SIViP (2017) 11:817–824
DOI 10.1007/s11760-016-1027-x

ORIGINAL PAPER

Sun position estimation and tracking for virtual object placement in time-lapse videos

Hasan Balcı[1] · **Uğur Güdükbay**[1]

Received: 19 June 2016 / Revised: 12 November 2016 / Accepted: 17 November 2016 / Published online: 30 November 2016

Abstract Realistic illumination of virtual objects placed in real videos is important in terms of achieving visual coherence. We propose a novel approach for illumination estimation on time-lapse videos and seamlessly insert virtual objects in these videos in a visually consistent way. The proposed approach works for both outdoor and indoor environments where the main light source is the Sun. We first modify an existing illumination estimation method that aims to obtain sparse radiance map of the environment in order to estimate the initial Sun position. We then track the hard ground shadows on the time-lapse video by using an energy-based pixel-wise method. The proposed method aims to track the shadows by utilizing the energy values of the pixels that forms them. We tested the method on various time-lapse videos recorded in outdoor and indoor environments and obtained successful results.

Keywords Sun position estimation · Light source estimation · Illumination estimation · Time-lapse video · Shadow tracking · Image/video editing

1 Introduction

One typical video editing application is to integrate virtual objects seamlessly into a real video, especially in a visual context, so that the user cannot differentiate the virtual objects from the real ones. To this end, consistent illumination of virtual objects within the real video is important to obtain visual coherence. This can be achieved in different ways, such as estimating the locations of light sources in the real environment, considering the shadows cast by/on virtual objects or calculating the global illumination.

Light source estimation in indoor environments is more complex than the one in outdoor environments because indoor environments may include different kinds of light sources while the main light source in outdoor environments is the Sun. However, light sources in indoor environments are generally static, whereas the Sun position changes over time. Regarding the environments that are lit mainly by sunlight, a user may want to observe the virtual objects on different times of the day where the Sun is in a different position each time. For example, an architect may want to analyze the appearance of a building in the course of the day. For this purpose, time-lapse videos can be formed by capturing frames of the real environment during the day and then they can be combined with Augmented Reality (AR) technology.

We introduce a new approach that provides visual consistency on time-lapse videos where the main light source is the Sun. First, we estimate the initial position of the Sun from the first frame of the time-lapse video by using a modified and enhanced state-of-the-art method. Then, by keeping track of the shadow length and direction found on the ground, we estimate the change in the Sun position and direction on each frame and adjust the illumination of the virtual objects accordingly. The contributions are as follows:

- We modify and enhance an existing method, which tries to estimate the illumination of a real scene from a single image, for estimating the Sun position.
- We propose a new algorithm that can estimate the Sun position fast and accurately on time-lapse videos. By

Electronic supplementary material The online version of this article (doi:10.1007/s11760-016-1027-x) contains supplementary material, which is available to authorized users.

✉ Uğur Güdükbay
gudukbay@cs.bilkent.edu.tr

[1] Department of Computer Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey

means of the energy-based pixel-wise algorithm we developed, we can track the hard ground shadows during the time-lapse video and estimate the changes in the Sun position. We use the estimated Sun position to insert virtual objects into the time-lapse video with correct lighting and shadows.

2 Related work

There are many studies related to illumination estimation for both outdoor and indoor environments. Because the main light source in outdoor environments is the Sun, the studies related to estimating illumination on outdoor environments mainly focus on finding the Sun position. In some studies, the skylight is considered as the ambient light of the scene in addition to the Sun [1–5]. Some studies assume that the Sun position is known and find the change in intensity of the sunlight during a period of time [6–8].

Illumination estimation in indoor environments is challenging because these environments may have more than one dominant light source and these sources may have different shapes. Most of the studies related to indoor environments calculate the radiance on the scene instead of estimating the light source positions [9–15]. For this purpose, they need the 3D model or geometric information of the scene and this is achieved mostly by using RGB-D cameras and depth maps.

There are not so many studies related to illumination estimation in time-lapse videos. Sunkavalli et al. [16] and Zhang et al. [17] both decompose each image in the time-lapse video into sunlight and skylight basis images by using the information from whole video sequence. They aim to relight the scene easily, to recover a portion of the scene geometry and to perform some image editing operations.

Lalonde et al. [18] transfer appearance and illumination from time-lapse sequences to other time-lapse sequences or single images. They have a Webcam database that contains time-lapse sequences. To transfer appearance to an original image, they evaluate illumination conditions that are the sun position, sky color and weather conditions of the original image. They find an image from the Webcam database with similar illumination conditions and transfer an object from that image to the original image. For illumination transfer, they propose a model to obtain the high dynamic range environment maps of the images. They use the acquired environment maps to illuminate the virtual objects into the images.

