Visual Comput (2005) 21: 639–648
DOI 10.1007/s00371-005-0335-x

ORIGINAL ARTICLE

Xiaochun Cao
Yuping Shen
Mubarak Shah
Hassan Foroosh

Single view compositing with shadows

Published online: 1 September 2005

X. Cao (✉) · Y. Shen · M. Shah · H. Foroosh
School of Computer Science, University of Central Florida, Orlando, FL, 32816
{xccao, ypshen, shah, foroosh}@cs.ucf.edu

Abstract In this paper, we describe how geometrically correct and visually realistic shadows may be computed for objects composited into a single view of a target scene. Compared to traditional single view compositing methods, which either do not deal with the shadow effects or manually create the shadows for the composited objects, our approach efficiently utilizes the geometric and photometric constraints extracted from a single target image to synthesize the shadows consistent with the overall target scene for the inserted objects. In particular, we explore (i) the constraints provided by imaged scene structure, e.g. vanishing points of orthogonal directions, for camera calibration and thus explicit determination of the locations of the camera and the light source; (ii) the relatively weaker geometric constraint, the planar homology, that models the imaged shadow relations when explicit camera calibration is not possible; and (iii) the photometric constraints that are required to match the color characteristics of the synthesized shadows with those of the original scene. For each constraint, we demonstrate the working examples followed by our observations. To show the accuracy and the applications of the proposed method, we present the results for a variety of target scenes, including footage from commercial Hollywood movies and 3D video games.

Keywords Matting and compositing · Image-based rendering · Computer vision · Shadow matte

1 Introduction

Matting and compositing are important operations in the production of special effects [7]. During matting, foreground objects are extracted from a single image or video sequence. During compositing, the extracted foreground objects are placed over novel background images. In the film industry, this process is time-consuming, especially for video sequences. Recently, many techniques have been proposed aiming to improve the matting quality or to automate the process. Typical methods for pulling alpha mattes are blue screen matting [20] and natural image/video matting [1, 2, 7, 14, 18, 22]. While these methods have proven to be quite successful in pulling mattes in difficult natural backgrounds, they focus on the matting issues and simply apply alpha blending in the composition process without dealing with the effects of shadows. However, shadows provide important visual cues for depth, shape, contact, movement, and lighting in our perception of the world [11, 12].

In other efforts, shadows for compositing are typically either created manually or composited using matted shadows from the source scenes. The first kind of approach is commonly called “faux shadow” in the film industry [25]. In this technique, artists use the foreground object’s alpha matte to create its shadow manually. Consequently, the geometric accuracy of the “faux shadow” highly depends on the experience of the artists, and the color characteristics of the “faux shadow” are interactively adjusted by the compositor. In the recent work by [17], a semiautomatic method for creating shadow mattes in cel animation is presented. Their system creates shadow mattes based on hand-drawn characters, given high-level guidance from the user regarding the depths of various objects. The method employs a scheme for “inflating” a 3D figure based on hand-drawn art. It provides simple tools for adjusting object depths, coupled with an intuitive interface by which the user specifies object shapes and the relative positions in a scene. Their system obviates the tedium of drawing shadow mattes by hand, and provides control over complex shadows falling over interesting shapes. However, the method has difficulties in obtaining models in matting and compositing operations. The second kind of approach is to extract shadows from the source scenes using luma keying or alpha matting [7, 20], which are efficient when both the target and source scenes are accessible. The accessibility here means that the target scene setup is controllable and known, i.e. we are able to obtain the information about the camera and light source. Nevertheless, these methods would not simply apply to our case since we have only a single perspective view of the target scene, e.g. a previously captured image or a frame in a commercial DVD, which is inaccessible.

In this paper, we introduce a new framework for compositing that aims to relieve the above limitations for a single view of target scenes. Toward this goal, we show how to extract both geometric and photometric constraints, that are necessary for correct and realistic shadow synthesis, from a single view of a target scene. Particularly, we focus on the exploration of (i) the constraints provided by imaged scene structure for camera calibration and thus explicit determination of the locations of the camera and the light source; (ii) the planar homology, which models the imaged shadow relations when camera calibration is not possible; and (iii) the photometric constraints that are required to match the color characteristics of the synthesized shadows with those of the original scene. There are three advantages of this framework: first, this is a pragmatic framework for a single view case where the auto-calibration based on motion alone is impossible and where, however, the ubiquitous presence of buildings, vertical objects and other man-made structures often provides sufficient geometric constraints for the determination of the correct shadow location of an inserted object. Second, the framework is flexible in that various techniques suitable for different scenes can be easily integrated in it. Third, it is simple and easy to implement.

We begin the paper with an introduction of the background material related to the projective geometry in Sect. 2. In Sect. 3, we explore the geometric constraints when the camera calibration is possible, and when it is not. Section 4 describes the photometric constraint to match the color characteristics of the synthesized shadows as those of the original scene. We then demonstrate in Sect. 5 the results of our method applied to various real images and applications in film production. Finally, Sect. 6 concludes this paper with observations and proposed areas of future work.

2 Background

2.1 Pinhole camera model

A pinhole camera, based on the principle of collinearity, projects a region of $\mathbb{R}^3$ lying in front of the camera into a region of the image plane $\mathbb{R}^2$. Consequently, a 3D point $\boldsymbol{M} = [X\ Y\ Z\ 1]^T$ and its corresponding projection $\boldsymbol{m} = [u\ v\ 1]^T$ in the image plane are related by a 3×4 matrix $\boldsymbol{P}$ as

$$\boldsymbol{m} \sim \underbrace{\boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]}_{\boldsymbol{P}}\boldsymbol{M}, \quad \boldsymbol{K} = \begin{bmatrix} f & \gamma & u_0 \\ 0 & \lambda f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where $\sim$ indicates equality up to multiplication by a nonzero scale factor, $\boldsymbol{R}$ is the 3×3 orthonormal rotation matrix, $\boldsymbol{t} = -\boldsymbol{RC}$, with $\boldsymbol{C} = [C_x\ C_y\ C_z]^T$ representing the coordinates of the camera center in the world coordinate frame, is the translation vector, and $\boldsymbol{K}$ is a nonsingular 3×3 upper triangular matrix containing the five camera intrinsic parameters: the focal length f, the aspect ratio λ, the principal point $(u_0,\ v_0)$ and the skew factor γ accounting for nonrectangular pixels.

