Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.7529

Enhanced Directional Light Vector Estimation From a Virtual AR Object and its 2D Shadow Mask

Aleksander Yacovenco [**Federal University of Juiz de Fora** | ***aleksander.yacovenco@ufjf.br***]
Luiz Maurílio da Silva Maciel [**Federal University of Juiz de Fora** | ***luiz.maciel@ufjf.br***]
Marcelo Bernardes Vieira [**Federal University of Juiz de Fora** | ***marcelo.bernardes@ufjf.br***]
Rodrigo Luis de Souza da Silva [**Federal University of Juiz de Fora**| ***rodrigo.luis@ufjf.br***]

Departamento de Ciência da Computação, Universidade Federal de Juiz de Fora, Rua José Lourenço Kelmer, s/n, São Pedro, Juiz de Fora, MG, 36036-900, Brazil.

Received: 30 January 2026 • **Accepted:** 25 June 2026 • **Published:** 10 July 2026

Abstract. This work presents a novel method for inferring the main directional light source in a 3D scene, given only 2D inputs, namely a camera image and a rough shadow mask. A two stage algorithm is proposed, in which the first stage handles the inputs and makes an initial estimation for the light source. The second stage refines this first estimate to find a new vector assumed to be closer to the real directional light vector. One virtual object is rendered into the real scene in both stages. In the first stage, an external shadow estimator produces a coarse shadow for the virtual object, enabling the computation of an initial directional light vector. Next, our proposal is to compare the coarse shadow with the virtual shadow computed from the object. Thus, in the second stage, we seek to maximize the intersection over union (IoU) between both shadows. We assume that the best directional light vector provides the best shadow matching. The experiments are made both in virtual and real environments, in scenes with different levels of control and known data. Results show that our method is capable of finding the 3D light vector from the 2D scene, enhancing the initial rough shadow input.

Keywords: Directional Light estimation, Shadow Estimation, 3D lighting, Augmented Reality

1 Introduction

Augmented Reality (AR) has experienced significant growth in the past decades, together with related technologies on which AR can be based, such as computer vision and artificial intelligence [Chen *et al*., 2019]. AR can be defined as a variation of Virtual Reality (VR) in which the user can still see and interact with the real world environment, thus merging virtual objects and real ones in real time [Azuma, 1997]. A major challenge when dealing with AR is to correctly light the virtual objects in a scene.

AR applications must be capable of extracting the light properties of the real world and correctly placing them in the virtual scene. Usually, researchers employ more specialized hardware, such as a multiple camera setup [Frahm *et al*., 2005], depth cameras [Kán and Kaufmann, 2019], or more specialized methods, such as Simultaneous Localization and Mapping (SLAM) [Whelan *et al*., 2016], or a combination of multiple methods [Meilland *et al*., 2013]. In general, this can be seen as a costly problem.

With the advent of deep neural networks, there are recent works that, based on known shadowed objects into a real scene, estimate the shadow of one virtual object inserted in the scene [Liu and Wu, 2022]. In some works, only one image is needed for the neural network to extract the shadows of real objects [Liu *et al*., 2020]. However, as this is an ill-posed problem due to the use of 2D information to infer 3D projections, these works do not define well the shadow's shape for the virtual object. However, such methods are observed to be capable of approximating the direction and length of the shadow.

In this work, we insert a virtual object into a real image using a fiducial marker typical of AR systems. The marker plane defines the support onto which the virtual object's shadow should be cast. Therefore, the input to our method consists of: an image of a scene with real objects their respective real shadows and a fiducial marker that defines a plane and a position for a virtual object; the geometric model of a virtual object; and an image that contains its coarse shadow estimate in the real scene. We assume that any shadow estimator can be used to compute the image with a coarse shadow approximation of the virtual object.

Since we have the virtual object model, its coarse projected shadow and a fiducial marker that provides the plane of shadow projection, it is possible to estimate a 3D directional light vector of the real scene. With this vector, we can render the virtual object and its expected shadow over the plane to improve visual quality. Thus, the estimation of a suitable directional light vector is a contribution of our work.

The directional light vector is obtained from 3D geometry. In this work, the 3D plane given by the marker is the main reference for determining the light vector. The 2D pixels of the coarse shadow mask are projected onto this 3D plane by using camera calibration matrices. Combining these 3D points with the 3D model of the virtual object, it is possible to calculate which vector obtains the highest Intersection over Union (IoU) between the rendered shadow and the 3D points from the coarse shadow. The size of the search space is a major challenge, requiring a heuristic to find the best vector.

Our main contribution is a method to improve a coarse

shadow input and acquire the main 3D light vector of the real environment, using a single 2D image with a fiducial marker. The proposed method outputs a valid directional light vector for the input real scene given that: there is a single directional light producing the shadows of the real objects; the real shadows are cast on a single plane; the plane is coplanar with the fiducial marker's plane; both the shadow and the virtual object are within the visible frame and are not occluded by any other objects; finally, we assume that the object is placed exactly on top of the fiducial marker.

This paper extends the work presented at SVR 2025 [Yacovenco *et al*., 2025]. The main difference lies in the introduction of a more elaborate subdivision strategy. In our earlier approach, the search space was subdivided recursively into four regions, following a quadtree-like structure. In contrast, the new proposed method performs an initial subdivision by shifting a fixed-size section across the search space. Subsequent subdivisions then compute a new subsection centered on the selected region. These two refinements result in an improvement in the overall accuracy of the directional light vector.

2 Related works

Approaches for light estimation are very diverse and have many levels of complexity. Most methods retrieve either the directional light vector, point light positions in 3D space or the spherical harmonics (SH) representation of the whole illumination conditions of the scene. For example, it is possible to obtain global illumination maps from a whole scene using light probes [Nguyen and Le, 2012; Lopez-Moreno *et al*., 2013]. Other approaches use specialized hardware, as RGBD cameras, to reconstruct directional lighting vectors [Boom *et al*., 2017; Chotikakamthorn, 2015; Gruber *et al*., 2012]. The estimation of multiple light sources and their interaction with objects in the scene are beyond the scope of our work. Our proposal aims to find the directional light vector by using a single RGB image of a scene.

The first approaches to retrieve illumination information from the scene are mainly based on the geometry of the scene [Cao and Foroosh, 2007; Wang and Samaras, 2002]. They rely on known points of a controlled scene as well as one or more 2D images to be able to estimate the main light vector using triangulation or similar methods. Early approaches also make use of image processing techniques, such as image filtering and thresholding, to extract useful features from the image.

The work presented in [Cao and Foroosh, 2007] is based on two known 3D points to extract the directional light from the scene. Along with the projected shadow of these points and two 2D pictures taken from different angles, they can estimate not only the light source, but also the camera parameters.

Wang and Samaras [2002] also rely on geometric features of the scene. The authors work with an arbitrary object of known geometry. Their method searches for critical points in the geometry and for shadows cast from those points, excluding occluded and partially occluded shadows. The output of the method is a set of unit 3D light vectors in spherical space. Both methods need partial information of the scene geometry which, in most cases, is not a trivial task.

Recently, deep learning has become one of the most common approaches for light estimation in the literature. The most recent works employ some form of neural network in their methods [Kán and Kaufmann, 2019; Liu and Wu, 2022; LeGendre *et al*., 2019; Sun *et al*., 2020; Wang *et al*., 2022; Liu *et al*., 2022; Marques *et al*., 2022]. Deep learning methods provide better experimental results than previous approaches and can work in less constrained environments. For example, neural networks can usually estimate some information about the scene geometry. In addition, most deep learning approaches use a single monocular image as input.

