

# Augmented Reality Using Optical Flow

Konrad Koniarski

System Research Institute,  
Polish Academy of Science,  
ul. Newelska 6, 01-447 Warsaw, Poland  
Email: konrad.koniarski@gmail.com

**Abstract**—The paper deals with the application of Lucas-Kanade optical flow algorithm to develop an augmented reality (AR) system. Merging of a live view of the physical real world with context-related computer-rendered images to create a mixed image is a challenging problem. A virtual object has to be located in the correct pose and position in real time and perspective. Besides the occlusion problem need to be taken into consideration.

In the paper a computer-vision based method for AR systems based on the fiducial marker matching is proposed. For simplicity black square was used as the marker. This method consists of two main steps. The initial step uses Hough's transformation to detect the marker initial position and to select the marker tracked points. In the second step for each image frame these selected points are being tracked using Lucas-Kanade optical flow method. The positions of the selected points are used for calculating the pose and position of a virtual object. Unlike existing method proposed system using optical flow to increase speed performance. The examples of AR applications using the proposed algorithm are provided and discussed.

## I. INTRODUCTION

THE Augmented Reality (AR) is a technology supplemented people's perception of real world by using virtual world. It consists in the integration of the computer generated virtual information such as sound, video or graphics with the real-world environment. This technology is characterized by using real-virtual combinations, 3D registration as well as the real-time interaction. AR interfaces superimpose in real-time textural or pictorial virtual information onto real-world scenes registered in 3D and allow users to interact with real and virtual element simultaneously. The technical challenges in AR technology lie in determining in real-time which image elements should be shown, where and how. Since AR proposes that the user has not to be able to distinguish between real and virtual information it requires that the virtual elements show both geometric as well photometric consistency with the real part of the image. Geometric consistency includes correct placement and size of the virtual element as well as the identification of occlusions. Photometric consistency means suitable shadowing, mutual reflections, chromatic adaptation to scene illumination. The solution of these problems even under simplified assumptions is not trivial.

AR technology appeared in the 1990s. It finds a wide variety of applications like education, entertainment, architecture, medical, machinery manufacturing and maintenance, military, regional planning and many other fields. The implementation of AR technology significantly depends on the used devices

and/or user interfaces. The main devices for AR are head mounted or handheld displays, wireless wristbands or gloves as input devices, digital cameras or wireless sensors as tracking devices and powerful CPU with considerable amount of RAM memory to process camera images. The appropriate techniques for intuitive interaction between the user and the virtual content of AR applications are one of the most important aspects of AR. These techniques include tangible, collaborative, hybrid or multimodal AR interfaces.

Correctly positioning of virtual objects relative to the real environment is called registration. AR image registration uses different computer vision methods consisting usually from two steps: tracking and reconstruction or recognizing. Tracking is based on the detection in the camera images fiducial markers, optical images or other interest points. To interpret camera images tracking can make use of feature or edge detection methods or other image processing methods. Most of the available tracking techniques can be separated into two classes: feature-based and model-based methods [18]. Feature based methods consists in discovering the connection between 2D image features and their 3D world frame coordinates. Model based methods make use of model of the tracked object's features such as CAD models or 2D templates of the items based on distinguishable features. Once a connection is made between 2D image and 3D world frame it is possible to find the camera pose by projecting the 3D coordinates of the feature into the observed 2D image coordinates and by minimizing the distance to their corresponding 2D features. The constraints for camera pose estimation are most often determined using point features. The reconstruction or recognizing stage uses the data obtained from the first stage to reconstruct a real world coordinate system. There are different tracking technologies available, including among others optical, magnetic, movement, ultrasound sensors or thermal imaging. They capture features from the real world and based on this information the AR system determines when, where and how the virtual scene should be merged with a real one.

Due to low cost, accuracy as well as robustness requirements optical tracking is mostly used in AR applications. Two types of optical tracking are usually used: marker based and markerless. The first approach makes use of known artificial patterns, called markers, placed along the real environment in order to perform camera pose estimation. Markers are designed to be easy to recognize and the camera matrix can be recovered from the detected marker pose. The position of the virtual object is

computed using a camera matrix and a detected marker pose for correct virtual object projection. Marker based tracking is a more established approach for registration [15]. Marker based AR consists in the identification of their location with respect to camera as well as in the calculation of the projection and transformation matrices used to correct positioning and integrating of the virtual object with the real environment. The position of the virtual object is calculated based on the marker information. On the other hand markerless tracking differs from the former one by the method to place virtual objects in the real scene. In markerless AR natural objects in the real scene are used as markers. This approach exploits natural features of the real environment to perform tracking. Since markerless AR is based on natural features rather than on fiducial markers there are no ambient intrusive markers that are not really part of the world. Moreover markerless AR counts on specialized and robust trackers as well as provides possibility of extracting from the surroundings characteristic information that may be later used by the AR system for other purposes. Markerless AR is subject of intensive research in the last years and still presents important challenges to be overcome [18].

