.41	8.51	144.87 (47.07)	16.25
Caviar	10.82	13.94	14.08	243.07 (82.98)	24.70
Average	**9.21**	**12.80**	**12.12**	**211.93 (70.30)**	**21.78**

Fig. 4. Shadow detection performance when the colour information is gradually reduced. Each line shows the average performance of a shadow detection method ranging from the original sequences (0% desaturated) to the greyscale-converted sequences (100% desaturated). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

respectively. The large region texture-based method requires extra steps to generate the candidate shadow regions and calculate the gradients for each pixel, and is the slowest of these four. The amount of operations per pixel in the small region texture-based method depends on the size and number of kernels used for correlating the textures. This small region texture-based method has considerably higher computational load than the rest, even when the number of kernels and their size are minimised.

The large region texture-based method was designed firstly to increase the effectiveness of texture-based features by selecting large regions, and secondly to use spectral features to improve results in the absence of significant textures. However, it is hard to conclude how well the method achieves these two goals by simply observing the shadow detection performance on various sequences. For this reason, we tested the performance of each method while gradually decreasing the colour information of each sequence until reaching greyscale frames. This experiment is used for two purposes: (1) to make a fair comparison between both texture-based methods when colour information is lost and thus determine which one is using texture information more effectively; (2) to observe how the colour information is used when available, and how dependent are the methods on colour features.

Fig. 4 shows the average performance of each method across all sequences as the colour information is manually decreased. The performance results indicate the average between shadow

detection and discrimination rates. The desaturation rate indicates the reduction in colour information used to modify the frames as follows:

$$F_{desat} = (1-\lambda)F_{orig} + \lambda F_{grey} \qquad (16)$$

where λ is the desaturation rate, F_{orig} is the original frame and F_{grey} is the frame converted to greyscale. In other words, when the desaturation is 0% the original frames are used, when it is 100% the frames are converted to greyscale, and the rest are interpolations between the two, created by gradual blending to avoid colour distortions. Note that alternative ways of reducing the colour information can also be used. For example, it is possible to represent the images in HSV colour space and gradually decrease the saturation (S) component to remove colour.

Several things can be concluded from this figure. First, as expected, methods which use colour features perform better when all the colour information is available, and their performance drops as the frames are desaturated. Second, the decrease in performance of the physical method is lower since it is locally adaptive; the performance of the large region texture-based method also decreases, but with a magnitude lower than that of the chromacity method, since it is globally adaptive. Third, without colours the large region texture-based method depends on textures while performing considerably better than the small region texture-based method. Last, the large region texture-based method performs better than all others regardless of the degree of desaturation; while it uses colour information to improve performance, without such information its performance is nevertheless still good.

4.1.2. Qualitative results

We show the qualitative results in Fig. 5. The first column shows an example frame for each sequence, the second column shows the expected results, where foreground pixels are marked in blue and shadow pixels are marked in green. The remaining columns show the observed results with each shadow detection method. For symmetry, we show results for all sequences and all methods. However, it is important to note that the geometry-based method was designed for pedestrian shadow detection and, apart from mentioning the limitation, we do not take into account the sequences with objects other than pedestrians when comparing to this particular method. In general, as was also shown in the qualitative results, the large region texture-based method performs better in all the examples.

Specific observations can be also done for each method. The geometry-based method only works when each shadow has a unique orientation which differs to the object's orientation, as happens in the *Campus* example, but fails when the shadows have the same orientations as the objects or when shadows have multiple directions as in the *Lab* example. The chromacity-based method is affected by pixel-level noise, and fails when objects are darker and have similar colours to the background as in the *Campus* example. Although the physical method also uses chromacity features, it has the ability to adapt to each scene and it is less likely to fail when the chromacity-based method fails. The small region texture-based method works well in scenes with textured backgrounds but fails for pixels located in non-textured neighbourhoods (as can be seen in the *Highway* 1 example). Finally, the large region texture-based method works well in most cases, although it can distort the object contours as in the *Lab* example.

