Shadow Detection in Dynamic Scenes Using Dense Stereo Information and an Outdoor Illumination Model

Madsen, Claus B.; Moeslund, Thomas B.; Pal, Amit; Balasubramanian, Shankar

Published in:
Lecture Notes in Computer Science

DOI (link to publication from Publisher):
10.1007/978-3-642-03778-8_9

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Shadow Detection in Dynamic Scenes Using Dense Stereo Information and an Outdoor Illumination Model

Claus B. Madsen[1], Thomas B. Moeslund[1],
Amit Pal[2], and Shankkar Balasubramanian[2]

[1] Computer Vision and Media Technology Lab
Aalborg University, Aalborg, Denmark
cbm@cvmt.aau.dk
www.cvmt.aau.dk/~cbm
[2] Department of Electronics and Communication Engineering
Indian Institute of Technology Guwahati, Assam, India

Abstract. We present a system for detecting shadows in dynamic outdoor scenes. The technique is based on fusing background subtraction operations performed on both color and disparity data, respectively. A simple geometrical analysis results in an ability to classify pixels into foreground, shadow candidate, and background. The shadow candidates are further refined by analyzing displacements in log chromaticity space to find the shadow hue shift with the strongest data support and ruling out other displacements. This makes the shadow detection robust towards false positives from rain, for example. The techniques employed allow for 3Hz operation on commodity hardware using a commercially available dense stereo camera solution.

Keywords: Shadows, stereo, illumination, chromaticity, color.

1 Introduction

Shadows are an inherent part of images. Especially in outdoor vision applications shadows can be a source of grave problems for the processing and analysis of video data. For humans, though, shadows represent a significant cue to understanding the geometry of a scene, and to understanding the illumination conditions, which in turn helps processing the visual data. In this paper we present an approach to accurately identifying shadow regions in outdoor, daylight video data in near real-time (presently around 3 Hz, with potential for significant improvement). The main contributions of this work lie in utilizing a combination of color and dense depth data from a stereo rig for an initial, rough shadow detection, combined with a model-based chromaticity analysis for the final, precise shadow pixel identification.

Work on detection of shadows can be divided into techniques for detecting dynamic shadows (cast by moving objects) and static shadows (cast by static

R. Koch and A. Kolb (Eds.): Dyn3D 2009, LNCS 5742, pp. 110–125, 2009.

scene objects such as buildings). Static shadow detection work is based on single images and is typically more sophisticated in the use of physically based illumination/reflection. It is also typically computationally heavy techniques. Dynamic shadow detection work is naturally based on image sequences and utilizes somewhat simplistic illumination models which at best correspond poorly to real conditions, especially for outdoor scenery. These techniques all employ background subtraction based on a trained background model, a concept which is problematic for very long outdoor image sequences due to drastic illumination changes, precipitation, foliage changes, etc.

The ideas proposed in this paper can be operated in two modes: one based on background subtraction with a trained model, and one based on image differencing with no training. We shall focus on the former mode in the presentation and return to the latter mode in section 5.

Detecting static shadows is in principle difficult as it is theoretically impossible to definitively determine whether a region in an image is a bright surface in shadow or a dark surface in direct light. Regardless, promising results on single image shadow detection and removal has been presented over the recent years. A single image shadow removal technique is presented in [3] but requires a very high quality, chromatically calibrated camera, and does not handle soft shadows (penumbra regions). The technique presented in [11] distinguishes between cast shadows (on a plane) and self-shadowing, but is tested on somewhat simple scenarios, and it too does not handle soft shadows. Not being able to handle soft shadows is a severe problem for outdoor scenes in partly overcast conditions.

Single image shadow detection in scenes with soft shadows is addressed in [9,8], demonstrating successful shadow detection (and removal) on single images of non-trivial scenes. Unfortunately, the approach requires manual identification of training areas in the image (areas where the same material is visible in shadow as well as in direct sunlight).

So, the state-of-the-art in single image shadow work is that it does not really handle soft shadows, or requires manual training. Our method handles soft shadows very well, and we demonstrate the even quite subtle shadows in allmost overcast conditions can be detected. Furthermore we demonstrate that our method, in the no-background-model mode mentioned above, can generate the necessary input for the technique described in [8] thus eliminating the need for manual boot-strapping.

Dynamic shadow detection based on image sequences has recently received much attention especially in the surveillance literature. Here there is a need for detecting and tracking objects in a scene and one of the key problems has turned out to be false positives due to shadows [10,7]. Many of the approaches suggested for shadow segmentation are based on the idea that a pixel in shadow has the same color as when not in shadow, but at a lower intensity [10,6,4]. Such an illumination model is very simplistic (assumes all light sources to be white). This is a severe assumption, which is totally violated in outdoor scenes, and tests presented in in these works are also either on indoor scenes or on outdoor scenes in overcast conditions, where the assumption roughly holds.