3 Estimation of the initial Sun position

We need to estimate the initial Sun position before tracking shadows to calculate the change in the Sun position during a time-lapse video sequence. For this purpose, we use the approach of Chen et al. [19], which tries to estimate the scene illumination from a single image, and modify it according to our needs. The main advantage of this approach is that it tries to estimate the illumination in both outdoor and indoor environments. Because we are dealing with environments where the Sun is the main light source, these may include both outdoor and indoor environments whose main light source is the Sun (cf. Fig. 1).

Fig. 1 Sample outdoor environment (*left*), sample indoor environment with the Sun as the main light source (*right*)

The approach by Chen et al. consists of three stages. They construct the scene geometry from a single image using image features. They then decompose the image into its intrinsic components, which are reflectance and shading images. Finally, with an optimization process that uses the geometry information and the shading image, they estimate the environment illumination. Their approach does not estimate the exact positions of light sources; instead, they approximate the environment illumination with light sources placed symmetrically on a hemisphere covering the scene. We describe how we modify their approach for time-lapse videos.

3.1 Geometry extraction

Chen et al. require the geometric model of the scene represented by the image, which is used to calculate the normal vectors of surfaces for illumination computations. To this end, they use the approach by Saxena et al. [20], which estimates the scene geometry from a single image. We observed that the accuracy of this approach (65%) is not sufficient to estimate a correct position of the Sun because the extracted scene geometry includes a lot of incorrect surface normals.

We obtain the coarse 3D model of the scene in a preprocessing stage that requires some manual processing. In a study whose only aim is to estimate the Sun position in a single image, the automatic geometry extraction from the image generally gives acceptable results. However, for estimating the initial Sun position for time-lapse videos, inaccurate results at the initial stage will affect later stages severely.

3.2 Intrinsic components

On the second stage of the illumination estimation, Chen et al. decompose the input image into its intrinsic components, which are reflectance and shading images (Fig. 2b, c). An image can be considered as per-pixel product of its intrinsic components. The shading image obtained as a result of the

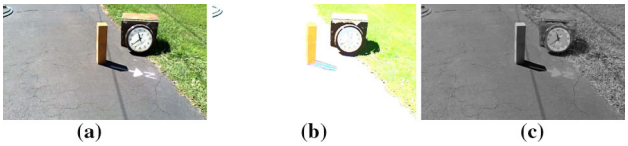

Fig. 2 Intrinsic components of an image. Starting from the input image (**a**), we compute the reflectance image (**b**), and the shading image (**c**)

decomposition can be accepted as the irradiance of the surface and used in the illumination model. We use the approach by Garces et al. [21] to decompose the images into their intrinsic components.

3.3 Illumination model

At this stage, our aim is to estimate the Sun position from the first frame of the time-lapse video by using the 3D model of the scene and the shading image that we acquire on the previous stage. To this end, we use the illumination estimation method proposed by Chen et al. tailored to our needs.

We first place eight spotlights over the 3D model of the scene homogeneously and symmetrically in such a way that these spotlights will construct a hemisphere over the scene. Each spotlight is directed to the center of the constructed hemisphere, and their lighting ranges are adjusted in such a way that they enclose the 3D model entirely. These spotlights approximate the real-world illumination caused by the main light sources in the scene. Apart from these spotlights, a point-light is also placed over the hemisphere in order to obtain the ambient light coming from the other points on the scene other than the main light sources. Figure 3 depicts a sample model with its light sources.

In the second step, we find the corresponding positions and normals of the pixels in the 3D model. For this purpose, we match the image with the 3D model of the scene by an edge-based camera tracking algorithm that uses control points assigned on the model edges (cf. Fig. 4). Then, a ray is cast from the camera position to each pixel in the image. The first point where the ray cast intersects the 3D model is the position of the pixel. The normal value of the pixel can also be found easily.

In the next step, we determine the spotlights that contribute to the illumination of each pixel. For this purpose, we cast a ray from each of eight spotlights to the geometric position of a pixel. If the first point where the ray cast from a spotlight intersects the 3D model is equal to the geometric position of the pixel to which the ray is cast, then the spotlight contributes to the illumination of the pixel. Otherwise, the pixel is blocked by some parts of the model and that spotlight does not contribute to the illumination of the pixel. If we subtract the position of a spotlight from the position of a pixel, we obtain the light vector from the spotlight to the pixel, which we use for illumination calculation. On Lambertian surfaces, the irradiance can be represented by [22]:

