Marcin AUGUSTYNOWICZ, Dariusz SAWICKI

Warsaw University of Technology

doi:10.15199/48.2016.01.49

Reconstruction of the relative coordinates of image using projective geometry

Abstract *Problem of determining the relative position often occurs in the process of image recognition. In general form, it can be described as quadrilateral to rectangle transformation. This paper describes and compares methods of simple relative coordinates calculation on the flat surface. The surface position can be set at any angle to the camera and in any rotation as well. The problem can be solved in efficient way using projective geometry, the new reconstruction method is introduced.*

Streszczenie *Problem określenia względnego położenia obiektów często występuje w procesie rozpoznawania obrazu. W ogólnej postaci, może to być opisane przekształceniem dowolnego czworokąta w prostokąt. W artykule przeanalizowano proste przekształcenia tego typu dla dowolnego ustawienia płaskiej powierzchni względem kamery. Został zaproponowany nowy algorytm, który rozwiązuje problem w prosty sposób z wykorzystaniem geometrii rzutowej.* ***(Rekonstrukcja względnych współrzędnych obrazu przy użyciu geometrii rzutowej)***

Keywords: global to local coordinate transform, quadrilateral to rectangle transformation, coordinates reconstruction on perspective image.
Słowa kluczowe: transformacja współrzędnych globalnych do lokalnych, przekształcenie czworokąta w prostokąt, rekonstrukcja współrzędnych w rzucie perspektywicznym.

Introduction

The main aim of this paper is to develop the simple method that allows determining the relative position on the fixed surface based on the camera view or other perspective projection. The camera field of view can be placed at any relation (angles) to the surface, where the analyzed rectangle lies. In this way in the captured image the rectangle can be deformed into practically any quadrilateral. The problem can be simplified to a determination of the proper transformation from quadrilateral to rectangle. This deformation is related to perspective projection.

The analysis of the position reconstruction problem was made for this paper in relation to application of camera usage in human computer interaction, but the problem is of high importance in many other fields, eg. it is the basic problem in machine vision [1, 2] and the image recognition [3]. Perspective images of planes allow making measurements of the world planes from their perspective images [4]. This way projection defined plane to plane homography can help in distance measurements, which is the basic problem in robot viewing and moving. Quadrilateral to quadrilateral transformation problem is known in texture mapping [5, 6], especially as linear mapping [7] and bilboarding [8].

Problem of quadrilateral to quadrilateral transformation is described in many books concerning image processing and analysis. Modern textbooks [9] propose the DLT algorithm (Direct Linear Transformation) as the best solution for this task. It is a powerful and versatile algorithm, in which a set of matrix operations allows finding a solution for any position of the input and output quadrilaterals. But the main goal of the task is to develop an algorithm for a specific HCI situation, where the resulting quadrilateral is always a rectangle. For this reason, after a brief analysis, the universal DLT algorithm, as too extensive, was not taken into account. A new effective and simple algorithm has been proposed. After testing it has been successfully implemented in the software of one of the exhibits in Warsaw Copernicus Science Centre, where it effectively works.

Main assumptions of the problem

The task consists in determining the transformation Ψ that allows converting the points of the quadrilateral $Q_0Q_1Q_2Q_3$ on the relevant points of the rectangle $P_0P_1P_2P_3$ assuming that the point Q_i is converted to P_i – figure 1.

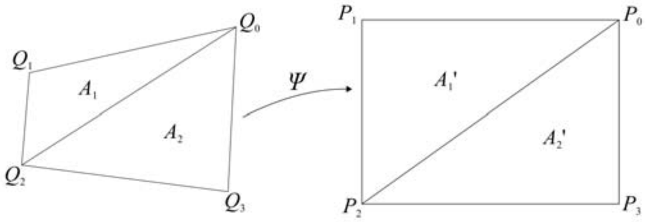

Fig. 1. Linear transformation from quadrilateral $Q_0Q_1Q_2Q_3$, to rectangle $P_0P_1P_2P_3$

Four vertexes of the rectangle determined local coordinates system. In this system position of any point inside the rectangle is defined as relative coordinate. When the vertexes are projected on the recorded, by camera, image, the coordinates are deformed by perspective projection. Reconstruction of the coordinates needs proper transformation from quadrilateral (camera view) to rectangle (original position). The process of determining the relative coordinates has been called calibration in this paper. For any result point in the rectangle, the relative coordinates are in the range from 0 to1.