## II. RELATED WORK

In this paper marker based AR is adopted to estimate the pose and position of the virtual object. Marker based AR has been investigated by many authors [3], [4], [17], [18], [19], [20], [21], [22], [23], [24]. The most frequently used method has been proposed by Kato in [3] where the matching technique for detection and tracking the square marker is developed. Kato's method has been implemented in the form of computational library ARToolKit. In ARToolKit marker position is computed for every frame of video sequence and that lower real time performance of Kato's method. Since Kato's method appeared sensitive to perturbations it has been improved in [4] where vision-based corner tracker has been used. Paper [23] presents many different types of fiducial markers in AR applications designed for having many image features that are easy to track by optical flow algorithms [4]. Eleven criterias for robustness and usefulness of marker are indicated where the most important are: false positive recognition, intermarker confusion, immunity to lighting conditions, immunity to occlusion and speed performance. New types of markers used for AR are proposed in [24]. For details concerning the construction and the application of markerless methods see [18]. Different aspects of using AR are indicated in [19], [20], [21], [22]. Paper [14] is concerned with the development of AR environment without specialized hardware or pre-calibration. AR environment is built using SURF method combined with bi-directional optical flow algorithm. In [14] Herakleous estimate SURF method as not sufficient for real time performance giving only 10 frames per second. Applying bi directional optical flow Herakleous improve performance of SURF for 50 percent. Paper [14] is not dealing with immunity of light conditions and immunity of occlusion.

This paper deals with the development of a marker based approach to AR. A new hybrid algorithm combining camera calibration, the selection of characteristic points on the marker surface as well as tracking the movement of these points using the optical flow technique and the estimation of the position pose of the virtual object is proposed and tested. As a initial step the camera calibration process is performed using moving plane techniques [8]. That step is done only once. Next step is to find marker position. Hough transformation [16] is used for the detection of the marker pattern features. For simplicity marker contain only one black square, but any recognizable pattern could be applied. Next block of presented method is tracking loop. A key element of the algorithm is the application of Lucas Kanade optical flow method [2] to track the movement of the selected marker points. Finally the pose of the virtual object is estimated using the updated camera projection matrix. The proposed algorithm has been programmed in C++ environment and tested. The obtained results are provided and discussed. They indicate that the proposed algorithm is robust and efficient tool in realizing AR applications.

## III. PROPOSED ALGORITHM

The proposed approach consists from two steps. At initial step camera is calibrated and marker is detected. In main part selected marker points are tracked using Optical Flow method and virtual object pose and position are calculated. Furthermore virtual object is rendered in front of image from camera. System overview diagram is presented on Fig. 1.

### A. Camera calibration

The aim of camera calibration is to determine the extrinsic and intrinsic parameters of the transformation between an object in real 3D space and the 2D image observed by the camera based on visual data acquired from the images. The extrinsic transformation parameters includes the rotation and the translation matrices of the camera. The coordinates of the principal point, the scale factors along the axes and the skew of the two image axes are called intrinsic camera parameters. These parameters, i.e., six extrinsic and 5 intrinsic are assembled in the form of the  $3 \times 4$  camera projection matrix.

In this paper Zhang method based on moving planes technique is used [8] to calibrate and rectify the camera. This algorithm consist of the following key steps:

- 1) The printed pattern is attached to a planar surface.
- 2) A few images of the model plane is taken under different orientations due to the movement either the camera or the plane.
- 3) The feature points in the images are detected.
- 4) Estimate all the parameters using the homography between the model plane and its image.
- 5) Estimate the coefficients of the radial distortion by solving the linear least-square problem.

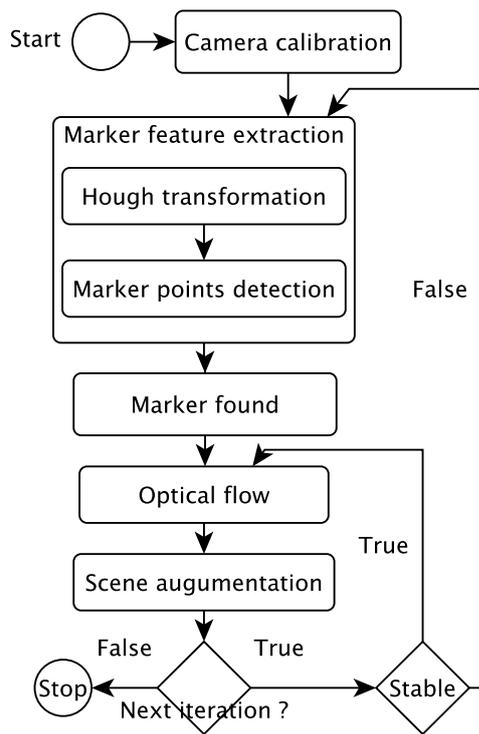


Fig. 1: System overview diagram.