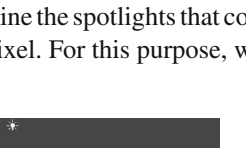

Fig. 3 A 3D model with eight spotlights and one point-light

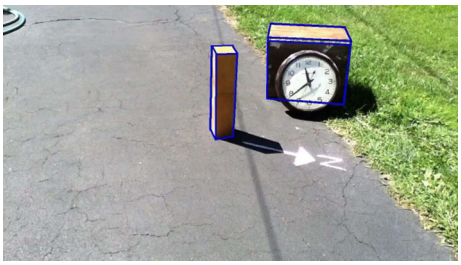

Fig. 4 Image matched with the 3D model

$$S = I_a + \sum_{i=1}^{m} v_i \; I_i \; (L_i \cdot N), \tag{1}$$

where S is the pixel values of the shading image, I_a is the ambient light, I_i is the intensity of the ith light source reaching to the pixel positions, L_i is the direction of the ith light source to the pixel positions, m is the number of light sources that illuminate the pixels, N is the normal of the pixel position, and v_i is the visibility of the ith light source at each pixel. The visibility term is 1 if the light source sees the pixel and 0 if it does not. Among these variables, we know S, L_i, N, and v_i and we need to find I_a and I_i. By applying Levenberg–Marquardt minimization using the shading image and the estimated irradiance, which is a robust algorithm to solve nonlinear least squares problems [23], we obtain the intensity values of the eight spotlight sources and the point-light source required to approximate the real-world illumination:

$$\underset{(I_a, I_1, I_2, \ldots, I_8)}{\arg\max} \sum_{j=1}^{n_s} \left(S_j - \left(I_a + \sum_{i=1}^{8} v_i \; I_i \; (L_i \cdot N) \right) \right), \tag{2}$$

where S_j is the value of the jth pixel on the shading image and n_s is the total number of pixels.

Although we obtain the intensity values of the spotlights and point-light we placed over the 3D model, in the environments where the main light source is the Sun, the source of illumination is mostly concentrated on one or two spotlights. For the next stage of our framework, we need to find a single position for the Sun so that we can easily track its movement. We determine the center of intensity formed by the spotlights on the hemisphere as the estimated position of the Sun and

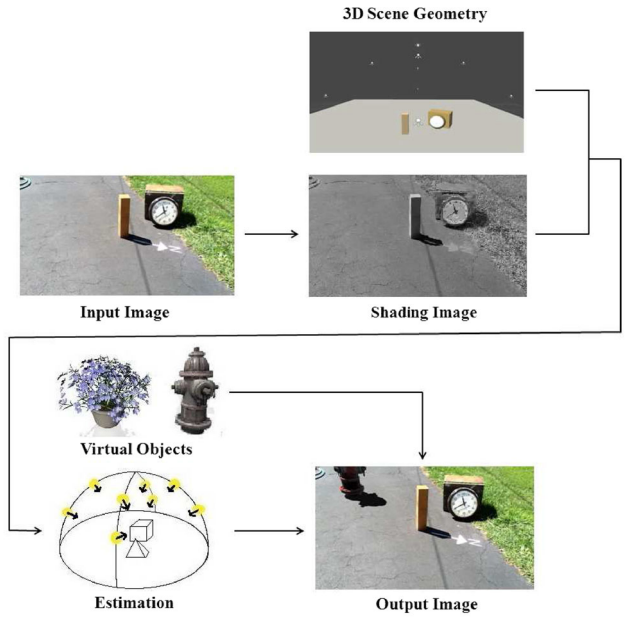

Fig. 5 The overview of the initial Sun position estimation

place a directional light source to that position instead of the eight spotlights. The point-light source that we place over the hemisphere is used to account for the ambient light. The overall framework is depicted in Fig. 5.

4 Shadow-tracking-based Sun position estimation

The minimization algorithm to estimate the initial Sun position takes around ten seconds for a video frame of 480×640 resolution. Because of its high computational cost, we cannot apply it on every frame. To estimate the Sun position for the remaining frames, we propose a method that uses hard ground shadows on the Sun direction, which we assume that there exists at least one hard shadow in the video. Our method estimates the Sun position at each frame based on the changes in the length and direction of the hard shadow.

First, we detect hard ground shadows in the first frame. Then, we test these shadows according to a set of criteria (described below), calculate the energy image of the first frame and determine the most appropriate shadow for tracking. On the following frames, we track this shadow by using a pixel-wise method that uses the energy images of these frames. According to the changes in the length of the shadow and its direction, we estimate the elevation and azimuth angles of the Sun and update the Sun position.

4.1 Shadow selection

In order to estimate the Sun position during a time-lapse video, we use hard ground shadows. Ground shadows are more informative than the shadows on the other surfaces for

Fig. 6 An output image with shadow edges shown in *blue* (*left*), and a sample energy image (*right*)

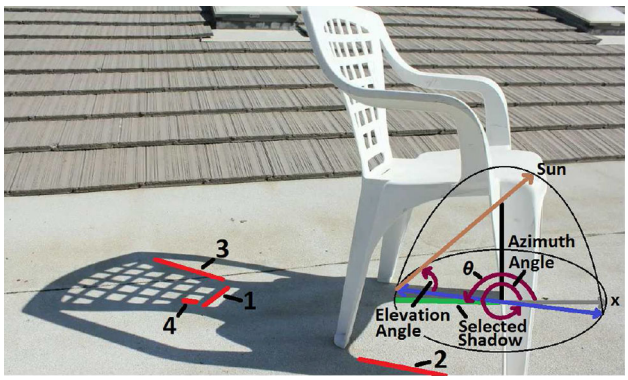

Fig. 7 Shadow edges eliminated in Steps 1, 2, 3, 4 and the selected shadow. The elevation angle depends on the shadow length. The azimuth angle is updated by adding the difference between the θ values in current and previous frames

estimating the elevation and azimuth angles of the Sun. Hard shadows are easier to track than soft shadows because soft shadows may be unstable during the video.