Three different methods of solving determination of the relative position in the original rectangle are analyzed in the paper. There are few main assumptions in these methods which include: simple calibration process, low computational complexity, stability, precision. Because of comparison made in presented paper, full description of each method with complete set of equations are presented.

Linear transformation (LT)

This method is based on the linear transformation (linear algebra) [10] and can be regarded as a simplified version of the DLT algorithm. Such method is often used in computer graphics for texture mapping [5]. For simplicity it is now assumed that reverse transformation to perspective is linear. The area of the quadrilateral can be divided into two triangles A_1 and A_2, which are shown in figure 1.

Let $A_1 : (Q_0, Q_1, Q_2)$ and $A_2 : (Q_0, Q_2, Q_3)$.

Let $Q_0=(x_0, y_0)$, $Q_1=(x_1, y_1)$, $Q_2=(x_2, y_2)$.

After transformation ψ points Q_0, Q_1, Q_2 correspond to the points P_0, P_1, P_2. Let $P_0=(u_0, w_0)$, $P_1=(u_1, w_1)$ $P_2=(u_2, w_2)$. This way ψ can be regard as $\psi =(\varphi_1, \varphi_2)$ where are two independent systems of three linear equations with three variables each (1).

$$(1)\qquad \begin{aligned} \varphi_1 : u_i &= \varphi_1(Q_i) = a x_i + b y_i + c \\ \varphi_2 : w_i &= \varphi_2(Q_i) = d x_i + e y_i + f \end{aligned} \qquad \text{for } i=0,1,2$$

Solutions are as follow:

$$(2)\qquad a = \frac{\begin{vmatrix} u_0 & y_0 & 1 \\ u_1 & y_1 & 1 \\ u_2 & y_2 & 1 \end{vmatrix}}{\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}},\ b = \frac{\begin{vmatrix} x_0 & u_0 & 1 \\ x_1 & u_1 & 1 \\ x_2 & u_2 & 1 \end{vmatrix}}{\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}},\ c = \frac{\begin{vmatrix} x_0 & y_0 & u_0 \\ x_1 & y_1 & u_1 \\ x_2 & y_2 & u_2 \end{vmatrix}}{\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}}$$

$$d = \frac{\begin{vmatrix} w_0 & y_0 & 1 \\ w_1 & y_1 & 1 \\ w_2 & y_2 & 1 \end{vmatrix}}{\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}},\ e = \frac{\begin{vmatrix} x_0 & w_0 & 1 \\ x_1 & w_1 & 1 \\ x_2 & w_2 & 1 \end{vmatrix}}{\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}},\ f = \frac{\begin{vmatrix} x_0 & y_0 & w_0 \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{vmatrix}}{\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}}$$

It is possible to make most of the calculation during precalculation process and get parameters value. The same calculation should be done for triangle A_2 to get parameters for transformation φ_3, φ_4.

There exist many implementations of the linear transformation in discussed task. Version presented in this paper allows calculating only one triangle from the quadrilateral, what can be useful in some applications. In quadrilateral to rectangle transformation this implementation builds 12 equations with redundant information because both triangles lie in the same plane. General problem how to transform a quadrilateral into a rectangle has 8 degrees of freedom and can be solved by system of 8 linear equations [5].

The advantage of this method is the simplicity of calculation algorithm.

Physical based model (PBM)

The second calibration method is based on the camera and surface physical setup. As the basis, the rules of projective geometry were used. [11, 12].

Image recorded by camera is created by the rays of light projected from object through the image sensor to the centre of projection (point S in figure 2).

Plane π_2 contains points Q_0,Q_1,Q_2,Q_3 that are vertexes of the original rectangle. Plane π_1 can be regard as an image sensor plane and contains points P_0,P_1,P_2,P_3 that are projections of Q_0,Q_1,Q_2,Q_3 respectively. d_{0i} are Euclidean distances between points Q_0 and Q_i for $i=1,2,3$.

The first step is to find parameters a, b, c of the plane equation $\pi_2 : x = ay+bz+c$.

From the physical model follow that $a \neq 0$ and $c > 0$.

Let $P_i = (\alpha_i, \beta_i, \gamma_i)$ and $Q_j = (x_j, y_j, z_j)$ for $i=0,1,2,3$, $j=0,1,2,3$.

The lines k_i through the points $S = (0,0,0)$ and $P_i = (\alpha_i, \beta_i, \gamma_i)$ are given by $k_i : x/\alpha_i = y/\beta_i = z/\gamma_i$ for $i=0,1,2,3$.