- 6) Refine all parameters including the lens distortion parameters by minimizing the distances between the image points and their predicted positions.

Remark the pattern could be anything as long as we know the metric on the plane. If pixels are square the minimum number of orientations is two but for better quality it should reach 4 or 5 different orientations. Either camera or the planar pattern can be moved. There is no need to know the motion but a pure translation should be excluded.

Figure 2 presents detection of marker pattern while Figure 4 shows rectified image.

*B. Tracking points selection*

Since we confine to the fiducial marker based AR let us choose a pattern of a marker. In literature [3], [17] there exists many examples of marker selection. A marker that is simple black square was chosen. The black square corners are perfect candidates for tracking points. For that algorithm any recognizable marker pattern can be used, but that required different method of selecting tracking point then applied Hough transformation. For details see Fig. 5. In second step of our algorithm our aim is to select the characteristic points of the fiducial marker allowing to determining its initial position and tracking it in the optical flow step.

First having acquired marker image we determined its edges using method of [10]. Next Hough method [16] is used to detect square edges characterizing the marker. Hough transform

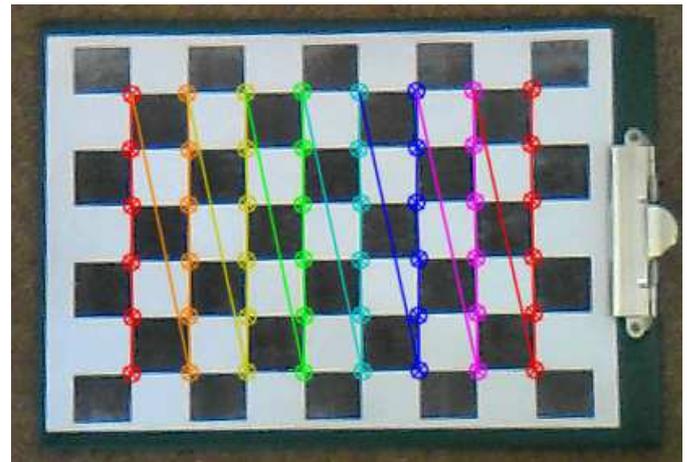


Fig. 2: Camera calibration using marker pattern and its geometrical representation (spherical distortion).

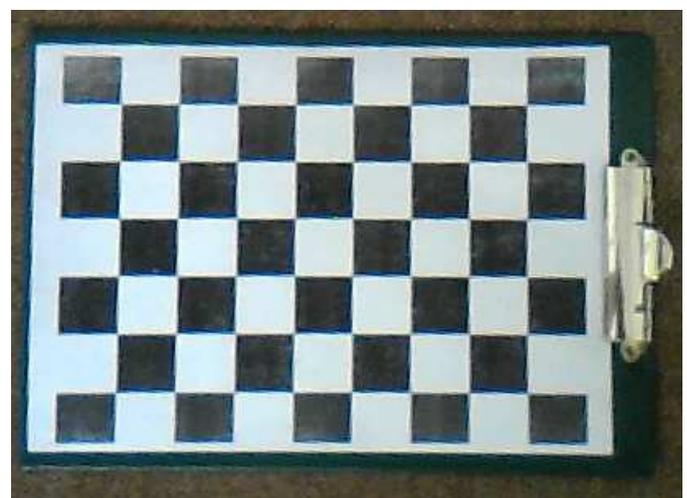


Fig. 3: Calibrated camera image (image without distortion).

is recognized as a powerful tool to extract parametrized curves. The idea behind the method is simple: parametric shapes in an image are detected by looking for accumulation points in the parameter space. If a particular shape is present in the image then the mapping of all of its points into parameter space must cluster around the parameter values which correspond to that shape. Usually this method consists from the following phases: voting, peaks localization, determining the actual parameters and verification. In our case black square generate 4 lines. I assume that in the initial position of the marker these lines are perpendicular. For our marker these set of lines is displayed in Fig. 5. Having known the parameters of each line I solve the system of linear equations to find their intersection points. Among these intersection points I choose square corners. These points are stable for Lucas-Kanade algorithm [2], i.e., such points where the image function takes minimum and maximum values with respect to black and white colours, respectively. The selected points are feature points, deployed