Although hard shadows are good at estimating the Sun position, every hard ground shadow may not work for our purpose. We determine an appropriate hard ground shadow to track during the time-lapse video. For this purpose, we detect the hard ground shadows on the first frame by using the method proposed by Lalonde et al. [24]. An example output of this method can be seen in Fig. 6. We also need to extract the energy image of the first frame. The energy image is useful to eliminate the hard shadows not used for Sun position estimation and to track the selected hard shadow in the following frames because the pixels that form the shadow edges have higher energy values than the other pixels. An energy image can be generated by applying an energy function to each pixel in an image. We use the energy function proposed by Avidan and Shamir [25]:

$$e(I) = \left|\frac{\partial}{\partial x} I\right| + \left|\frac{\partial}{\partial y} I\right|, \tag{3}$$

where I is the input image. We apply this energy function to the first frame of the time-lapse video and obtain its energy image (cf. Fig. 6). We use the energy images during the shadow tracking process in subsequent frames.

We eliminate all but one hard ground shadow and decide the most suitable one to be used in the subsequent frames according to the following steps in the given order (cf. Fig. 7):

1. If the angle between a shadow edge and the estimated initial position of the Sun is greater than 10°, then that shadow is eliminated. Such shadows do not give accurate information about the changes on the Sun position. Because the initial position of the Sun is estimated with at most 10° error, we do not want to miss a shadow that is on the direction of the Sun.
2. If at least one end of a shadow is on the boundary of the frame, that shadow is also eliminated. This is because the whole shadow is not seen in the frame. In other words, the visible part of the shadow may disappear or the invisible part may appear in the following frames and this may cause errors in the results.
3. The shadows whose energy lower than the average energy are eliminated. First, a shadow energy is calculated for each shadow by taking the average of the energy of the pixels that form the shadow. The energy of the pixels is taken from the energy image generated previously. Then the average of all shadow energies is calculated, and the shadows whose energy lower than the average value are eliminated. Tracking shadows with high energy is easier than tracking the ones with low energy.
4. The average shadow length is calculated for the remaining shadows and if a shadow is shorter than the average length, that shadow is eliminated. This is because it is hard to obtain significant information about the changes in the length and direction of short shadows and long shadows provide more precise information.
5. From the remaining shadows, if there exist shadows whose only one corner intersects with a corner of an object in the 3D model, we choose the shadow with the highest energy from these shadows as the final shadow. If there is no such shadow, we search for shadows whose both corners do not intersect any object in the 3D model and select the one with the highest shadow energy. If again there is no such shadow, we choose the shadow with the highest energy as the final shadow. To decide whether or not the corner of a shadow intersects with the corner of an object on the model, we use the matched image-model pair described previously.

4.2 Shadow tracking algorithm

We need to define some variables that will be used in the proposed shadow tracking algorithm. We define the pixel that is closest to the estimated position of the Sun from the shadow pixels as the start pixel (s). Related with this, s_x and s_y denote the horizontal and vertical pixel distances of the start pixel s to the top-left corner of the frame, respectively. Similarly, we define the pixel on the other corner of the shadow as the final pixel f. f_x and f_y denote the horizontal and vertical pixel distances to the top-left corner of the frame, respectively. We calculate the shadow length as l_s as $l_s = \sqrt{(f_x - s_x)^2 + (f_y - s_y)^2}$.

Let h and a denote the elevation and azimuth angles of the Sun, respectively. We measure the azimuth angle on the ground from the positive x-axis in counterclockwise direction. The angles h and a are initially equal to the elevation and azimuth angles of the Sun estimated for the first frame and updated in subsequent frames. The length of the object that generates the shadow we are tracking, l_v, is calculated as $l_v = l_s \tan(h)$. It is used to calculate the elevation angle of the Sun in subsequent frames.

Let θ denote the counterclockwise angle that the shadow edge makes with the positive x-axis, which is calculated as

$$\theta = \begin{cases} 360 - \arccos\left(\frac{\mathbf{t}\cdot\mathbf{v}}{\sqrt{(f_x-s_x)^2+(f_y-s_y)^2}}\right), & \text{if } s_y < f_y \\ \arccos\left(\frac{\mathbf{t}\cdot\mathbf{v}}{\sqrt{(f_x-s_x)^2+(f_y-s_y)^2}}\right), & \text{otherwise} \end{cases} \quad (4)$$

where $\mathbf{t}$ is a vector from (s_x, s_y) to $(s_x{+}1, s_y)$ and $\mathbf{v}$ is a vector from (s_x, s_y) to (f_x, f_y). The difference between the θ values in consecutive frames is used to update the azimuth angle at each frame (cf. Eq. 7).

