"Shadow Catcher": A Vision Based Illumination Condition Sensor using ARToolkit

Werner Hartmann*, Jürgen Zauner**, Michael Haller**, Thomas Luckeneder*, Wolfram Woess*

*Institute for Applied Knowledge Processing (FAW), University of Linz**
{hartmann | luckeneder}@faw.uni-linz.ac.at

*Upper Austria University of Applied Sciences (MTD) – Hagenberg***
{mhaller | jzauner}@fh-hagenberg.at

1. Introduction

Computing the correct illumination of virtual scenes is a well-explored area today. Most commonly used 3d graphic libraries like OpenGL or DirectX/Direct3d contain functions for these tasks.

Within the field of real time mixed reality, the problem of correct illumination (i.e. consistent with the real illumination conditions) has not been considered much, because there where, and still are, more severe tasks to cope with, like for instance commonly usable tracking solutions or conveniently usable end user devices. But one of the main reasons for this is that the properties of the real light source cannot be determined in real time. There are some approaches like [1] and [2] that try to overcome this problem by conventionally tracking the position of the light source, or by using projectors as light source, which also project shadowing effects. This approach works perfectly for well controllable in-door setups but it fails for out-door applications.

The "Shadow Catcher" approach uses the shadows that a light source casts on a so-called "sensor geometry" to determine the position of the light source by analyzing the shape of those shadows. Position and shape of this geometry are known by the system. ARToolKit [3] is currently used to track the position of the geometry, but the solution is independent of the underlying tracking system. The system uses computer vision methodologies to detect shadow silhouettes on the surface of this geometry. The knowledge about the geometry's shape, together with the information about the shadow silhouettes, is used to compute the position of the light source.

2. The Sensor Geometry

The Shadow Catcher approach is based on the idea of using texture changes that are caused by illumination changes to detect the position of the light source. A reference construction with known geometry, position and texture is used to interpret those texture changes.

The detection can either be performed by examining the shadows that the geometry casts on itself, or by considering diffuse reflection effects that the lights-source cause at the surface of the geometry.

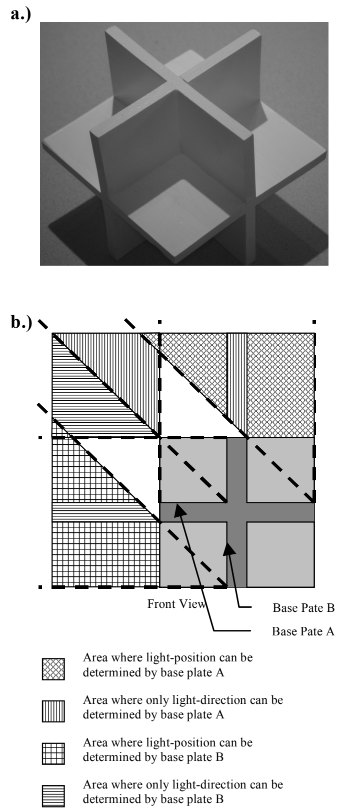

Figure 1. Sensor Geometry

Figure 1a shows the selected sensor geometry, which is designed for shadow based detection. It consists in three squares that are mounted orthogonal to each other. This geometry allows it to use each square either as base plate, where shadow patterns can be observed or as two adjacent shadow-generating plates. Figure 1b shows, how the sensitive areas of two base plates supplement each other. If the light source leaves the sensitive area of the current base plate, the new base plate can be determined by observing the shadow patterns perceived at the current base plate.

3. Implementation

The implementation of Shadow Catcher is based on the AMIRE framework [4]. The system uses a loader for the 3ds file format to load the virtual model of the sensor geometry, which is integrated into the AMIRE framework. OpenGL is used as 3d visualisation subsystem, and OpenCV [5] is used for the computer vision related methods and algorithms.

The building blocks of the algorithm used for the current implementation of Shadow Catcher are depicted at the flow chart showed in Figure 2. The algorithm uses a 3d geometry model of the whole sensor and sub models of each plate consisting in four squares that describe the four sensitive surfaces of each plate. Further, the position of each corner point of each sensitive surface and the position of every corner point of the shadow generating planes are stored. The position of each point is stored relative to the centre of the sensor geometry.

As an initial step, the visible sensitive areas of the geometry must be detected. The visible sensitive areas are those parts of the sensitive fields that are not occluded by other parts of the sensor. If parts of the sensitive area are visible, they must be isolated for the next step, analysis of the image with computer vision methods. First, the OpenCV implementation of the Canny edge detection algorithm is used to detect and enhance the edges within the rectangle containing the visible sensitive area. The output of the algorithm is a black-and-white image, which is used as input for a Hough transformation to detect the lines that form the shadow silhouettes. The intersections of those lines are the corners of the shadow silhouettes. Each of those corners must be examined, and the corresponding vertex of the sensor geometry that causes this corner point must be detected. The 3d position of a shadow corner and the 3d position of the vertex that causes this corner can be used to determine a ray that contains the light-source. The intersection of two of those rays is the position of the light source.

Figure 3 depicts the current state of the implementation. The image shows the sensor geometry tracked by ARToolkit. The lower left corner of the image shows the isolated visible sensitive surfaces, after applying the canny edge detection and the hough transformation. The detected shadow corner is depicted as a black point.

4. Constraints

The proposed method strongly depends on clearly interpretable shadow patterns that the sensor casts on itself. Thus, the method relies on a clear view between the sensor and the light source, which does not contain any obstacles that would cast additional shadows onto the sensor.

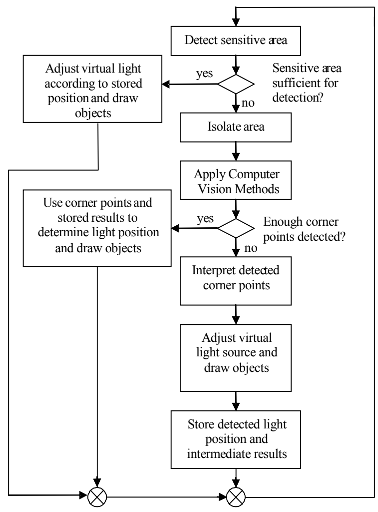

Figure 2. Flow chart of current implementation

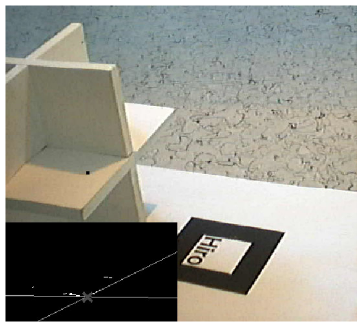

Figure 3. Screenshot with edge detection results

The method is not capable to detect the flare angle of spotlights and it cannot handle spotlights correctly that light either the virtual object or the sensor. Further, the method is currently unable to detect more than one light source.

5. References

[1] T. Naemura, T. Nitta, A. Mimura and H. Harashima, „Virtual Shadows in Mixed Reality Environment Using Flashlight-like Devices“, *TVRSJ*, 2002, 7(2)
[2] R. Raskar, G. Welch and K. L. Low, „Shader lamps: Animating real object with image-based illumination”, *Proceedings of European Workshop on Rendering*, 2001
[3] ARToolKit site: http://www.hitl.washington.edu/artoolkit
[4] AMIRE site: http://www.amire.net
[5] OpenCV site: http://www.intel.com/research/mrl/research/opencv