Online Tracking of Outdoor Lighting Variations for Augmented Reality with Moving Cameras

Yanli Liu and Xavier Granier

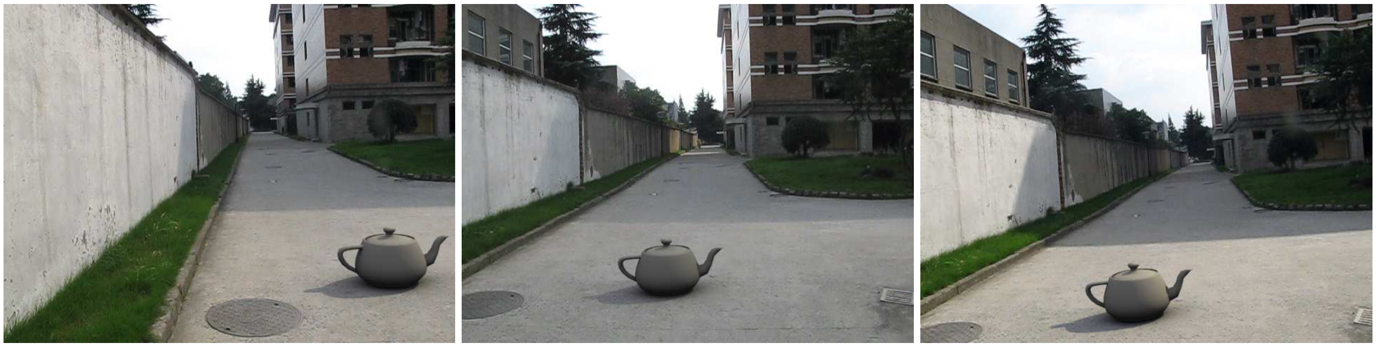


Fig. 1. Resulting integration of a virtual teapot into the `building` video sequence with moving viewpoints. The virtual teapot is lighted using a illumination model composed of sunlight and skylight. Thanks to our online tracking of lighting variations, the illumination is consistent from frame to frame between the real scene and the teapot.

Abstract—In augmented reality, one of key tasks to achieve a convincing visual appearance consistency between virtual objects and video scenes is to have a coherent illumination along the whole sequence. As outdoor illumination is largely dependent on the weather, the lighting condition may change from frame to frame. In this paper, we propose a full image-based approach for online tracking of outdoor illumination variations from videos captured with moving cameras. Our key idea is to estimate the relative intensities of sunlight and skylight via a sparse set of planar feature-points extracted from each frame. To address the inevitable feature misalignments, a set of constraints are introduced to select the most reliable ones. Exploiting the spatial and temporal coherence of illumination, the relative intensities of sunlight and skylight are finally estimated by using an optimization process. We validate our technique on a set of real-life videos and show that the results with our estimations are visually coherent along the video sequences.

Index Terms—Augmented reality, illumination coherence, moving cameras.

1 Context and Introduction

Augmented reality is a very challenging task since it requires achieving at least visual consistency between known virtual objects and real scenes from which most of the parameters are unknown. These parameters include 3D geometry, material properties and lighting of the real scene. While many efforts in augmented reality have been devoted to geometric calibration of cameras, by either extracting parameters directly from videos (e.g., [2, 33, 13]), or from new captors embedded in nowadays mobile devices (such as accelerometers for tracking virtual cameras [18]), less attention has been put on illumination consistency between virtual objects and real scenes. This is especially important for uncontrolled outdoor scenes where lighting variations over time (due to the time of the day and weathering conditions) may drastically change a scene's appearance. Without tracking illumination variations, the inconsistent variations of lighting between rendered virtual objects and real ones (such as flickering) will definitely destroy the feeling of realism of the system, making users aware that the virtual objects are not real parts of the scene. In this paper, we will focus on tracking such lighting variations of outdoor scenes.

Nowadays, handhold devices with video capabilities are easily accessible. This leads to the possible development of augmented reality applications for general users. Such applications have to take into account two main constraints. First, the visualization of the resulting video at the moment of the capture improves the user's experience. To achieve such a goal, the processing has to be done in a streaming-like (online) manner, leading to the fact that the system can be only aware of the past frames: solutions based on a global estimation on the whole video sequence are thus unsuitable. Online processing is a first step toward real-time solutions. To develop an online algorithm, we reduce the complexity by relying on a sparse set of pertinent pixels. We also re-use as much as possible information from previous frames by taking into account frame-to-frame coherency.

The second constraint is the fact that the viewpoint is moving freely. The few existing algorithms [16, 17, 29] designed to estimate outdoor illumination are tailored for fixed viewpoints. In case of moving viewpoints, feature-points misalignment is almostly inevitable even using the state-of-art techniques to track features. Furthermore, some feature-points may enter or leave the view frustum from one frame to another. Obviously, it is more challenging to achieve a stable process for moving viewpoints.

In this paper, we focus on online tracking of outdoor lighting variations from videos with moving viewpoint. For this purpose, we introduce some assumptions. First, we consider that the camera calibration and tracking are done online by another process (such as in [26]). Secondly, we can reasonably assume that consumer video camera have or will have access to information like time of the day and GPS coordinates. Such information greatly simplifies the overall process. Finally, we assume that during the initialization, it is possible to detect some planar surfaces [12], such as ground, building surfaces. Our approach

- *Yanli Liu is with College of Computer Science, Sichuan University, P.R.China, E-mail: yanliliu@scu.edu.cn. She is also with INRIA Bordeaux Sud-Ouest, France, and National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, P.R.China.*
- *Xavier Granier is a research scientist at INRIA Bordeaux Sud-Ouest, with LaBRI (CNRS : UMR5800 – Université de Bordeaux) and LP2N (Institut d'Optique Graduate School – CNRS : UMR5298 – Université de Bordeaux) - F-33400 Talence, France E-mail: xavier.granier@inria.fr*

Manuscript received 15 September 2011; accepted 3 January 2012; posted online 4 March 2012; mailed on 27 February 2012.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

will only track such planar patches over the frames. Note also that we do not intend to accurately reconstruct outdoor lighting, but to be consistent enough for our final purpose that is, the changes due to lighting are visually consistent between real and virtual objects.

This goal is achieved thanks to the following contribution of the paper: (1) we develop a novel framework, based on an optimized process, to track lighting changes of outdoor videos with moving viewpoints. To the authors' knowledge, there is no published algorithm yet for online tracking the illumination of outdoor videos with moving cameras. (2) we estimate the lighting via a sparse set of well-chosen planar feature-points, which makes the tracking high efficient and suitable for online processing. (3) to deal with moving viewpoints and resulting misalignment of feature-points, we make use of some constraints in-between frames based on the observed characteristics of outdoor lighting. These contributions leads to a stable estimation on a reduced set of feature-points. The presented algorithm achieves nearly real-time processing with an un-optimized Matlab implementation.