In non-overcast outdoor scenes regions in shadow (blocked from direct sunlight) exhibit a blue hue shift due to the differences in the spectrum of the light coming from the sky, and the light coming from the sun. This fact is incorporated into the work presented in [5], which utilizes the blue shift for outdoor scenes for separating foreground from shadow using using background subtraction.

All previous work on dynamic shadow detection utilized a trained background model in some form. By combining depth information with color information the techniques presented in this paper makes it much simpler to distinguish between foreground and shadow, allowing us to operate with a much less well-trained background model (and thus more robust towards illumination changes, precipitation, etc.). And, as mentioned, we can even detect a substantial part of the shadows without any kind of background model.

The inspiration for the work presented in this paper came from two sides: 1) in dynamic outdoor scenes (time sequence video of scenes with moving objects) some shadows will move, which provides a unique opportunity to study the same pixel under both shadow *and* non-shadow conditions (making it possible to estimate the shadow hue shift automatically), and 2) when combining color information with dense depth information shadow candidate regions can be identified from the observation that a cast shadow represents a change in color channel values but not in depth (for shadows that fall on static surfaces in the scene).

The paper is organized as follows. In section 2 we present the setup used and give a brief introduction to the methods presented later in the paper. Section 4 describes how the color and depth information is combined to detect shadow candidate regions, and how chromaticity analysis is used to finally identify shadow pixels. We then present and discuss some results in section 5, followed by conclusions.

2 Setup and Overview of Approach

The setup for this work is centered around a commercial stereo rig from Point Grey Research INC., [12], see figure 1. The Bumblebee XB3 real-time dense stereo camera delivers rectified stereo image pairs via FireWire at up to 16 frames per second, depending on resolution. In this work we operate with a 640x480 resolution, resulting in a stereo frame rate of approx. 10 Hz. Using the accompanying SDK for the stereo camera disparities can be computed at a per pixel level using correlation techniques. On an Intel Core Duo 2 2.3 GHz machine running Windows XP SP2, equipped with 2 GByte RAM, the disparities are computed in around 50 milliseconds, so the limiting factor is the 10 Hz transfer of rectified stereo images from the camera. All RGB and disparity images shown in this paper represent the view of the right camera of the stereo rig. The disparity values employed in this work are in 16 bit resolution (subpixel disparity information) as the Bumblebee XB3 SDK offers this functionality.

It is assumed that the camera is static relative to the scene. It is also assumed that the scene contains a substantial amount of static surfaces (objects that do

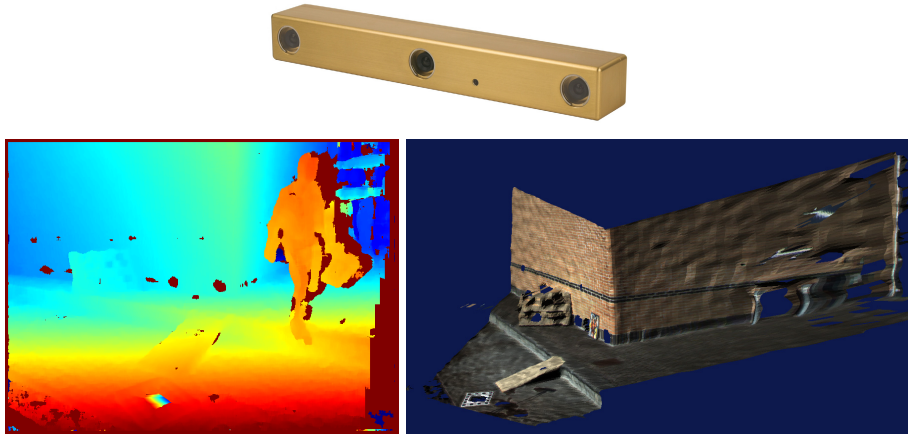

Fig. 1. Top: The commercially available Bumblebee XB3 stereo rig from Point Grey Research, Inc. at approximately 3000 USD. Bottom left: Pseudo-colored disparity image of an outdoor scene. The dark red patches represent undetected disparities due to pixels being over-exposed or under-exposed, lack of texture, or pixels representing scene points that are not visible in both cameras (occlusion). Bottom right: 3D mesh constructed from disparity image and textured with the color information from the RGB image. This scene is a frame with no person.

not move), and that dynamic objects are also present in the scene. In general the scene will contain shadows cast by static objects, as well as shadows cast by dynamic objects. The techniques presented in this paper are able to detect shadows cast by the dynamic objects, although section 5 demonstrates how the generated results can be used to detect the static shadows as well.