The results from both the quantitative experiments and qualitative observations are summarised in Table 4. We assign scores to each method according to several criteria. The geometry-based method has strong assumptions regarding the object and shadow shape, but when these assumptions are met the method works well independently on the quality of spectral and
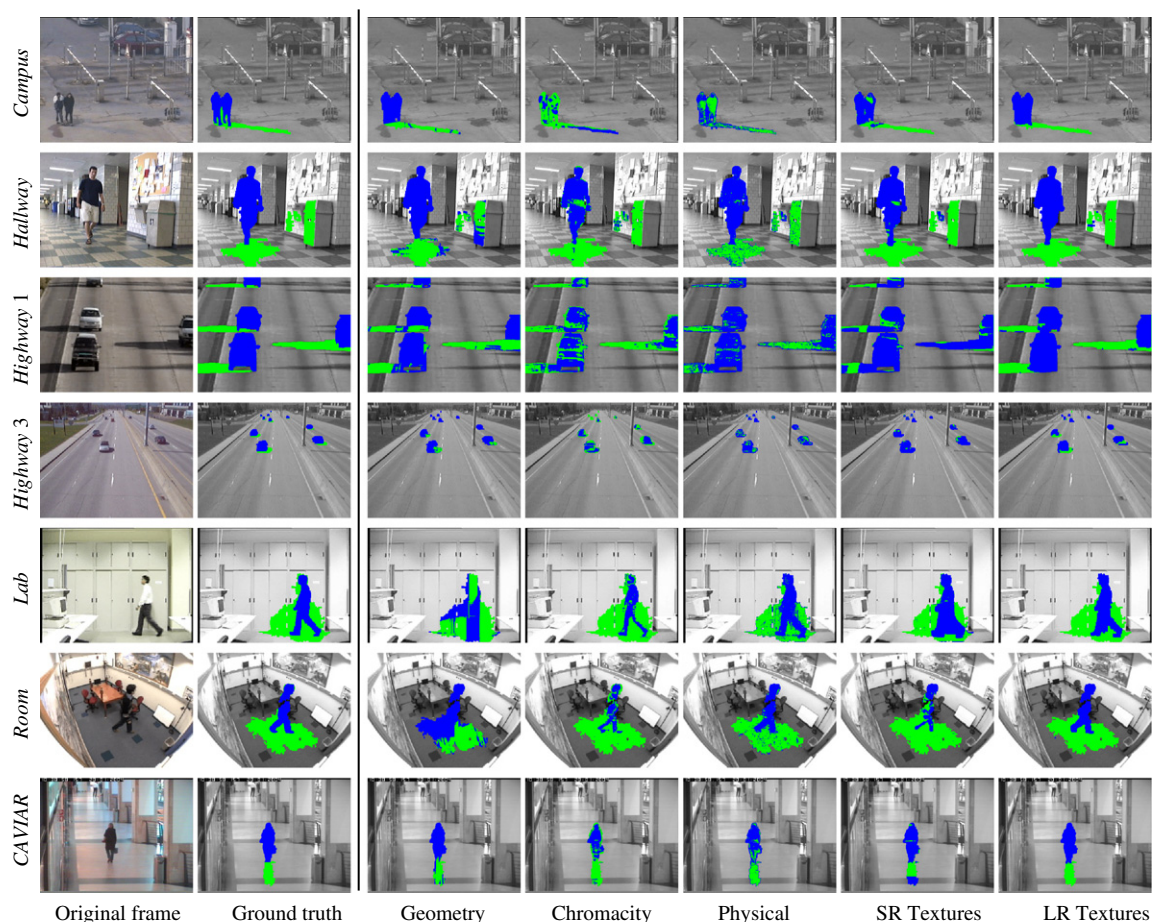
Fig. 5. Qualitative shadow detection results. The first column shows an example frame for each sequence. The second column show the expected results where object pixels are marked in blue and shadow pixels are marked in green. The remaining columns show the observed results for each method. The geometry-based method works well for long shadows with a well defined orientation (*Campus* example) but does not work well for spread shadows (*Lab* example). Both the chromacity-based and physical methods are affected by pixel-level noise, and fail when objects are darker and have similar colours to the background (*Campus* example), although the physical method adapts better to the different scenes. The small region (SR) texture-based method fails for pixels located in non-textured neighbourhoods (*Highway* 1 example). The large region (LR) texture-based method works reasonably well in all cases. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4

