Automatic environment map registration

Ulysse Larvy[1,2], Céline Loscos[1] and Yiorgos Chrysanthou[2]

[1]Université de Reims Champagne-Ardenne, [2]University of Cyprus

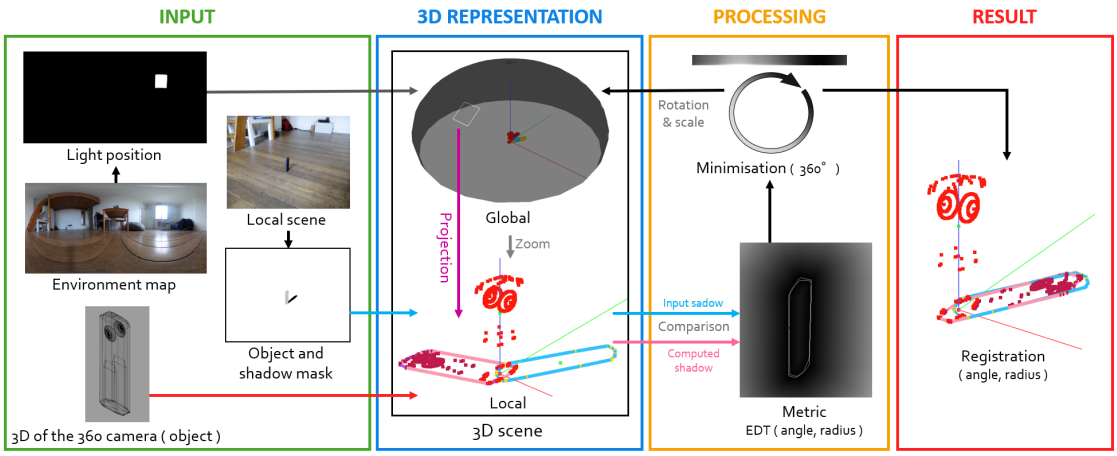

Figure 1: ***Overview of the presented method*** *INPUT : Local scene and environment map (EM), masks of the main object and its shadow, and geometry of the main object. 3D REPRESENTATION of the global scene (top) using the EM image, and local scene (bottom) using a shadow mask (blue) and 3D geometry of the main object (red). Computation of the projection on the ground of the object points using light present in the EM (purple). PROCESSING : EM (and so the light source position) rotation to obtain different projections of a computed shadow. Comparison, using Euclidean distance transform (EDT), of the shadows. RESULT : Minimisation of the EDT to obtain the EM registration (the input shadow equal to the computed shadow).*

Abstract

In this paper, a method to automatically register an environment map (EM) around a local scene is presented. In the literature, this step is most of the time manually processed by a user. However, it is an essential step when lighting and/or background coherence is needed. We present a method to find the coherent spatial organization between a main light source present in the EM and a couple object/shadow in a local scene. We automatically recover the EM orientation which corresponds to the local scene illumination. We proceed to a 3D representation of the scene using the EM mapped on a hemisphere as a background scene, a simplified geometry description of the reference object and its shadow outline. As a first step, we compute a projection of the main object shadow to compare it against the real acquired shadow. In a second step, we minimize a metric based on Euclidean Distance Transform (EDT), to compare both shadows and to recover the EM orientation. We demonstrate that we can automatically find rotation and scaling parameters that position in a coherent manner the background around a local scene.

1. Introduction

Nowadays, many different computer-graphics-based applications use environment maps (EMs) as a representation of far geometry. These maps can also be used as input for lighting conditions. While they are now well-acknowledged, their use requires user manipulation from their acquisition to their integration into a 3D scene.

In this paper, we propose an automatic registration method of an EM which consist in finding its orientation angle and scaling. An originality of the approach is that we do not require a 3D model of the scene. This constraint was set to answer to the most usual conditions of EM use. Our assumptions are threefold: (i) the local 3D scene photograph and the corresponding EM are known, (ii) the rendered image of the 3D scene can be segmented between objects and shadows for at least one of the objects of the scene, (iii) the main light sources of the environment could be identified in the EM, and the ground plane is flat around the main object to ensure the object shadow to be on a planar surface. Our method has three steps:

DOI: 10.2312/egs.20191010

1. Create a 3D representation of the scene using an EM, a scene image and the 3D geometry of a known object present in the scene image;
2. Compute a virtual shadow and compare it to the input shadow;
3. Solve by minimization of the EDT to adjust the best position.