2.2 Image of the absolute conic

The *Image of the Absolute Conic* [10], $\boldsymbol{\omega}$, is an imaginary point conic directly related to the camera internal matrix, $\boldsymbol{K}$, as $\boldsymbol{\omega} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$, which can be expanded as:

$$\boldsymbol{\omega} \sim \begin{bmatrix} 1 & -\frac{\gamma}{f\lambda} & \frac{\gamma v_0 - \lambda f u_0}{f\lambda} \\ * & \frac{f^2+\gamma^2}{f^2\lambda^2} & -\frac{\gamma^2 v_0 - \gamma\lambda f u_0 + v_0 f^2}{f^2\lambda^2} \\ * & * & \frac{v_0^2(f^2+\gamma^2) - 2\gamma v_0 \lambda f u_0}{f^2\lambda^2} + f^2 + u_0^2 \end{bmatrix}, \tag{2}$$

where the lower triangular elements are denoted by “$*$” to save the space, since $\boldsymbol{\omega}$ is symmetric. Instead of directly determining $\boldsymbol{K}$, it is possible to first compute $\boldsymbol{\omega}$ and, then, compute $\boldsymbol{K}$ uniquely as described in Sect. 3.1.

2.3 Planar homology

In the error-free case, imaged shadow relations (illuminated by a point light source) are modeled by a *planar homology* [21] (see Fig. 1). A planar homology is a planar projective transformation that has a line of fixed points,

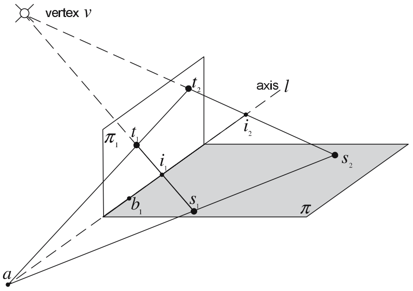


Fig. 1. Geometrically, a planar object, $\boldsymbol{\pi}_1$, and its shadow (cast on a ground plane $\boldsymbol{\pi}$) are related by a planar homology

called the *axis*, and a distinct fixed point $\boldsymbol{v}$, not on the axis $\boldsymbol{l}$, called the *vertex* of the homology.

In our case, the vertex $\boldsymbol{v}$ is the image of the light source, and the axis, l, is the image of the intersection between planes $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}$. Under this transformation, points on the axis are mapped to themselves. Each point off the axis, e.g. $\boldsymbol{t}_2$, lies on a fixed line $\boldsymbol{t}_2\boldsymbol{s}_2$ through $\boldsymbol{v}$ intersecting the axis at $\boldsymbol{i}_2$ and is mapped to another point $\boldsymbol{s}_2$ on the line. Note that $\boldsymbol{i}_2$ is the intersection in the image plane, although the light ray $\boldsymbol{t}_2\boldsymbol{s}_2$ and the axis, $\boldsymbol{l}$, are unlikely to intersect in the 3D world.

One important property of a planar homology is that the corresponding lines, i.e. lines through pairs of corresponding points, intersect with the axis: for example, the lines $\boldsymbol{t}_1\boldsymbol{t}_2$ and $s_1 s_2$ intersect at $\boldsymbol{a}$, a point on the axis $\boldsymbol{l}$. Another important property of a planar homology is that the cross ratio defined by the vertex, $\boldsymbol{v}$, the corresponding points, $\boldsymbol{t}_i$ and s_i, and the intersection, $\boldsymbol{i}_i$, of the line $\boldsymbol{t}_i\boldsymbol{s}_i$ with the axis, is the characteristic invariant of the homology, and is the same for all corresponding points. For example, the cross ratios of the four points $\{\boldsymbol{v}, \boldsymbol{t}_1;\ \boldsymbol{s}_1, \boldsymbol{i}_1\}$ and $\{\boldsymbol{v}, \boldsymbol{t}_2;\ \boldsymbol{s}_2, \boldsymbol{i}_2\}$ are equal.

3 Geometric constraints

In this section, we first describe how to determine the position and orientation of the light source and camera when the imaged scene structure provide enough calibration constraints, which is called "strong geometric constraints". Then we show it is still likely to generate the shadow of an inserted object, provided that it is planar or distant, when the camera calibration is not possible. We refer to this constraint as "weak geometric constraint". For each case, we give some working examples followed by discussions.

3.1 Strong geometric constraints

The orthogonal vanishing points, the intersections of imaged parallel lines, are ubiquitously present in man-made structures and can be used for the camera calibration [4, 5, 15]. For example, given a typical frame (Fig. 2a), we are able to extract three sets of parallel lines for camera parameter estimation shown in red, green and blue, respectively.

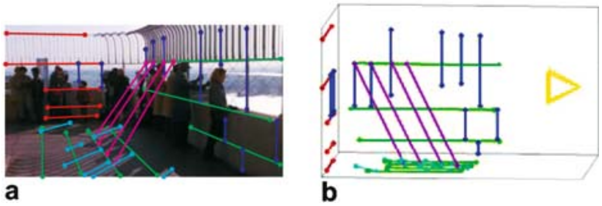


Fig. 2. a One frame of the movie "Sleepless in Seattle" (1993). The extracted feature lines are plotted by five different colors: red, green and blue lines are along X, Y, and Z directions, respectively in the 3D world coordinate system, cyan lines are shadow lines of some spiked fence, and magenta lines are along the light source direction. **b** The recovered 3D scene, where the yellow square pyramid on the right indicates the camera location that captures **a**

Therefore, we can compute the three mutually orthogonal vanishing points, denoted as $\boldsymbol{v}_x$, $\boldsymbol{v}_y$ and $\boldsymbol{v}_z$, along the world X-axis, Y-axis and Z-axis, respectively, by intersecting the image projections of the lines along each direction. Since $\boldsymbol{v}_x$, $\boldsymbol{v}_y$ and $\boldsymbol{v}_z$ are along mutually orthogonal directions, we have three linear equations in term of $\boldsymbol{\omega}$ in Eq. 2:

$$\boldsymbol{v}_x^T\boldsymbol{\omega}\boldsymbol{v}_y = 0,\ \ \boldsymbol{v}_x^T\boldsymbol{\omega}\boldsymbol{v}_z = 0,\ \ \boldsymbol{v}_y^T\boldsymbol{\omega}\boldsymbol{v}_z = 0. \tag{3}$$

In addition to the calibration constraints provided by the imaged structure, other constraints can be obtained from the priors of a normal camera used in graphics, including zero skew, $\gamma = 0$, and known aspect ratio λ, usually 1 (square), 1.2 (widescreen) or 0.9,

$$\omega_{12} = 0,\ \ \omega_{22} = 1/\lambda^2, \tag{4}$$

where ω_{ij} denotes the element in thi^{th} row and j^{th} column of $\boldsymbol{\omega}$ in Eq. 2. The five linear constraints in Eq. 3 and Eq. 4 are sufficient to solve the five unknowns in $\boldsymbol{\omega}$. After the camera calibration, we can compute the camera internal and external parameters, i.e. the projection matrix $\boldsymbol{P}$ in Eq. 1, up to a scale as shown in [4]. The scale is related to the camera location, and can be eliminated as follows. The fourth column, $\boldsymbol{p}_4$, of $\boldsymbol{P}$ is nothing but the projection of the 3D world origin, $[0\,0\,0\,1]^T$, since (denoting the i^{th} column of $\boldsymbol{P}$ as $\boldsymbol{p}_i$),

$$[\boldsymbol{p}_1\,\boldsymbol{p}_2\,\boldsymbol{p}_3\,\boldsymbol{p}_4][0\,0\,0\,1]^T = \boldsymbol{p}_4.$$

Without loss of generality, we use the corner of visible walls, at image location (313.4, 206.4), as the imaged world origin for the view shown in Fig. 2a. In other words, $\boldsymbol{p}_4 = \alpha[313.4\ 206.4\ 1]^T$, where α is the similarity ambiguity. If we specify the height of the upper most green line from the ground plane as the unit distance, which removes the unknown similarity ambiguity α, the computed camera location is (1.83, 8.13, 0.96).