The approach presented here has 4 main steps. First we employ a background subtraction method on the color information (the RGB image). Next we perform a similar step on the dense per pixel disparity information, where the background image in this case represents an acquired depth model of the scene (disparities are proportional to metric depth, so there is no reason to spend computational resources on scaling disparities to metric depth unless metric information is required for some other processing step unrelated to the core shadow detection). The third step is to combine the results of the two background subtractions which allows us to interpret the nature of each pixel: foreground, background, shadow. This is illustrated in figure 2. The fourth and final step is to evaluate some chromaticity (normalized color channel information) characteristics of the segmented shadow pixel population to eliminate those pixels that do not conform to an illumination model which predicts the overall behaviour of regions as they transition from being exposed to direct sunlight to being in shadow.

Subsequently, the different steps are elaborated in further detail. While we have an operational C++ real-time implementation of the described approach some of the results shown in this paper are generated by a similar Matlab

Fig. 2. Processing results from frame 137 out of a recorded sequence of 200 frames. Left: The current frame where the person has entered the scene and has splashed water from the bucket onto several surfaces in the scene. Middle: result from background subtraction, segmented into foreground object (grey) and shadow candidate pixels (white), which include all the detected water splashes since the water has caused the surfaces to change appearance. Left: pixels identified as being in shadow after chromaticity analysis. Note that the shadow cast by the camera tripod is a static shadow, and as such is not detected by the presented approach.

implementation from which we have easier access to specific intermediate results and can generate illustrative visuals for the paper.

3 Theoretical Framework and Fundamental Assumptions

The work presented in this paper rests on a fundamental radiometric model of the radiance of points in a scene illuminated by a combination of sunlight and sky light. This model, together with some assumptions that are made, are described in this section.

It is assumed that the images represent an outdoor scene subjected to daylight illumination. It is also necessary to assume that the materials represented in the scene are predominantly diffuse (exhibit Lambertian reflectance distribution characteristics). We do *not* require the albedos (diffuse reflectances) of the surfaces in the scene to be constant over time. In fact we clearly demonstrate that our approach can avoid errornously detecting/hallucinating shadows in areas where the surface has simply changed appearance from suddenly becoming wet (as they would in rainy conditions).

Concerning the images it is furthermore assumed that they are properly exposed, i.e., important areas in the image are allowed to be neither severely over-exposed (color channel values near 255) nor severely under-exposed (values near 0). We will also assume that it is a fair assumption that the camera is linear, in the sense that there is a linear relationship between the radiance of a surface and the pixel value assigned to the image point of that surface.

For a linear camera the pixel value in some color channel is proportional to the reflected radiance of the surface being imaged, and radiance is measured in $W/(m^2 \cdot sr)$. In a setting as described above it is possible to formulate the value, P_r, of a pixel as follows, using subscript r to indicate elements particular to the red channel (green and blue channel being similar):

$$P_r = c_r \cdot \rho_r \cdot E_r \cdot \frac{1}{\pi} \quad (1)$$

where ρ_r is the diffuse albedo of the surface point being imaged (ratio of outgoing radiosity to incoming irradiance), and E_r is the incoming irradiance in the red channel. Thus, $\rho_r \cdot E_r$ is the reflected radiosity. Dividing this by π $[sr]$ yields the reflected radiance of the surface (since the radiosity from a diffuse surface is π times the radiance of the surface). Finally, c_r is the (typically unknown) scaling factor translating the measured radiance into pixel value (0 to 255 range for an 8 bit camera) for a linear camera. This scaling value depends on the aperture of the lens, the shutter speed, the gain, the white-balancing etc. of the camera.

In the kind of outdoor daylight setting we are addressing in this paper the total incoming irradiance at a point is a sum of two contributions, $E_r = E_r^{\text{sun}} + E_r^{\text{sky}}$, again using subscript r for red color channel as example. The amount of irradiance received from the sun, E_r^{sun}, depends on several factors: the radiance of the sun, how large a fraction of the sun's disk is visible from the point in interest (if the sun's disk is completely occluded the point is in full shadow, also called umbra), and on the angle between the surface normal at the point and the direction vector to the sun from the point. If the sun's disk is only partially occluded the point is in the penumbra (soft shadow).

We shall return to this formulation in section 4.3, where we use it to justify our approach to letting shadow candidate pixels vote for a shadow hue shift which can be used to dismiss pixels that are in fact not in shadow in a particular frame.

4 Methods

As described in section 2 we initially segment each frame into background, foreground, and shadow. This is performed by combining the results from a background subtraction process on both the color image information and on the disparity image information. The two background subtraction processes are described below.