After reviewing related work in section 2, we present the scene set up in sections 3 and 4 and the method in section 5. We present different results 6, before concluding in section 7.

```
Data: image EM, point2D p_light_pos2D, image SDW_mask,
      image OBJ_mask, vector<point3D> 3D_object
Result: best_angle
number best_score ← ∞, best_angle, best_radius;
forall point2D p in SDW_mask do
    point3D p_i ← 3D_recovery(p, OBJ_mask);
    vector<point3D> sdw_i ← add(sdw_i, p_i);
end
vector<point2D> sdw_input ← Convex_hull(sdw_i_z=0);
image im_sdw_input ← image_of(sdw_input);
image im_EDT_input ← EDT(im_sdw_input);
for i=0 to 360, step 1 do
    for j=200 to 500, step 10 do
        point3D p_light_pos3D ← 3D_pos(p_light_pos2D, i, j);
        forall point3D p in 3D_object do
            r3D ← Ray(p_light_pos3D, p);
            point2D p_floor ← Floor_extended(r3D);
            vector<point2D> sdw_c ← add(sdw_c, p_floor);
        end
        image im_sdw_c ← image_of(Convex_hull(sdw_c));
        image overlap_sdw ← im_EDT_input * im_sdw_c;
        number score_ij ← Σ_pixels_values overlap_sdw;
        if score_ij < score then
            score ← score_ij, best_angle ← i, best_radius ← j;
        end
    end
end
```

Algorithm 1: Computation of the best angle to register the EM. Each necessary time, we explicit when we use *2D* or *3D* element.

The algorithm (1) presents the main steps of our method. The different cited functions refer to article sections (*3D_recovery, Convex_hull, EDT, Ray, Floor_extended*). The *3D_pos* function computes the 3D position of the light source from the known 2D position, taking into account the EM rotation and radius, and the *image_of* function computes a mask using a vector of 2D points. The * operation multiplies images pixel per pixel.

2. Related work

In computer graphics, EMs are often used to simulate the light coming from a surrounding background scene to each point of a local scene. EM also helps for virtual object insertion or light changes, while keeping a coherent illumination. Traditionally, EMs are captured and represented as 180° or 360° panoramic/spherical images. Photographic cameras mounted with fish-eye lenses or 360° camera are the most used devices. Another way to capture EM is to use a light probe sphere as in [Deb08, JNVL10]. More recent devices allow the automatic capture of 360° spherical panorama, for example, a Ricoh Theta camera as in [BTH15] or a four-GoPro camera system in [LN15].

When no dedicated device is available, 360° images are often created by stitching sequences of images, as proposed by Perazzi *et al.* [PSHZ*15], where a set of unstructured low field of view video images are used to recreate a panoramic video avoiding artifacts. Other significant approaches use SIFT features [BL07] to be fully automatic. More recently, Chang *et al.* [CCYS13] present a method based on ASIFT features and energy map to avoid problems due to moving objects. It is not always possible to directly recover the far scene. For example, data can be collected from unknown sources or the environment do not permit to take such an EM picture. Assuming that only an approximation is necessary to simulate a coherent illumination, Lalonde *et al.* [LE10] propose a method to synthesize a plausible EM using a single image. Bousseau *et al.* present in [BCRA11] a method to synthesize an EM which enhance material visual features, placing a light area at chosen position, as for transparent or translucent material. Original approaches deduce an EM from other sources such as specular planar surfaces like a book cover [JND12], or an eye ball reflection [NN04].

The use of EMs in lighting and re-lighting applications can be introduced at a very early stage in illumination research methods. In [Pel10] and later with [BCD*13] the author presents an interface that allows a user to edit lighting effects of HDR EM in a scene. More recently Gardner *et al.* [GSY*17] aims to predict lighting of indoor scene using a CNN trained with a dataset of EMs. To get a correct and coherent illumination of the scene, EM requires to be orientated. Currently, in many works, this orientation is done manually as in [Laf12] or in [Pel10].