After the preprocessing on the first frame of the time-lapse video, we track the shadow and estimate the Sun position in the subsequent frames (cf. Algorithm 1).

Algorithm 1 Shadow tracking and estimation of elevation and azimuth angles of the Sun

INPUT: time-lapse video, start pixel s, azimuth angle a, obj. length l_v
OUTPUT: updated Sun position

```
while not the end of the video do
  fetch the frame into a matrix
  extract the energy image of the frame
  determine the search direction according to a
  update s using s of the previous frame
  check ← true
  while check do
    search the next pixel, p_n, which belongs to the shadow
    according to the search direction determined using a
    if ((energy value of p_n < energy threshold) or
        (slope of shadow > slope threshold)) then
      check ← false
      f ← p_n
    end if
  end while
  calculate h using f, s and l_v
  calculate a using f, s of the current frame and the a value
  of the previous frame
  update Sun position according to new h and a values
end while
```

First, we fetch the next frame on the video sequence and extract its energy image. We check whether the start pixel s that comes from the previous frame has still the same energy value in the current frame. If its energy value does not change, we continue to use s as the start pixel. Otherwise, we search

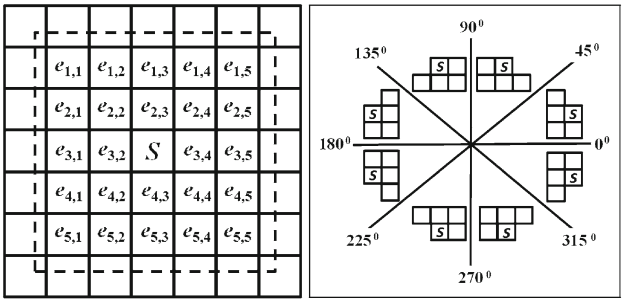

Fig. 8 Shadow construction. **a** Searching for the start pixel for the current frame. **b** Determining the search direction. According to the value of the azimuth angle a, the appropriate search direction is chosen to form the shadow in the current frame. A pixel with label s denotes the start pixel, and the other four pixels are the candidates for shadow pixels

for a new start pixel to construct the shadow on the current frame. We determine the new position of the start pixel by searching the pixel that is the center of mass of a 5×5 pixel area in terms of energy values, where the center pixel is the old s (cf. Fig. 8a). $e_{i,j}$ denotes the energy value of the pixel on the ith row and the jth column inside the pixel area. We find the center of mass of the pixel area as follows:

$$m_i = \text{round}\left(\sum_{i=1}^{5}\sum_{j=1}^{5} j\, e_{i,j}\right), \quad m_j = \text{round}\left(\sum_{j=1}^{5}\sum_{i=1}^{5} i\, e_{i,j}\right), \tag{5}$$

where m_i and m_j denote the row and column indices of the center of mass within the 5×5 pixel area. We assign the center of mass as the new start pixel by using Eq. 6:

$$s = (s_x - 3 + m_j, s_y - 3 + m_i). \tag{6}$$

Before we start the shadow construction in the current frame, we need to determine the search direction for possible shadow pixels. For this purpose, we use the azimuth angle a calculated for the previous frame and move in the chosen search direction, starting from s (cf. Fig. 8b). While selecting the next shadow pixel p_n, we choose the pixel with the maximum energy from four possible pixels on the search direction. We continue by constructing the shadow edge that we track to estimate the elevation and azimuth angles of the Sun until one of two termination conditions occurs:

1. We stop finding the next pixel if the energy value of a pixel is less than the energy threshold. We define the energy threshold as the half of the energy of the initial shadow in the first frame. A pixel whose energy is below the threshold means that we reach the other corner of the shadow edge (cf. Fig. 9 left).
2. While constructing the shadow edge in the current frame, we calculate the slope of the edge after the 15th pixel. Then, at every ten pixels, we calculate the slope of the last ten pixels and if the last calculated slope differs at least one-third of the slope of the first 15 pixels, we stop the shadow edge construction (cf. Fig. 9 right). It means that we reach the other corner of the edge and probably a new edge in another direction starts, where the yellow pixel is the 15th pixel of the shadow edge. As it is seen, because the slope of the last 10 pixels after the green pixel differs from the slope of the first 15 pixels at least by its one-third, we stop searching new pixels and accept the green pixel as the final pixel f.