2 Previous Work

Sunlight and skylight are distant lights and may thus be assumed as almost constant over the real scene observed from a frame. Direct measurements of illumination may be obtained by putting makers or light probes [27, 22] into the real world. The use of light probes leads to an accurate estimation of directional distribution of light. However, it also results in large computational cost due to the dense directional sampling that has to be processed. Recent works have shown that it is possible to create such an environment map dynamically [7] and to have a dynamic [9] or real-time [6] acquisition. Unfortunately, for a context where the camera is moving freely, we need a sufficiently dense set of makers in order to ensure at least one of them is visible in each frame to capture lighting intensities [6]. In this paper, we estimate illumination directly from video frames and thus our system does not require any supplemental devices.

Estimating the lighting parameters is a complex inverse problem. Assuming a correct photometric calibration, a pixel directly measures the intensity resulting from the influence of geometry, reflective property, and incident lighting. In a controlled environment, it is possible to add further assumptions that allow such a recovery of the three components, namely BRDFs, 3D geometry and lighting ([21, 4]). It relies in general on an iterative optimization process to estimate one after the other two components. However, outdoor scenes are un-controlled environments. By assuming the sole knowledge of 3D geometry, it is possible to estimate the two other components that are BRDFs and lighting (e.g., light position and reflectance in [5]). In order to estimate lighting only, previous works generally assume a known BRDFs and 3D geometry. Based on this a-priori knowledge, illumination estimation may be computed from intensity variations such as shading (e.g. [1]), shadows (e.g. [31]). All the above techniques mostly rely on a full 3D reconstruction or on dense 3D information [20]. In our approach, in order to lower down the overall complexity for online processing, we want to reduce the number of samples on which the computation will be done. We also need to reduce the amount of required information that we have to estimate.

Recently, two main trends have emerged to estimate parameters from images and videos. The first one is based on time-lapse sequences. Lalonde et al. [15] propose an approach to estimate GPS coordinates and camera parameters. Such an approach is based on the assumptions that a large amount of sky is visible in each frames and that the viewpoint is fixed. Similarly, Junejo and Hassan [11] estimate the same parameters from shadows. Based on a daylight model, Sunkavalli et al. [29] propose to estimate the sunlight and the skylight from a video sequence. The second trend is data-driven estimation of illumination. As an example, Lalonde et al. [14] have collected a large set of time-lapse sequence from webcam located all around the United State. Based on this data, they demonstrate that it is possible to recover parameters for both appearance and illumination. Unfortunately, all these approaches are not suitable for online processing since they require the use of the whole video sequence or a large database.

Recently, Liu et al. [17] have shown that, for a fixed camera, and with a coarse knowledge of 3D scene, it is possible to recover the illumination in a video sequence with online computation. To achieve such a result, a set of reference images is acquired through a training process from the initial video. Illumination is computed as a combination of these images. Unfortunately, the training step may require a long video sequence. Alternatively, exploiting the relation between image statistics and lighting parameters, the authors have later introduced [16] a statistics-based approach but still limited to a fixed viewpoint.

3 Overview

To design our algorithm, we assume a strong correlation in lighting over large spatial and temporal extents [10]. With such an assumption, we combine information of current frame with the previous one to estimate the relative variations of sunlight and skylight. Since using all the pixels would make the underlying optimization process too highly constrained and too computational expensive, we thus perform the estimation on a limited set of clustered feature-points.

Illumination Model As suggested by dichromatic reflection model [25], most of real-world surfaces exhibit a mixture of interface reflection (specular reflection) and body reflection (diffuse reflection). Based on this model, researchers in computer vision have derived the widely-used assumption of neutral interface reflection (see [3]). This assumption states that the color of the specular reflection is the same as the color of the incident lighting. In our approach, we also exploit this statement.

Moreover, like existing works on outdoor illumination estimation (e.g., [24, 28, 16]), we model the sunlight as a time-varying directional light which colored intensity is $\boldsymbol{L}^{\text{sun}}(t)$ and its direction $\boldsymbol{l}(t)$, and the skylight as a time-varying ambient light with colored intensity $\boldsymbol{L}^{\text{sky}}(t)$.

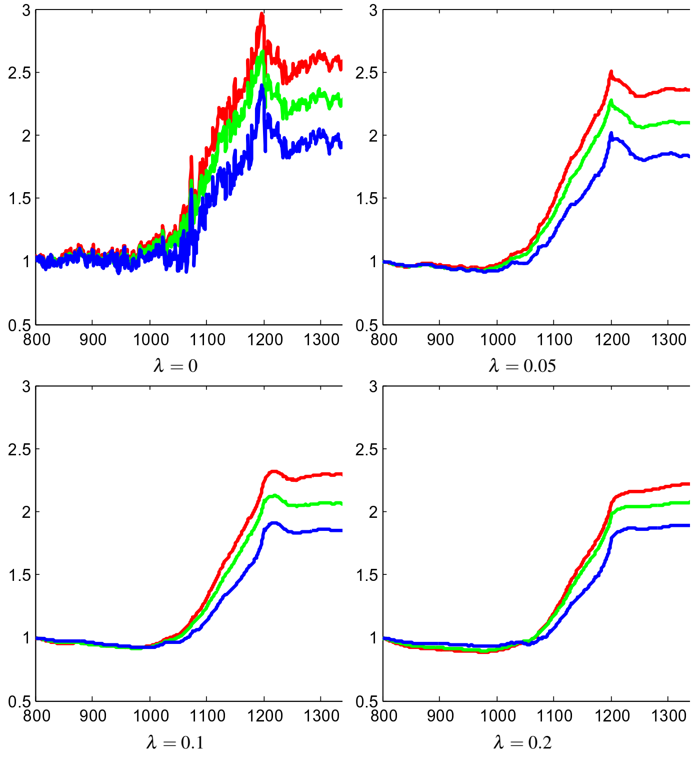
Under these two assumptions, the trichromatic (RGB) color $\boldsymbol{I}_p(t)$ of pixel p in the frame t is computed as follows:

$$\begin{aligned}\boldsymbol{I}_p(t) = &\left[k_p \left\langle \boldsymbol{n}_p, \boldsymbol{h}_p(t)\right\rangle^{m_p} + \boldsymbol{\rho}_p \left\langle \boldsymbol{n}_p, \boldsymbol{l}(t)\right\rangle\right] s_p^{\text{sun}} \boldsymbol{L}^{\text{sun}}(t) \\ &+ \boldsymbol{\rho}_p s_p^{\text{sky}} \boldsymbol{L}^{\text{sky}}(t)\end{aligned} \tag{1}$$

where $\boldsymbol{n}_p$ is the surface normal at p, $\boldsymbol{h}_p(t)$ the half-vector between $\boldsymbol{l}(t)$ and the view direction at p, $\boldsymbol{\rho}_p$ is trichromatic diffuse coefficients of surface at pixel p, k_p and m_p are the specular properties of surface, and $s_p^{\text{sun}} \in [0,1]$ and $s_p^{\text{sky}} \in [0,1]$ are the shadowing term due the 3D scene geometry for the sunlight and the skylight (ambient occlusion).