3. Input data

Our method registers an input image of an EM and an image of local scene (figure 2, left). We use an equirectangular image EM representation. Other representations are compatible with our method, only the projection on the 3D representation will change. Two mask images of the same size as input EM and local scene images are created: a mask of the EM light and of the main object and its shadow corresponding to the light in the EM. The mask of the main object shadow is considered to encompass umbra and soft umbra. A last input data is the 3D geometry of a reference object (figure 2, right). For instance, we have modelled a Ricoh theta 360° camera (303 vertices, 605 edges, 306 faces). A rough convex hull of the object is sufficient.

4. Scene setup

In our method, we assume that at least one light source is visible in the EM, and an object and its shadow are visible in the input image. We do not have clues about light sources' features (color, size, etc.).

Environment map and local scene To set up the scene, we first project the image of the EM on a half-sphere (figure 3, left). Then we place at the center of the sphere a representation of the local scene (the main object and of the input shadow). We use the 3D geometry of the main object (figure 3, right) to recreate it visual representation mapping in the local scene image. The 3D points are scaled to obtain an object size of 10 (no unit, it is only a referenced ratio for EM scaling).

Figure 2: ***Input data*** *left column: local scene and masks of the main object and its shadow, corresponding to the light in the EM (middle column). Right: 3D geometry of the main object (i.e., 360° camera) present in the local scene.*

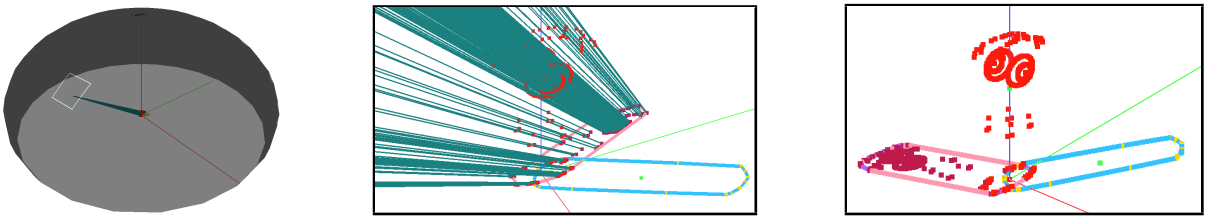

Figure 3: ***Scene representation*** *Left: global scene (EM) with light position and projected rays. Middle: local scene. Ray projection (turquoise) of the object 3D points to the ground plane. Right: local scene based on the input image (blue) and the 3D geometry of the main object (red). The projected shadow is in pink. The projected rays have been hidden for better comprehension of the local scene.*

Input shadow 3D recovery To obtain the direction of the input shadow relatively to the main object we use the hypothesis that the ground is flat around the main object, so its shadow is projected on a horizontal planar surface. Using that, the upper a pixel is in the image space hindmost it is in the space, the lower a pixel is in the image, the closer it is from the camera (see figure 4). If we consider the lower pixel of the object on the image plane as touching the ground, we can then assume that a shadow pixel higher (respectively lower) than the object basis in the image plane, is behind (resp. in front of) the object relative to the camera. We can then obtain an *input shadow* (from input image). At the end of this step, a 3D representation of the scene (global and local) is obtained (figure 3).

5. Environment map registration

In this section, we develop our approach to find an overlap between a computed virtual shadow and the input shadow.

Computed shadow definition

Using the 3D scene representation, we project rays using the EM light position and main object 3D points (figure 3, pink) to define a computed shadow. We trace rays from the light source position (corresponding to an EM orientation and scale) to each 3D points of the main object. These rays are extended to the ground plane ($z = 0$). We obtain a 2D set of points (figure 3, purple points).

Convex hull This set of points is not directly comparable to the input shadow, so we compute the convex hull of this set. Using this approach allows us to work with a simplified geometry and avoid dealing with difficult cases such as concave objects or holes. Of course, other types of shadow computation methods could be used. In our case, with a compact object as 360° camera, the convex hull allows describing the object in a good way, without deformation and loss of information.