Finally, the partially reconstructed scene of the image Fig. 2a is shown in Fig. 2b. Note that we do not need to fully reconstruct the target scene for image compositing applications since it is already present in the existing image. Note also that we reconstruct the Fig. 2b using the planar homographies that map the world planes to the image planes, since the traditional triangulation method [10] would not work in our case due to the fact that there is only a single view. For example, we compute the planar homography, $\boldsymbol{H}_z$ that maps the world plane $Z = 0$, i.e. the ground plane, to the image plane as $\boldsymbol{H}_z = [\boldsymbol{p}_1\ \boldsymbol{p}_2\ \boldsymbol{p}_4]$.

The imaged light source position, $\boldsymbol{v}$, can be computed by intersecting the images of the parallel lines along the light source direction, e.g. the magenta lines in Fig. 2a. Note that $\boldsymbol{v}$ is not visible in Fig. 2a. Since the target scene in Fig. 2 is illuminated by an infinite light source (the sunlight), the angles ϕ_x (ϕ_y and ϕ_z, respectively) between the light source direction and the world X-axis (Y and Z-axis, respectively) can be computed as,

$$\phi_j = \cos^{-1} \frac{\boldsymbol{v}_j^T \boldsymbol{\omega} \boldsymbol{v}}{\sqrt{\boldsymbol{v}^T \boldsymbol{\omega} \boldsymbol{v}} \sqrt{\boldsymbol{v}_j^T \boldsymbol{\omega} \boldsymbol{v}_j}}, \ j \in \{x, y, z\}. \tag{5}$$

For the image in Fig. 2, the computed angles are $\phi_x = 119.6°$, $\phi_y = 138.9°$ and $\phi_z = 64.4°$.

The second computation example is the single view shown in Fig. 3, available from the University of Washington. The same process for the camera calibration and the computation of $\boldsymbol{P}$ as described above can be simply applied using the feature lines in Fig. 3a. The difference is that Eq. 5 can not be used to compute the light source orientation since it is difficult to identify more than one line along the light source direction and, consequently, is unlikely to compute the imaged light source, $\boldsymbol{v}$, from a single view shown in Fig. 3. In this case, however, we can still compute the light source orientation by using two feature points along the lighting direction: $\boldsymbol{t}$ on the person's head and its cast shadow position $\boldsymbol{s}$ on the ground shown in Fig. 3b. Without any difficulty, we can identify the corresponding bottom point $\boldsymbol{b}$ of $\boldsymbol{t}$ on the ground plane ($X - Y$ plane), such that it has the same 3D X and Y coordinates as $\boldsymbol{t}$. Then, we compute the points, $\boldsymbol{b}$ and $\boldsymbol{s}$, in the world coordinate system as

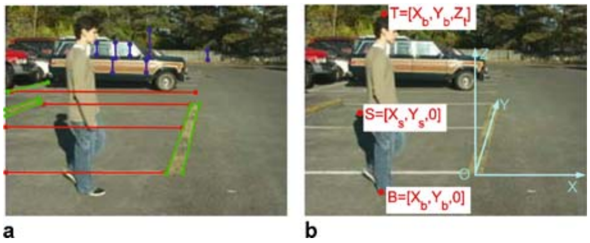


Fig. 3. **a** A view with the extracted feature lines for camera calibration plotted in three different colors, each of which has the same definition as that in Fig. 2. **b** The 3D world coordinate system and the 3D coordinate values, denoted by corresponding uppercase characters, of three image feature points ($\boldsymbol{t}$, $\boldsymbol{b}$ and $\boldsymbol{s}$) used to compute the orientation of the light source

$$[X_b\ Y_b\ 1]^T \sim \boldsymbol{H}_z^{-1} \boldsymbol{b}, \quad [X_s\ Y_s\ 1]^T \sim \boldsymbol{H}_z^{-1} \boldsymbol{s}.$$

The Z coordinate, Z_t, of the feature point $\boldsymbol{t}$ is then estimated from the equation (two equations, one unknown), $\boldsymbol{t} \sim \boldsymbol{P}[X_b\ Y_b\ Z_t\ 1]^T$. For the distant light source, the sunshine, two 3D points, $\boldsymbol{t}$ and $\boldsymbol{b}$, along the lighting direction are enough to give us light source direction as $[X_b - X_s,\ Y_b - Y_s,\ Z_t]^T$.

Before we end this section, we want to make a few observations. First, it might be difficult to approximate the camera pose and lighting direction by simply looking at shadow lines in a perspective view. For example, it is difficult to tell the angle in 3D world, β, between the cyan shadow lines (Fig. 2a) and the red lines, since the imaged shadow lines intersect at a finite point, (678.8, 82.5), and hence each of them gives different values of that angle β. However, our method is able to compute β as 56.3°. Second, single view calibration is necessary in most cases, even for a clip from a commercial movie, in that the static shot is the basic camera shot of all motion pictures. An example of this would be the original shot at time instance around 01:28:30 containing the frame (Fig. 2). The last remark is that the new findings of camera calibration objects, such as parallel shadows [4] and co-planar circles [6], could be easily integrated into this framework, although we do not give examples here.

3.2 Weak geometric constraints

For the cases where single view camera calibration and the explicit geometric light source estimation are not possible, we utilize a relatively weak constraint, the planar homology, which makes it possible to synthesize the correct shadows of an inserted object if the object is planar or distant.