Feature-points Similarly to lighting estimation, estimating illumination variations would requires in theory the knowledge of 3D geometry (at least normals) and BRDFs properties of each pixel. Since we consider a moving viewpoint, these properties would have to be also estimated at each frame, making the process impracticable for nearly real-time solution. Instead, we prefer to select a subset of pertinent pixels, called feature-points, on which the normal and coarse BRDFs model may be easily estimated: this is the case for non-shadowed part of planar surfaces. Indeed, the identification of planar points may be easily done by estimating a homography and by computing re-projection error in-between frames. The selection of feature-points and the tracking of their attributes (normal and BRDFs) are described in Section 5.

Online Processing The core process of our approach works as an out-of-core solution. Assuming that visually coherent lighting, normals, and BRDFs have been estimated in a calibration step, for each frame t, we combine the information provided by the feature points of the current and previous frame with the estimated lighting parameters of the previous frame $t-1$. For newly appeared planar feature points, we use current lighting parameters to compute their appearance attributes (BRDFs and normals). We detail the process in Section 4.



$\lambda = 0$

$\lambda = 0.05$

$\lambda = 0.1$

$\lambda = 0.2$

Fig. 2. Relative RGB lighting intensities of video `building` for different value of smoothness factor λ. For simplicity, only the values for the sunlight are shown. Since the estimation is relative to the initial guess, we use 1 as reference intensity in the first frame.

4 Tracking Lighting Parameters

One of the main assumptions of our approach, is that the outdoor lighting is changing very slowly compared with the video frame rate. To confirm this assumption, we have experimented lighting estimation technique of Liu et al. [16] and have observed that there was no noticeable changes during time intervals smaller than $1/5$ seconds, even for days with drastically changing weather. Therefore, for video frame rate higher than 15fps, we can track sunlight changes in a new frame by minimizing the intensity difference between two consecutive frames. The energy to minimize is defined as the sum of a data term ($E^{data}(t)$, which minimization provides a coarse estimation of lighting in frame t) and a smoothness term ($E^{smooth}_{t,t\text{-}1}$, which tries to preserve previously estimated lighting):

$$E(t) = (1-\lambda)E^{data}(t) + \lambda\, E^{smooth}_{t,t\text{-}1}$$

with $\lambda \in [0,1]$. The influence of smoothness factor λ is shown in Figure 2. With $\lambda = 0$, we only use the coarsely estimated lighting: there is no tracking of the lighting changes, resulting in a lot of oscillations and thus, flickering in the final rendering. With $\lambda = 1$, there will be no lighting changes over the frames.

In order to simplify the notation, we will only notice the dependency on t when required for the comparison between two consecutive frames. By default, a denoted value is valid for the current frame only.

$$E = (1-\lambda)E^{data} + \lambda\, E^{smooth}_{t,t\text{-}1}\,. \tag{2}$$

Data Term Denoting $\widetilde{\boldsymbol{I}}_p(t)$ the intensity of pixel p in frame t computed using Equation (1), and $\boldsymbol{I}_p(t)$ the actual intensity, the data term is defined as the sum over all frame's pixels of the difference between actual pixel intensity and computed one

$$E^{data} = \sum_p |\boldsymbol{I}_p - \widetilde{\boldsymbol{I}}_p|^2\,.$$

Performing such a minimization on all the frame's pixels may be prohibitively expensive. We thus prefer to rely on a reduced set of reliable feature-points clustered according to their color and planarity similarities. All feature-points of a resulting cluster Ω_{ij} will likely share the same BRDFs and normal. Reader may refer to Section 5 for details. The minimization is computed only on these points, leading to a reduced data term

$$E^{data} = \sum_{i,j}\sum_{p\in\Omega_{ij}} |\boldsymbol{I}_p - \widetilde{\boldsymbol{I}}_p|^2\,.$$

Due to the inevitable misalignments of feature-points resulting from camera movements, the intensity difference of a feature-point between two consecutive frames is mainly composed of the lighting changes and the feature misalignments. The selection of reliable features which can best reveal lighting changes is thus essential. Our key assumption for this selection is that lighting changes are global. On the contrary, feature misalignment is a local error. Hence, we score the reliability (i.e., minimize the influence of potential outliers) of every planar feature-point p in cluster Ω_{ij} as:

$$\omega_{ijp} = \frac{1}{Z_{ij}} e^{-(I_p - \bar{I}_{ij})^2/\sigma^2}\,, \tag{3}$$

where $\bar{I}_{ij}$ is the mean intensity of all the selected features in Ω_{ij} and Z_{ij} is the following normalization factor

$$Z_{ij} = \sum_{p\in\Omega_{ij}} e^{-(I_p - \bar{I}_{ij})^2/\sigma^2}\,. \tag{4}$$

The final data term is defined as the weighted distance between image values and fitted ones:

$$E^{data} = \sum_{i,j}\sum_{p\in\Omega_{ij}} \omega_{ijp} |\boldsymbol{I}_p - \widetilde{\boldsymbol{I}}_p|^2\,. \tag{5}$$

Smooth Term Since in real-world videos, the lighting change also smoothly, and in order to track the light intensities over frames, we introduce a smooth term into our energy function. This smooth term is simply defined as the sum of differences of sunlight and skylight between two consecutive frames:

$$E^{smooth}_{t,t\text{-}1} = |\boldsymbol{L}^{\text{sun}}(t\text{-}1) - \boldsymbol{L}^{\text{sun}}(t)|^2 + \gamma|\boldsymbol{L}^{\text{sky}}(t\text{-}1) - \boldsymbol{L}^{\text{sky}}(t)|^2 \tag{6}$$

We introduce the factor $\gamma > 1$, since we have noticed in our experiments that the variations of skylight are smaller than that of sunlight, which is in accordance with common sense.

Implementation Details The nonlinear least-squares minimization can be solved iteratively using standard optimization techniques such as Levenberg-Marquadt or `fminsearch` in MATLAB. We have experimented different value of λ (see Figure 2) and found that $\lambda = 0.1$, $\gamma = 5$ and $\sigma = 1$ are good default settings.