The approaches of [LeGendre *et al*., 2019], [Sun *et al*., 2020], [Wang *et al*., 2022], [Liu *et al*., 2022] and [Marques *et al*., 2022] are very similar: they retrieve the SH light from a single monocular image using one or more CNNs. Each work uses a neural network architecture that is trained from a specific dataset, which is the key part of the proposals. The work of [LeGendre *et al*., 2019] uses spheres of known materials to collect image samples for the dataset. The resulting neural network can estimate a relative light map for arbitrary input images. Similarly, [Wang *et al*., 2022] uses outdoor HDR panoramas to estimate light fields to train their architecture. The resulting neural networks are used to insert virtual objects into street scenes. In a more specific scenario, the work [Kán and Kaufmann, 2019] relies on RGBD input and employs a residual convolutional neural network to extrapolate the position of light. In general, works based on Deep Learning use the strategy of training neural networks from multiple images that provide an extensive variation of light fields for a target application. Considering that arbitrary scenes have arbitrary lighting conditions, the datasets end up being very restricted to specific scene contexts.

A different but related approach is to train a neural network to learn the shadows cast for the objects in the scene [Liu *et al*., 2020] instead of its light sources. The task of the neural network is to generate the shadow of virtual objects inserted into the scene. This strategy is promising since the cast shadows furnishes hints about the scene geometry and light source. Our work is based on this possibility and is designed to estimate the directional light vector from a generated shadow.

Finally, in our previous work [Yacovenco *et al*., 2025], the 3D light vector was computed based on an initial shadow mask approximation, where a neural network was employed to generate the input shadow mask. The initial directional light vector estimation was improved through a recursive space subdivision strategy based on a quad tree.

3 Proposed method

In this section, we discuss our proposal to solve the problem of light vector estimation. To ensure our method works properly, we need to enforce the following conditions: we assume there is a single directional light acting on a known virtual object; the shadow cast by that directional light is projected on a single plane; the plane is coplanar with the fiducial marker's plane; both the shadow and the virtual object are within the visible frame and not occluded by any other objects; finally,

we assume the object is placed exactly on top of the fiducial marker.

To begin the process, our method needs two inputs: a 2D frame of the scene with the fiducial marker present; and the 3D virtual object. These inputs are then passed onto a shadow estimator and an AR system. The AR system is responsible for the 3D scene reconstruction and it provides important data such as the marker's 3D position and orientation, as well as the camera position relative to the marker. The shadow estimator is responsible for estimating the shadow in 2D image space. It retrieves the initial shadow estimation M, which is a shadow mask. Since we know the shadow is cast on the same plane where the object is placed, denoted as plane P, we can retrieve that shadow's 3D points given the marker's position and orientation is over P. The 3D points of the shadow compose a set that will be denoted as B.

The virtual object and its cast shadow points in 3D space are used to make the initial light vector estimation. Essentially, we separate some points over the object's surface (blue dots) and all shadow points (green dots) in two sets (Figure 1), respectively A and B, and those points are combined to make vectors that start somewhere on the object's surface and end somewhere on the cast shadow. This combination set C can be expressed as $C = \{\mathbf{c_0}, \mathbf{c_1}, ..., \mathbf{c_k}\}, k = m \cdot n$ where $\mathbf{c_i} = \mathbf{a}_{\lceil \mathbf{i/n} \rceil} - \mathbf{b_i} \mod \mathbf{n}$. m, n and k are respectively the number of elements in sets A, B and C. Each vector in this new set C is a valid candidate for an initial directional light vector $\mathbf{v}$, and $\mathbf{v}$ is updated to match the vector $\mathbf{c_i}$ which casts the shadow that covers the most area compared to the initial shadow estimation M.

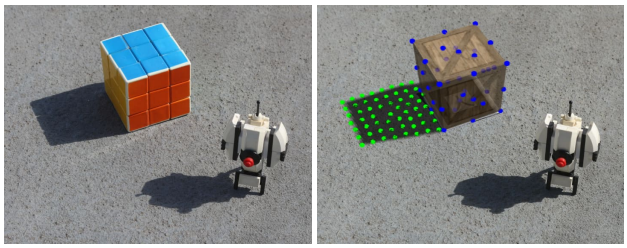


Figure 1. Pictures from the same scene taken with Rubik's cube (a) and a virtual cube (b). The blue dots on the cube's surface and the green dots on the shadow are a subset of the ones which are paired to compute the preliminary candidates, presented in Section 3.2.1. To aid the visualization, not all shadow points are shown, as they would be overly dense.

We cannot guarantee this estimation is the best available since we have worked with discrete points to find $\mathbf{v}$. Therefore, to further improve the directional light estimation, the preliminary vector $\mathbf{v}$ corresponding to that initial estimation is set as the center of a radial search space S in which we search for the best available directional light vector $\mathbf{v}'$. We repeat this process recursively by diminishing the search space until we reach the maximum number of repetitions or until the directional light can no longer be improved.

Figure 2 is an overview of our method's steps and data flow. The colored rectangles and all processing steps are explained in the following sections.

3.1 Scene reconstruction and shadow estimation

The scene with the fiducial marker and the 3D virtual object are used to acquire an estimation of the shadow cast by the virtual object. This object is rendered on top of the marker, which must be present in the scene. Also, both the object and its estimated shadow must fit inside the image.

First, the 2D image of the scene with the marker present is passed onto the AR system. The AR system returns the marker's position and orientation relative to the camera. We render the 3D virtual object on top of the marker by using the projection matrices from the AR system.

We need the 2D mask of the virtual object as input for the shadow estimator. The shadow estimator then outputs a 2D shadow mask. By using the AR system output and the fiducial marker's plane, these points form 3D points in camera reference space. A list of all pairs containing 3D points of the shadow combined with the vertices of the virtual object is produced. These data pairs are then used to compute the preliminary candidate directional light vectors, which is discussed in Section 3.2.

In our method, we assume the marker to be at ground level and coplanar to the ground. The marker's center defines the origin of the plane where the shadow is projected. We need the camera parameters, the plane position, and the orientation relative to each other to define the virtual 3D scene. The camera and the plane are used to map points from 3D to 2D and vice versa.

When the virtual object is placed on top of the marker, we can compute: 1) the 3D virtual object in camera's coordinates; 2) the 2D image with the virtual object inserted in the scene from the estimated camera's perspective; 3) the virtual object's 2D mask by coloring it white on a black background. The object's mask is needed for the shadow estimation.

3.1.1 Shadow estimation

In this step we employ a known shadow estimator to compute the 2D shadow of a virtual object, given one or more real shadows of true objects of the scene image.

The particular shadow estimator chosen in our study, ARShadowGAN [Liu *et al.*, 2020], outputs the estimated shadow mask and only needs the image with the virtual object present and the object's mask as input. It has a pre-trained model and its inputs are easier to handle: the real scene image, the real scene image with the virtual object rendered, and the 2D virtual object mask. Any other shadow estimator can be used instead.

The ARShadowGAN provides a gray scale image where black pixels are background and all other pixels form estimated shadow regions. The closest the pixel is to white, the closest is the confidence that it is indeed a shadow point. We employ Otsu's thresholding method [Otsu, 1979] to get a binary shadow mask. Next, we employ Suzuki's algorithm [Suzuki *et al.*, 1985] to find the multiple shadow contours whose inner pixels are set to white. The region with highest average intensity in the shadow estimation image is selected, and all other pixels are set to black.