Parameters of plane π_2 are given by system (3):

$$(3)\quad \begin{cases} d_{0j}^2 = (x_0 - x_j)^2 + (y_0 - y_j)^2 + (z_0 - z_j)^2 & j = 1,2,3 \\ x_k = a y_k + b z_k + c & k = 0,1,2,3 \\ \dfrac{x_k}{\alpha_k} = \dfrac{y_k}{\beta_k} = \dfrac{z_k}{\gamma_k} & k = 0,1,2,3 \end{cases}$$

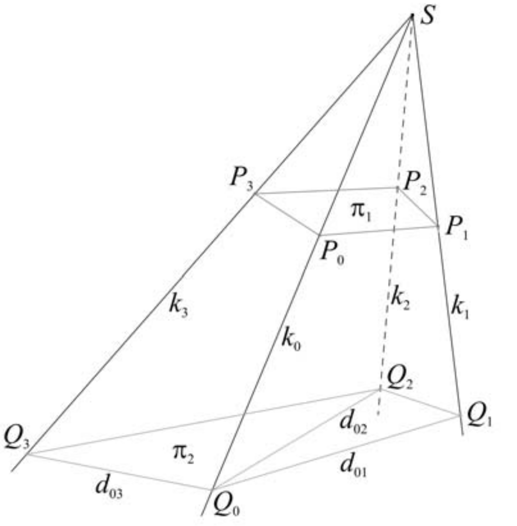

Fig. 2. Schema of physical model

Let $r = \beta_0\alpha_j - \beta_j a_0$, $s = \gamma_0\alpha_j - \gamma_j a_0$, $t = \gamma_0\beta_j - \gamma_j\beta_0$

Then for $j = 1,2,3$:

$$(4)\quad F_j(a,b,c) = c^2((ar+bs)^2 + (bt+r)^2 + (-at+s)^2) - d_{0j}^2(\alpha_0 - a\beta_0 - b\gamma_0)^2(\alpha_j - a\beta_j - b\gamma_j)^2$$

To get a, b, c one needs to solve system of three nonlinear equations: $F_j(a,b,c)=0$. It can be done using Newton's method:

$$(5)\quad \begin{vmatrix} a_{n+1} \\ b_{n+1} \\ c_{n+1} \end{vmatrix} = \begin{vmatrix} a_n \\ b_n \\ c_n \end{vmatrix} - MdF \cdot \begin{vmatrix} F_1(a_n,b_n,c_n) \\ F_2(a_n,b_n,c_n) \\ F_3(a_n,b_n,c_n) \end{vmatrix} \quad \text{for } j = 1,2,3 .$$

$$MdF = \begin{vmatrix} \dfrac{\partial F_1(a_n,b_n,c_n)}{\partial a} & \dfrac{\partial F_1(a_n,b_n,c_n)}{\partial b} & \dfrac{\partial F_1(a_n,b_n,c_n)}{\partial c} \\ \dfrac{\partial F_2(a_n,b_n,c_n)}{\partial a} & \dfrac{\partial F_2(a_n,b_n,c_n)}{\partial b} & \dfrac{\partial F_2(a_n,b_n,c_n)}{\partial c} \\ \dfrac{\partial F_3(a_n,b_n,c_n)}{\partial a} & \dfrac{\partial F_3(a_n,b_n,c_n)}{\partial b} & \dfrac{\partial F_3(a_n,b_n,c_n)}{\partial c} \end{vmatrix}^{-1}$$

where for $j=1,2,3$

$$\frac{\partial F_j(a_n,b_n,c_n)}{\partial a} = c^2(2(ar+bs)r - 2(-at+s)t) + d_{0\alpha\beta\gamma}(2a\beta_0\beta_j + b(\beta_0\gamma_j + \gamma_0\beta_j) - \beta_0\alpha_j - \alpha_0\beta_j)$$

$$\frac{\partial F_j(a_n,b_n,c_n)}{\partial b} = c^2(2(ar+bs)r - 2(-at+s)t) + d_{0\alpha\beta\gamma}(2a\beta_0\beta_j + b(\beta_0\gamma_j + \gamma_0\beta_j) - \beta_0\alpha_j - \alpha_0\beta_j)$$

$$d_{0\alpha\beta\gamma} = -2d_{0j}^2(\alpha - a\beta_0 - b\gamma_0)(\alpha_j - a\beta_j - b\gamma_j)$$

$$\frac{\partial F_j(a_n,b_n,c_n)}{\partial c} = 2c((ar+bs)^2 + (bt+r)^2 + (-at+s)^2)$$

To calibrate point there is a need to find cross-cut of plane π_2 and line that contains the point $S=(0, 0, 0)$ and the calibrated point. Next, there is a need to make transformation exactly like in previous chapter.