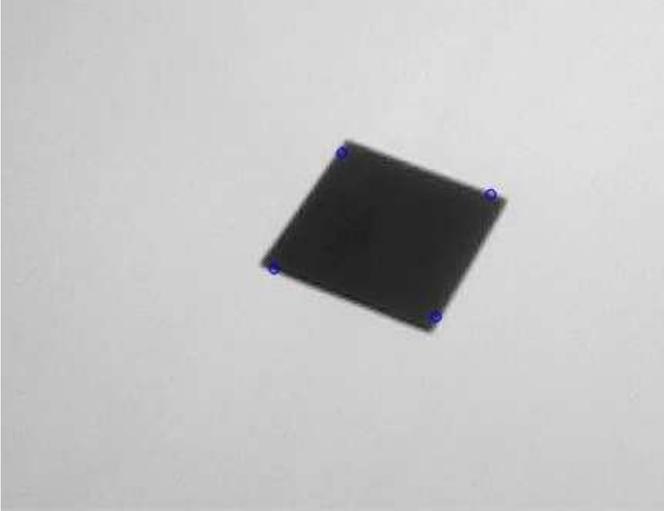


Fig. 4: Marker pattern.

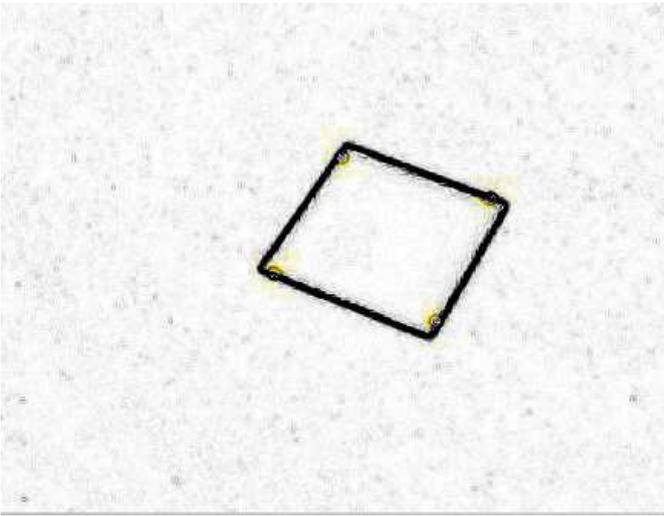


Fig. 5: Marker detection. Black lines are detected by Hough line detector.

at squares corners, which improve tracking stability during the next optical flow step.

### C. Lucas-Kanade Optical Flow

The core area of computer vision is the analysis of the sequence of images to approximate motion of 3D objects. Optical flow field is a convenient and useful image motion representation. It is used to approximate the movement of 3D objects based on a sequence of 2D images [12]. I expect that optical flow is not too different from the motion field. The optical flow calculations are based on two main assumptions: the image brightness constancy and the small motion. The first assumption means that while the object may change the position in short time interval the reflectivity and illumination will remain constant. The second assumption means that the movements less than one pixel are considered. There are

numerous methods to calculate optical flow. These methods include differential-based methods, frequency-based methods, correlation techniques, multiple motion methods, feature based method, as well as hierarchical approaches [12]. Optical flow approach is widely used for object tracking, camera stabilization, image mosaics, 3D shape reconstruction or obtaining a structure from motion. Since tracked features points can move in any direction or be obscured by other scene objects, tracking is a challenging investigation problem.

The optical flow methods are very often used in Augmented Reality applications to track the movement of the extracted features points [2], [14], [15]. Lucas and Kanade's local differential algorithm is one of the most popular method for optical flow computations [13]. This method is based on the assumption that in a given pixel the flow vector will be similar to a small neighbourhood surrounding this pixel. A weighted least square method is used to approximate the optical flow at a given pixel. For each pixel an optical flow vector consistent with the neighbouring spatial and temporal gradients is found. This method is very popular these days, but it still has some drawback. The first disadvantage of this method is challenging computational problem and result image is not dense. Only selected points can be computed for the real time applications. The selected points for tracking can't be placed on edges or low textured regions where this method doesn't give stable results. The best candidates for tracking are points in high textured regions. Lucas-Kanada's optical flow method use correlation window for tracking near pixel movement. For point movement tracking Gaussian pyramid with different scales of images is used [9].