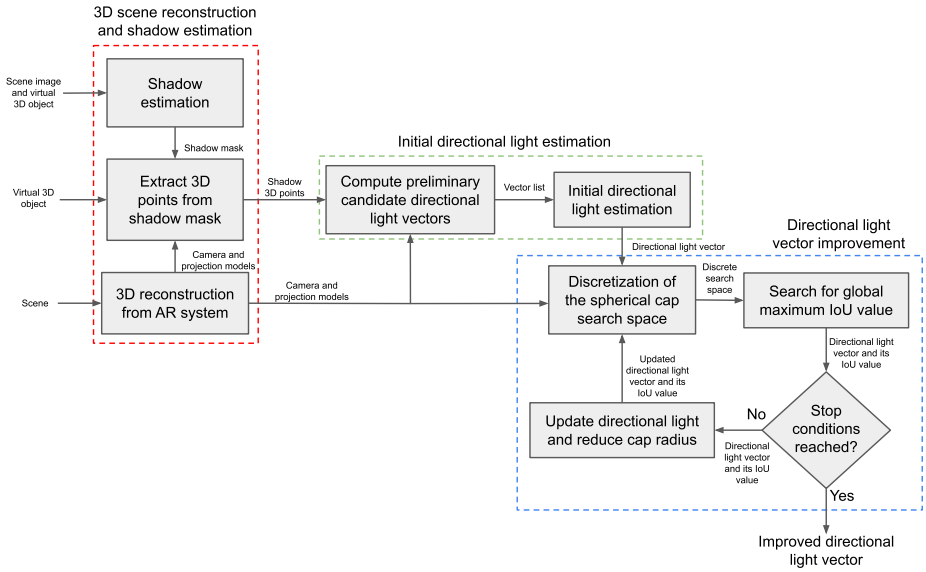


Figure 2. Diagram indicating the steps of the method and data flow.

3.1.2 Extract 3D points from shadow mask

The final shadow needs to be passed from 2D space to 3D space. This is done by mapping the white pixels of the shadow mask to the corresponding points on the marker's 3D plane. We denote this set of 3D points as B.

3.2 Initial directional light estimation

In this step, the 3D object and its 3D estimated shadow are already available. From here, we start our main contribution, which is composed of multiple steps. First, we need a set of candidate directional light vectors. We compare the shadows produced by each vector with the initial shadow estimation mentioned in Section 3.1.1. Then, the best directional light vector is chosen and the next and final step of the method can take place.

In principle, if there is an object projecting a shadow on a plane, the directional light vector can be computed by finding the correct correspondence of a point on the object's surface and a point on the shadow. This is valid, of course, if we assume a continuous space and that the shadow is perfectly cast. With discrete objects and imperfect shadow estimation, the directional light vector cannot be easily retrieved by looking for correspondences. However, it is still a reasonable starting point.

After acquiring the candidate directional light set, we need a criterion to compare each vector in the set and decide objectively which one is a reasonable initial guess. Every candidate vector provides a shadow of the virtual object over the marker's plane. We propose to evaluate them based on how they intersect the estimated shadow. A suitable directional light vector must maximize the area of intersection but also minimize the area of union. In other words, the best candidate vector generates a shadow that best fits the estimated shadow mask.

3.2.1 Compute preliminary candidates

We combine the 3D points of the set B (Sec. 3.1.2) with a set of points A sampled from the virtual object's surface. The set of points A is composed of the virtual object's vertices, edges midpoints and faces midpoints. We then connect all the points sampled from the virtual object's surface to all the 3D points from the shadow mask, forming set C. The list of candidates are the unit vectors from this list of pairwise connections.

3.2.2 Initial directional light estimation

Each vector in the candidate set is used to cast a shadow of the virtual object over the marker's plane. These shadows are then compared to the initial estimated shadow mask, considered as a subset of the true shadow. The vector that casts the shadow that is closest to the estimated shadow mask is selected as the initial directional light vector.

We need to objectively measure the similarity of two shadow masks by computing the overlap of the estimated shadow and a rendered shadow projected from a given candidate directional light vector from C. We classify the pixels marked as shadow in both masks as false positive (FP), false negative (FN), and true positive (TP).

The union of true positives, false positives and false negatives are the union of the two shadow sets. True positives are the intersection. Thus, we use the intersection over union (IoU) as comparison criterion:

$$IoU = \frac{TP}{TP + FP + FN}. \quad (1)$$

3.3 Improving the directional light vector

The vicinity of the initial directional light vector estimation $\mathbf{v}$ can be used to find a vector that improves the IoU criterion. Consider a unit sphere S containing the unit vector $\mathbf{v}$. We define the search space as the sphere cap $S'_{\mathbf{v}}$, over the unit

sphere S, centered at $\mathbf{v}$ with azimuthal angle $\phi = \alpha$, where α is a parameter that defines the size of the cap. A sample of this cap is seen as blue in Figure 3.

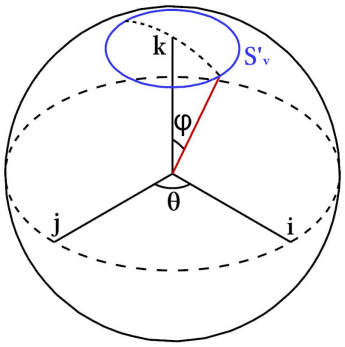


Figure 3. Sphere cap $S'_{\mathbf{v}}$ (blue) in $\mathbf{i}, \mathbf{j}, \mathbf{k}$ reference system.

We parametrize $S'_{\mathbf{v}}$ using a local coordinate system $F = (\mathbf{i}, \mathbf{j}, \mathbf{k})$:

$$\mathbf{i} = \frac{\mathbf{z} \times \mathbf{v}}{||\mathbf{z} \times \mathbf{v}||}, \quad \mathbf{j} = \mathbf{i} \times \mathbf{v}, \quad \mathbf{k} = \mathbf{v}, \tag{2}$$

where vector $\mathbf{z}$ is a unit vector in the positive z axis. When $\mathbf{v}$ is aligned with $\mathbf{z}$, the canonical coordinates are used. The parametric equation for the sphere cap $S'_{\mathbf{v}}$ is:

$$S'_{\mathbf{v}} = \{(i,j,k) \mid i = \sin\phi \cdot \cos\theta,\ j = \sin\phi \cdot \sin\theta, \\ k = \cos\phi,\ 0 \le \phi \le \alpha, 0 \le \theta \le 2\pi\}, \tag{3}$$

where we need to compute the IoU. Consider we have a smooth function $\mu_{\mathbf{v}}$ that computes the IoU for any point in $S'_{\mathbf{v}}$. We propose to search regions over the cap where the total IoU w given by the integral:

$$w = \int_{\theta_1}^{\theta_2} \int_{\phi_1}^{\phi_2} \mu_{\mathbf{v}}(\sin\phi \cdot \cos\theta, \sin\phi \cdot \sin\theta, \cos\phi)\, d\theta\, d\phi, \tag{4}$$

are a local maximum, where the angles θ_1, θ_2, ϕ_1, ϕ_2 define a sector over the cap. The total IoU w is the density of IoU in the sector. Higher the w, higher the chances to find a better directional light vector inside the sector. Unfortunately, the function $\mu_{\mathbf{v}}$ cannot be solved in continuous space. We need to proceed with a numerical solution for the integration (Eq. 4) and a method to search for a local maximum IoU. Therefore, the sphere cap is discretized for IoU computation.

3.4 Discretization of the spherical cap search space

Consider C a matrix of $m \times n$ elements. We sample $S'_{\mathbf{v}}$ by mapping the C onto the sphere cap, where m represents the phi with angles from 0 to α, and n represents θ with angles from 0 to 2π. The ranges $[0, \alpha]$ and $[0, 2\pi]$ are mapped to $A = \{\frac{1}{m} \cdot \alpha, \frac{2}{m} \cdot \alpha, ..., \frac{m-1}{m} \cdot \alpha, \alpha\}$ and $B = \{\frac{1}{n} \cdot 2\pi, \frac{2}{n} \cdot 2\pi, ..., \frac{n-1}{n} \cdot 2\pi, 2\pi\}$, respectively. This sampling method is illustrated by Algorithm 1.