Summary of the comparative evaluation. The shadow detection methods are rated as *low*, *medium* or *high* according to seven criteria. For the first five criteria, *high* indicates that the method performed well in all cases. A method with low scene, object or shadow independence, only works well for a narrow set of scene, object or shadow types. Low penumbra detection means that a method is not appropriate for detecting shadow borders. Low robustness to noise means that shadow detection performance is significantly affected by scene noise. For the last two criteria (i.e. 6 and 7) lower means better. A high detection/discrimination trade-off means that tuning a method to increase one of the rates will significantly decrease the other. In terms of computational load, the methods are ranked by their average processing time per frame.

	Geometry	Chromacity	Physical	SR textures	LR textures
1. Scene independence	Medium	Medium	Medium	Medium	High
2. Object independence	Low	High	High	High	High
3. Shadow independence	Low	High	High	High	High
4. Penumbra detection	Medium	Low	Medium	High	High
5. Robustness to noise	Medium	Low	Medium	High	High
6. Detection/discrimination trade-off	Low	High	Medium	High	Low
7. Computational load	Low	Low	Low	High	Medium

texture features. The chromacity-based method is simple and fast, and as Prati et al. [31] concluded, its few assumptions lead it to work reasonably well in most scenes. However, it has a strong trade-off between shadow detection and discrimination. The physical method reduces the limitations of the chromacity-based method, provided there are sufficient examples to learn and adapt to the appearance of moving cast shadow in the background. However, as the chromacity-based method, it is sensitive to pixel-level noise and scenes with low saturated colours. The small region texture-based method is robust to various illumination conditions and is easy to implement, but it requires the background to be textured and needs a costly texture correlation operation per pixel. The large region texture-based method is not sensitive to pixel-level noise, and it is independent to the type of shadows, objects and scenes. It presents the best results at the cost of additional computational load.

4.2. Effect on tracking performance

Given that several methods perform relatively well (i.e. detection and discrimination rates above 75%), will pursuing higher rates translate into improved object detection and tracking? Although measuring shadow detection accuracy pixel-by-pixel is an objective measure, applications rarely have the need of explicitly detecting shadow pixels. Rather, shadow detection methods are typically used in tracking applications to clean the detection results and improve tracking performance. It is important to prove that increasing the detection rate and/or discrimination rate of shadow pixels will result in better tracking results. For this reason, we performed a second set of experiments to measure tracking performance after applying each method to remove the shadows in the foreground masks (i.e. setting the shadow pixels to zero).

Using various tracking algorithms is important since they may use the shadow-removed foreground masks differently. For instance, the masks can be used prior to tracking (e.g. for initialisation of particle filters [51]) or during tracking (e.g. for updating appearance models in blob matching [10,50]). We used five tracking algorithms implemented in the video surveillance module of OpenCV v2.0 [3]: blob matching (CC), mean-shift tracking (MS), mean-shift tracking with foreground feedback (MSFG), particle filtering (PF), and blob matching/particle filter hybrid (CCPF). The foreground masks are used while tracking in CC, prior to tracking in MS and PF, and both while and prior to tracking in MSFG and CCPF.

For the experiments, we used the tracking ground truth data available for the 50 sequences in the second set of the CAVIAR dataset. We performed 30 tracking evaluations by combining six shadow removal options (no shadow removal, geometry-based method, chromacity-based method, physical method, small and large region texture-based methods) with the five tracking algorithms. The tracking performance was measured with the two metrics proposed by Bernardin and Stiefelhagen [2], namely multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP):

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \qquad (17)$$

$$\text{MOTP} = \frac{\sum_{i,t}(d_t^i)}{\sum_t c_t} \qquad (18)$$

Briefly, MOTA is an inverse measure of the number of missed objects (m_t), false positives (fp_t) and mismatches (mme_t). The higher the MOTA, the better. MOTP measures the average pixel distance (d_t) between the ground-truth locations of objects and their locations according to a tracking algorithm. The lower the MOTP, the better. Ground truth objects and hypotheses are matched using the Hungarian algorithm [23].