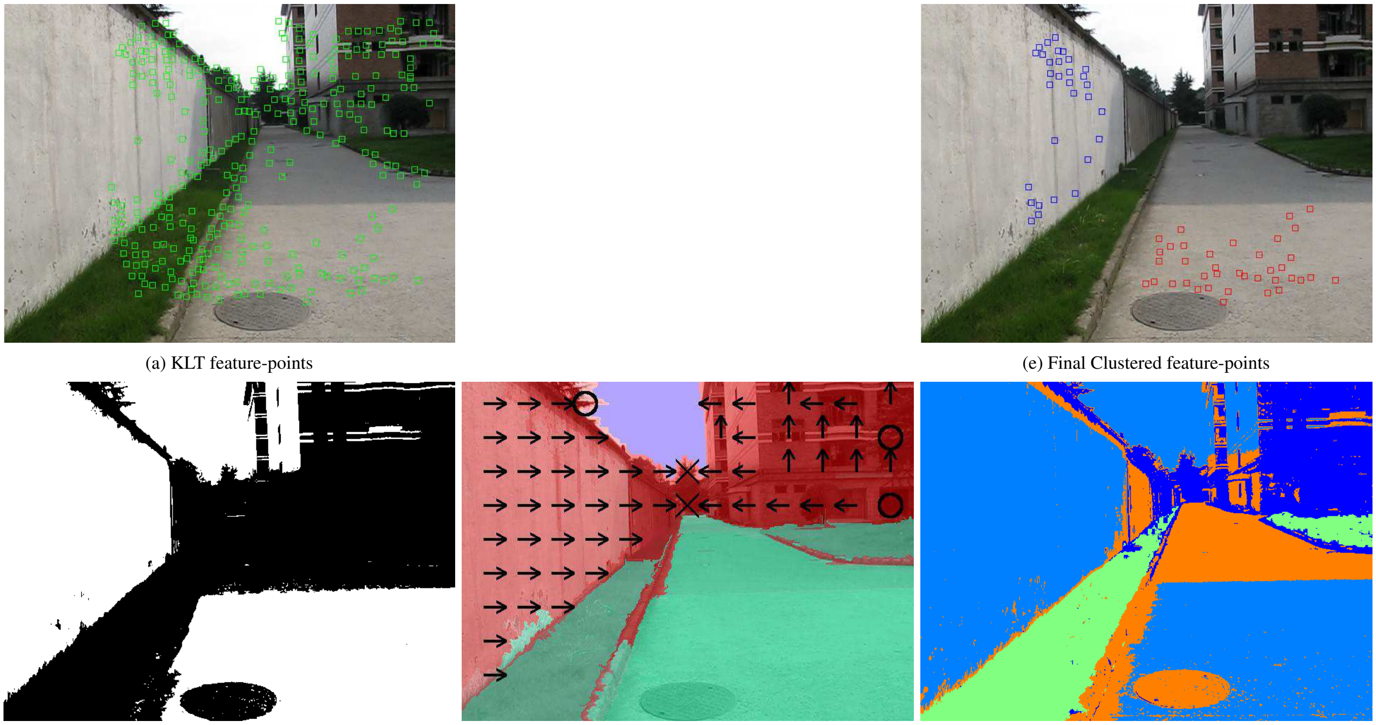
In this paper, we track lighting changes on non-shadowed areas (see next section). In these regions, illumination and its changes are mainly due to sunlight. To ease the optimization, we thus first use $\boldsymbol{L}^{\text{sun}}(t\text{-}1)$ and $\boldsymbol{L}^{\text{sky}}(t\text{-}1)$ to estimate $\boldsymbol{L}^{\text{sun}}(t)$, and then fix $\boldsymbol{L}^{\text{sun}}(t)$ to estimate $\boldsymbol{L}^{\text{sky}}(t)$.

5 Selection of Feature Points

Our approach to detect and track feature-points is based on the KLT algorithm [30]. We further add some specific constraints for the feature-points to be more pertinent, to ease the online update of their attributes and to lower down the overall computational cost.

5.1 Definition and Initialization of Clusters

Our system is based on a sparse set of feature-points that have strong normals and BRDFs coherence, as illustrated in Figure 3. Constructed into an initialization step, this clustering is transferred from frame to frame in order to track the corresponding properties of feature-points.



(a) KLT feature-points

(e) Final Clustered feature-points

(b) Lighted Areas

(c) Planar Surfaces

(d) Color Clustering

Fig. 3. System initialization. On the whole set of feature points (a), we keep only the sunlit ones (b). We cluster the remaining feature-points according to their belonging to quasi-planar surfaces (c) and to their color (d). The remaining sparse feature-points (e) have strong normal and BRDFs properties. Note that only features on the large planar surfaces are selected.

When estimating illumination from every frame, the occlusion term s_p is also required. It is difficult to accurately and consistently detect shadows under dynamically changing lighting: the most accurate techniques rely on the knowledge of 3D geometry [20, 23]. Recent shadow detection techniques are also still quite expensive to reach a sufficiently accurate estimation []. In our approach, we rely only on sparse un-shadowed feature-points ($s^{\text{sun}} = s^{\text{sky}} = 1$) that are easier to detect with a simple threshold on the luminance []. Another solution is to manually mark-out shadowed regions.

To ease the estimation of normal that we attribute to each feature-point, we assume that the scene contains some planar (or quasi-planar) regions such as building facades, ground... These planar regions may be detected using automatic algorithms (e.g. [12, 8]) or manually marked out and tracked over frames. Thus, we first cluster the feature points according to their appurtenance to a set of quasi-planar surfaces $\mathscr{P}_i, (i = 1...N)$. For traditional outdoor scenes containing some buildings, the most representative surfaces are the vertical ones and the ground [].

To ensure coherent BRDF inside a cluster, we further cluster the feature-points according to their color (by using a mean-shift algorithm). Denoting M_i the number of color clusters in $\mathscr{P}_i$, we finally obtain the set of clusters $\Omega_{ij}, (i = 1...N, j = 1...M_i)$.

5.2 Tracking of Feature Points: Clustering

Similarly to [30], we extract feature-points for each frame t and match them with feature-points of frame t-1, using similarity of their descriptor. Three categories of points are defined: (i) points are not paired; (ii) points are paired with previously clustered points; (iii) points are paired with previously un-clustered points (we denote them as *new feature-points*).

The two first categories are easy to deal with. In (i), we keep these points, but do not use them for the tracking of light parameters. In (ii), each new feature-point is attached to the cluster of its paired one: their normal and reflectance attributes are thus directly transferred. Since we have selected quasi-planar feature-points during the initialization, we preserve this characteristic over the frames.

For the third category, we need more steps to ensure that the feature-points have the required properties and can be clustered with a sufficient reliability. For this purpose, for each planar region $\mathscr{P}_i$, we first estimate a homograph transformation $\mathbf{H}_i$ between already clustered feature-points from frame t and those from frame t-1, using the techniques []. Based on $\mathbf{H}_i$, we compute re-projection error between feature-point p from frame t and to its paired feature-point q_p from frame t-1:

$$h_i(p) = |q_p - \mathbf{H}_i \cdot p|^2 \tag{7}$$

Low $h_i(p)$ ensure that p belongs to the quasi-planar region $\mathscr{P}_i$.