The Lucas-Kanade's Optical Flow algorithm [2] is used for tracking the marker feature points in camera captured images. This method is faster than detection image features analyzing all pixels. Consider small window  $\Omega \subset D \subset R^2$  occupied by the pixels  $\mathbf{p}$  of an image  $D$ . The pixel image  $p$  is characterized by  $p = [x, y]$  where  $x, y$  denotes coordinates of a point in  $\Omega$ . I denote the intensity of the image at position  $(x, y)$  and at time  $t$  by  $I(x, y, t)$  as well as the velocity  $v$  of the image pixel  $p$  by

$$\dot{p} = v = (v_x, v_y) = \left( \frac{dx}{dt}, \frac{dy}{dt} \right) \quad (1)$$

Based on the brightness constancy assumption of the pixel  $p$  during the time increment  $dt$  I can write

$$I(x + v_x dt, y + v_y dt, t + dt) \approx I(x, y, t) \quad (2)$$

If the brightness changes smoothly with  $x, y, t$  I expand the left-hand side of (2) by a Taylor series and obtain

$$I(x + v_x dt, y + v_y dt, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} v_x dt + \frac{\partial I}{\partial y} v_y dt + \frac{\partial I}{\partial t} dt + O(dt^2), \quad (3)$$

where  $\frac{\partial I}{\partial x}$  denotes the first derivative of the function  $I$  with respect to  $x$  and  $O(dt^2)$  denotes the higher order terms. Due to the small motion assumption I ignore in (3) the higher order

terms. Substituting (3) into (2) I obtain optical flow constraint equation

$$\nabla I \cdot v + I_t = 0 \quad (4)$$

where  $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$  denotes a spatial gradient,  $I_t = \frac{\partial I}{\partial t}$  is a temporal gradient and  $v = (v_x, v_y)$  denotes the optical flow vector.

Remark, for each pixel, the equation (4) provides only a single constraint for the two components of the optical flow vector  $v_x$  and  $v_y$ . Only the component of the vector in the direction of the spatial gradient is provided. It means that I cannot determine optical flow uniquely only from such optical flow constraint equation. We need some other constraints.

Lucas-Kanade's local differential technique is one of the most popular methods to calculate the optical flow. This method assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration and solves the basic optical flow equation (4) for all the pixels in that neighborhood. Therefore we obtain the system consisting from many equations (4) where the number of unknowns is lower than the number of equations. The least square method is used to find a solution to this system of equations. The optical flow is a field which describes the movement of pixels between two frames, often used for image sequences like movies. It often forms the lowest level for motion analysis and is used as input to higher level systems like segmentation and tracking for interpretation. This system follows the same approach. The optical flow field is determined for each frame of the movie and is used to segment the frame into separate regions. These regions are fed as input to an object tracking system which determines information about the regions in the image and matches them with regions seen in previous frames. This allows us to track various objects through the scene.

Let  $\Omega$  denotes a surrounding neighbourhood of a pixel  $[x, y]$  under consideration of size  $N$  where each neighbour pixel is represented as  $p_i$  and  $\mathbf{p} = (p_1, \dots, p_N)$ . Let us define a window function  $W(\mathbf{p})$ ,  $\mathbf{p} \in \Omega$ . We let the weight of the center bigger than others, i.e., the window function favors the center. Usually  $W(\mathbf{p})$  is assumed to be Gaussian window. The size of the window  $W$  is  $5 \times 5$ , i.e., usually  $N = 25$  pixels are used. Let us introduce the error function for the optical flow vector  $v$  for a pixel  $[x, y]$

$$E(v) = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p})(\nabla I(\mathbf{p}) \cdot v + I_t(\mathbf{p}))^2 \quad (5)$$

This function simply sums the error of applying the flow vector  $v$  to the spatial and temporal gradients of all surrounding neighbours using the optical flow equation (4). The error is higher if the considered flow vector is not consistent with the spatial and temporal gradients for the surrounding pixels. The weights are used to diminish the importance of distant neighbours. We are looking for  $v$  minimizing the error function (5), i.e.,

$$\min_v E(v) = E(v^*) \quad (6)$$

Differentiating the error function (5) with respect to  $v_x$  and  $v_y$  we get the following necessary optimality condition to the minimization problem (6)

$$\frac{\partial E}{\partial v_x} = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p}) \left( \frac{\partial I}{\partial x}(\mathbf{p}) v_x + \frac{\partial I}{\partial y}(\mathbf{p}) v_y + \frac{\partial I}{\partial t}(\mathbf{p}) \right) = 0 \quad (7)$$

$$\frac{\partial E}{\partial v_y} = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p}) \left( \frac{\partial I}{\partial x}(\mathbf{p}) v_x + \frac{\partial I}{\partial y}(\mathbf{p}) v_y + \frac{\partial I}{\partial t}(\mathbf{p}) \right) = 0 \quad (8)$$