The tracking results are presented in Fig. 6. Each bar represents the performance result of a particular tracking algorithm after removing the shadows detected by one of the shadow detection methods, averaged for the 50 test sequences. In all cases, the large region texture-based method results in better tracking performance, with considerable improvements over the next best method (small region texture-based).

Three other things can be observed in these results. First, in most cases, tracking performance is significantly improved with shadow removal regardless of the tracking algorithm. In some cases, the geometry-based method led to a decrease in tracking performance. Recall that the geometry method relies on the assumption that objects and shadows have different orientations, which is not met in the CAVIAR dataset. Second, the improvement in tracking performance by using a better shadow remover
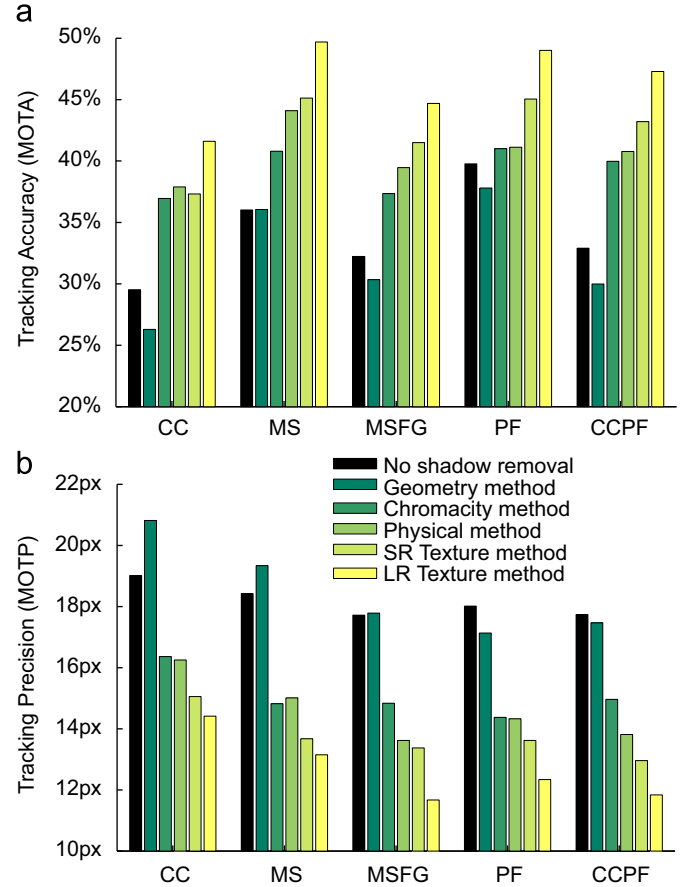

Fig. 6. Effect of shadow removal methods on: (a) multiple object tracking accuracy (MOTA), where taller bars indicate better accuracy; and (b) multiple object tracking precision (MOTP), where shorter bars indicate better precision. Results are grouped by tracking algorithm: blob matching (CC), two mean shift trackers (MS and MSFG), particle filter (PF) and hybrid tracking (CCPF).

(e.g. large region texture-based method instead of the chromacity-based method) is comparable to the improvement by using a better tracking algorithm (e.g. particle filter instead of blob matching). Last, improving shadow detection performance (as shown for the CAVIAR dataset in Fig. 3) leads to a proportional improvement in tracking performance. Therefore, regardless of the tracking algorithm, it is worth pursuing better shadow detection methods to obtain more accurate tracking results.