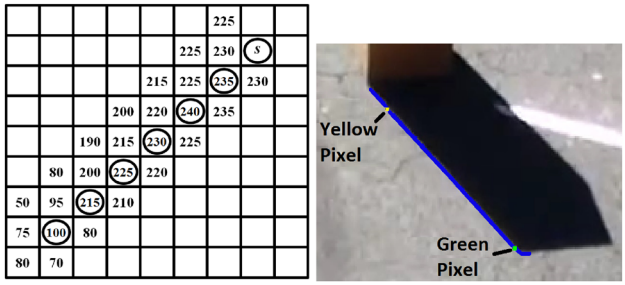

Fig. 9 The termination condition based on energy (*left*) and slope (*right*)

We determine the energy and slope thresholds as a result of the repeated experiments. We observe that the second termination condition is encountered more frequently than the first one. After we find the starting and ending pixels of the shadow edge in the current frame, we calculate the elevation angle, h, and the azimuth angle, a, as follows:

$$h = \arctan(l_v/l_s), \quad a = a_{\text{prev}} + (\theta_{\text{curr}} - \theta_{\text{prev}}), \tag{7}$$

where a_{prev} denote the azimuth angle for the previous frame, θ_{curr} and θ_{prev} denote the values of the angle θ (Eq. 4) for the current and previous frames, respectively. The Sun position is updated using the new elevation and azimuth angles.

5 Evaluation and results

To the best of our knowledge, this is the first approach that aims to estimate the Sun position in each frame and to insert a virtual object into a time-lapse video. We present still images from the time-lapse videos with virtual objects placed seamlessly to show the qualitative results. We compare the estimated elevation and azimuth angles of the Sun with ground truth values for quantitative evaluation. We used a notebook computer with Intel i7-4700MQ (2.4GHz Clock) processor, 6Gb RAM, AMD Radeon HD 8750M GPU for the experiments. We implemented the initial Sun position estimation on the Unity game engine [26] and the shadow-tracking-based Sun position estimation using MATLAB [27]. We use Levenberg–Marquardt minimization algorithm to

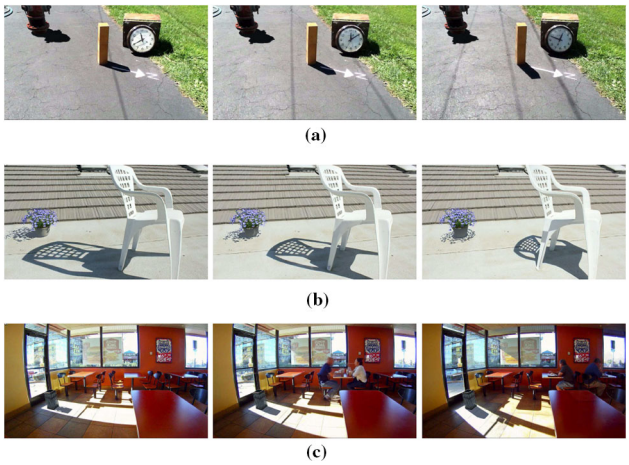

Fig. 10 Still frames from the time-lapse videos. The illumination of the virtual object changes synchronously with the Sun position. **a** fire hydrant, **b** flower, and **c** trash bin

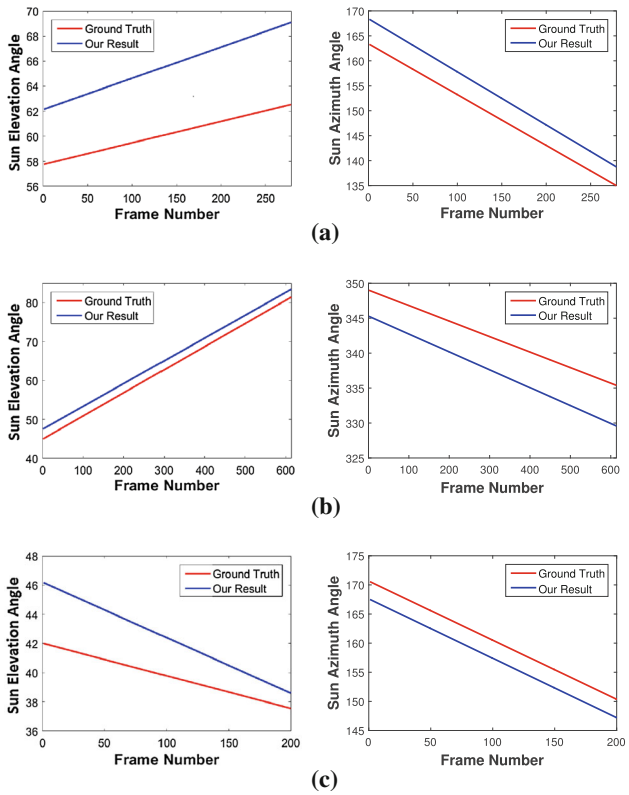

Fig. 11 Comparison of the elevation (*left*) and azimuth angles (*right*) of the Sun with ground truth values for the time-lapse videos. **a** Sun dial, **b** chair, and **c** fast food

estimate the initial Sun position using the first video frame, which takes around 10 seconds for a video frame of 480×640 resolution. We then identify hard ground shadows and select the one to be used for shadow tracking on subsequent video frames, which takes around approximately 1–2 min. After this preprocessing stage, subsequent frames can be processed in real time by smoothing estimated elevation and azimuth angle values. In this way, the algorithm can also tolerate if hard shadow disappears in a few frames. We obtain frame rates of more than 30 frames per second (fps).