One example target scene is shown in Fig. 4a, and the computation process is as follows. We first choose three pairs of corresponding points under a planar homology, e.g. $\langle \boldsymbol{t}_2, \boldsymbol{s}_2 \rangle$, $\langle \boldsymbol{t}_1, \boldsymbol{s}_1 \rangle$ and $\langle \boldsymbol{b}_1, \boldsymbol{b}_1 \rangle$ as input. Then, we compute the planar homology, $\boldsymbol{H}$, directly as [10]:

$$\boldsymbol{H} = \boldsymbol{I} + (\mu - 1) \frac{\boldsymbol{v} \boldsymbol{l}^T}{\boldsymbol{v}^T \boldsymbol{l}}, \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix, μ, $\boldsymbol{v}$ and $\boldsymbol{l}$ are given as

$$\boldsymbol{v} = (\boldsymbol{t}_2 \times \boldsymbol{s}_2) \times (\boldsymbol{t}_1 \times \boldsymbol{s}_1),$$
$$\boldsymbol{l} = ((\boldsymbol{t}_2 \times \boldsymbol{t}_1) \times (\boldsymbol{s}_2 \times \boldsymbol{s}_1)) \times \boldsymbol{b}_1,$$
$$\mu = \{\boldsymbol{v}, \boldsymbol{t}_1;\ \boldsymbol{s}_1, \boldsymbol{i}_1\}.$$

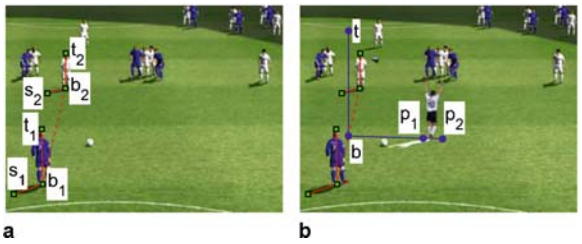


Fig. 4a,b. Example to find the correct shadow pixels of an inserted real soccer player into a snapshot of the game FIFA 2003. **a** shows three pairs of corresponding points under a planar homology, i.e. $\langle \boldsymbol{t}_2, \boldsymbol{s}_2\rangle$, $\langle \boldsymbol{t}_1, \boldsymbol{s}_1\rangle$ and $\langle \boldsymbol{b}_1, \boldsymbol{b}_1\rangle$. The point $\boldsymbol{b}_2$ is used for the computation of the vertical vanishing point, but not used for computing $\boldsymbol{H}$. **b** plots the the computed shadow pixels, marked in white, of the inserted object

Note that $\boldsymbol{i}_1$ is the intersection in the image plane, although the light ray $\boldsymbol{t}_1\boldsymbol{s}_1$ and the axis, $\boldsymbol{l}$, are unlikely to intersect in the 3D world.

Now we describe how to determine the correct shadow position of the inserted real player in Fig. 4b, which is matted from Fig. 11. Obviously, the distant standing person can be approximated as a planar object in some vertical plane, denoted by $\boldsymbol{\pi}_2$ as shown in Fig. 5. We have already computed the planar homology, $\boldsymbol{H}$, from the plane $\boldsymbol{\pi}_1$ to $\boldsymbol{\pi}$ and, hence, its vertex $\boldsymbol{v}$ and axis $\boldsymbol{l}_1$. The new homology, $\boldsymbol{H}_2$, that maps the points on the plane $\boldsymbol{\pi}_2$ to their shadows on plane $\boldsymbol{\pi}$, has the same vertex $\boldsymbol{v}$ as that of $\boldsymbol{H}$. The axis, $\boldsymbol{l}_2$, of $\boldsymbol{H}_2$ is the intersection of the plane $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}$, and can be found by specifying two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ on the intersection, i.e. $\boldsymbol{l}_2 = \boldsymbol{p}_1 \times \boldsymbol{p}_2$. The manually specified points are shown in Fig. 4b. Denote by $\boldsymbol{b}$ the intersection of two axis $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$, and by $\boldsymbol{v}_y$ the vertical vanishing point. The $\boldsymbol{v}_y$ can be computed by intersecting two vertical objects, such as $\boldsymbol{t}_2\boldsymbol{b}_2$ and $\boldsymbol{t}_1\boldsymbol{b}_1$ in Fig. 4a. We then randomly choose one point $\boldsymbol{t}$ on the vertical line $\boldsymbol{l}_v$ passing through $\boldsymbol{b}$ and $\boldsymbol{v}_y$. Finally, the $\boldsymbol{H}_2$ is computed as:

$$\boldsymbol{H}_2 = \boldsymbol{I} + (\mu_2 - 1)\frac{\boldsymbol{v}\boldsymbol{l}_2^T}{\boldsymbol{v}^T\boldsymbol{l}_2},$$

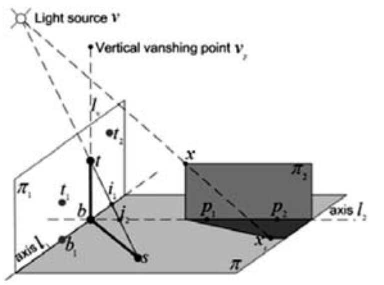


Fig. 5. The geometry for computing the shadow positions for any points on plane $\boldsymbol{\pi}_2$, given the planar homology, $\boldsymbol{H}$, that relates the planar object, $\boldsymbol{\pi}_1$, and its shadow (cast on a ground plane $\boldsymbol{\pi}$)

where

$$\mu_2 = \{\boldsymbol{v}, \boldsymbol{t}; \boldsymbol{Ht}, (\boldsymbol{t} \times (\boldsymbol{Ht})) \times \boldsymbol{l}_2\}.$$

Note that the four points should be scaled to inhomogeneous coordinates to compute μ_2. Consequently, given any point, $\boldsymbol{x}$, in the plane $\boldsymbol{\pi}_2$, it is easy to compute its shadow position, $\boldsymbol{x}_s$, on the plane, $\boldsymbol{\pi}$, by simply applying the 2D transformation, $\boldsymbol{x}_s \sim \boldsymbol{H}_2\boldsymbol{x}$. The computed shadow pixels corresponding to the inserted object in Fig. 4b are marked as white.

The technique described in this section is mostly related to the popular technique, commonly called "faux shadow" in the film industry [25], for which artists also use the foreground object's alpha matte to create its shadow by warping or displacement-mapping the shadow. Compared to "faux shadow" created by hand, however, the proposed approach has two advantages. First, our method models the imaged shadow relations by the planar homology and is able to obtain geometrically correct shadow positions, while the geometric accuracy of the "faux shadow" highly depends on the experience of the compositors. Second, our method infers the most likely rendered shadows from the original shadows in the target scene, while color characteristics of the "faux shadow" are manually adjusted by the compositor. This photometric constraint is described in the next section. In addition, the proposed method is simple and easy to implement. The major interactions consist of specifying three pairs of points $\langle \boldsymbol{t}_2, \boldsymbol{s}_2\rangle$, $\langle \boldsymbol{t}_1, \boldsymbol{s}_1\rangle$ and $\langle \boldsymbol{b}_1, \boldsymbol{b}_1\rangle$ as shown in Fig. 4a, and two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ as shown in Fig. 4b. Notice that the above given examples are for vertical objects, since we observe that vertical objects are ubiquitous in the real world, and also typically we are interested in inserting a new actor, which is mostly standing and thus again is vertical, into some target scene.