5. Main findings and future directions

In this paper we presented a review and a comparative evaluation of shadow detection methods published during the last decade. The survey follows the work of Prati et al. [31] in 2003 but with recent publications, a more comprehensive set of test sequences and more detailed experiments.

In the review the shadow detection methods are classified in a feature-based taxonomy. Methods that use mainly spectral features are classified as either chromacity-based or physical methods. Methods that use mainly spatial features are classified as either geometry-based or texture-based methods. For the comparative evaluation, we selected a prominent method from each group, except for the texture-based category, where we selected two: one that uses small regions and one that uses large regions for the texture correlation. We compared the shadow detection performance of these methods using a wide range of test sequences. Additionally, we observed the effect of low

saturation on shadow detection performance and evaluated the practical link between shadow detection and tracking performance.

The quantitative and qualitative comparative results can serve as a guide (for both practitioners and researchers) to select the best method for a specific purpose—all shadow detection approaches make different contributions and all have individual strengths and weaknesses. Out of the selected methods, the geometry-based technique has strict assumptions and is not generalisable to various environments, but it is a straightforward choice when the objects of interest are easy to model and their shadows have different orientation. The chromacity-based method is the fastest to implement and run, but it is sensitive to noise and less effective in low saturated scenes. The physical method improves upon the accuracy of the chromacity method by adapting to local shadow models, but fails when the spectral properties of the objects are similar to that of the background. The small-region texture-based method is especially robust for pixels whose neighbourhood is textured, but may take longer to implement and is the most computationally expensive. The large-region texture-based method produces the most accurate results, but has a significant computational load due to its multiple processing steps.

Traditionally, simple and fast shadow detection and removal methods have been favoured in computer vision applications such as tracking systems [9,16,19,44]. It is hence pertinent to note that when the shadow detection performance is relatively poor (e.g. $< 60\%$ for the geometry-based technique on the CAVIAR dataset), shadow removal can in fact lead to tracking performance which is worse than not using shadow removal. In contrast, more elaborate shadow detection algorithms lead to considerably better tracking performance, regardless of the tracking algorithm.

A logical future direction is to use extra features in the existing methods, as all the currently used features largely provide independent contributions. For instance, geometry and temporal features can be added to the physical or texture-based approaches. Alternatively, physical or texture features can be used to pre-select candidate shadow pixels and feed them to geometry-based methods for shadow remodelling. Lastly, even if there are considerable differences in the computational load across the various shadow detection methods, all of them can be optimised to meet real-time requirements. In particular, all the compared approaches are parallelisable at either the pixel or region level.

Acknowledgements

References

[1] F. Bergeaud, S. Mallat, Matching pursuit of images, International Conference on Image Processing, vol. 1, 1995, pp. 53–56.

[2] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics, EURASIP Journal on Image Video Processing, 2008, doi:10.1155/2008/246309.

[3] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008.

[4] A. Cavallaro, E. Salvador, T. Ebrahimi, Shadow-aware object-based video processing, IEE Proceedings on Vision, Image and Signal Processing 152 (4) (2005) 398–406.

[5] C.-C. Chen, J. Aggarwal, Human shadow removal with unknown light source, in: International Conference on Pattern Recognition, 2010, pp. 2407–2410.

[6] C.-T. Chen, C.-Y. Su, W.-C. Kao, An enhanced segmentation on vision-based shadow removal for vehicle detection, in: International Conference on Green Circuits and Systems, 2010, pp. 679–682.

[7] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, Detecting moving objects, ghosts, and shadows in video streams, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1337–1342.

[8] L.Z. Fang, W.Y. Qiong, Y.Z. Sheng, A method to segment moving vehicle cast shadow based on wavelet transform, Pattern Recognition Letters 29 (16) (2008) 2182–2188.

[9] P. Forczmanski, M. Seweryn, Surveillance video stream analysis using adaptive background model and object recognition, Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 6374, 2010, pp. 114–121.

[10] L.M. Fuentes, S.A. Velastin, People tracking in surveillance applications, Image and Vision Computing 24 (11) (2006) 1165–1171.