The advantage of this method is that it corresponds to the analyzed problem in vivid way. But it is the academic only, not practical feature.

Antiperspective projection (AP)

Both methods described above allow solving the problem. Nevertheless the question if it is the optimal solution still remains. To improve computational complexity, dependencies between projective coordinates were analyzed one more time [13].

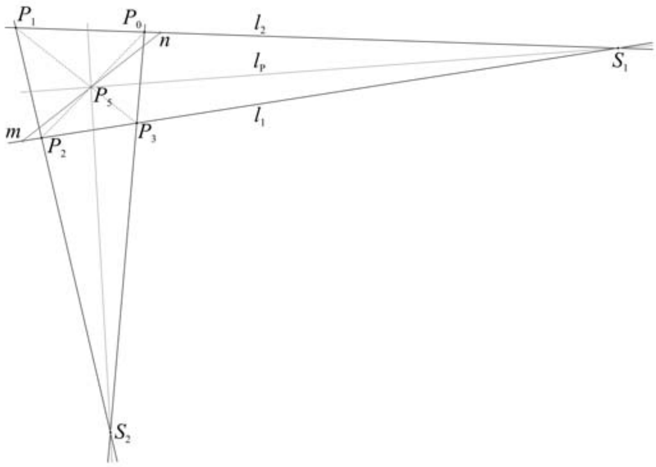

Fig. 3. Schema of antiperspective projection model

The experiments done after analysis allow introducing a new method as a solution of the problem. The method has been called “antiperspective projection” because it is based on the standard perspective projection but calculations are conducting in the opposite way to original projection. There exists the theorem that "every quadrilateral can be considered as the projective image of a square" [14]. Of course such operation is not invertible because many different quadrilaterals can be transformed to the same square, but it has no impact on problem described in this paper. The main task is, therefore, to find quadrilateral to rectangle transformation as opposite way to perspective projection.

At the original rectangle points, where one of coordinates is common, parallel lines (rays) could be created. These lines (rays) meet in vanishing point at image that is the projection of the rectangle.

There is a line that crosses the image of the rectangle. Using cross-cut of this line and ray enables to conduct an operation that is opposite to original projection.

Let points P_0, P_1, P_2, P_3 are input data. Let point P_5 – can be given as an input or can be calculated as an intersection of $\overline{P_1P_4}$ and $\overline{P_0P_2}$ line segments.

Line l_1 contains P_2, P_3 and l_2 contains P_0, P_1

Point S_1 is an intersection point between l_1 and l_2. It is also a cross of lines which has the same value of coordinate at original rectangular.

Let $n \in l_1 \Rightarrow n_y = a_1 n_x + b_1$, $m \in l_2 \Rightarrow m_y = a_2 m_x + b_2$,

and $\left|\overline{P_5 n}\right| = \left|\overline{P_5 m}\right|$ then:

$$(n_x - P_{5x})^2 + (n_y - P_{5y})^2 = (m_x - P_{5x})^2 + (m_y - P_{5y})^2$$

This equation have two solutions, but it is easy to reject one solution where $n = m = S_1$. The same operations must be done with the use of S_2 to get the abscissa values. The intersection between $\overline{mn}$ segment and line that goes through the calibrated point and S_1 is known as k. The y-coordinate can be calculated as a proportion $\left|\overline{nk}\right| / \left|\overline{mn}\right|$. Analogical operations need to be done in order to designate x-coordinate.

Tests and comparison

All three methods were tested on the same test images. Test image is an 800x600px checkerboard splitted into 100x100px squares (Fig. 4f). Test image has been recorded in 800x600px from 7 directions as shown in figures 4a – 4e.