4 Photo-realistic constraints

While the geometric constraints help us to place shadows and reflections at correct positions, we also need to match the color characteristics of the shadows as those of the real scene. In order to create visually realistic shadows over the target image, we enforce the shading image values [3] of the synthesized shadows of the inserted objects to be the same as those of shadows cast by the existing objects in the original target image. The shading image (or illumination image), $\boldsymbol{S}(x, y)$, together with the reflectance image, $\boldsymbol{I}_{unshadow}(x, y)$, are called the intrinsic images. Generally, the observed image, $\boldsymbol{I}_{shadow}(x, y)$, can be modelled as the product of these two intrinsic images:

$$\boldsymbol{I}_{shadow}(x, y) = \boldsymbol{S}(x, y)\boldsymbol{I}_{unshadow}(x, y). \tag{7}$$

Therefore, our problem reduces to recovering the shading image, $\boldsymbol{S}(x, y)$, from the input image $\boldsymbol{I}_{shadow}(x, y)$.

Recently, many approaches [9, 23, 24] have been proposed to derive illumination image and reflectance image from either a single image or a video of an object under different illumination conditions. Theoretically, those decomposed light maps could be used as shadow mattes in our work. However, the method [24] involves recording a sequence of images of a fixed outdoor scene over the course of a day, while the strategy [23] requires a trained classifier that must incorporate the knowledge about the structure of the surface in the target scene and how it appears when illuminated. Therefore, these are not practical for our case, since we are provided with only a single view of the target scene, which is inaccessible. On the other hand, the more recent method [9] aims to recover intrinsic images by entropy minimization from a single image, but is based on the assumptions, e.g. narrow-band (or spectrally-sharpened) sensors and Planckian lights, and is not able to handle the compression effects such as JPEG effects. Nevertheless, our challenge is compositing objects into a given image or a frame of a video, which are typically compressed.

Our approach takes advantage of the property that changes in color between pixels indicate either reflectance changes or shading effects [23]. In other words, it is unlikely that significant shading boundaries and reflectance edges occur at the same point. Therefore, we make the assumption that every color change along the shadow boundaries, the edges caused by illumination difference only (e.g. Fig. 6b), is caused by shading only, i.e. the reflectance image colors across the shading boundaries should be the same or similar. In practice, considering the gradual change along the normal direction of the shadow boundaries, due to either compression effects or soft shadows, the input image pixel value, $\boldsymbol{I}_{shadow}(x, y)$, and the reflectance image pixel value, $\boldsymbol{I}_{unshadow}(x, y)$, of boundary pixel, (x, y), are obtained as

$$\boldsymbol{I}_{shadow} = median\{\boldsymbol{I}_{shadow}(m, n) : (m, n) \in \mathcal{N}_i\}, \quad (8)$$

$$\boldsymbol{I}_{unshadow} = median\{\boldsymbol{I}_{shadow}(m, n) : (m, n) \in \mathcal{N}_o\}, \quad (9)$$

where $\mathcal{N}_i$ and $\mathcal{N}_o$ are subsets of the set of neighbor pixels of (x, y), and the subscripts denote whether the pixels are inside ($\mathcal{N}_i$) or outside ($\mathcal{N}_o$) of the shadows. Previous methods also use color differences on two sides of a shadow boundary for estimating the influence of an illumination source [13, 19]. In our implementation, we use a 7×7 neighborhood, e.g. the pixels bounded by the red box in Fig. 6a. From Eq. 8, we can compute the shading image value as, $\boldsymbol{S}(x, y) = \boldsymbol{I}_{shadow}(x, y)/\boldsymbol{I}_{unshadow}(x, y)$, for each pixel (x, y) along the shadow boundaries. Notice that for each color channel (red, green and blue), $\boldsymbol{S}$ is computed independently.

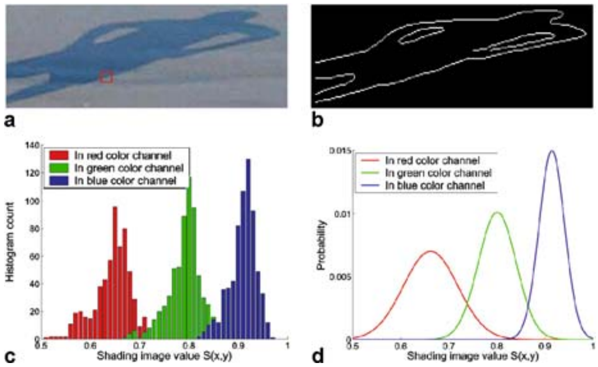


Fig. 6. **a** A shadow patch in a target scene shown in Fig. 10b. **b** The shadow boundary detected by Canny edge detector. **c** The histograms and **d** the fitted Gaussian distributions of the computed shading image values, for all pixels along the shadow boundary

For example, the computed $\boldsymbol{S}(x, y)$ along the boundaries of the shadow map (Fig. 6b) is plotted in Fig. 6 (c,d). The interesting observation is that the changes in a color image due to shading affect three color channels disproportionally. Obviously, the shading affects the red color channel the most and the blue color channel the least. This coincides with the observations in [16] that shadow pixels appear more "blueish". While there are some richer models to model this effect, we simply use different shading image values along three color channels to approximate the effect, i.e. $\boldsymbol{S} = diag(\beta_R, \beta_G, \beta_B)$. The $\boldsymbol{S}$ matrix is assumed approximately constant, and computed using the median value of all computed $\boldsymbol{S}(x, y)$ for pixels (x, y) along the shadow boundary. For each computed shadow pixel, (u, v), of the inserted object (e.g. white pixels in Fig. 4b), we are able to compute its pixel value after shading as

$$\boldsymbol{I}_{shadow}(u, v) = diag(\beta_R, \beta_G, \beta_B)\boldsymbol{I}_{unshadow}(u, v).$$

In the scene in Fig. 6, the computed shading image values are $\beta_R = 0.67$, $\beta_G = 0.78$ and $\beta_B = 0.91$. Provided that the ground surface is locally flat and partially under shadow, which is mostly true in the real world, our experiments show that this approximation works well.

5 Results

The proposed method has been tested on an extensive set of target scenes. The shown target scenes vary from frames in commercial movies, frames in videos available on the internet, snapshots in 3D video games to images taken by the authors. In Sect. 5.1, we apply our method to two target images that can be calibrated. Then, in Sect. 5.2, we demonstrate the performance of our method on target images where strong geometric constraints are not available. Finally, we show the applicability of our method to film production in Sect. 5.3.