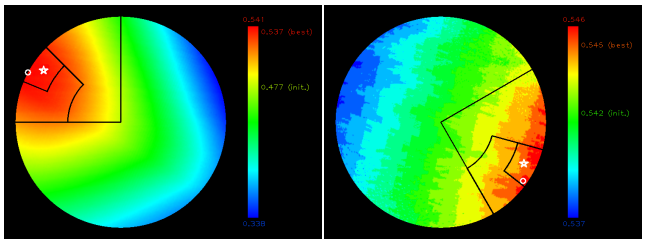
Algorithm 1 Sphere cap sampling

Input: center vector $\mathbf{v}$, opening angle α, matrix dimensions m and n
Output: discrete search space C
Require: $\mathbf{v}$ is a unit vector; $\alpha, m, n > 0; m, n \in \mathbb{N}$

```
function sphereCapSampling(v, α, m, n)
    let C be a m × n matrix
    let u be a unit vector in ℝ³ perpendicular to v
    for i ← 0 to m do
        c_{i,0} ← v · cos(i·α/m) + (u × v) · sin(i·α/m) +
                  u · (u · v) · (1 − cos(i·α/m))
        for j ← 1 to n do
            c_{i,j} ← c_{i,0} · cos(j·2π/n) + (v × c_{i,0}) · sin(j·2π/n) +
                      v · (v · c_{i,0}) · (1 − cos(j·2π/n))
        end for
    end for
    return C
end function
```

3.4.1 Search for a maximum IoU value

We propose evaluating regions around the highest IoU values rather than considering sampled points in isolation. Initially, a fixed-size section of 90° is shifted across the search space by the parameter β, and the section with the highest IoU integration per unit area is selected. Then this section is subdivided a number of times as defined by the parameter s. After the final subdivision, a smaller cap centered on the selected section is sampled and the process is repeated recursively up to r times. This subdivision strategy is illustrated in Figure 4.



(a) **(b)**

Figure 4. Top view of the subdivision of the sphere cap $S'_{\mathbf{v}}$ during the first (a) and last (b) recursions for scene r14. Parameters are: $m = 257, n = 513, \alpha = 3.0, r = 3, s = 3, \beta = 30°$. The white circle indicates the local maximum of $S'_{\mathbf{v}}$ and the white star indicates the point selected by our method.

For each recursion, the center point of the new cap $\mathbf{v}'$ is set at the center of the previous section, and its radius α' is the distance from its center to its furthest vertex. This ensures that the new cap encompasses the previous section.

The Algorithm 2 provides an overview of how the method searches for the maximum value, denoted as g. Each point of the sphere cap $S'_{\mathbf{v}}$, when connected to the origin, represents a vector, and each vector is associated with an IoU value, as mentioned in Section 3.2.2. A discrete implementation of the function $\mu_{\mathbf{v}}$ was used to acquire the IoU value of a vector by comparing the shadow mask produced by it to the first estimation of he shadow mask M.

We search for the maximum IoU by using the IoU integral image computed for a set of sectors. Each sector comprises a region of area inside the cap. In this work, the cap is partitioned into sectors regularly distributed in $\phi, \theta \in [0, \alpha] \times [0, 2\pi]$. Using an integral image of the sam-

Algorithm 2 Enhanced directional light vector

Input: directional light estimation vector $\mathbf{v}$; opening angle α; number of resamples r; number of subdivisions s; matrix dimensions m and n; shadow mask M; shift angle β
Output: enhanced directional light estimation vector $\mathbf{v}'$
Require: $r, m, n > 0; r, s, m, n \in \mathbb{N}; \beta \mid 360°$

function enhanceDirectionalLight($\mathbf{v}$, α, r, s, m, n, M, β)
  $\mathbf{v}' \leftarrow \mathbf{v}$
  // Compute sampling C and its quantization Q
  let C and Q be $m \times n$ matrices
  $C \leftarrow \text{sphereCapSampling}(\mathbf{v}, \alpha, m, n)$
  for each vector $c_{i,j}$ in C **do**
    $q_{i,j} \leftarrow \text{computeIoU}(M, c_{i,j})$
  end for
  // Compute integral image T
  let T be a $m \times n$ matrix
  for each element $t_{i,j}$ in T **do**
    $t_{i,j} \leftarrow q_{i,j} - q_{i-1,j} - q_{i,j-1} + q_{i-1,j-1}$
  end for
  // Find vector with best IoU in the most dense IoU section
  $\mathbf{v}', \alpha' \leftarrow \text{retrieveSector}(c_{0,0}, s, C, T, m, n, \beta)$
  // If reached recursion limit, return this vector
  if $r \leq 0$ **then**
    return $\mathbf{v}'$
  else
    // Reduce radius and repeat until conditions are met
    return enhanceDirectionalLight($\mathbf{v}'$, α', $r-1$, s, m, n, M)
  end if
end function

pled cap $S'_{\mathbf{v}}$, we can easily compute the IoU integral for each sector in O(1). We search for the sector with the highest integrated IoU integral per unit area, i.e. the sector with highest average IoU. We assume that this sector is likely to contain the local maximum IoU g.

The Algorithm 3 computes the initial sector by shifting a fixed-size region across the search space a predefined number of times, as determined by the shift angle parameter β. It then selects the sector corresponding to the region with the highest IoU values and calls Algorithm 4 to perform subsequent subdivisions.

Algorithm 3 Retrieve first sector

Input: center of space vector $\mathbf{v}$; number of subdivisions s; discrete search space C; integral image T; boundaries m, n; shift angle β
Output: center vector of most dense subsection $\mathbf{v}'$, opening radius α'
Require: unit vector $\mathbf{v}$; $s \in \mathbb{N}$

function retrieveSector($\mathbf{v}, s, C, T, m, n, \beta$)
  $\delta \leftarrow \frac{360°}{\beta}$
  for $i \leftarrow 0$ **to** δ **do**
    $x_i \leftarrow t_{m,\delta(i+1)} - t_{0,\delta(i+1)} - t_{m,\delta i} + t_{0,\delta i}$
  end for
  $x_{max} \leftarrow \max(x_i)$
  let k be the index of x_{max}
  return retrieveSubSector($\mathbf{v}', s-1, C, T, 0, m, k\delta, k\delta + \frac{\theta}{4}$)
end function

The Algorithm 4 recursively searches for an improved local maximum IoU value g. At each iteration, the area is subdivided into four sectors using a quad-tree scheme, along with a fifth sector centered in the current region. The vertices of this fifth sector are the center of the four quad-tree quadrants, defining a centered region. Figure 12b shows one example of this sector being selected twice in the recursive process. Of the five sectors, the one with the highest average IoU is selected. The process continues until the IoU value g' no longer improves or the maximum number of recursion steps is reached.

Algorithm 4 Retrieve center vector of the most dense subsection from integral image

Input: center of section vector $\mathbf{v}$; number of subdivisions s; discrete search space C; integral image T; boundaries $\phi_1, \phi_2, \theta_1, \theta_2$
Output: center vector of most dense subsection $\mathbf{v}'$, opening radius α'
Require: unit vector $\mathbf{v}$; $\delta, \phi_1, \phi_2, \theta_1, \theta_2 \in \mathbb{N}; \phi_2 > \phi_1; \theta_2 > \theta_1$