To judge if p belongs to cluster Ω_{ij}, we also need to check the BRDF consistency and we thus introduce a criteria based on color difference. For each cluster Ω_{ij}, we compute the mean value $\bar{\boldsymbol{I}}_{ij}$ of its already-paired feature-points from current frame t. For each new feature-point p, we thus calculate the distance $c_{ij}(p)$ between its pixel value $\boldsymbol{I}_p$ and the mean value:

$$c_{ij}(p) = |\boldsymbol{I}_p - \bar{\boldsymbol{I}}_{ij}|^2 \tag{8}$$

The final distance function of p to cluster Ω_{ij} is defined as

$$d_{ij}(p) = \alpha\, h_i(p) + (1-\alpha)\, c_{ij}(p)\,, \tag{9}$$

with α is a user-selected parameter to weight the two criteria. The cluster p belongs to is thus defined by the property

$$p \in \Omega_{\underset{(i,j)}{\operatorname{argmin}} d_{ij}(p)}\,. \tag{10}$$

To improve the robustness, we need to take into account spatial consistency. We thus introduce a new criterion that rejects feature-points for which the following condition is not fulfilled:

$$(\underset{ij}{min}\, d_{ij}(p)) < \varepsilon \text{ and } T_{ij}(\mathscr{N}_p) > N_{min}\,, \tag{11}$$

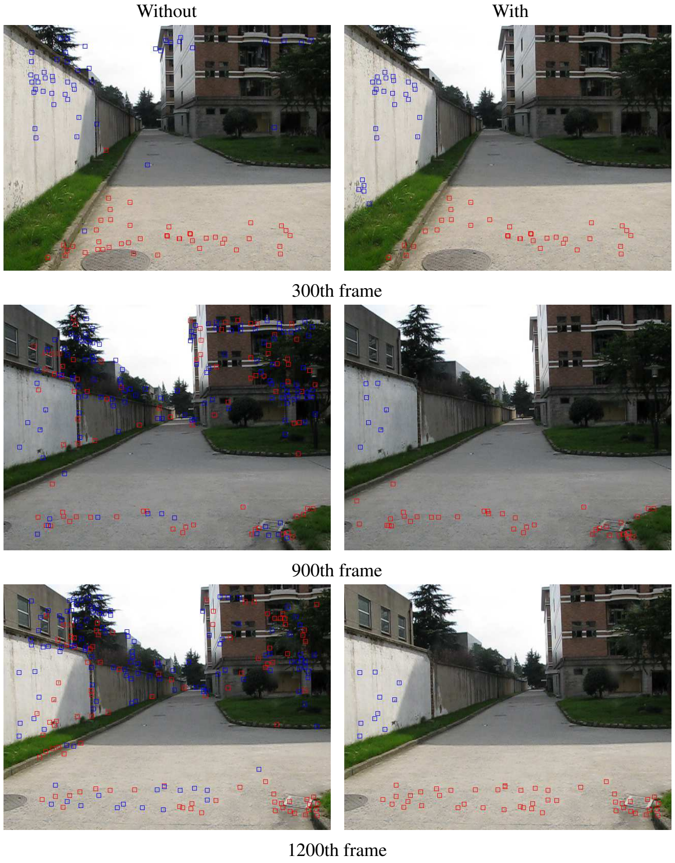


Fig. 4. Online selection of planar feature-points without (left column) and with (right column) spatial consistency and shadow detection. For our experiments, we set $\alpha = 0.3, \varepsilon = 0.8$. Note that in the frames of the lower row, almost all the misclustered feature-points including the red points on the left wall, the points on the building and trees, and the points on the shadow ground are removed.

where ε is a user-selected parameter to ensure that the labeling does not introduce too large errors, where $T_{ij}(\mathscr{N}_p)$ is the number of feature-points which has been labeled as Ω_{ij} in p's neighborhood $\mathscr{N}_p$ and finally, and where N_{min} is another user-selected parameter that chooses what is the reasonable minimal neighborhood size to ensure a sufficiently accurate estimation. We set $N_{min} = 3$ for all the examples of the paper.

5.3 Tracking of Un-shadowed Feature Points

After the initialization and for each frame, we check the likelihood of a feature-point to turn into shadow in order to still keep only the most likely sunlit feature-points for tracking the lighting parameters. For convenience, we still denote the selected sunlit feature-point sets that have same normal i and color j as Ω_{ij}. For system stability, we reject the clusters Ω_{ij} size is less than a threshold $\#\Omega_{max}$. We simply set $\#\Omega_{max} = 5$ for the examples shown in the paper.

Before the adjustment of light parameters, we first check whether every feature-points in Ω_{ij} has turned into shadow in the current frames. For this purpose, we use a similar approach than for BRDF clustering (see previous section). We set a feature-point as un-shadowed if it fulfills the two following conditions:

1. The feature-points have to be correlated to un-shadowed feature-points. First, we test the set of points that are paired with clustered points of previous frame: they cannot be labeled as un-shadowed if their paired points are not un-shadowed ones. Secondly, we test the points that are paired with previously unclustered points: they cannot be declared un-shadowed if their neighboring feature-points are not un-shadowed.

Scene	Characteristics	# Feature Points	fps
`building`	1 color cluster 2 planes	58	12.5
`laboratory`	1 color cluster 1 plane	97	16.4
`wall`	2 color clusters 2 planes	181	10.7

Table 1. Test scenes and corresponding average frame per second (fps) and average number of feature points estimated on $1{,}000$ frames. The time estimation only includes the processes described in this paper.

2. The feature-point has not turned into shadow. For this we check the intensity change inside the cluster Ω_{ij}. Denoting $I_{ij}^{\text{mean}}(t\text{-}1)$ the average luminance of the cluster in previous frame t-1, we keep the feature points as un-shadowed only if their intensity $I_p(t)$ verify
$$|I_p(t) - I_{ij}^{\text{mean}}(t\text{-}1)| > \mu\,.$$
In our experiments, we set $\mu = 0.6$

6 Tracking the Reflectance Attributes

The reflectance attributes (normals and BRDFs) are transferred through the clustering process described in the previous section. We thus need to only initialize their values. This initialization is done on the first two frames.

Still assuming the lighting does not change significantly in-between two consecutive frames, the color difference of a feature-point p is only caused by the change of viewpoint. As camera movement is known, according to Equation (1), the image difference at p in lit regions can be formulated in terms of specular properties k_p and m_p:

$$\begin{aligned}\widehat{I}_p^{diff} &= \boldsymbol{I}_p(2) - \boldsymbol{I}_p(1)\\ &= k_p \boldsymbol{L}^{\text{sun}}\left[\langle \boldsymbol{n}_p, \boldsymbol{h}_{2\,p}\rangle^{m_p} - \langle \boldsymbol{n}_p, \boldsymbol{h}_{1\,p}\rangle^{m_p}\right].\end{aligned} \tag{12}$$

Similar to [32], we approximate surface BRDFs as piecewise constant specular component and spatially varying diffuse component that is, all the feature point of $\Omega_{i,j}$ has the same k_i and m_i. This leads to

$$\widehat{I}_p^{diff} = k_i \boldsymbol{L}^{\text{sun}}\left[\langle \boldsymbol{n}_p, \boldsymbol{h}_p(2)\rangle^{m_i} - \langle \boldsymbol{n}_p, \boldsymbol{h}_p(1)\rangle^{m_i}\right]. \tag{13}$$

k_i and m_i are thus estimated by minimizing the following objective function

$$E(k_i, m_i) = \sum_j \sum_{p\in\Omega_{ij}} |\boldsymbol{I}_p(t) - \boldsymbol{I}_p(t\text{-}1) - \widehat{I}_p^{diff}|^2\,. \tag{14}$$

To obtain k_i and m_i through above equation, L^{sun} is required to be known. Note that for videos, the relative lighting to the first frame is enough. To track the relative variations of lighting, we assume L to be 1 in the first frame. With L^{sun} and initials of k_i and m_i, by minimizing about function, the relative k and m to the sunlight are obtained. Then, for every feature point p, the relative diffuse property $\boldsymbol{\rho}_p$ can be estimated:

$$\boldsymbol{\rho}_p = \frac{1}{T}\sum_{q\in N_p} \frac{\boldsymbol{I}^t_{qt} - k_i\langle \boldsymbol{n}_q, \boldsymbol{h}_{qt}\rangle^{m_i} \boldsymbol{L}^{\text{sun}}_t}{\langle \boldsymbol{n}_q, \boldsymbol{l}_t\rangle \boldsymbol{L}^{\text{sun}}_t + \boldsymbol{L}^{\text{sky}}_t} \tag{15}$$

where N_p is the neighborhood of p and T is the number of pixels of N_p.