Let us introduce notation:

$$W = \text{diag}(W(p_1), \dots, W(p_N))_{N \times N}, \quad v = [v_x, v_y]^T \quad (9)$$

$$A = \begin{bmatrix} \frac{\partial I}{\partial x}(p_1) & \frac{\partial I}{\partial y}(p_1) \\ \vdots & \vdots \\ \frac{\partial I}{\partial x}(p_N) & \frac{\partial I}{\partial y}(p_N) \end{bmatrix}_{N \times 2} \quad (10)$$

$$b = - \begin{bmatrix} \frac{\partial I}{\partial t}(p_1) \\ \vdots \\ \frac{\partial I}{\partial t}(p_N) \end{bmatrix}_{N \times 1} \quad (11)$$

Using this notation the system (7)-(8) can be written in the matrix form

$$A^T W^2 A v = A^T W^2 b \quad (12)$$

So, assuming the matrix

$$A^T W^2 A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}_{2 \times 2} \quad (13)$$

$$a_{11} = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p}) \left( \frac{\partial I}{\partial x}(\mathbf{p}) \right)^2$$

$$a_{12} = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p}) \frac{\partial I}{\partial x}(\mathbf{p}) \frac{\partial I}{\partial y}(\mathbf{p})$$

$$a_{21} = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p}) \frac{\partial I}{\partial x}(\mathbf{p}) \frac{\partial I}{\partial y}(\mathbf{p})$$

$$a_{22} = \sum_{\mathbf{p} \in \Omega} W^2(\mathbf{p}) \left( \frac{\partial I}{\partial y}(\mathbf{p}) \right)^2$$

is invertible, the flow  $v$  for the image pixel  $p$  equals

$$v = (A^T W^2 A)^{-1} A^T W^2 b \quad (14)$$

#### D. Pose estimation of the augmented object

In the last step of the proposed method the projection matrix of the augmented object is computed using the approach from section III-A. However here this matrix projects [8] 3D virtual object points positions on the 2D image plane in global coordinates. First based on the intrinsic camera parameters and the homography matrix the estimations of the rotation and translation matrices as well as the projection matrix are calculated. The homography matrix is calculated using

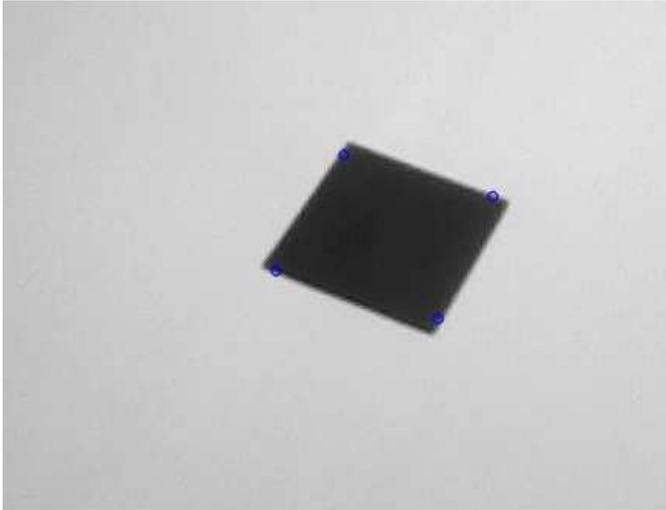


Fig. 6: Blue circles marked tracked marker points.

maximum likelihood criterion and the coordinates of the model and image points. For details see [8]. Next the components of the projection matrix are refined by minimizing the square of the Euclidean norm distance between the positions of the geometric reference points on the marker and the positions of the tracked marker points.

#### IV. RESULTS

The algorithm has been implemented in C/C++ and MinGW environments. Moreover OpenCV 2.4 development library was used for image processing and manipulations [11]. Then OpenGL library is used for rendering 3D objects and composing it into real scene. All tests were run on 8GB RAM machine equipped with Intel i7 2640M 2.80GHz processor. The processed images have 640 by 480 pixels resolution. The marker presented in Fig. 4 was printed and displayed on flat moving surface. Webcam was static. Results are presented on Fig. 7. Right part of images show tracked marker, left part present marker and the virtual object - standard OpenGL teapot. All of those images presents marker from different perspective and proper projection of virtual object.

#### V. EVALUATION

For evaluation of proposed method five criterias were used: false/positive recognition, marker confusion, immunity of light conditions, immunity of occlusion and speed performance.