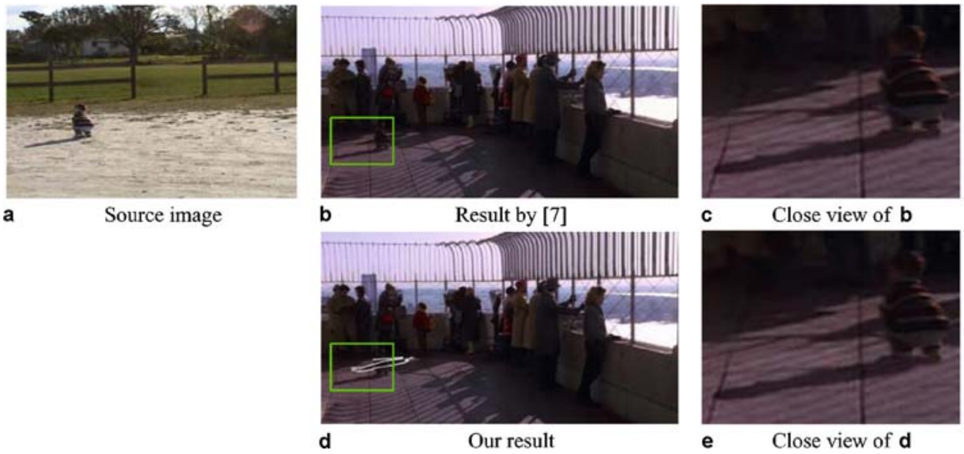


Fig. 7a–e. The comparison of our method with the shadow matting method proposed by [7]. We aim to composite the child in **a** into one frame from the commercial movie "Sleepless in Seattle" (1993), as shown in Fig. 2

5.1 Scenes where strong geometric constraints are available

The first target frame is from the commercial movie "Sleepless in Seattle" (1993), as shown in Fig. 2. We first compute the positions and orientations of the camera and the light source using the method described in Sect. 3.1. Then, we use the shadow edges marked by white lines shown in Fig. 7d to obtain the shading image values ($\beta_R = 0.48$, $\beta_G = 0.43$ and $\beta_B = 0.46$). The color characteristics of our synthesized shadow in Fig. 7d and zoomed in (e) is comparable to that by [7] in (b) and (c). However, our result is obtained by using a single frame, while their method involves 512 frames, from which we find the darkest and brightest value at each pixel as shown in Fig. 8. In addition, it is difficult for [7] to ensure that the relationship of the light source, reference plane, and camera in the source environment is the same as that in the target due to the potential perspective distortions. Note that we multiply the R, G and B values of each pixel of the composited foreground object, the child, by manually specified constant scales (0.50, 0.38 and 0.38) to match the intensity differences between the source and target image and the "reddish" effects in the target scene.

In the second experiments, we aim to expand a small statue (Fig. 9a) into the target image (Fig. 3). It is based on the computed relative position and orientation with its principal axis coinciding with the computed lighting direction. The result shown in (c) demonstrates that our method is able to synthesize shadows with correct geometric relationship and realistic color characteristics. We use the shadow edges marked by red lines shown in Fig. 9c to obtain the shading image values ($\beta_R = 0.34$, $\beta_G = 0.42$ and $\beta_B = 0.51$). Note that we multiply the intensity of each pixel of the composited foreground object, the small statue, by constant scales 1.28 to match the intensity differences between the source and target image.

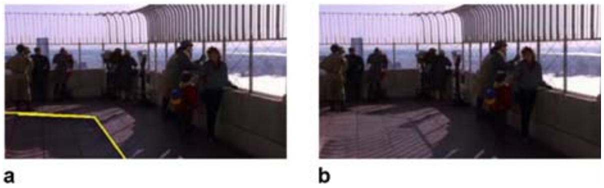


Fig. 8a,b. The lit and shadow images of the *Seattle sequence*. Note that we are only interested in areas where the inserted foreground might cast shadows, which in our case is located on the left bottom separated by the yellow lines. **a** Target shadow image **b** Target lit image

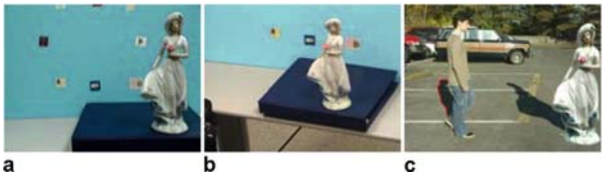


Fig. 9a–c. We expand a small statue **a** into the target shown in Fig. 3. **a** The image from the camera's view point. **b** The image taken by a camera whose principal axis coincides with the computed lighting direction. **c** shows the final composite with convincing shadows obtained by our new compositing method

5.2 Scenes where only weak geometric constraints are available

We also created shadows for planar objects, e.g. a parking sign post as shown in Fig. 10. The input feature points, including three pairs of points that are corresponding under a planar homology and one point used for the computation of the vertical vanishing point, are plotted in the same color in Fig. 10b as those in Fig. 4a. In this experiment, we demonstrate the advantage of using different shading image values along each color channel. Our fi-

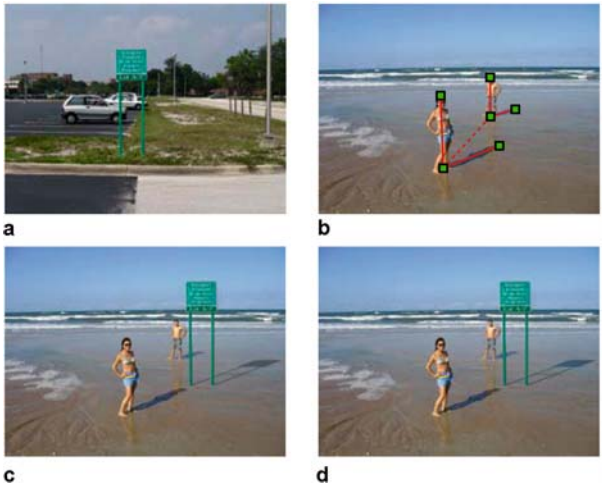


Fig. 10. c The result by assuming that any changes in a color image due to shading should affect all three color channels proportionally. **a** Source image. **b** Target image. **d** Our result

nal result shown in Fig. 10d is noticeably more realistic than the result in (c) in that shadow regions are illuminated by the sky, and the sky is assumed to be blue and the only source of illumination on shadowed regions [16]. The shading image for (d) is computed in Sect. 4, while for the result (c) we use the intensity image and compute the shading image value as $\beta_R = \beta_G = \beta_B = 0.72$.

For all of the above experiments, the light source is the sunshine. We also demonstrate our method for the synthetic light source from the snapshot of a 3D video game, plotted in Fig. 4. No matter whether the light source is finite or not, our method is able to synthesize the correct and realistic shadow for an inserted real player, shown in Fig. 11. Note that the very impressive shadows are generated even for the raised hand of the real player. The weak geometric constraint is computed in Sect. 3.2. We use the shadow edges marked by red lines shown in Fig. 11, right, to obtain the shading image values ($\beta_R = \beta_G = \beta_B = 0.50$).