4.1 RGB Background Subtraction

We apply the Codebook method [6] since it has been shown to outperform other background subtraction methods [2]. The method contains three steps: modeling the background, pixel classification and model updating.

Each pixel is modeled as a group of codewords which constitutes the codebook for this particular pixel. Each codeword is a cylindrical region in RGB-space and for each new frame each pixel is compared to its codebook. If the current pixel value belongs to one of the codewords it is classified as background, otherwise foreground.

The codebooks are built during training and updated at run-time. The training phase is similar to the pixel classification except that a foreground pixel results in the construction of a new codeword and a background pixel is used to

Fig. 3. Left: Pixels (frame 137) determined by the Codebook method as being different from the trained color background model. Right: Pixels (also frame 137) determined as being different from the disparity background model.

modify the codeword it belongs to using a standard temporal weighting scheme. The codebooks generated in this way during training will typically fall into three categories:

Static codebook. For example a pixel representing a road with no shadows or occlusions. Typically only one codeword is used.

Quasi-static codebook. For example a pixel containing the sky, but sometimes occluded by vegetation due to wind gusts. During training typically two codewords will be constructed for this codebook, one for the sky and one for the vegetation.

Noisy codebook. One of the above combined with noise in the form of a pedestrian, car etc. passing by the pixel or noise due to incorrect segmentation. The result will be an often high number of codewords for this codebook.

To handle the noisy codebooks a temporal filter is applied. It is based on the Max Negative Run-Length (MNRL), which is the longest time interval in which a codeword has not been activated. The filter effectively removes codewords with little support during the training phase, such as passing pedestrians.

Normally it is difficult to tune the sensitivity of the Codebook method (and other background subtraction methods for that matter) particularly due to problems with shadows. In this work, however, this is less of a problem since oversegmentation is actually a desired effect. We therefore tune the method to detect even small changes, and we do not train with shadow regions. This effectively results in a segmentation of the dynamic foreground object including its shadow, see figure 3 for an example.

4.2 Disparity Background Subtraction

We also apply the Codebook method for depth-based background subtraction. Here we only apply one codeword per pixel and its value is not the actual depth

Fig. 4. Classification of pixels into foreground (grey), background (black) and shadow candidates (white)

value, but rather the disparity. The disparity background model is learned as the Median of a number of training images.

Normally a disparity map contains undefined pixels, due to e.g., noise. We therefore smooth the disparity map to obtain more consistent data, and the pixels classified as different from the background also need some clean-up using standard morphological operations. We have employed erode followed by dilate with a radius 5 pixels disk structuring element.

4.3 Shadow Classification

For each pixel in an input image we now have two TRUE/FALSE values with respect to whether the pixel is different from the RGB background model and whether it is different from the disparity background model. From this we can infer the pixel's type (foreground, background, shadow) as shown in table 1. In figure 4 a labeling based on table 1 is shown. The rational behind this table can be formulated as follows: if the color of a pixel has changed, but there is no change in disparity, then the pixel has gone from direct light to shadow. If there is a change in disparity but not in color it can be argued whether to classify it is foreground or background. We have chosen background, since disparity data is less robust than color data, at any rate if there is not change in color it cannot represent a shadow.

Table 1. Classification scheme based on results of background subtraction on color and depth data. It should be noted that only color changes that represent *decreased* intensity are valid shadow candidates.

		Change in RGB?	
		Yes	No
Change in depth?	Yes	Foreground	Background
	No	Shadow	Background

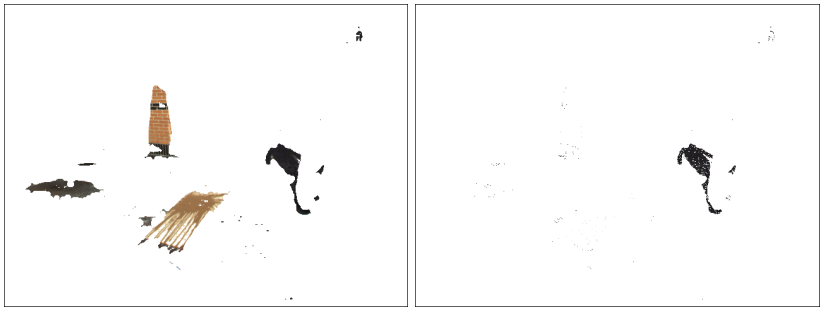

Fig. 5. Left: population of shadow pixel candidates after logical classification based on background subtraction on RGB and disparity data. Right: population of shadow pixels after analysis of the permissible shift in the log chromaticity plane.