- 1) False/positive recognition: This criteria indicate sensitiveness for not recognize marker. First step of the presented method detects marker tracking points and passing marker points as input for second stage. If second stage loose it's stability in tracking marker points methods fallback to detection stage. This robust solution works stable until brightness constraint is violate or view angle is to acute.
- 2) Marker confusion: indicates sensitiveness if marker is recognized correctly. In presented case marker have four

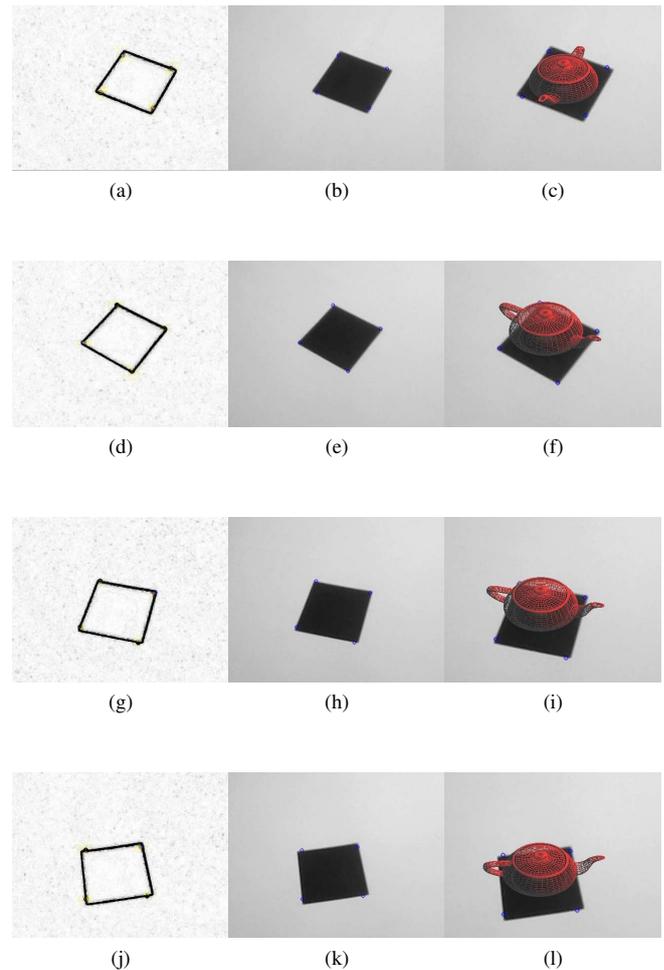


Fig. 7: Augmented virtual image - First column - marker detection. Middle column marker points tracking. Right column augmented virtual image

symmetric points any of them can be used as first feature to track.

- 3) Immunity of light conditions: Lucas-Kanade method use brightness constraint to track pixel movement between two frames. This constraint is also required for presented method. Selected marker has big contrast, so small changes in image brightness not disturb stability of presented method.
- 4) Immunity of occlusion: Optical flow is sensitive on occlusion. When tracking point is vanished by to acute angle or by other item occlusion Lucas-Kanade method is not stable and observed point usually move to other local minimum. Proposed method using only 4 points to track marker and if any of them is not tracked correctly methods loose stability. That triggers step back to marker detection block.
- 5) Speed performance: One of the most important criteria

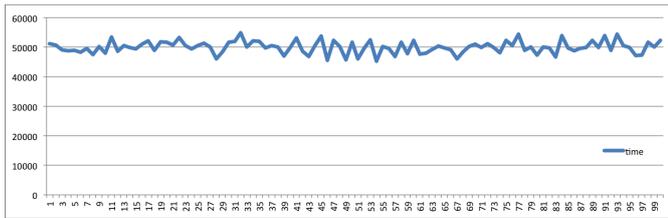


Fig. 8: Time in microseconds.

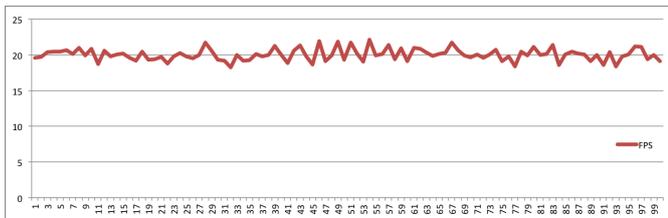


Fig. 9: Frames per second.

for AR is real time performance. Figure 8 presents time in microseconds to render each frame for 100 frames video. Figure 9 showing that system works with average 20 frames per second. That is comparable with real time performance.