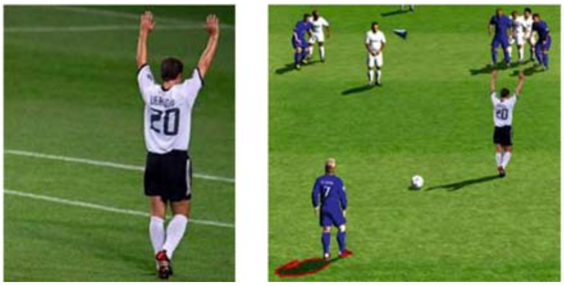


Fig. 11. The performance of our method for the synthetic light source from the snapshot of a 3D video game as shown in Fig. 4. Left: the source image. Right: a zoomed view of our result

5.3 Application in film production

To demonstrate the strength of our method, we apply it on two commercial Hollywood movies: "Sleepless in Seattle" (Fig. 12) and "The Pianist" (Fig. 13). The camera and light source parameters of "Sleepless in Seattle" are recovered as described in Sect. 3.1. For the "The Pianist", we first calibrate the camera using the feature lines shown in Fig. 14a. To recover the light source geometry, we click twelve correspondences, $\boldsymbol{t}$, $\boldsymbol{s}$ and $\boldsymbol{b}$ as shown in Fig. 14b and c, in the first twelve frames of the original clip.

Based on the analysis of the target scenes, we observe that the light source of the "Seattle sequence" is typical daytime sunlight, and the camera is at a location above the ground plane at a height of a typical person. We capture the source videos at some park with similar lighting condition using an off-the-shelf Sony video camera off the shelf. For the "Pianist sequence", it is very difficult to place a light at the computed light source position and then transfer the shadows extracted from the camera view to the target video, since, in order to do this, one would need a very dark (otherwise double shadows might appear) and huge studio (around 100 meters according to our computation) to ensure the geometry match as the target backgrounds. In our case, alternatively, we place two video

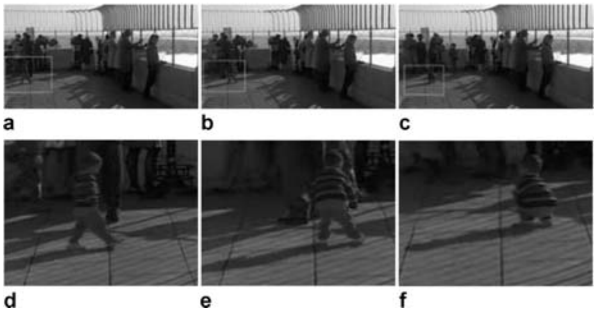


Fig. 12a–f. Three example frames **a**, **b** and **c** of our composite of the *Seattle sequence*. The bottom row plots the zoomed in views of the frames above them

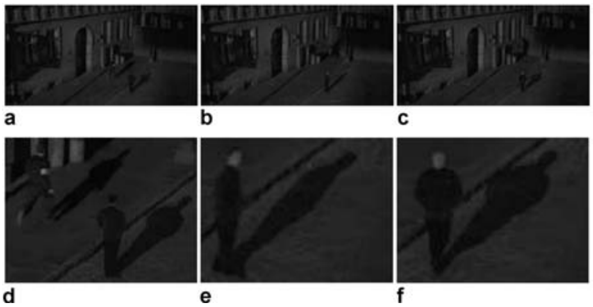


Fig. 13a–f. Three example frames **a**, **b** and **c** of our composite of *The Pianist*. The bottom row plots the zoomed in views of the frames above them

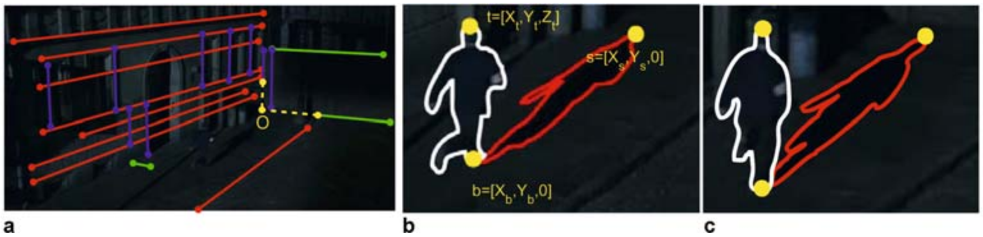


Fig. 14. **a** is one frame from the movie *The Pianist*, with the extracted feature lines for camera parameters estimation plotted in three different colors with the same definitions as that in Fig. 2a. The two yellow dashed lines intersect at a point, O, which is specified as the image of the world coordinate origin. **b** & **c** are two close views of two frames with extracted features ($\boldsymbol{t}$, $\boldsymbol{b}$ and $\boldsymbol{s}$) used to compute the light source geometry

cameras on the 3^{rd} and 2^{nd} floors of a parking garage to capture the videos that would be seen from the original camera and light source. In other words, the shadow map [8] in 3D graphics is implemented using a real camera and applied in film production.

In both clips, our method successfully composites not only the foreground objects, but also its geometrically correct and visually convincing shadows into the commercial movies without special setup. The results are shown in Fig. 12 and Fig. 13. We treat the case where the shadows of the added sequence overlap with the original shadows in the original sequence as follows. When the inserted objects cast shadows on some existing shadow areas, we retain the original appearance. The existing shadow areas can be computed based on the comparison of the input frame and the target shadow image shown in Fig. 8a.

6 Conclusion and future work

This paper presents a framework for rendering shadows of objects composited into a single target view. This method is especially useful when the target scene is not accessible, e.g a given image or frame from a commercial movie. We explore both the geometric and photometric constraints, and utilize them for correct and realistic shadow synthesis. The experimental results demonstrate that this method is efficient and can be applied to a variety of target scenes.

Compared to the shadow matting methods, our framework is able to handle cases where one aims to transfer objects into inaccessible scenes and requires only a single view. In addition, our framework has advantages over the traditional "faux shadow" in that it increases the geometric accuracy and visual reality of the synthesized shadows. As a pragmatic and flexible framework, it is also simple and easy to implement.

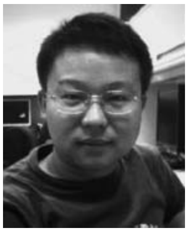
Our method has a number of important restrictions. First, the alpha blending of shadows is an approximation and only valid for scenes with one dominant, point-like light source, and it does not model potentially complex effects arising from interreflections. Second, we require the target background to be planar. Despite these restrictions, we believe that in many settings the shadows created by our approach are more plausible than the manually generated "faux shadow". Traditional shadow generating methods typically share our requirement for a single matched key light source and have difficulties to model complex effects arising from interreflections. However, they cannot synthesize a geometrically correct silhouette of the shadow when the view from the lighting direction is too far from the camera's viewpoint.

Acknowledgement This material is based upon work funded in part by the U.S. Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government. The authors would like to thank Mr. Yunjun Zhang and Ms. Min Hu for their help in preparing partial results presented in this paper.