After the initialization, and for each frame, k_i and m_i are simply transferred according to the planar type $\mathscr{P}_i$ of feature points. For new ones, we calculate the diffuse coefficient $\boldsymbol{\rho}_p$ according to Equation (15) after lighting estimation.

7 Results and Discussions

7.1 Test Setups

We have tested our approach on three real videos, with different complexities of the contained real scene (see Table 1 for the characteristics

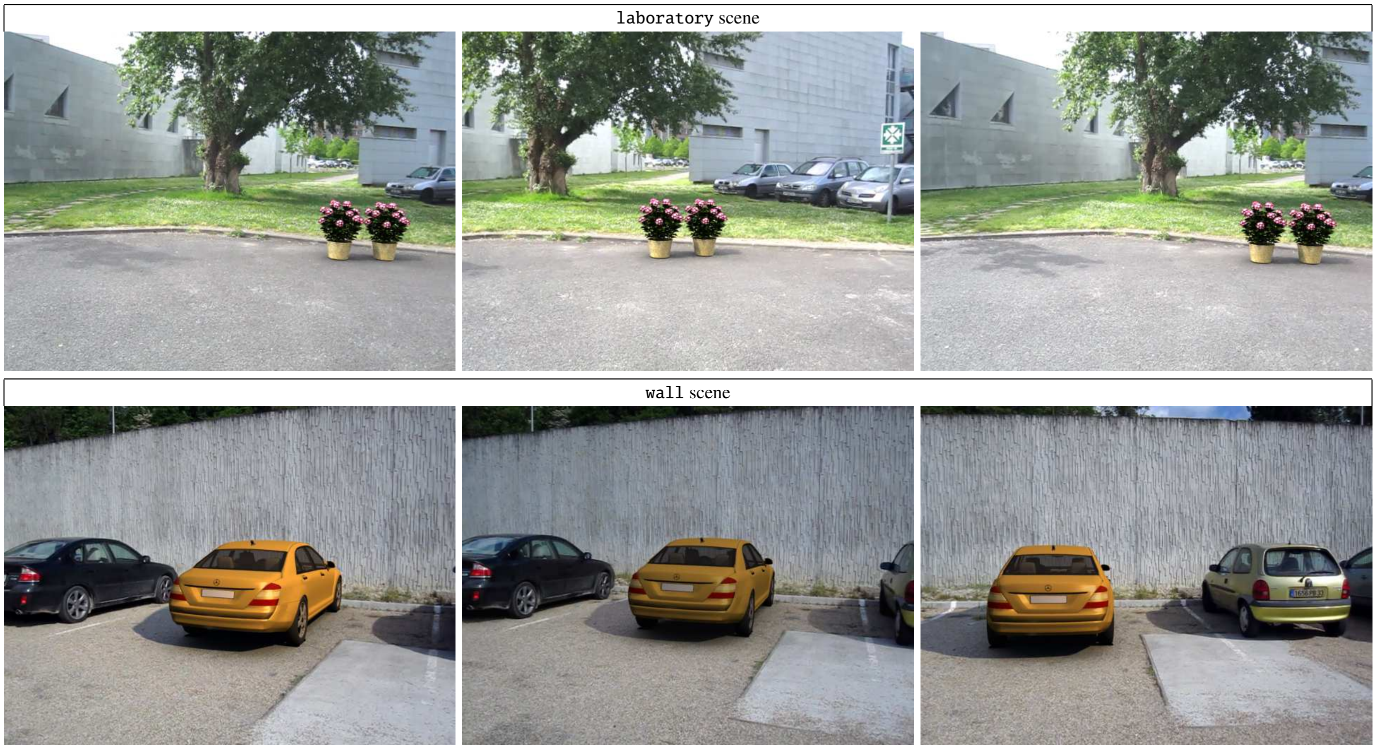
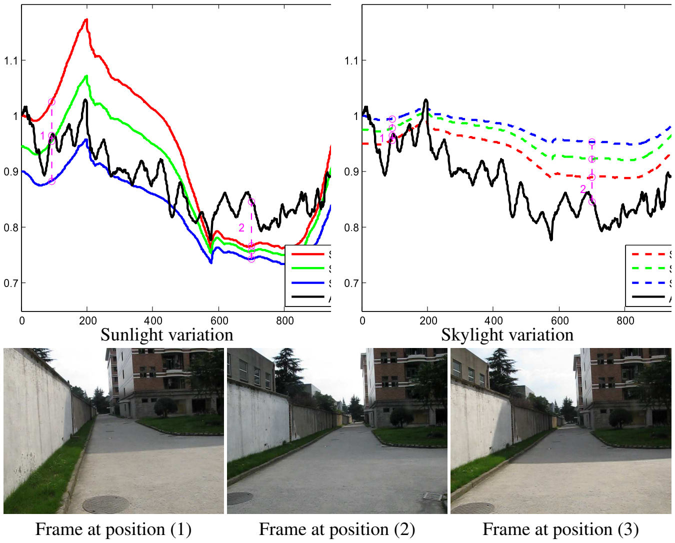


Fig. 5. Integration results of two test videos: `laboratory` (upper row) and `wall` (lower row). The virtual objects in these scenes are respectively the two flowerpots and a yellow car. The whole sequences are shown in companion videos.

Scene	All Feat. Points		New Feat. Points	
	light. est.	shad. det.	hom. estim.	spac. cons.
`building`	9.82%	0.09%	89.38%	0.71%
`laboratory`	13.27%	0.12%	84.94%	1.67%
`wall`	8.01%	0.2%	91.1%	0.69%

Table 2. Average proportion of the different steps in the total computational cost. From left to right: optimization process for lighting estimation (see Section 4), shadow detection (see Section 5.3), homography estimation and spatial consistency test. New features points are points that requires further process to be clustered (see Section 5.2).