## VI. CONCLUSION

This paper presents a new approach for Augmented Reality system based on Optical Flow method. The presented method uses a marker for obtaining the position, scale and pose of the virtual object. The proposed method runs in real time. The marker is detected correctly when there is no brightness change between image frames. All part tracking point from the marker must be visible during tracking otherwise method become unstable. The performed experiment shows that method is working correctly when marker was moved.

Using Optical Flow method for Augmented Reality is considered by the author as the natural step forward in combining marker and markerless method, and it will certainly become much popular in the near future.

## VII. FUTURE WORK

As presented methods gives good results in Augmented Reality application, the next step should improve algorithm stability when one of tracked points is obscured. This method is also sensitive to changing light conditions. Future work should improve presented method to be less sensitive for changing light conditions.

## REFERENCES

[1] Chari V, Singh J M, Narayanan P J, (2008) Augmented Reality using Over-Segmentation, National Conference on Computer Vision Pattern Recognition Image Processing and Graphics.

[2] Lucas B D, Kanada T (1981) An Iterative Image Registration Technique with an Application to Stereo Vision, *IJCAI*, 81, 674–679.

[3] Kato H, Billinghurst M (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system, *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, 85–94.

[4] Malik S, Roth G, McDonald C (2002) Robust Corner Tracking for Real-Time Augmented Reality, *Vision Interface 2002*, Calgary, Alberta, Canada, May 2002, 399-406. (National Research Council of Canada, Report No 45860).

[5] Koniarski K (2011), Image features detection methods in multiframe analysis (in Polish), in: *Techniki informacyjne: teoria i zastosowania*, eds: J. Hołubiec, 1(13), 68–82.

[6] Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: An efficient alternative to SIFT or SURF, *2011 IEEE International Conference on Computer Vision (ICCV)*, 2564–2571.

[7] Klein G, Murray D (2009) Parallel Tracking and Mapping on a Camera Phone, *8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, 83-86.

[8] Zhang Z (2000) A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330-1334.

[9] Bouguet J Y (2001) Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm, Intel Corporation, 5.

[10] Sun D, Roth S, Lewis J P. (2008) Learning Optical Flow, *Computer Vision - ECCV 2008*, Springer, 83-97.

[11] Computer Vision Library: OpenCV (2012) <http://opencv.willowgarage.com> version 2.4.

[12] Beauchemin S S, Baron J L (1995) The computation of optical flow, *ACM Computing Surveys (CSUR)*, 27(3), 433–466.

[13] Baker S, Scharstein D, Lewis J L, Roth S, Black M, Szeliski M (2011) A database and Evaluation Methodology for Optical Flow, *International Journal of Computer Vision*, 92(1), 1-31.

[14] Herakleous K, Poullis C H (2013) Improving augmented reality applications with optical flow, *IEEE International Conference on Image Processing 2013*, 3403-3406.

[15] Li H, Qi M, Wu Y (2012) A Real-Time Registration Method of Augmented Reality based on SURF and Optical Flow, *Journal of Theoretical and Applied Information Technology*, 42(2), 281–286.

[16] Ji J, Chen G, Sun L (2011) A novel Hough transform method for line detection by enhancing accumulator array, *Pattern Recognition Letters*, 32(11), 1503-1510.

[17] Hirzer M (2008) Marker detection for augmented reality applications, Institut for Computer Graphics and Vision, Graz University, Technical Report, ICG-TR-08/05.

[18] Fuhr B (2011) *Handbook of Augmented Reality*, Springer Science+Business Media, New York, New York.

[19] Lee T, Hollerer T (2009) Multithreaded Hybrid Feature Tracking for Markerless Augmented Reality, *IEEE Transactions on Visualization and Computer Graphics*, 15(3), 355–368.

[20] Gedik O S, Alatan A A (2013) 3-D Rigid Body Tracking Using Vision and Depth Sensors, *IEEE Transactions on Cybernetics*, 43(5), 1395–1405.

[21] Cheok A D, Qiu Y, Xu K, Kumar G K (2007) Combined Wireless Hardware and Real-Time Computer Vision Interface for Tangible Mixed Reality, *IEEE Transactions on Industrial Electronics*, 54(4), 2174–2189.

[22] Chen Z, Li X (2010) Markless Tracking based on Natural Feature for Augmented Reality, *IEEE International Conference on Educational and Information Technology (ICEIT 2010)*, 2, 126-129.

[23] Fiala M (2010) Designing highly reliable fiducial markers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7), 1317–1324.

[24] Demuyneck O, Menendez J M (2013) Magic Cards: A New Augmented Reality Approach, *IEEE Computer Graphics and Applications*, 33(1), 12–19.