Table 1 Error rates of the elevation and azimuth angles

	Sun dial	Chair	Fast food
Elevation min error	4.38°	2.01°	1.04°
Elevation max error	6.54°	2.55°	4.17°
Elevation mean error	**5.46°**	**2.28°**	**2.60°**
Azimuth min error	3.7°	3.73°	3.05°
Azimuth max error	5.02°	5.81°	3.16°
Azimuth mean error	**4.38°**	**4.77°**	**3.10°**

Figure 10 presents the visual results of the application of the method on three time-lapse videos [28–30]. The still frames show that the virtual objects are properly illuminated and seamlessly integrated into the real scenes (please confirm the accompanying video). Figure 11 depicts graphs that compare the estimated elevation and azimuth angles with the ground truth values. The initial differences between our results and the ground truth values show that the errors are caused from the first stage that estimates the initial Sun position. Starting with these initial error values, our shadow tracking approach calculates the Sun position in subsequent frames with reasonable error.

Table 1 shows the error rates of the elevation and azimuth angles. The mean error rates on both elevation and azimuth angles (in bold) are less than 6°. According to our observations, Sun position estimation is easier for the scenes with more light-shadow content. These scenes also include more alternatives to choose the shadow to be used for tracking. Please see the electronic appendix for further results.

6 Conclusions and future work

We propose a new method for illumination estimation in time-lapse videos. We mainly target videos of indoor and outdoor environments where the only light source is the Sun. We estimate the Sun position during a time-lapse video and calculate the illumination of the virtual objects placed in the real video accordingly. Our method first estimates the initial position of the Sun from the first frame of the video by modifying an existing illumination estimation method. Then by tracking a hard ground shadow in the scene with an energy-based pixel-wise method for the rest of the frames, it estimates the changes in the Sun position. The proposed method gives successful results on videos found on the Internet.

The proposed method has some drawbacks such as preparing a coarse 3D model of the environment requires manual processing during the preprocessing stage. Automatic or semi-automatic approaches requiring minimal user interac-

tion for scene reconstruction, such as cuboid-proxies [31], could be exploited to generate the coarse model of the environment. We also assume that there exists at least one appropriate hard ground shadow during the time-lapse video. Soft ground shadows may be used in order to track some other light sources that may have effect on the environment. Moreover, the shadows other than the ones on the ground may be used additionally to increase the accuracy.

Another future extension would be to evaluate the energy changes on the pixels that belong to the tracked shadow in the time-lapse video. This evaluation gives the opportunity to track the changes on the light intensity of the Sun as well as its position. We could exploit graphics processor unit (GPU) to speed up the processing of video frames.

Acknowledgements This research is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 112E110. The first author is supported by TÜBİTAK under BİDEB 2210 Graduate Scholarship. We gratefully acknowledge Gordon Bates, Matthew Davies and Tom Gazda for permitting us to use their videos.