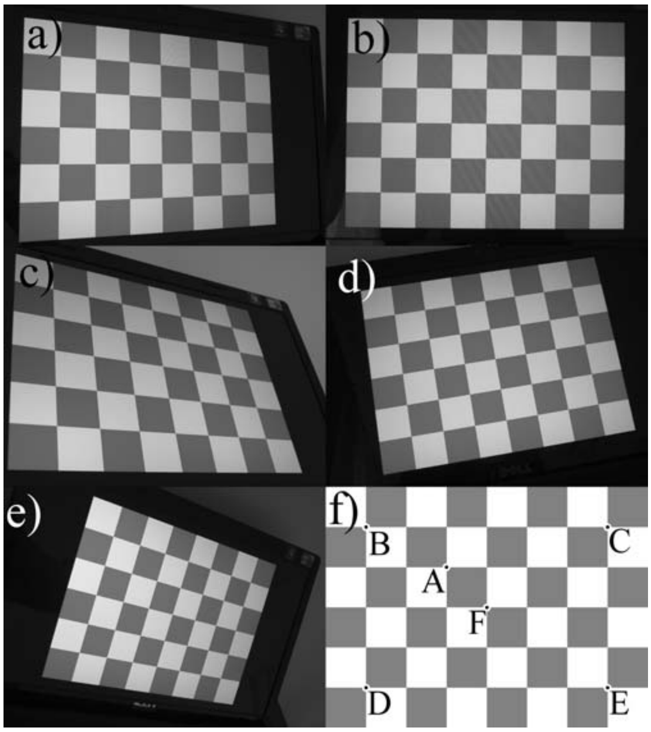

Fig. 4. Recorded test images. Selected test points are highlighted in the f) image

There were 6 points selected to test on each image. Each point corresponds to one of the following original points (Fig. 4f): A=(300;200), B=(100;100), C=(700;100), D=(100;500), E=(700;500), F=(400;300). Deformed view of the checkerboard needs transformation (calibration) to obtain proper position. Three discussed methods for calculation of the transformation were used for the set of test points. The most important problem in application of such transformation is the calculation error. The results of the physical based method and antiperspective projection were very close, so there are practically the same values of the errors. Different results we obtain using linear transformation.

Overall results of the calculations experiments are combined in Fig. 5. Each groups represents result of calibration of the same point on each image. “+” signs represents results of the linear transformation. “x” signs represents results of the physical based model and the antiperspective projection as well. Circle is the reference value. The maximum errors of the simplest, linear transformation reached 6.9%. The error of this method can be acceptable only when calibrated surface is almost perpendicular to camera (rotation has small influence).

The results of methods based on physical model and antiperspective projection are exactly the same. Maximum value of percent error is smaller than 0.9%. That suggests correctness of these two methods. The error is caused by measurement uncertainty and it is connected with image recognition quality.

In order to shorten the calculating time, algorithms can be divided into two steps. The first one is called precalculation and it should be executed only once to calculate parameters for specific test image. Calibration, the second step, is based on the precalculated parameters. It must be executed for each of the calibrated points. Algorithms were implemented in Matlab on Windows operating system.

The fastest during precalculation is the antiperspective method and it is ten times faster than the slowest physical based method. Results are reversed in test of calibration calculation time, but time proportion is smaller. If there is a need to make precalculation as many times as calibration, then the shortest total time gives antiperspective projection. The results of calibration with the use of physical based model and antiperspective projection give identical values