After this logical classification we are left with a population of dynamic (cast by a dynamic object) shadow candidate pixels for each frame. These pixels are shadow *candidates* only. Predominantly two things can cause pixels to falsely be labeled as shadow pixels: 1) the albedo of the surface changed (for example due to the surface becoming wet), or 2) imperfections of the disparity data causes the disparity background subtraction to produce sub-optimal results (see for example figure 3). All pixels labeled as shadow candidates are shown with their RGB values in figure 5.

We address the problem of rejecting the non-shadow pixels by returning to the formulation of the value of a pixel as given in section 3. In log chromaticity space the pixel values become:

$$\begin{aligned} r &= \log(P_r/P_g) \\ &= \log(P_r) - \log(P_g) \\ &= \log(c_r) - \log(c_g) + \log(\rho_r) - \log(\rho_g) + \log(E_r) - \log(E_g) \qquad (2) \\ b &= \log(P_b/P_g) \\ &= \log(c_b) - \log(c_g) + \log(\rho_b) - \log(\rho_g) + \log(E_b) - \log(E_g) \qquad (3) \end{aligned}$$

The next observation is that the background image depicts the scene free of dynamic shadows. Thus, if for a given frame a pixel has been classified as a shadow candidate by applying the rules in table 1, we have the same pixel in two different versions: 1) a version in direct sun light from the background image, where $E = E^{\text{sky}} + E^{\text{sun}}$, and 2) a version in shadow from the current frame, where $E = E^{\text{sky}}$ (with appropriate indexes for respective color channels). If we subtract the chromaticity values of these two versions for a given pixel:

$$r^{\text{sky}} - r^{\text{sun + sky}} = \log\left(\frac{E_r^{\text{sky}}}{E_r^{\text{sky + sun}}}\right) - \log\left(\frac{E_g^{\text{sky}}}{E_g^{\text{sky + sun}}}\right) \qquad (4)$$

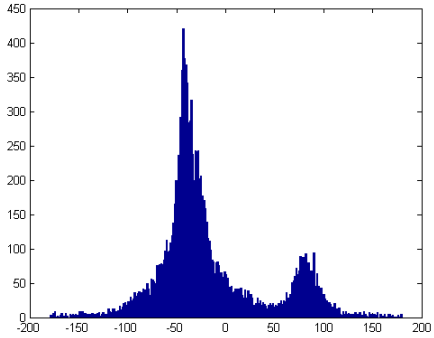

Fig. 6. Histogram over chromaticity displacement vector orientations, measured in the range from -180 degrees to +180 degrees. In this case the histogram has two peaks, one at around -45 degrees, corresponding to the darkening of the wooden plate as a result of a water splash, and one at around 90 degrees, corresponding to the actual blue shift of the shadow.

$$b^{\text{sky}} - b^{\text{sun + sky}} = \log\left(\frac{E_b^{\text{sky}}}{E_b^{\text{sky + sun}}}\right) - \log\left(\frac{E_g^{\text{sky}}}{E_g^{\text{sky + sun}}}\right) \tag{5}$$

Looking at eqs. 4 and 5 we see that by subtracting the log chromaticity coordinates of the two different versions of a pixel we get a 2D vector in log chromaticity space which is independent of camera properties (the $c_{r/g/b}$ scaling constants), and independent of the material properties (the $\rho_{r/g/b}$ albedos). The only things that influence these log chromaticity displacement/hue shift vectors are the irradiances. Furthermore the depth of the shadow (determined by the amount of sun disk occlusion and the angle between the surface normal and the direction vector to the sun) only influences the length of this displacement vector, not the direction.

With these observations we compute the orientations of all these displacement vectors (one for each of the shadow pixel candidates) and form a histogram of the orientations in the range from -180 to +180 degrees, see figure 6. The number of bins in the histogram is set to the number of shadow candidate pixels divided by 50 to ensure a reasonable number of candidates in each orientation bin. The minimum number of bins is 10, though, to handle the case of very few detected shadow candidates.

Since the spectrum of light from the sky is dominated by wavelengths in the blue channel pixels that go from direct sun light to shadow conditions will undergo a blue shift, which in terms of the histogram in figure 6 corresponds to displacement orientation near +90 degrees (in rb chromaticity space blue chromaticity is upwards). We therefore search the histogram to find a local maximum near 90 degrees. The chosen peak corresponds to the chromaticity shift pixels undergo when they transition from shadow to direct light. All pixels whose displacement vectors are not close to (in our system within 20 degrees of) this