function retrieveSubSector($\mathbf{v}, s, C, T, \phi_1, \phi_2, \theta_1, \theta_2$)
  // Subdivide section, return subsection with highest IoU
  $\overline{\phi} \leftarrow \lfloor \frac{\phi_1+\phi_2}{2} \rfloor, \quad \phi_3 \leftarrow \lfloor \frac{\phi_1+\overline{\phi}}{2} \rfloor, \quad \phi_4 \leftarrow \lfloor \frac{\phi_2+\overline{\phi}}{2} \rfloor$
  $\overline{\theta} \leftarrow \lfloor \frac{\theta_1+\theta_2}{2} \rfloor, \quad \theta_3 \leftarrow \lfloor \frac{\theta_1+\overline{\theta}}{2} \rfloor, \quad \theta_4 \leftarrow \lfloor \frac{\theta_2+\overline{\theta}}{2} \rfloor$
  $b_0 \leftarrow t_{\overline{\phi},\overline{\theta}} - t_{\phi_1,\overline{\theta}} - t_{\overline{\phi},\theta_1} + t_{\phi_1,\theta_1}$
  $b_1 \leftarrow t_{\phi_2,\overline{\theta}} - t_{\overline{\phi},\overline{\theta}} - t_{\phi_2,\theta_1} + t_{\overline{\phi},\theta_1}$
  $b_2 \leftarrow t_{\overline{\phi},\theta_2} - t_{\phi_1,\theta_2} - t_{\overline{\phi},\overline{\theta}} + t_{\phi_1,\overline{\theta}}$
  $b_3 \leftarrow t_{\phi_2,\theta_2} - t_{\overline{\phi},\theta_2} - t_{\phi_2,\overline{\theta}} + t_{\overline{\phi},\overline{\theta}}$
  $b_4 \leftarrow t_{\phi_3,\theta_3} - t_{\phi_4,\theta_3} - t_{\phi_3,\theta_4} + t_{\phi_4,\theta_4}$
  $b_{max} \leftarrow \frac{\max(b_0,b_1,b_2,b_3,b_4)}{(\overline{\phi}-\phi_1)\cdot(\overline{\theta}-\theta_1)}$
  let $\phi'_1, \phi'_2, \theta'_1, \theta'_2$ be the boundaries of b_{max}
  $\overline{\phi}' \leftarrow \lfloor \frac{\phi'_1+\phi'_2}{2} \rfloor, \quad \overline{\theta}' \leftarrow \lfloor \frac{\theta'_1+\theta'_2}{2} \rfloor, \quad \mathbf{v}' \leftarrow c_{\overline{\phi}',\overline{\theta}'}$
  // If reached subdivision limit, return this vector and the new opening radius
  if $s \leq 0$ **then**
    $\alpha'_1 \leftarrow \arccos\left(\frac{\mathbf{v}' \cdot c_{\phi'_1,\theta'_1}}{|\mathbf{v}'| \cdot |c_{\phi'_1,\theta'_1}|}\right)$,
    $\alpha'_2 \leftarrow \arccos\left(\frac{\mathbf{v}' \cdot c_{\phi'_1,\theta'_2}}{|\mathbf{v}'| \cdot |c_{\phi'_1,\theta'_2}|}\right)$,
    $\alpha'_3 \leftarrow \arccos\left(\frac{\mathbf{v}' \cdot c_{\phi'_2,\theta'_1}}{|\mathbf{v}'| \cdot |c_{\phi'_2,\theta'_1}|}\right)$,
    $\alpha'_4 \leftarrow \arccos\left(\frac{\mathbf{v}' \cdot c_{\phi'_2,\theta'_2}}{|\mathbf{v}'| \cdot |c_{\phi'_2,\theta'_2}|}\right)$,
    $\alpha'_{max} \leftarrow \max(\alpha'_1, \alpha'_2, \alpha'_3, \alpha'_4)$
    return $\mathbf{v}', \alpha'_{max}$
  else// Otherwise, continue search
    return retrieveSubSector($\mathbf{v}', s-1, C, T, \phi'_1, \phi'_2, \theta'_1, \theta'_2$)
  end if
end function

The proposed heuristic proved to be effective and straightforward to implement. Despite that the search for the global maximum could be improved, the time cost is prohibitive. Our method provides a fair result with a low time complexity. It is important to notice, however, that the cost for computing IoU for each sample dominates the time complexity.

4 Experimental results

In this section, our goal is to evaluate the proposed method's accuracy when finding the highest IoU and the corresponding directional light vector within the search space $S'_{\mathbf{v}}$.

4.1 Scene sets

We assembled real world scenes, gathered from photos, and synthetic scenes, assembled virtually, to evaluate the method in multiple environments. The synthetic scenes are divided into two subsets: synthetic scenes with solid color backgrounds; and synthetic scenes with textured background, as explained in Section 4.1.2. Both scene sets and our source code are available in `https://github.com/yaleksander/ar-server-app`.

4.1.1 Real world scenes

For real world scenes, the photos were taken around 3p.m. on a sunny day. The camera used to take the photos was a Canon T5i with 18-55mm lenses. The directional light vector is unknown for all images.

We empirically estimate the directional light vector from the images. The Estimated Ground Truth (EGT) is defined by the vector that best matches the shadow of the real objects in the scene. The environment must allow for two pictures to be taken in sequence, one with a fiducial marker present, and one with a real cube of same size as the virtual cube placed exactly on top of the marker. We use the Rubik's cube and a larger wooden cube as real objects. Everything in the scene must remain the same in both pictures, except the real cube and its shadow. The pictures were taken on a tripod. The real cube's shadow must be cast on a single plane. The EGT is the vector that best replicates the real shadow by using a virtual cube model in the same pose of the real one. In the following experiments, the EGT is compared to our method's output to evaluate the estimation error.

This process can be visualized on Figure 5. Figure 5a is the real world picture representing the ground truth (GT). Figure 5b is the picture taken immediately after removing the Rubik's cube from the fiducial marker. Figure 5c is the same picture but with the virtual cube placed on top of the marker, with the EGT shadow.

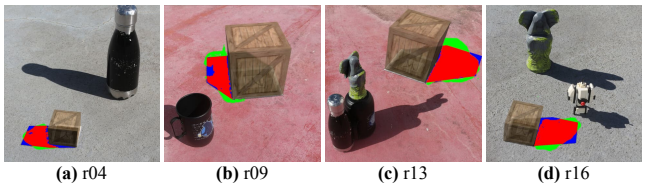
(a) GT **(b)** Empty scene **(c)** EGT

Figure 5. Visualization of the GT (a), fiducial marker (b), and EGT (c) for the real world scene r08.

The experiments are executed by submitting an image with a fiducial marker to the method. There are 16 images in total, labeled from r01 to r16. Each picture must also have an associated initial shadow estimation, which is generated by ARShadowGAN [Liu *et al*., 2020]. A subset of the real world images dataset is presented in Figure 6.

4.1.2 Synthetic scenes

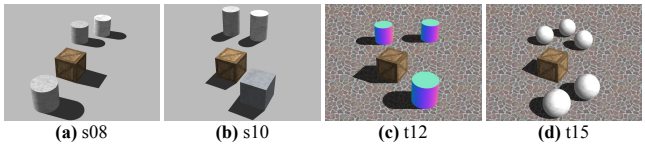
The synthetic scenes complement the evaluation of the method's accuracy. It consists of 32 virtual scenes modeled to mimic the real ones. We developed a simple ThreeJS application to control the directional light and place objects of basic shapes on the scene, such as cubes, spheres, and cylinders, to generate the synthetic scenes.



(a) r04 **(b)** r09 **(c)** r13 **(d)** r16

Figure 6. Subset of the real world dataset. The green area is the input shadow mask computed by the ARShadowGAN, the blue area is the shadow cast by the initial directional light vector $\mathbf{v}$ and the red area is the intersection between them.

In this set the ground truth directional light vector (GT) is known and the object's shadow cast by the GT can be used as input for our method. Its purpose is to evaluate the method's accuracy under nearly ideal conditions. The synthetic scenes were divided into two sets: scenes with solid color background; and scenes with textured background. The purpose is to assess the impact of the type of background. The two sets are identical except for their background. Figure 7 presents a subset of these datasets.



(a) s08 **(b)** s10 **(c)** t12 **(d)** t15

Figure 7. Subset of the synthetic datasets. The object's shadow is cast by the GT.