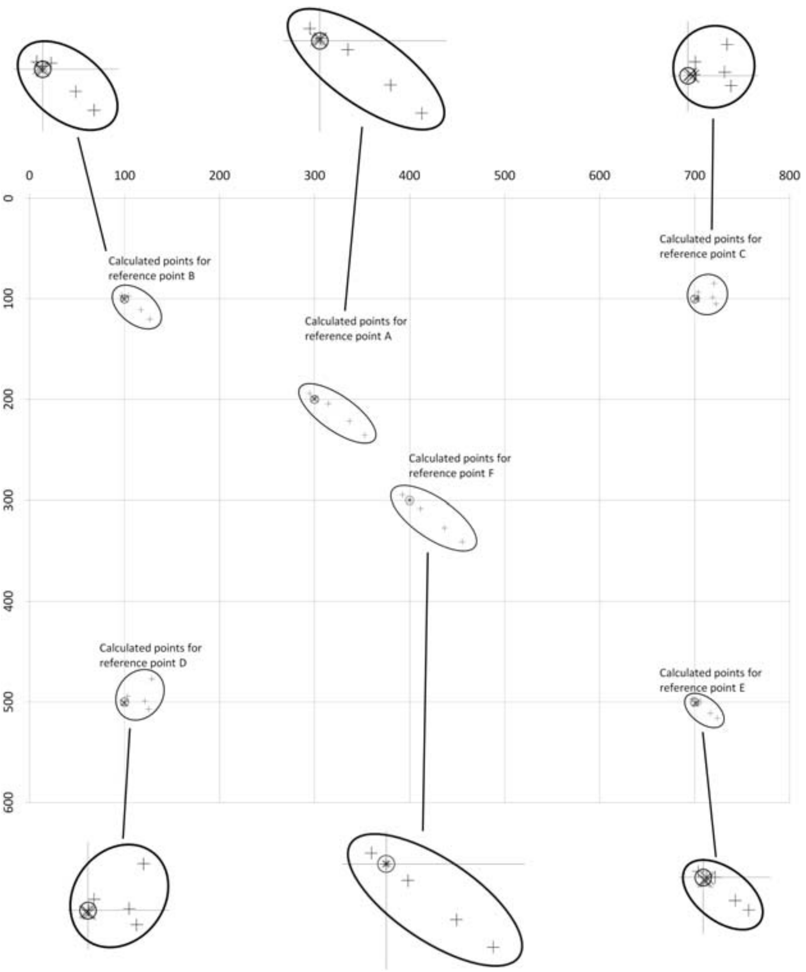

Fig. 5. Result of calibration. “+” signs: results of the linear transformation. “x” signs: results of the physical based model and the antiperspective projection as well. Circle: the reference value

Summary

Two known methods are analyzed: linear transformation and physical model based on perspective projection. The simplest linear transformation is easy to implement. It does not generate a lot of calculations so is very fast, it works sufficiently with rotated image, but it works correctly only when the camera is almost perpendicular to the surface.

Physical based model is the method that is not so easy to implement (especially in low level application – without simple matrix library). The complexity of the implementation relates to solving of the non-linear equations system and matrix inversion in Newton method. It generates a lot of calculations and algorithm is not stabile, because of the fact that the results can depend on (selection of) Newton’s method starting point.

The new method called antiperspective projection is introduced. Precision of physical based model and the new method is equal but the antiperspective projection has less implementation problems. Additionally new algorithm allows preprocessing application in case of most of calculation. Comparison of the presented methods shows that new antiperspective projection is approximately 10 time faster than other method analyzed in this paper.

Correct operation of the implemented algorithm in the exhibit of Warsaw Copernicus Science Centre confirms the validity of the proposed solution.

***Authors**: Marcin Augustynowicz MSc Eng, prof. Dariusz Sawicki PhD DSc, Eng, Warsaw University of Technology, Institute of Theory of Electrical Engineering, Measurements and Information Systems, Koszykowa 75, 00-661 Warsaw, Poland, e-mails: augustynowicz.marcin@gmail.com, dasa@iem.pw.edu.*

REFERENCES

[1] Davies E.R.: *Machine Vision. Theory, Algorithms, Practicalities*. Morgan Kaufmann 2005
[2] Wu Y.H., Hu Z.Y.: The Invariant Representations of a Quadric Cone and a Twisted Cubic. *IEEE Trans on Pattern Anal and Machine Intel.* 25 (10), 2003, 1329-1332
[3] Bahram J. (ed.): *Image Recognition and Classification: Algorithms, Systems and Applications*. CRC Press 2002
[4] Criminisi A., Reid I, Zisserman A.: A Plane Measuring Device. University of Oxford 1997. http://www.robots.ox.ac.uk/~vgg/presentations/bmvc97/criminispaper/ (retrieved 27.10.2014)
[5] Heckbert P.S.: *Fundamentals of Texture Mapping and Image Warping*. Master’s Thesis. Univ of California, Berkeley 1989
[6] Ebert D.S. (et al.): *Texturing & Modeling. A Procedural Approach*. Morgan Kaufmann 2003
[7] Schneider P.J., Eberly D.H.: *Geometric Tools for Computer Graphics*. Morgan Kaufmann 2003
[8] Lengyel E.: *Mathematics for 3D Game Programming and Computer Graphics*. Charles Rover Media Inc. 2002
[9] Hartley R., Zisserman A.: *Multiple View Geometry in Computer Vision (sec ed)*, Cambridge University Press 2004
[10] Dorst L., Fontijne D., Mann S.: *Geometric Algebra for Computer Science*. Morgan Kaufmann 2007
[11] Carlborn J.B., Paciorek J., 1978. Planar Geometric Projections and Viewing Transformations, *ACM Comput Surv.* 10 (4), 465-502
[12] Salomon D.: *Transformations and Projections in Computer Graphics*. Springer Verlag 2006
[13] Augustynowicz M.: *Multipoint Touch Screens* (in Polish), MS Thesis. Warsaw University of Technology 2008
[14] Dorie H.: 100 Great Problems of Elementary Mathematics. Dover Pub. 1965