References

1. Agarwala, A., A., Salesin, D.H., Seitz, S.M.: Keyframe-based tracking for rotoscoping and animation. ACM Trans. Graph. 23(3), 584–591 (2004)
2. Apostoloff, N.E., Fitzgibbon, A.: Bayesian video matting using learnt image priors. In: Proc. IEEE CVPR, pp. 407–414 (2004)
3. Barrow, H., Tenenbaum, J.: Recovering intrinsic scene character-istics from images. Academic Press (1978)
4. Cao, X., Shah, M.: Camera calibration and light source estimation from images with shadows. In: Proc. IEEE CVPR, pp. 918–923 (2005)
5. Caprile, B., Torre, V.: Using vanishing points for camera calibra-tion. Int. J. Comput. Vision 4(2), 127–140 (1990)
6. Chen, Q., Wu, H., Wada, T.: Camera calibration with two arbitrary coplanar circles. In: Proc. ECCV, pp. 521–532 (2004)
7. Chuang, Y.: New models and methods for matting and compositing. Ph.D. thesis, University of Washington (2004)
8. Crow, F.C.: Shadow algorithms for computer graphics. In: Proc. of SIGGRAPH, pp. 242–248 (1977)
9. Finlayson, G., Drew, M., Lu, C.: Intrinsic images by entropy minimization. In: Proc. ECCV, pp. 582–595 (2004)

10. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
11. Hasenfratz, J., Lapierre, M., Holzschuch, N., Sillion, F.: A survey of real-time soft shadows algorithms. Computer Graphics Forum 22(4), 753–774 (2003)
12. Kersten, D., Mamassian, P., Knill, D.C.: Moving cast shadows induce apparent motion in depth. Perception 26(2), 171–192 (1997)
13. Li, Y., Lin, S., Lu, H., Shum, H.: Multiple-cue illumination estimation in textured scenes. In: Proc. IEEE ICCV, pp. 1366–1373 (2003)
14. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Trans. Graph. 23(3), 303–308 (2004)
15. Liebowitz, D., Zisserman, A.: Combining scene and auto-calibration constraints. In: Proc. IEEE ICCV, pp. 293–300 (1999)
16. Nadimi, S., Bhanu, B.: Physical models for moving shadow and object detection in video. IEEE Trans. Pattern Anal. Mach. Intell. 26(8), 1079–1087 (2004)
17. Petrovic, L., Fujito, B., Williams, L., Finkelstein, A.: Shadows for cel animation. In: Proc. ACM SIGGRAPH, pp. 511–516 (2000)
18. Rother, C., Kolmogorov, V., Blake, A.: grabcut : interactive fore-ground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
19. Sato, I., Sato, Y., Ikeuchi, K.: Stability issues in recovering illumination distribution from brightness in shadows. In: Proc. IEEE CVPR, pp. 400–407 (2001)
20. Smith, A.R., Blinn, J.F.: Blue screen matting. In: Proc. ACMSIG-GRAPH, pp. 259–268 (1996)
21. Springer, C.E.: Geometry and Analysis of Projective Spaces. Free-man (1964) 23. Tappen, M.F., Freeman, W.T., Adelson, E.H.: Recovering intrinsic images from a single image. In: Advances in Neural Information Processing Systems (2002)
22. Sun, J., Jia, J., Tang, C., Shum, H.: Poisson matting. ACM Trans. Graph. 23(3), 315–321 (2004)
23. Tappen, M.F., Freeman, W.T., Adelson, E.H.: Recovering intrinsic images from a single image. In: Advances in Neural Information Processing Systems (2002)
24. Weiss, Y.: Deriving intrinsic images from image sequences. In: Proc. IEEE ICCV, pp. 68–75 (2001)
25. Wright, S.: Digital Compositing for Film and Video. Focal Press (2001)



XIAOCHUN CAO received the BE and ME degrees in computer engineering from Beihang University, Beijing, China, in 1999 and 2002, respectively. He is currently a PhD student at the University of Central Florida. He received the Best Student Paper Award at the International Conference on Pattern Recognition in 2004. His research interests include computer vision, computer graphics and pattern recognition. He is a student member of the IEEE.

YUPING SHEN was born in Zhangzhou, China, in 1982. He received the B.Eng. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2004. He is currently a PhD student in the School of Computer Science, University of Central Florida. His research interests include computer graphics and computer vision. Current research focuses on non-photorealistic rendering and image and video synthesis.

MUBARAK SHAH received the BE degree in 1979 in electronics from Dawood College of Engineering and Technology, Karachi, Pakistan and was awarded a five year Quaid-e-Azam (Father of Nation) scholarship for the PhD degree. He spent 1980 at Philips International Institute of Technology, Eindhoven, The Netherlands, where he completed the EDE diploma. He received the MS and PhD degrees, both in computer engineering, from Wayne State University, Detroit, Michigan in 1982 and 1986, respectively. Since 1986, he has been with the University of Central Florida, where he is currently Agere Chair professor of computer science and the director of the Computer Vision Lab. He is a coauthor of the books Video Registration (2003) and Motion-Based Recognition (1997), both by Kluwer Academic Publishers. He has published more than 150 papers in leading journals and conferences on topics including visual motion, tracking, video registration, and activity and gesture recognition. He has served as a project director for the national site for REU (Research Experience for Undergraduates) in Computer Vision, funded by the US National Science Foundation since 1987. He is a fellow of the IEEE, was an IEEE Distinguished Visitor speaker in 1997-2000. He received the Harris Corporation Engineering Achievement Award in 1999, the TOKTEN awards from UNDP in 1995, 1997, and 2000; the Teaching Incentive Program award in 1995 and 2003, the Research Incentive Award in 2003, and the IEEE Outstanding Engineering Educator Award in 1997. He is an editor of an international book series on video computing, editor-in-chief of Machine Vision and Applications, and an associate editor of Pattern Recognition. He was an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and a guest editor of the special issue of the International Journal of Computer Vision on video computing.

HASSAN FOROOSH is an assistant professor of computer science at the University of Central Florida (UCF). Prior to joining UCF, he was a senior research scientist at the University of California, Berkeley (2000-2002), and prior to that an assistant research scientist at Center for Automation Research, University of Maryland, College Park (1997-2000). He received the MS and PhD degrees in Computer Science, specializing in computer vision and Image Processing, from INRIA-Sophia Antipolis in France in 1993, and 1996, respectively. He has authored and co-authored over 40 peer-reviewed journal and conference papers, and has been in the organizing and the technical committees of several international colloquia. Dr. Foroosh is a senior member of IEEE, and an Associate Editor of the IEEE Transactions on Image Processing. He has served on the committes of the IEEE Int. Conf. on Image Processing, the IEEE Workshop on Image and Video Registration, the IEEE Workshop on Motion and Video Computing, the IEEE Workshop on Applications of Computer Vision, and the IEEE Workshop on Stereo and Multi-baseline Vision. In 2004, he was a recipient of an academic excellence award from Sun MicroSystems, and the Pierro Zamperoni best paper award in the International Conf. on Pattern Recognition.