[11] T. Horprasert, D. Harwood, L. Davis, A statistical approach for real-time robust background subtraction and shadow detection, in: IEEE ICCV'99 Frame-Rate Workshop, 1999.

[12] J.-W. Hsieh, W.-F. Hu, C.-J. Chang, Y.-S. Chen, Shadow elimination for effective moving object detection by Gaussian shadow modeling, Image and Vision Computing 21 (6) (2003) 505–516.

[13] J.-B. Huang, C.-S. Chen, Moving cast shadow detection using physics-based features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2310–2317.

[14] O. Javed, M. Shah, Tracking and object classification for automated surveillance, Proceedings of European Conference on Computer Vision, vol. 4, 2002, pp. 343–357.

[15] A. Joshi, N. Papanikolopoulos, Learning to detect moving shadows in dynamic environments, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (11) (2008) 2055–2063.

[16] B. Lei, L.-Q. Xu, Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management, Pattern Recognition Letters 27 (15) (2006) 1816–1825.

[17] A. Leone, C. Distante, Shadow detection for moving objects based on texture analysis, Pattern Recognition 40 (4) (2007) 1222–1233.

[18] Z. Liu, K. Huang, T. Tan, L. Wang, Cast shadow removal combining local and global features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[19] L. Maddalena, A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, IEEE Transactions on Image Processing 17 (7) (2008) 1168–1177.

[20] N. Martel-Brisson, A. Zaccarin, Learning and removing cast shadows through a multidistribution approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (7) (2007) 1133–1146.

[21] N. Martel-Brisson, A. Zaccarin, Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[22] B. Mitra, R. Young, C. Chatwin, On shadow elimination after moving region segmentation based on different threshold selection strategies, Optics and Lasers in Engineering 45 (11) (2007) 1088–1093.

[23] J. Munkres, Algorithms for the assignment and transportation problems, Journal of the Society for Industrial and Applied Mathematics 5 (1) (1957) 32–38.

[24] S. Nadimi, B. Bhanu, Physical models for moving shadow and object detection in video, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (8) (2004) 1079–1087.

[25] K. Nakagami, T. Nishitani, The study on shadow removal on transform domain GMM foreground segmentation, in: International Symposium on Communications and Information Technologies, 2010, pp. 867–872.

[26] A. Nghiem, F. Bremond, M. Thonnat, Shadow removal in indoor scenes, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, 2008, pp. 291–298.

[27] H. Nicolas, J.-M. Pinel, Joint moving cast shadows segmentation and light source detection in video sequences, Signal Processing: Image Communication 21 (1) (2006) 22–43.

[28] J. Panicker, M. Wilscy, Detection of moving cast shadows using edge information, International Conference on Computer and Automation Engineering, vol. 5, 2010, pp. 817–821.

[29] L. Pei, R. Wang, Moving cast shadow detection based on PCA, International Conference on Natural Computation, vol. 2, 2009, pp. 581–584.

[30] F. Porikli, J. Thornton, Shadow flow: a recursive method to learn moving cast shadows, Tenth IEEE International Conference on Computer Vision, vol. 1, 2005, pp. 891–898.

[31] A. Prati, I. Mikic, M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 918–923.

[32] R. Qin, S. Liao, Z. Lei, S. Li, Moving cast shadow removal based on local descriptors, in: International Conference on Pattern Recognition, 2010, pp. 1377–1380.

[33] T. Randen, J. Husoy, Filtering for texture classification: a comparative study, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4) (1999) 291–310.

[34] V. Reddy, C. Sanderson, B.C. Lovell, A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts, EURASIP Journal on Image and Video Processing, in press, doi:10.1155/2011/164956.

[35] V. Reddy, C. Sanderson, A. Sanin, B.C. Lovell, Adaptive patch-based background modelling for improved foreground object segmentation and tracking, in: International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2010, pp. 172–179.

[36] E. Salvador, A. Cavallaro, T. Ebrahimi, Cast shadow segmentation using invariant color features, Computer Vision and Image Understanding 95 (2) (2004) 238–259.