4.2 Quantitative results

In this section, we present and discuss the quantitative aspects of our method. Two evaluation criteria are relevant when analyzing the results: the angle between the ground truth light vector and the output vector, and the IoU (Eq. 1) between the shadow cast by the GT and the estimated shadow.

Our method has five parameters that can affect the final result: cap opening angle α, shift parameter β, number of IoU resamplings r, number of subdivisions s, and IoU sampling resolution $m \times n$. To evaluate which set of parameters provides the best and most consistent results on average, an ablation study was performed by combining different values for each parameter, as explained in Section 4.2.1.

4.2.1 Ablation

The ablation seeks to understand how various sets of parameters α, β, r and s affect the output directional light estimate. We have empirically chosen the angles for $\alpha \in \{5, 15, 30, 45\}$, and for $\beta \in \{15, 30, 45\}$. Both r and s assume values in the set $\{0, 1, 2, 3\}$. The combination of this set of parameters was applied to reconstruct the real world scenes r01 to r16. In a preliminary investigation, we used three different resolutions for the IoU sampling: 97×449, 129×513 and 161×577. In our experimental context, the best average accuracy was found with 129×513 for all scenes. This resolution is used in the following results.

The IoU tends to increase as α, β, r and s increase to a certain value. The angle $\alpha = 5°$ gives the worst result in most cases because the EGT is usually further away from the initial vector $\mathbf{v}$. The broader angle $\alpha = 45°$ tends to map more cap area to each point of the fixed resolution 129×513 IoU sampling matrix. Thus, different α angles result in different IoU sampling, possibly providing slightly different maximum IoU values. The best overall α is $30°$ for this experiment.

Taking into account all 16 scenes, the combination that provided the most consistent results was $\alpha = 30°$, $\beta = 30°$, $r = 3, s = 2$. All ablation results can be reproduced from the source code and datasets available in `https://github.com/yaleksander/ar-server-app`.

Setting $\beta = 15°$ results in more sectors shifted to find the highest integral IoU in the first iteration. Interestingly, it includes all the sectors obtained by making $\beta = 30°$. However, this parameter plays a role only for the first iteration. Thus $\beta = 15°$ provides a better initial IoU that does not lead to the highest IoU in the subsequent subdivisions.

Among all 16 images of the dataset, we highlight the results for three specific scenes: one considered an example with low difficulty (r13), one of medium difficulty (r09), and one considered a hard case for our method (r16).

In r13, the input is considered easier since the shadow generated by ARShadowGAN is geometrically close to the expected one, pointed to the correct direction (Fig. 6c). The best result yielded a light direction only $2.28°$ from the EGT vector. In r09, the input has a concave shape that extends beyond the expected shadow. This complicates the search for the directional light. However, the error decreases as the IoU increases because the EGT shadow is roughly a subset of the input shadow. With $\alpha = 30°$, the cap is large enough to include the best IoU result, with an angle error of $11.87°$. In r16, the input shadow (green area of Fig. 6d) has a different direction than the EGT shadow (blue area of Fig. 6d). The input is distributed towards the top right direction whereas the real objects' shadow points to the bottom right direction.

The accuracy of the method depends directly on the input shadow estimation. The r16 example shows that the final direction light vector diverges from the EGT as IoU increases. The angle error worsened from $14.63°$ (initial estimate) to $16.26°$ (maximum IoU). Taking into account the entire dataset with 16 scenes, only two (r02 and r16) had angle error worsened as IoU increased.

4.2.2 Results for synthetic scenes

The experiments on synthetic scenes were executed in two scenarios: with the GT cast shadow as input; and with the shadow estimated by the ARShadowGAN as input. The goal is to evaluate the impact of our method applied to the true shadow, and applied to the ARShadowGAN output shadow.

Results for ground truth shadow input As Section 4.2.1 shows, the best average results are recorded for $\alpha = 30°, \beta = 30°, r = 3, s = 2$ and resolution 129×513.

The experiments were executed on all synthetic scenes, both with solid color and textured background. The results were identical for both sets because the cast shadow is identical for their corresponding scenes.

In this case, our method obtained directional light vectors with an angular error below 1° when the true shadow is given as input.

Results for estimated shadow input The results for estimated shadow shows that the method's accuracy depends on the input shadow quality. Even with average angle errors of 10.07° (solid background) and 10.98° (textured background), the shadow cast in the virtual scene is plausible and appealing as discussed in Section 4.3.

4.2.3 Execution times

The experiments were executed on a computer with a GeForce GTX 1050Ti Nvidia GPU, i7-11700K Intel processor and 48GB RAM. The average runtime for each loop of the method given a certain $m \times n$ resolution is presented on Table 1.

Table 1. Average execution time in seconds for each loop of the method's implementation.

$m \times n$ resolution	Rendered frames	Avg. time (seconds)
10×36	360	2
97×449	43553	228
129×513	66177	337
161×577	92897	487
257×1025	263425	1447

The execution times were measured by running the method on the real world scenes with fixed parameters $\alpha = 30°$, $\beta = 90°$, $r = 1$, $s = 0$ and resolution according to the left column of Table 1. The parameters that impact the execution time are the resolution $m \times n$ and the number of resamplings r. A higher resolution means more frames to render, and more resamplings mean more loops to the method. The time cost increases linearly as the resolution increases for a fixed r.

4.3 Qualitative results

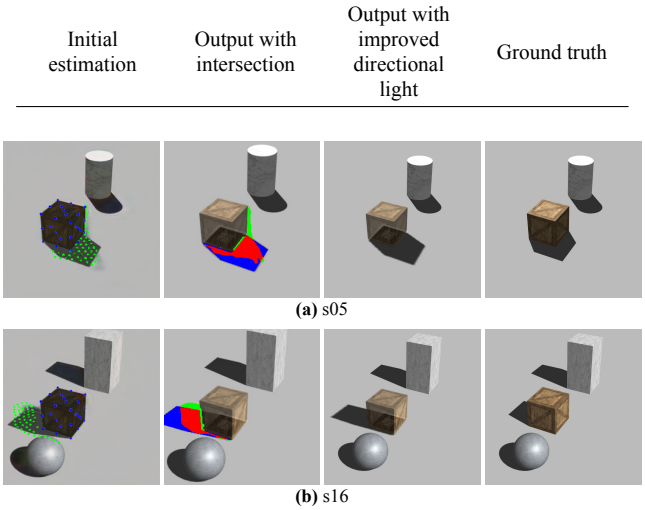
To qualitatively evaluate our method's graphic output, we compare the shadow cast by the output vector $\mathbf{v}'$ to the input. The results are divided in sections for each dataset: synthetic with solid color background and real world pictures. Most of our results are visually similar to the GT, for synthetic scenes, or EGT, for real world scenes.

4.3.1 Synthetic pictures with solid color background

The objective of qualitative experiments is to evaluate how our method improves on a rough input shadow shape. Therefore, the estimated input shadow, and not the GT shadow, is compared to the output shadow (Figure 8).

Figure 8 presents the results of two synthetic scenes. It is divided in four columns: the initial estimation, the intersection between input and output shadows, the output scene improved by the computed directional light vector, and the GT shadow. In the leftmost column, only a subset of the shadow points are shown. That is because otherwise they would be too dense to be seen as individual points.

In all experiments, the wooden cube is the known virtual object, and the rest of the scene is treated as the background.