Sunlight variation

Skylight variation

Frame at position (1)

Frame at position (2)

Frame at position (3)

Fig. 6. Estimated variations of sunlight and skylight for the `building` video. As a reference, we also display the average intensity of each frame (black curve).

and Figure 1, Figure 5 and the companion videos for the visual results). The videos were taken by a Canon SD780 IS camera with a non-experts user handholding it. During the whole sequence, we assume a constant focal length while the camera may move freely. To show our integration results, the camera movement has to ensure that the virtual objects are in the field of view. Our algorithm is implemented without any optimization on Matlab on an Intel i7 2.67GHz CPU with 6GB of RAM. The resolution of three videos is 640 × 480.

Note that, since the only 3D information that we have are the regions that correspond to quasi-planar geometry, we position the virtual object on such surfaces and we assume that it can not receive any shadow casted by the real scene geometry. For lighting the virtual object, we directly use the Equation (1). To transfer the shadows (both s^{sun} and s^{sky}) casted by the virtual object on the real planar surface, we adopt a standard solution by evaluating two illumination values from Equation (1) based on our estimated lights and BRDFs: $\tilde{\boldsymbol{I}}_p^{\text{shad}}$ (resp. $\tilde{\boldsymbol{I}}_p^{\text{unshad}}$) with (resp. without) the rendered shadow and ambient occlusion due to the virtual object on the planar surface. In newly shadowed regions, the original pixel intensity $\boldsymbol{I}_p$ is adjusted according to:

$$\boldsymbol{I}'_p = \frac{\tilde{\boldsymbol{I}}_p^{\text{shad}}}{\tilde{\boldsymbol{I}}_p^{\text{unshad}}}\boldsymbol{I}_p$$

7.2 Results

Our approach requires an initialization step. We illustrate the possible automatic process that we introduce in Section 5.1 on the `building` scene in Figure 3. In the scene presented in this figure, there are two kinds of planar surfaces (the ground and the wall on the left of image). The final clustered feature-points are show in Figure 3 (e), where different colors represent different clusters. The feature-points and their attributes are tracked over frame, as illustrated in Figure 4. Our feature selection results in a low number of points, which is sufficient for our lighting estimation. For the two videos in Figure 5, the selection of planar surfaces and of un-shadowed regions was done manually, illustrating that our approach is robust to different initialization strategies.

To estimate the variations of sunlight and skylight, we have introduced an optimization process that try to minimize the spatial variation and the variation over frames using two energy functions. The coefficient λ balances between these two energy functions. We have tested different values of λ on the resulting estimation of lighting. As it can be seen in Figure 2, without minimization of changes over time, the variations of sunlight has too many oscillations, which would cause noticeable appearance flicker on virtual objects. With the increase of λ, the lighting is more stable. Since there is no published algorithms for online tracking outdoor illumination under moving cameras, in Figure 6, we compared our approach with estimation based on averaging all the pixel intensities of each frame, which is simple yet easy to come up with. Our approach provide a richer and more accurate information since we show that the variations of sky and sky lighting are correlated but different, with stronger variations for sunlight and smoother one for skylight.

The integration of a virtual object into videos is not only an important goal of our algorithm, but also an effective way to examine the consistency of the estimation. Figures 1 and 5 show the integration results of the three test-scenes. The three videos are all captured under cloudy days, in which the sunlight is sometimes blocked and unblocked by clouds. In these cases, the appearances of the virtual objects match well the overall illumination of the video. It demonstrates that although our approach uses a sparse set of feature points spanning on one or two planar surfaces, the estimated results are stable and consistent. Please see the supplementary videos to this paper for the complete results.

We have evaluated the performance and shown the results in Table 1 and Table 2. From Table 1, we show that even with a non-optimized version of our approach, our solution reach quasi real-time performances. Indeed, as shown in Table 2, the vast majority of time is spent on the fitting of the homography. So it is no wonder that the algorithm efficiency strongly related to the number of different number of planes (see Table 1). The minimization of energy function to estimate the sunlight and skylight, usually takes only 10% $\sim$ 15%. We strongly believe that with an optimized and parallelized version of our algorithm which will be suitable for multicore architectures such as GPUs, we can achieve real-time. For such a goal, we have to focus first on the homography estimation. An optimized version would allow also the use of a larger number of planar surfaces to extend the range of possible scenes that our approach could deal with. The memory complexity of the proposed system is $O(n)$, where n is the number of selected feature-points. Theoretically, at least 4 feature-points for every plane cluster are required to fit a homography. To make the algorithm robust, the numbers of feature-points actually used for three examples are listed in Table 1. We also tested the performance of our algorithm with SIFT [19], another classical feature-points detection algorithm. Since we have optimized the selection of feature-points, we found our algorithm is not very sensitive to feature-points detection algorithms. As KLT is much faster than SIFT on both CPUs and GPUs, especially on CPUs, in the paper we chose KLT to detect feature-points. One can also use SIFT and other feature-points detection algorithms if not taking into account time efficiency.

7.3 Limitations and Future Work

The main goal of the presented algorithm is to provide a user with an Augmented Reality experience with its mobile and general purpose camera. This leads to the assumptions that we do not have the full 3D reconstruction of the real scene, and to the fact that the estimation of shadows can only rely on a pure image-processing approach. To estimate the lighting variations on reliable pixels, we focus on fully lighted ones. Sufficient to track the variations with a visual quality, such an approach is not suitable for a fully accurate estimation. Indeed, the skylight is dominant in shadowed areas, and taking into account such values would improve the separation between sky and sun lighting. One solution for online estimation would be to use stereo cameras [20], but unfortunately, this provides the system with only partial 3D information about the visible scene, and shadows may be due to geometry not visible in the current viewpoint. As we are relying on a reduce set of feature points, future improvements need to deal with this partial 3D information.

Secondly, since our system is tracking the lighting variations, the overall quality depends on the quality of the initialization. We have both experimented manual and automatic steps and shown that your algorithm is working in both cases. However, the automatic solution is fast but may fail in some cases, and the manual initialization may be tedious for a non-expert user. Generally speaking, to reach a physically accurate technique, not a visual consistent one as introduced here, we need to improve the quality of this initialization. An assisted semi-automatic calibration step may help in creating a more friendly and robust algorithm.

Currently, our tracking relies on independent evaluation on R, G and B channels. Once again, our results show that this is sufficient for visual quality. A third possibility to increase accuracy is to take into account the color or spectra characteristics into the lighting estimation. However, as noticed by Sunkavalli et al. [29], the current color models are not suited for optimization. They introduce a new model as a linear model, but it is not an a-priori model and it still requires a global optimization on the whole video sequence.

8 Conclusion

In this paper, we have introduced a fully image-based pipeline for online tracking of lighting variations of outdoor videos. This approach does not require any a-priori knowledge of the 3D scene and works with moving viewpoint. It is based on the selection of a sparse set of feature points that are sufficiently reliable to perform the tracking. The tracking itself is done through an energy minimization that takes into account the temporal and spatial consistency. Our approach manages the changes and thus misalignment of feature points with the camera movement, ensuring a stable estimation on this sparse set. We have experimented our approach on a set of real-life videos, and we have shown the efficiently of such a pipeline. We have also discussed some of the limitations and possible future work.