Shadow overlap detection

The degree of overlapping of the computed shadow and the input shadow is estimated using a Euclidean Distance Transform (EDT) metric. We first compute the distance map of the input shadow. Then, we compare the computed shadow and the input shadow to consider pixels belonging to only one shape (equation 1 and 2). We obtain a mask image representing the difference between the two shapes. In practice, we apply an XOR to the image pixel by pixel. Finally, we sum the result pixel values multiplied by the corresponding distance map pixel values to obtain the metric.

$$\begin{aligned}\zeta_{\theta,\eta} &= \sum_{i,j} (S_C \cup S_I \setminus S_C \cap S_I) \star I_{EDT} \\ &= \sum_{i,j} XOR(S_C, S_I) \star I_{EDT}\end{aligned} \tag{1}$$

In equation 1 ζ is the metric in function of (θ,η), with θ the angle and η the scale factor of the EM; S_C, S_I are respectively the binary image of the computed shadow and the binary image of the input shadow; I_{EDT} is the distance map; (i, j) the coordinates of pixel in the image.

$$S_C \cup S_I \setminus S_C \cap S_I = \begin{cases} 0 & if \quad S_C = S_I (= 0 or 1) \\ 1 & if \quad S_C \neq S_I \end{cases} \tag{2}$$

EM orientation

To obtain the EM orientation, we minimize the metric ζ. If the shapes are the same, the resulting image of the equation 2 is full of zero values and the associated metric is $\zeta_{\theta,\eta} = 0$. Reversely, if the two shapes present few pixels in common, the resulting metric is high. The couple of angle and scale values (θ,η) that minimizes the EDT metric is thus considered as the best to orient the EM regarding to the local scene. To obtain a correct minimization of the metric, we consider a couple (θ,η). The η parameter is affecting the computed shadow size. However, we noticed that only θ has a physical meaning, because η allows only a coherent size of the computed shadow, but is not in direct relation to the true distance of the light source.

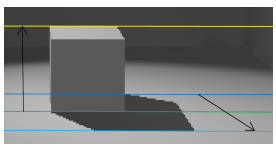

Figure 4: ***Shadow orientation recovery*** *from the dark-blue line (behind the object) to the light-blue line (in front of the object).*

6. Results

To validate our method, we use a set of eight images of the same local scene taken at different viewpoints. The camera viewpoints rotate with a regular angle of 45 degrees around the reference object (figure 5). The results of the eight input shadows (blue) are presented in the bottom center image, in a top view of the local scene. We observe that the resulting shapes are not well aligned with the ground truth because of missing information when a part of the shadow is hidden by the main object (local scene 5 and 6, on figure 5 on the top of top view).

Numerical results are presented in table 1. Our expectation is to find exactly 45 degrees between each pair of consecutive images. We estimate the overall error using two error metrics (oriented angles). The first one, $error_{1_n}$, uses one scene as a reference and computes the difference between its current angle and the computed scene angle (plus $n*45$, modulo 360), with n the scene number:

$$error_{1_n} = (angle_1 - angle_n + n*45) \,\%\, 360 \tag{3}$$

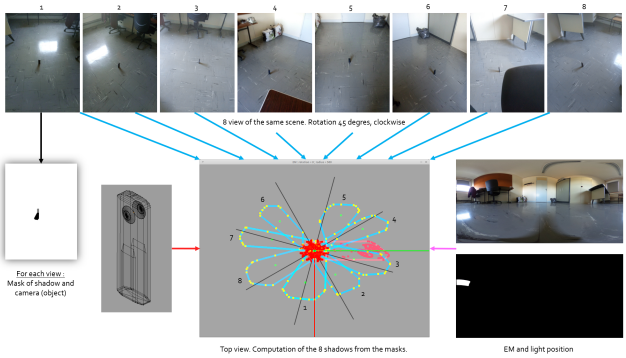

Figure 5: ***Validation of our method*** *Top line: 8 viewpoints of a same local scene. The EM (bottom, right) and main object (bottom, left) are the same for all the local scenes. The results of the eight input shadows are shown in the bottom center image (blue), a top view of the local scene. Black lines have been affixed and represent the ground truth (45 degrees angles).*

scene	expected θ	computed θ	computed η	$error_1$	$error_2$
1	189	189	455	0	14
2	144	140	455	4	4
3	99	106	410	-7	-11
4	54	60	320	-6	1
5	9	26	455	-17	-11
6	-36	-58	455	22	-6
7	-81	-96	230	15	-7
8	-126	-140	455	14	-1

Table 1: *Orientation results of the eight local scenes of figure 5.*

The $error_{2_n}$ computes the relative error to the previous scene:

$$error_{2_n} = (angle_{n-1} - angle_n + 45) \,\%\, 360 \quad (4)$$

Mean errors are respectively 10.625 and 6.875 degrees. Larger errors are associated with viewpoints 5 and 6 due to incomplete recovery of the input shadow, because behind the object. The maximum error is 22 degrees for the $error_1$ and 11 degrees for $error_2$.