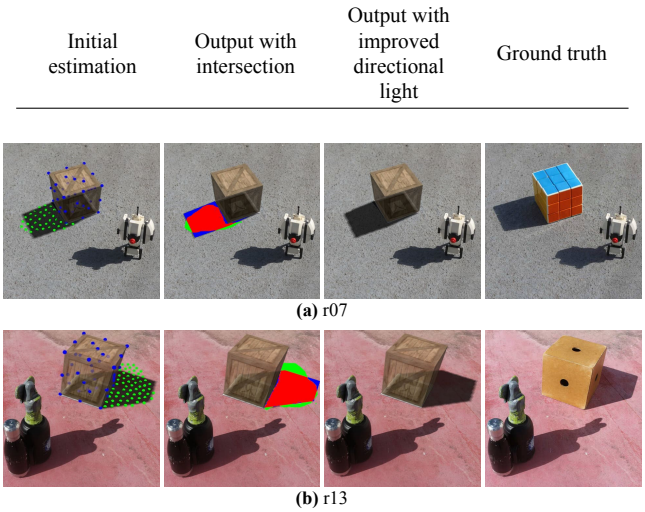
(a) s05

(b) s16

Figure 8. Shadow estimation for two scenes with solid color background. The leftmost column represents the object points (blue) and estimated shadow points (green). The next column from left to right is the intersection (red) between the estimated shadow (green) and the final output shadow (blue). The last two columns are respectively the final output shadow and the ground truth.

4.3.2 Real world pictures

The estimated shadows for the real world scenes have more consistent direction and length, and it tends to produce suitable output shadows.



(a) r07

(b) r13

Figure 9. Shadow estimation of a subset of the real world scenes. The leftmost column represents the object points (blue) and estimated shadow points (green). The next column from left to right is the intersection (red) between the estimated shadow (green) and the output shadow (blue). The last two columns are respectively the final output shadow and the ground truth.

The input shadow in Figure 9a is almost aligned with the GT, and the output fits it very closely (angular error relative to the EGT of 13.16°). It also has good visual quality. Figure 9b (angular error relative to the EGT of 11.87°) also produced a shadow of good quality, although the shadows are slightly shorter than the GT.

The results show that our method is capable of enhancing the visual quality of the shadow. When examining the shape of the input and output shadows, our method significantly improves the ARShadowGAN shadow estimation.

4.4 Comparison to the previous method

Compared with our previous method, the current approach yields improved results (Table 2). For example, Figure 10 presents the first and last subdivisions for scene with the previous method (Fig. 10a and 10b) and the proposed method (Fig. 10c and 10d). The IoU obtained with the previous approach was 0.677, whereas the new approach achieved an IoU of 0.721.

Table 2. Improvement in IoU values for the last subdivision of the last recursion.

Scene	Old method IoU	New method IoU	Improvement
r01	0.719	0.717	−0.28%
r02	0.613	0.612	−0.16%
r03	0.745	0.784	5.28%
r04	0.604	0.598	−0.99%
r05	0.664	0.690	3.92%
r06	0.704	0.703	−0.14%
r07	0.641	0.641	0.00%
r08	0.705	0.714	1.26%
r09	0.618	0.635	2.79%
r10	0.529	0.549	3.81%
r11	0.677	0.721	6.55%
r12	0.593	0.616	3.94%
r13	0.703	0.716	1.87%
r14	0.522	0.546	4.60%
r15	0.502	0.517	3.03%
r16	0.724	0.699	−3.41%
AVG	**0.641**	**0.654**	**2.00%**

The distance between the center of the selected sector and the local maximum is slightly larger in Figure 10a than in Figure 10c. Another distinction can be observed when comparing Figures 10b and 10d. Figure 10d has a higher local maximum than Figure 10b. This can be verified by comparing the lowest and highest IoU values of each space, indicated by the gradient scale on the right side (0.657 and 0.679 vs 0.718 and 0.721, as seen in Figures 10b and 10d, respectively). At this stage, the search space becomes sufficiently restricted that any further improvements are of marginal significance.

4.5 Limitations

The main problem of our method is when the length and/or direction of the input shadow are very distinct from the ground truth. In these cases, our method tends to output an incorrect shadow.

In some cases, the results might be perceived as incoherent when visually compared with the ground truth. Some examples are shown in Figure 11. Our method finds the directional light vector that casts the shadow with highest IoU.

Figures 11a and 11b depict a scene in which the output shadow has a visibly shorter length than expected when compared side by side with the GT. In Figures 11c and 11d, the visual error between the output shadows and the corresponding GT is very clear. The output result when the input shadow has a wrong direction is visually odd.

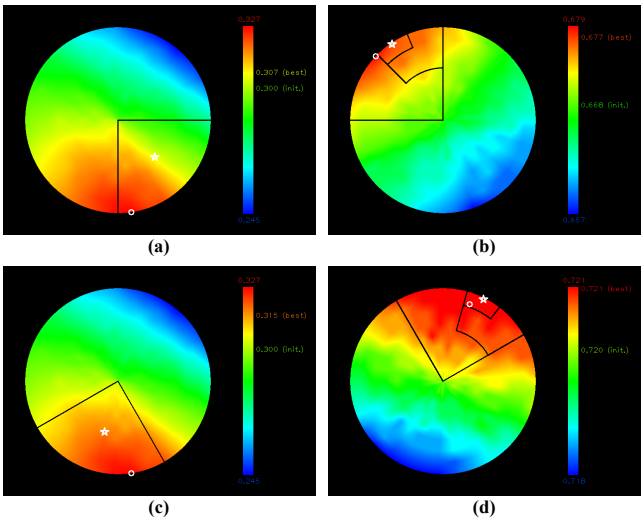


Figure 10. Comparison between the previous and current subdivision strategies for scene r11. The white circle indicates the local maximum of $S'_{\mathbf{v}}$ and the white star indicates the point selected by our method. (a) and (b) are the first and last recursions for the previous method, while (c) and (d) are the first and last recursions for the current method, respectively.

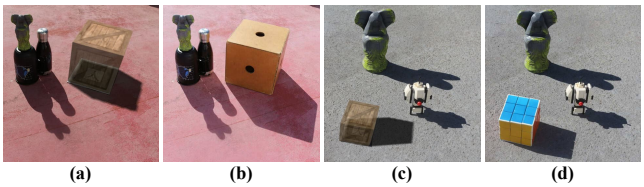


Figure 11. Results with input shadows with incorrect lengths (a) and (b); and with a input shadows with incorrect directions (c) and (d).

5 Discussion

Compared to our previous work [Yacovenco *et al.*, 2025], the proposed method demonstrates a general improvement in finding the directional light vector. The average improvement in IoU in the 16 evaluated scenes was **2**% (Table 2). This is due to more flexible subdivisions that tend to encompass the local maximum.

In the previous method, the spherical cap $S'_{\mathbf{v}}$ was iteratively subdivided into four sections in a quad-tree scheme, repeated a number of times defined by the subdivision parameter s. However, during the analysis of the results, it became evident that the region with the highest average IoU was occasionally partially excluded, as illustrated in Figure 12a.

To address this issue, the proposed method evaluates a larger set of overlapping sections by considering regions that intersect with their neighbors. The section size remains unchanged relative to the previous approach; however, during the first subdivision at each recursion level, starting from the first quadrant, a 90° region is shifted across the search space a predefined number of times until it returns to the initial quadrant. In this work, the number of shifts of the best result was 12, resulting in angular shifts of 30°.

From the second subdivision onward, the selected section is subdivided into five subsections in a enhanced quad-tree scheme. The average IoU of a fifth subsection is computed, positioned at the center of the parent section (center of the quad-tree), as illustrated in Figure 12b.

The proposed subdivision strategy yields notably improved results during the selection of the initial section. However,

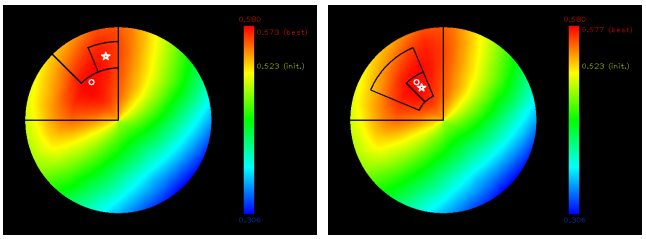


Figure 12. Difference in subdivision strategy between the previous (a) and current (b) methods for scene r09. The white circle indicates the local maximum of $S'_{\mathbf{v}}$ and the white star indicates the point selected by our method.

from the second subdivision onward, the accuracy gains become less important, since the heuristic might select a suboptimal region with a lower IoU compared to a neighboring region.