This work is the first step of a long march to a seamless and real-time AR solution for videos with moving viewpoint. First, an optimized version of our algorithm that is suitable for massively parallel processor such as GPUs would help in reaching real-time performances. Secondly, our approach relies on the quality of the process initialization that may be improved by the combination of simplified manual inputs and automatic processes. Finally, an improved shadow-detection would certainly help in increasing the quality of sky lighting and in allowing real objects to cast shadows on virtual objects. This last improvement would certainly require more 3D information: obtaining such information for the whole scene is still challenging for online processing.

Acknowledgments

This work has been initiated during the post-doctoral fellowship of first author at INRIA Bordeaux Sud-Ouest. The work is supported by 973 program of China (No. 2009CB320800) and NSFC of China (No. 61103137). The first author is also supported by the Open Project Program of the State Key Lab of CAD&CG (Grant No.A1112), Zhejiang University. We also want to thanks Guanyu Xing for his help in modeling.

References

[1] M. Andersen, T. Jensen, and C. Madsen. Estimation of Dynamic Light Changes in Outdoor Scenes Without the use of Calibration Objects. In *Int. Conf. Pattern Recognition (ICPR)*, pages 91–94. IEEE Computer Society, 2006.

[2] K. Cornelis, M. Pollefeys, M. Vergauwen, L. V. Gool, and K. U. Leuven. Augmented reality using uncalibrated video sequences. In *Lecture Notes in Computer Science*, volume 2018, pages 144–160, 2001.

[3] E. Eibenberger and E. Angelopoulou. Beyond the neutral interface reflection assumption in illuminant color estimation. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*, number 2, pages 4689–4692, 2010.

[4] T. Haber and P. Bekaert. Image-Based Acquisition of Shape and Spatially Varying Reflectance. In *British Machine Vision Conference (BMVC) - Poster*, 2008.

[5] K. Hara, K. Nishino, and K. Ikeuchi. Light Source Position and Reflectance Estimation from a Single View without the Distant Illumination Assumption. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:493–505, 2005.

[6] V. Havran, M. S. G. Krawczyk, K. Myskowski, and H.-P. Seidel. Interactive System for Dynamic Scene Lighting using Captured Video Environment Maps. In *Proc. Eurographics Symposium on Rendering 2005*, pages 31–42, 2005.

[7] T.-Y. Ho, L. Wan, C.-S. Leung, P.-M. Lam, and T.-T. Wong. Unicube for dynamic environment mapping. *IEEE Trans. Visualization and Comp. Graph.*, 17(1):51–63, 2011.

[8] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *Int. J. Comp. Vision*, 75(1):151–172, Oct. 2007.

[9] K. Jacobs, A. H. Nielsen, J. Vesterbaek, and C. Loscos. Coherent radiance capture of scenes under changing illumination conditions for relighting applications. *The Visual Comp.*, 26(3):171–185, 2010.

[10] N. Jacobs, N. Roman, and R. Pless. Consistent Temporal Variations in Many Outdoor Scenes. In *IEEE conf. Comp. Vision and Pattern Recognition (CVPR)*, pages 1–6. IEEE Computer Society, 2007.

[11] I. N. Junejo and H. Foroosh. GPS coordinates estimation and camera calibration from solar shadows. *Comp. Vis. Image Underst.*, 114:991–1003, 2010.

[12] O. Kähler and J. Denzler. Detection of planar patches in handheld image sequences. In *Proc. Photogrammetric Computer Vision*, volume 36, pages 37–42, 2006.

[13] A. D. C. Ke Xu, Kar Wee Chia. Real-time camera tracking for markerless and unprepared augmented reality environments. *Image and Vision Computing*, 26(5):673–689, 2008.

[14] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam Clip Art: Appearance and Illuminant Transfer from Time-lapse Sequences. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 28(5):1–10, 2009.

[15] J.-F. Lalonde, S. G. Narasimhan, and A. A. Efros. What Do the Sun and the Sky Tell Us About the Camera? *Int. J. Comp. Vision*, 88(1):24–51, 2010.

[16] Y. Liu, X. Qin, G. Xing, and Q. Peng. A new approach to outdoor illumination estimation based on statistical analysis for augmented reality. *Comp. Anim. Virtual Worlds*, 21(3-4):321–330, 2010.

[17] Y. Liu, X. Qin, S. Xu, N. Eihachiro, and Q. Peng. Light source estimation of outdoor scenes for mixed reality. *The Visual Comp. (Proc. CGI)*, 25(5-7):637–646, 2009.

[18] J. Lobo and J. Dias. Fusing of image and inertial sensing for camera calibration. In *Int. Conf. Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 103–108. IEEE Computer Society, 2001.

[19] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International conference on Computer Vision(ICCV)*, volume 2, pages 1150–1157, 1999.

[20] C. Madsen and B. Lal. *Probeless Illumination Estimation for Outdoor Augmented Reality*. INTECH, 2010.

[21] B. Mercier, D. Meneveaux, and A. Fournier. A Framework for Automatically Recovering Object Shape, Reflectance and Light Sources from Calibrated Images. *Int. J. Comp. Vision*, 73(1):77–93, 2007.

[22] Miika Aittala. Inverse lighting and photorealistic rendering for augmented reality. *The Visual Comp. (Proc. CGI)*, 26(6-8):669–678, 2010.

[23] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios. Illumination Estimation and Cast Shadow Detection through a Higher-order Graphical Model. In *IEEE conf. Comput. Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2011.

[24] Y. Sato and K. Ikeuchi. Reflectance analysis under solar illumination. In *Workshop on Physics-Based Modeling in Computer Vision*, pages 180–187. IEEE Computer Society, 1995.

[25] S. A. Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985.

[26] S. N. Sinha, J. michael Frahm, M. Pollefeys, and Y. Genc. Gpu-based video feature tracking and matching. Technical report, In Workshop on Edge Computing Using New Commodity Architectures, 2006.

[27] J. Stumpfel, C. Tchou, A. Jones, T. Hawkins, A. Wenger, and P. Debevec. Direct hdr capture of the sun and sky. In *Proc. ACM AFRIGRAPH*, pages 145–149. ACM, 2004.

[28] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz. Factored time-lapse video. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26:101:1–101:8, 2007.

[29] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister. What do color changes reveal about an outdoor scene? In *IEEE conf. Comp. Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE Computer Society, 2008.

[30] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, CMU, 1991.

[31] Y. Wang and D. Samaras. Estimation of multiple directional light sources for synthesis of augmented reality images. *Graph. Models*, 65:185–205, 2003.

[32] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *Proc. ACM SIGGRAPH*, pages 215–224. ACM Press/Addison-Wesley Publishing Co., 1999.

[33] G. Zhang, X. Qin, W. Hua, T.-T. Wong, P.-A. Heng, , and H. Bao. Robust Metric Reconstruction from Challenging Video Sequences. In *IEEE conf. Comp. Vision and Pattern Recognition (CVPR)*, volume 36, pages 1–8, 2007.