We have not identified a method we can directly compare with due to different input data. We can however compare our results to a recent method [JSZ*19] which finds the orientation of an EM using deep learning. In this approach, the average error of the found angle is much higher (around 30 degrees) than our recovery error (10 degrees), and most importantly, they do not consider EM scale. We found in our experiments that, although scale does not have a determined physical meaning, it has a clear impact on the results.

7. Conclusion

In the literature, many methods need to orient the EM to obtain coherence with a near scene. This orientation is done manually by a user. We present a method to automatically register an EM with a local scene. A characteristic of our approach is that we do not need to create a full 3D model of the scene or any user interaction, and we make a few assumptions. Our method provides correct results even in cases where part of the shadow is hidden by the object. In future work, we want to improve orientation accuracy. Firstly, optimising the orientation research by using more efficient algorithms as gradient descent should improve the computation time while increasing orientation accuracy. Secondly, our approach should be extended to more cases such as taking into account more light sources, area light sources, and more orientation cases of EMs, to enable out-centered EM capture such as by a video surveillance camera placed high up a scene.

References

[BCD*13] BANTERLE F., CALLIERI M., DELLEPIANE M., CORSINI M., PELLACINI F., SCOPIGNO R.: Envydepth: An interface for recovering local natural illumination from environment maps. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 411–420.

[BCRA11] BOUSSEAU A., CHAPOULIE E., RAMAMOORTHI R., AGRAWALA M.: Optimizing environment maps for material depiction. In *Computer graphics forum* (2011), vol. 30, Wiley Online Library, pp. 1171–1180.

[BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *International journal of computer vision 74*, 1 (2007), 59–73.

[BTH15] BODINGTON D., THATTE J., HU M.: *Rendering of Stereoscopic 360 Views from Spherical Image Pairs*. Tech. rep., 2015.

[CCYS13] CHANG S.-M., CHANG H.-H., YEN S.-H., SHIH T. K.: Panoramic human structure maintenance based on invariant features of video frames. *Human-Centric Computing and Information Sciences 3*, 1 (2013), 1.

[Deb08] DEBEVEC P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes* (2008), ACM, p. 32.

[GSY*17] GARDNER M.-A., SUNKAVALLI K., YUMER E., SHEN X., GAMBARETTO E., GAGNÉ C., LALONDE J.-F.: Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090* (2017).

[JND12] JACHNIK J., NEWCOMBE R. A., DAVISON A. J.: Real-time surface light-field capture for augmentation of planar specular surfaces. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2012), IEEE, pp. 91–97.

[JNVL10] JACOBS K., NIELSEN A. H., VESTERBAEK J., LOSCOS C.: Coherent radiance capture of scenes under changing illumination conditions for relighting applications. *The Visual Computer 26*, 3 (2010), 171–185.

[JSZ*19] JIN X., SUN X., ZHANG X., SUN H., XU R., ZHOU X., LI X., LIU R.: Sun orientation estimation from a single image using shortcuts in dcnn. *Optics & Laser Technology 110* (2019), 191–195.

[Laf12] LAFFONT P.-Y.: *Intrinsic image decomposition from multiple photographs*. PhD thesis, Université Nice Sophia Antipolis; INRIA Sophia-Antipolis, 2012.

[LE10] LALONDE J.-F., EFROS A. A.: Synthesizing environment maps from a single image. *Technical Report CMU-R I-TR-10–24* (2010).

[LN15] LHUILLIER M., NGUYEN T.-T.: Synchronization and self-calibration for helmet-held consumer cameras, applications to immersive 3d modeling and 360 video. In *3D Vision (3DV), 2015 International Conference on* (2015), IEEE, pp. 434–442.

[NN04] NISHINO K., NAYAR S. K.: Eyes for relighting. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 704–711. doi:10.1145/1015706.1015783.

[Pel10] PELLACINI F.: envylight: an interface for editing natural illumination. In *ACM Transactions on Graphics (TOG)* (2010), vol. 29, ACM, p. 34.

[PSHZ*15] PERAZZI F., SORKINE-HORNUNG A., ZIMMER H., KAUFMANN P., WANG O., WATSON S., GROSS M.: Panoramic video from unstructured camera arrays. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 57–68.