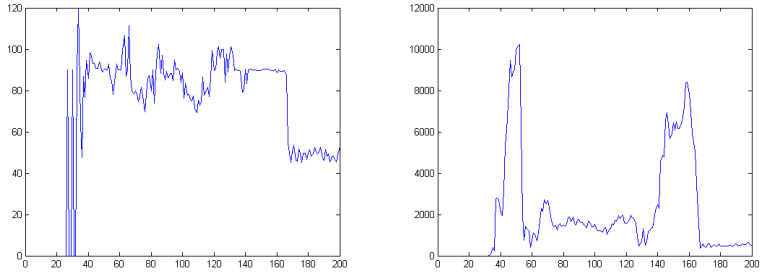

Fig. 7. Left: The detected log chromaticity blue shift direction (in degrees) as a function of frame number. Right: The number of verified shadow pixels in the scene as a function of frame number. When there are too few shadow pixels (less than around 500) the direction cannot be detected robustly.

"most voted for direction" are classified as not shadow pixels and are removed from the shadow pixel population. The remaining pixels all exhibit the same behaviour in log chromaticity space and are thus all consistent with the illumination model. In figure 5 it can be seen how this chromaticity analysis removes wrongly detected shadow pixels, especially those corresponding to all the water splashes (which have changed the albedo of the surfaces).

5 Results and Discussion

This section will address a set of relevant issues in relation to the presented techniques. We take a closer look at the estimated blue shift direction over a time sequence, we address overlapping shadows, then discuss long time sequences and demonstrate how this work can be operated in a mode with no background model, and finally we discuss some of the assumptions made in this work.

5.1 Blue Shift Direction

The proposed automatic approach to finding the blue shift direction is remarkably robust. Figure 7 plots the chosen direction for the 200 frame sequence used in the above description. The same figure also shows the development in the number of verified shadow pixels per frame through the sequence, and it is essentially seen that the blue shift direction is found robustly whenever there is a sufficient number of shadow pixels available in the scene.

5.2 No Background Model Mode

Our approach as described this far is based on background subtraction in both color and disparity data. As a result shadows that are static, i.e., part of the background image model, do not get detected. Another drawback of using background subtraction is that for very long image sequences (days, weeks, months

Fig. 8. In partly overcast, windy conditions illumination changes can be drastic. These two frames are 15 frames apart, corresponding to a time difference of 1.5 second. On average the latter image is 50% brighter than the former.

...) it can be difficult to maintain the background model due to highly varying illumination, precipitation, seasonal changes, etc. Figure 8 shows how drastically the illumination can change from one to second to the next, making classical background subtraction very difficult, if not impossible.

To address this problem we demonstrate that image differencing can be employed instead of the background subtraction step. The sequences used in this paper are recorded with 10 frames per second. If we perform image differencing on color and disparity images by subtracting frame T (current frame) and frame $T - \Delta T$ (some frames old), and perform everything else similar to what has been described in the paper, we can detect dynamic shadows with no training of background models at all. Δ can be adjusted to find a compromise between being very robust towards rapidly changing illumination (small ΔT) and detecting all of the shadow area (large ΔT). If ΔT is small compared to the movements of shadows in the scene the shadows in the two frames will overlap, and only part of the shadow will be detected (the part which is in shadow in frame T but not in frame $T - \Delta T$). In this paper we have used a ΔT of 0.5 second, i.e., we perform image differencing with a 5 frame delay. Figure 9 shows some detection results from a sequence acquired under highly varying illumination conditions.

5.3 Detecting All Shadows

By definition our approach only detects the dynamic shadows, regardless of using the background model or the image differencing mode. To address this problem the technique presented here can be combined with an implementation of the technique described in [8]. That method, as described in section 1, requires manual initialization (ratios of sky to sun-plus-sky irradiances for each color channel).

A by-product of the shadow detection technique described here is that it can provide those ratios. These ratios are straight forward to compute, as they are just the per color channel averages of the ratios of detected dynamic shadow

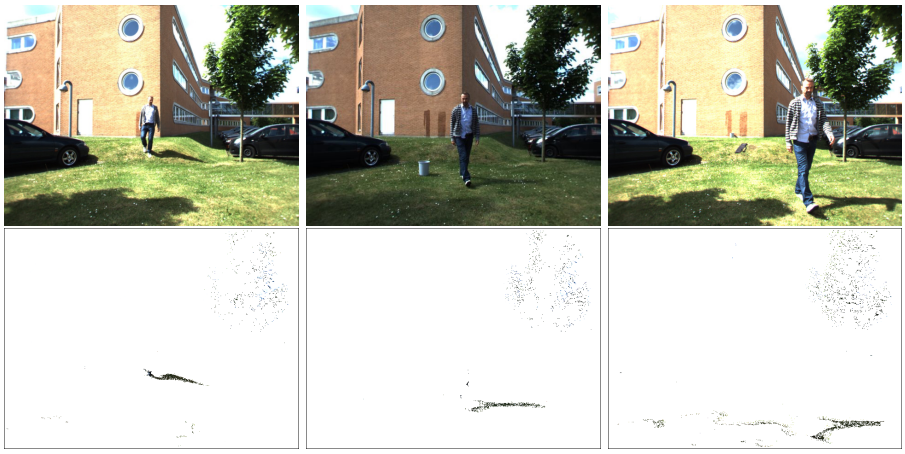

Fig. 9. Dynamic shadow detection based on image differencing (frames 180, 520, and 1741).