6 Conclusions

In this work, we presented an extension of our previous method for estimating the main directional 3D light vector of a scene from a 2D shadow mask input. The proposed approach introduces a modified subdivision strategy aimed at improving estimation accuracy.

The method needs an image with a fiducial marker present and a 2D shadow mask as input. Assuming the shadow is cast on the marker's plane, it acquires the 3D contour of the shadow. The shadow points are combined with the points of a virtual object that is assumed to cast that shadow, generating a set of candidates for the initial directional light estimation. From this set, the vector that casts the shadow closest to the input shadow mask with maximum IoU is chosen as the initial estimation.

From this first estimation, our method looks for the best directional light vector within a search space. The search space is represented as a sphere cap $S'_{\mathbf{v}}$ of the unit sphere S centered at the origin (Section 3). $S'_{\mathbf{v}}$ is centered on the vector $\mathbf{v}$, which represents the initial estimation of the directional light. The method then recursively enhances the directional light by finding the maximum by recursively subdividing and resampling the sphere cap.

The results show that our method can improve object shadows by finding an overlap with high IoU with the input shadow. Through this process, we can retrieve a suitable directional light vector, producing coherent shadows of the virtual object. Qualitative results show that our method produces a suitable output even when the angle error is high ($\geq 10°$). In general, the experimental results indicate a gain of **2**% compared to the previous method.

The method proposed in this work is limited by its dependence on the accuracy of the shadow input, particularly with respect to its length and direction. This limitation may be addressed in future work by exploring the use of multiple shadow estimation techniques. An ensemble-based approach could then be employed to select the shadow representation and corresponding vector that best fit the scene.

Declarations

Authors' Contributions

AY, RS and MV contributed to the conception of this study. AY performed the experiments. AY, RS and MV analyzed the results. RS, MV and LM reviewed the Materials and Methods. AY and RS are the main contributors and writers of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets and software used in the current study are available at https://github.com/yaleksander/ar-server-app

The images corresponding to the scenes used for the experiments are available at https://github.com/yaleksander/ar-server-app/tree/main/threejs/my-images

The tables with the full results of the ablation study are available at https://github.com/yaleksander/ar-server-app/blob/main/ablation-new-method.pdf

References

Azuma, R. T. (1997). A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4):355–385. DOI: 10.1162/pres.1997.6.4.355.

Boom, B. J., Orts-Escolano, S., Ning, X. X., McDonagh, S., Sandilands, P., and Fisher, R. B. (2017). Interactive light source position estimation for augmented reality with an rgb-d camera. *Computer Animation and Virtual Worlds*, 28(1):e1686. DOI: 10.1002/cav.1686.

Cao, X. and Foroosh, H. (2007). Camera calibration and light source orientation from solar shadows. *Computer Vision and Image Understanding*, 105(1):60–72. DOI: 10.1016/j.cviu.2006.08.003.

Chen, Y., Wang, Q., Chen, H., Song, X., Tang, H., and Tian, M. (2019). An overview of augmented reality technology. In *Journal of Physics: Conference Series*, volume 1237, page 022082. IOP Publishing. DOI: 10.1088/1742-6596/1237/2/022082.

Chotikakamthorn, N. (2015). Near point light source location estimation from shadow edge correspondence. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 30–35. IEEE. DOI: 10.1109/iccis.2015.7274543.

Frahm, J.-M., Koeser, K., Grest, D., and Koch, R. (2005). Markerless augmented reality with light source estimation for direct illumination. In *Conference on Visual Media Production CVMP, London*, pages 211–220. IET Stevenage. Avaialble at:`https://www.researchgate.net/publication/228737475_Markerless_augmented_reality_with_light_source_estimation_for_direct_illumination`.

Gruber, L., Richter-Trummer, T., and Schmalstieg, D. (2012). Real-time photometric registration from arbitrary geometry. In *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 119–128. IEEE. DOI: 10.1109/ismar.2012.6402548.

Kán, P. and Kaufmann, H. (2019). Deeplight: light source estimation for augmented reality using deep learning. *The Visual Computer*, 35:873–883. DOI: 10.1007/s00371-019-01666-x.

LeGendre, C., Ma, W.-C., Fyffe, G., Flynn, J., Charbonnel, L., Busch, J., and Debevec, P. (2019). Deeplight: Learning illumination for unconstrained mobile mixed reality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5918–5928. DOI: 10.1109/cvpr.2019.00607.

Liu, C., Wang, L., Li, Z., Quan, S., and Xu, Y. (2022). Real-time lighting estimation for augmented reality via differentiable screen-space rendering. *IEEE Transactions on Visualization and Computer Graphics*, 29(4):2132–2145. DOI: 10.1109/tvcg.2022.3141943.

Liu, D., Long, C., Zhang, H., Yu, H., Dong, X., and Xiao, C. (2020). Arshadowgan: Shadow generative adversarial network for augmented reality in single light scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8139–8148. DOI: 10.1109/cvpr42600.2020.00816.

Liu, D. S.-M. and Wu, S.-J. (2022). Light direction estimation and hand touchable interaction for augmented reality. *Virtual Reality*, 26(3):1155–1172. DOI: 10.1007/s10055-022-00624-8.

Lopez-Moreno, J., Garces, E., Hadap, S., Reinhard, E., and Gutierrez, D. (2013). Multiple light source estimation in a single image. In *Computer Graphics Forum*, volume 32, pages 170–182. Wiley Online Library. DOI: 10.1111/cgf.12195.

Marques, B. A. D., Clua, E. W. G., Montenegro, A. A., and Vasconcelos, C. N. (2022). Spatially and color consistent environment lighting estimation using deep neural networks for mixed reality. *Computers & Graphics*, 102:257–268. DOI: 10.1016/j.cag.2021.08.007.

Meilland, M., Barat, C., and Comport, A. (2013). 3d high dynamic range dense visual slam and its application to real-time object re-lighting. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 143–152. IEEE. DOI: 10.1109/ismar.2013.6671774.

Nguyen, R. M. and Le, M. N. (2012). Light source estimation from a single image. In *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1358–1363. IEEE. DOI: 10.1109/icarcv.2012.6485343.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66. DOI: 10.1109/tsmc.1979.4310076.

Sun, Y.-k., Li, D., Liu, S., Cao, T.-C., and Hu, Y.-S. (2020). Learning illumination from a limited field-of-view image. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE. DOI: 10.1109/icmew46912.2020.9105957.

Suzuki, S. *et al*. (1985). Topological structural analysis of

digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46. DOI: 10.1016/0734-189x(85)90136-7.

Wang, Y. and Samaras, D. (2002). Estimation of multiple directional light sources for synthesis of mixed reality images. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pages 38–47. IEEE. DOI: 10.1016/s1524-0703(03)00043-2.

Wang, Z., Chen, W., Acuna, D., Kautz, J., and Fidler, S. (2022). Neural light field estimation for street scenes with differentiable virtual object insertion. In *European Conference on Computer Vision*, pages 380–397. Springer. DOI: 10.1007/978-3-031-20086-1_22.

Whelan, T., Salas-Moreno, R. F., Glocker, B., Davison, A. J., and Leutenegger, S. (2016). Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716. DOI: 10.1177/0278364916669237.

Yacovenco, A., da Silva, R., Vieira, M., and Maciel, L. (2025). Directional light vector estimation from a virtual ar object and its 2d shadow mask. In *Anais do XXVII Simpósio de Realidade Virtual e Aumentada*. SBC. DOI: 10.1109/SVR67689.2025.00025.