References

1. Panagopoulos, A., Wang, C., Samaras, D., Paragios, N.: Illumination estimation and cast shadow detection through a higher-order graphical model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), pp. 673–680 (2011)
2. Lalonde, J.-F., Efros, A.A., Narasimhan, S.G.: Estimating the natural illumination conditions from a single outdoor image. Int. J. Comput. Vis. **98**(2), 123–145 (2012)
3. Guo, R., Dai, Q., Hoiem, D.: Paired regions for shadow detection and removal. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2956–2967 (2013)
4. Liu, Y., Gevers, T., Li, X.: Estimation of sunlight direction using 3D object models. IEEE Trans. Image Proc. **24**(3), 932–942 (2014)
5. Lalonde, J.-F., Matthews, I.: Lighting estimation in outdoor image collections. In: Proceedings of the International Conference on 3D Vision, vol. 1, Tokyo, Japan, pp. 131–138 (2014)
6. Liu, Y., Granier, X.: Online tracking of outdoor lighting variations for augmented reality with moving cameras. IEEE Trans. Vis. Comput. Graph. **18**(4), 573–580 (2012)
7. Andersen, M.S., Jensen, T., Madsen, C.B.: Estimation of dynamic light changes in outdoor scenes without the use of calibration objects. In: Proceedings of the International Conference on Pattern Recognition, vol. 4, 91–94 (2006)
8. Xing, G., Liu, Y., Qin, X., Peng, Q.: On-line illumination estimation of outdoor scenes based on area selection for augmented reality. In: Proceedings of the 12th International Conference on Computer-Aided Design and Computer Graphics (CADGRAPHICS'11), pp. 439–442 (2011)
9. Ikeda, T., Oyamada, Y., Sugimoto, M., Saito, H.: Illumination estimation from shadow and incomplete object shape captured by an RGB-D camera. In: Proceedings of the 21st International Conference on Patten Recognition Series (ICPR'12), Tsukuba, Japan, pp. 165–169 (2012)
10. Gruber, L., Langlotz, T., Sen, P., Hoherer, T., Schmalstieg, D.: Efficient and robust radiance transfer for probeless photorealistic augmented reality. In: Proceedings of the IEEE Virtual Reality, pp. 15–20 (2014)
11. Lensing, P., Broll, W.: Instant indirect illumination for dynamic mixed reality scenes. In: Proceedings of the 11th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'12), pp. 109–118 (2012)
12. Neverova, N., Muselet, D., Trémeau, A.: Lighting estimation in indoor environments from low-quality images. In: Proceedings of the 12th International Conference on Computer Vision (ICCV'12), vol. 2, pp. 380–389 (2012)
13. Yoo, J., Lee, K.: Light source estimation for realistic shadow using segmented HDR images. In: Hong, D., Jeon, S. (eds.) Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR'07), vol. 260 (2007)
14. Lopez-Moreno, J., Garces, E., Hadap, S., Reinhard, E., Gutierrez, D.: Multiple light source estimation in a single image. Comput. Graph. Forum **32**(8), 170–182 (2013)
15. Zhao, W., Zheng, Y., Wang, L., Peng, S.: Lighting estimation of a convex Lambertian object using weighted spherical harmonic frames. Signal Image Video Proc. **9**(1), 57–75 (2015)
16. Sunkavalli, K., Matusik, W., Pfister, H., Rusinkiewicz, S.: Factored time-lapse video. In: ACM Transactions on Graphics. Proceedings of the SIGGRAPH'07, vol. 26, no. 3, Article No. 101 (2007)
17. Zhang, R., Zhong, F., Lin, L., Xing, G., Peng, Q., Qin, X.: Basis image decomposition of outdoor time-lapse videos. Vis. Comput. **29**(11), 1197–1210 (2013)
18. Lalonde, J.-F., Efros, A.A., Narasimhan, S.G.: Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. In: ACM Transactions on Graphics. Proceedings of the SIGGRAPH Asia'09, vol. 28, no. 5, Article No. 131 (2009)
19. Chen, X., Wang, K., Jin, X.: Single image based illumination estimation for lighting virtual object in real scene. In: Proceedings of the 12th International Conference on Computer-Aided Design and Computer Graphics, pp. 450–455 (2011)
20. Saxena, A., Sun, M., Ng, A.: Make3D: learning 3D scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 824–840 (2009)
21. Garces, E., Munoz, A., Lopez-Moreno, J., Gutierrez, D.: Intrinsic images by clustering. In: Computer Graphics Forum. Proceedings of the Eurographics Symposium on Rendering, vol. 31, no. 4, pp. 1415–1424 (2012)
22. Phong, B.T.: Illumination for computer generated pictures. Commun. ACM **18**(6), 311–317 (1975)
23. Moré, J.J.: The Levenberg–Marquardt algorithm: implementation and theory. In: Watson, G.A. (ed.) Numerical Analysis, pp. 105–116. Springer, Berlin (1977)
24. Lalonde, J.-F., Efros, A., Narasimhan, S.: Detecting ground shadows in outdoor consumer photographs. In: Proceedings of the European Conference on Computer Vision (ECCV'10), LNCS, vol. 6312, pp. 322–335 (2010)
25. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. In: Proceedings of the 34th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'07). New York, NY, USA: ACM (2007)
26. Unity Technologies: Unity Game Engine. http://unity3d.com/. Accessed 2016-11-12
27. MathWorks: MATLAB. http://www.mathworks.com/products/matlab/. Accessed 2016-11-12
28. Gazda, T.: Sun's shadow time lapse. https://www.youtube.com/watch?v=3B7KLstUZbI. Accessed 2016-11-12
29. Davies, M.: Shadows timelapse. https://www.youtube.com/watch?v=Lvhjbrr5GI8. Accessed 2015-11-12
30. Bates, G.: Fast food shadows timelapse. https://www.youtube.com/watch?v=mdhS6pds8VY. Accessed 2015-11-12
31. Zheng, Y., Chen, X., Cheng, M.-M., Zhou, K., Hu, S.-M., Mitra, N.J.: Interactive images: Cuboid proxies for smart image manipulation. In: ACM Transactions on Graphics. Proceedings of the SIGGRAPH'12, vol. 31, no. 4, Article No. 99 (2012)