pixel values to their non-shadow pixel values. In the case of using the image differencing mode: if image differencing between frame T and frame $T - \Delta T$ results in a pixel being classified as dynamic shadow, then compute the per color channel ratio of pixel value in frame T to the pixel value in frame $T - \Delta T$. For a diffuse surface this ratio equals the sky to sun plus sky irradiance ratio, see section 3. The average of these ratios for all dynamic shadow pixels in a given frame provides the initialization information for the graph cuts technique from [8].

Figure 10 shows the results from using the dynamically detected shadows to boot-strap the graph cuts based shadow segmentation and removal technique, which is capable of handling soft shadows.

5.4 Assumptions Revisited

As described in section 3 this work rests on a number of assumption that are worth discussing. The assumption of the camera being static makes it possible to employ background models or to use simple image differencing as shown above. It would be possible to extend this work to a camera placed on a pan-tilt unit. Omni-directional depth background models are employed in e.g., [1]. If the mounted on a pan-tilt unit a spherical representation of the color and depth background model could be composed by scanning in all directions. If using the image differencing mode optical flow techniques could be employed to compute the overlap between the current frame and the delayed frame used for subtraction. This way the dynamic shadows in the overlap region could be detected and the information from the shadow pixels could then be used for detecting all shadows in the current frame as described in section 5.3.

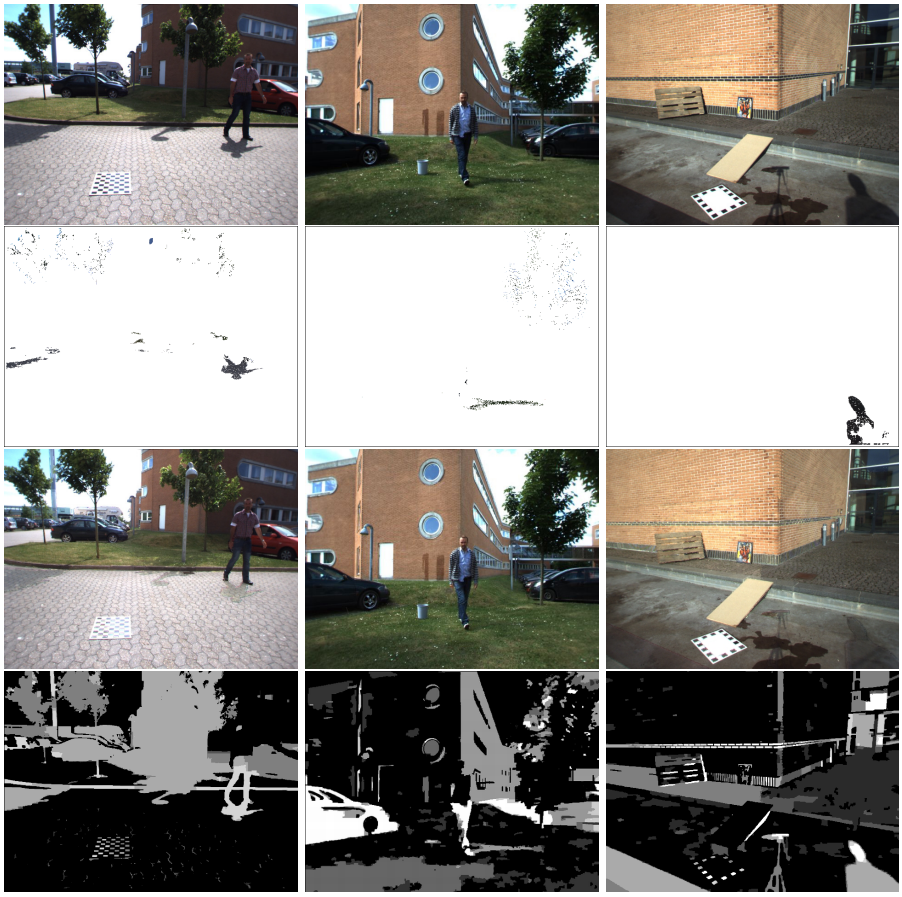

Fig. 10. First row: original images from three different sequences. Second row: shadows detected by approach described in this paper. Third row: shadows removed with graph cuts based approach. Fourth row: level of shadow (brighter areas represent a deeper shadow level).