[37] A. Sanin, C. Sanderson, B. Lovell, Improved shadow removal for robust person tracking in surveillance scenarios, in: International Conference on Pattern Recognition, 2010, pp. 141–144.

[38] Y. Shan, F. Yang, R. Wang, Color space selection for moving shadow elimination, in: International Conference on Image and Graphics, 2007, pp. 496–501.

[39] M. Shoaib, R. Dragon, J. Ostermann, Shadow detection for moving humans using gradient-based background subtraction, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 773–776.

[40] C. Stauffer, W. Grimson, Learning patterns of activity using real-time tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 747–757.

[41] B. Sun, S. Li, Moving cast shadow detection of vehicle using combined color models, in: Chinese Conference on Pattern Recognition, 2010, pp. 1–5.

[42] Y.-L. Tian, M. Lu, A. Hampapur, Robust and efficient foreground analysis for real-time video surveillance, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 1182–1187.

[43] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, IEEE International Conference on Computer Vision, vol. 1, 1999, pp. 255–261.

[44] H. Wang, D. Suter, A consensus-based method for tracking: modelling background scenario and foreground appearance, Pattern Recognition 40 (3) (2007) 1091–1105.

[45] Y. Wang, K.-F. Loe, J.-K. Wu, A dynamic conditional random field model for foreground and shadow segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2) (2006) 279–289.

[46] D. Xu, X. Li, Z. Liu, Y. Yuan, Cast shadow detection in video segmentation, Pattern Recognition Letters 26 (1) (2005) 91–99.

[47] A. Yoneyama, C. Yeh, C. Kuo, Moving cast shadow elimination for robust vehicle extraction based on 2d joint vehicle/shadow models, in: IEEE Conference on Advanced Video and Signal Based Surveillance, 2003, pp. 229–236.

[48] W. Zhang, X.Z. Fang, Y. Xu, Detection of moving cast shadows using image orthogonal transform, International Conference on Pattern Recognition, vol. 1, 2006, pp. 626–629.

[49] W. Zhang, X.Z. Fang, X. Yang, Q. Wu, Moving cast shadows detection using ratio edge, IEEE Transactions on Multimedia 9 (6) (2007) 1202–1214.

[50] Q. Zhou, J. Aggarwal, Object tracking in an outdoor environment using fusion of features and cameras, Image and Vision Computing 24 (11) (2006) 1244–1255.

[51] S.K. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, IEEE Transactions on Image Processing 13 (11) (2004) 1491–1506.

Andres Sanin is currently a Ph.D. student at NICTA and the University of Queensland, Australia. He has a Bachelor in Science (2004) and an M.Sc. (2005) in Biology, from the Universidad de Los Andes (Colombia). He received an M.Sc. in Computer Science (2008) from the University of Queensland.

Conrad Sanderson is a senior research scientist at NICTA, Australia. He received the Ph.D. degree in 2003 from Griffith University, Queensland, Australia. He has previously worked on robust speech recognition at the Advanced Telecommunication Research Laboratories (Japan), audio-visual biometrics and non-frontal face recognition at the IDIAP Research Institute (Switzerland), ship classification in infra-red images at the Centre for Sensor Signal and Information Processing (Australia), as well as natural language processing and bioinformatics at NICTA. He has served as a reviewer for a number of international conferences and scientific periodicals (such as the Proceedings of the IEEE journal). His current research interests are in the areas of pattern recognition and computer vision, with applications such as automated video surveillance.

Brian Lovell is a Research Leader at NICTA, Australia. From 1983 to 1986, he worked in Saudi Arabia and Egypt with Schlumberger SA. He then returned to the University of Queensland to pursue his Ph.D. While studying for his Ph.D., he was offered a lecturing position in 1989. In 2000, Professor Lovell was visiting professor at the MIT Media Lab, Michigan State University, and the University of Alberta. In 2004 he was appointed as Director of Engineering in the School of ITEE at the University of Queensland. He is a Past-President of the International Association for Pattern Recognition (IAPR).