A fundamental assumption for this work is that the surfaces are Lambertian (diffuse). We have demonstrated on a number of real outdoor sequences that our model works on a large range of naturally occurring materials in outdoor scenes. Material such as concrete and grass are far from Lambertian when viewed close-up, but at a certain distance they overall display diffuse reflection behaviour because of the surface roughness. Glass and metal surfaces pose a real problem, but the stereo camera can typically not produce valid disparity information from such surfaces and the risk of falsely detecting shadows on such materials is not high (we do not allow a pixel to be classified as shadow if there is no valid disparity value for it). We are presently working on developing a technique for detecting, over long image sequences, pixels that do not conform to a diffuse

reflection assumption (i.e., do not over the entire sequence consistently vote for the same illumination model as the majority of the pixels in the scene).

Finally, the theoretical framework is based on an assumption of the camera having a linear response curve, and that the color channels are independent. This is typically only true for very high quality cameras, and certainly the cameras in the Bumblebee stereo rig are not designed for color vision applications. Regardless, we have demonstrated that the model works quite well even with cameras of such low quality.

6 Conclusion

We have presented a technique for detecting shadows in dynamic scenes. The main contributions lie in the combination of color and disparity data analysis, and in the use of qualitative chromaticity plane descriptors for ruling out false positives in the shadow pixel populations. A powerful feature of the proposed approach is its ability to handle albedo changes in the scene, e.g., its robustness towards falsely labeling pixels as shadow in situations where surfaces in the scene have become wet. Another promising feature of the work is that the techniques employed allow for 3Hz operation on commodity hardware using a commercially available dense stereo camera solution.

We conjecture that by enabling vision systems to estimate information about the illumination conditions in the scene the vision systems can be made more robust. In this paper we have demonstrated that it is possible to estimate powerful information concerning the scene illumination in terms of the illuminant direction which can be utilized to verify dynamic shadows and to detect static ones, as well.

Future work will include combining this work with static shadow detection, working with temporal analysis of the detected shadows and illumination information, and using this illumination estimation for realistic augmentation of synthetic objects into the scenes.

Acknowledgments

This research is funded by the CoSPE (26-04-0171) and the BigBrother (274-07-0264) projects under the Danish Research Agency. This support is gratefully acknowledged.

References

1. Bartczak, B., Schiller, I., Beder, C., Koch, R.: Integration of a time-of-flight camera into a mixed reality system for handling dynamic scenes, moving viewpoints and occlusions in real-time. In: Proceedings of the 3DPVT Workshop, Atlanta, GA, USA (June 2008)

2. Chalidabhongse, T.H., Kim, K., Harwood, D., Davis, L.: A Perturbation Method for Evaluating Background Subtraction Algorithms. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Nice, France, October 11-12 (2003)
3. Finlayson, G.D., Hordley, S.D., Drew, M.S.: Removing shadows from images. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 823–836. Springer, Heidelberg (2002)
4. Hu, J.-S., Su, T.-M.: Robust Background Subtraction with Shadow And Highlight Removal for Indoor Surveillance. Journal on Advanced in Signal Processing 2007(1), 108–132 (2007)
5. Huerta, I., Holte, M.B., Moeslund, T.B., Gonzàlez, J.: Detection and removal of chromatic moving shadows in surveillance scenarios. In: Proceedings: IEEE ICCV 2009, Kyoto, Japan (September 2009)
6. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time Foreground-Background Segmentation using Codebook Model. Real-time Imaging 11(3), 167–256 (2005)
7. Moeslund, T.B., Hilton, A., Krüger, V.: A Survey of Advances in Vision-Based Human Motion Capture and Analysis. Journal of Computer Vision and Image Understanding 104(2-3) (2006)
8. Nielsen, M., Madsen, C.B.: Graph cut based segmentation of soft shadows for seemless removal and augmentation. In: Proceedings: Scandinavian Conference on Image Analysis, Aalborg, Denmark, June 2007, pp. 918–927 (2007)
9. Nielsen, M., Madsen, C.B.: Segmentation of soft shadows based on a daylight- and penumbra model. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 341–352. Springer, Heidelberg (2007)
10. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting Moving Shadows: Algorithms and Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 918–923 (2003)
11. Salvador, E., Cavalarro, A., Ebrahimi, T.: Shadow identification and classification using invariant color models. Computer Vision and Image Understanding 95, 238–259 (2004)
12. Point Grey Research: Bumblebee, `http://www.ptgrey.com/products/bumblebee/index.html`