Proceedings of the RAAD 2011
20th International Workshop on Robotics in Alpe-Adria-Danube Region
October 5-7, 2011, Brno, Czech Republic

On Performance Evaluation of 3D Scene Reconstruction Systems

Tiberiu T. Cocias[a], Sorin M. Grigorescu[a] and Florin Moldoveanu[a]

[a] *Department of Automation, Transilvania University of Brasov, Romania*
E-mail: {tiberiu.cocias, s.grigorescu, moldof}@unitbv.ro
URL: www.rovis.unitbv.ro

Abstract. Nowadays reconstruction techniques of unknown and dynamic scenes in a virtual 3D space have to provide a reliable and precise pose of the target objects, at a low reconstruction error rate. In this sense, object recognition and 3D reconstruction methods are classified into two categories: *marker based* and *a-priori free methods*. In this paper, two such methods are combined in order to evaluate the accuracy of a real-time 3D scene and shape reconstruction system. We consider as ground truth the robust pose estimation delivered by the ARToolKit library, which has an average pose error of 6%, to which we relate the calculated unknown object poses. The evaluation scenario begins with the accurate detection of the ARtoolKit marker fallowed by a manual measure of the real pose. The object of interest is reconstructed in a virtual space using a region based recognition method and a 3D pose estimation approach. Finally, a statistical evaluation of the errors between the estimated objects and the marker is performed.

Keywords. Pose Estimation, Object Recognition, 3D Reconstruction, Augmented Reality

1 Introduction

Today's autonomous robotic systems aim to mimic the mechanisms and the complex behaviour of humans. The need to move, grab or understand an unknown scene represents a crucial requirement for a robotic platform. In this process, a key task is the robust estimation of the robot's *position and orientation* (pose) together with the understanding of the environment. In order to achieve this purpose, the robot must be able to reliably sense the surrounding 3D environment. The 3D sensing problem is usually solved through *stereo imaging* (Hartley and Zisserman, 2004), *laser scanning* (Surmann et al., 2003), or 3D scene estimation with *Time of Flight* (ToF) cameras (May et al., 2006). Each of these three methods has advantages and disadvantages. The stereo vision approach has difficulties in providing reliable real-time 3D information to the robot, as it is also highly sensitive to changing lighting conditions. On the other hand, a laser scanner or ToF camera has a higher precision, but are usually limited to a low sensing area around the robot. Also, because of their active components, lasers and ToF sensors are mainly used indoors.

In this paper, a performance evaluation system for estimating the precision of a real-time 3D sensing platform is proposed. Mainly, such a system includes two components: an *experimental test-bed*, along with ground truth information, and *the vision methods* to be evaluated (Ahmed and Farag, 2004). In order to obtain a good performance evaluation, all the results must be quantified for a reliable objective comparison (Szeliski, 1999), (Szeliski and Zabih, 1999). The work presented in this paper is aimed at the development of a testing setup for evaluating the precision of 3D pose estimation for recognized objects of interest, together with the virtual 3D reconstruction of the imaged scene. The block diagram of the proposed architecture can be seen in Fig. 1. Our performance evaluation approach is based on the ARToolKit library, used as a reliable ground truth, to which the poses of the objects are related.

The rest of this paper is organised as follows. In Section II, a description of the stereo 3D scene reconstruction system is presented, followed by the description of the marker-based pose recognition in Section III. Finally, before conclusions, experimental results are given in Section IV.

2 Stereo scene reconstruction

The stereo vision system enables, through a disparity map, an easy and low time computation mechanism to estimate the depth of a scene. The primary problems to be solved in a stereo system are *camera calibration*, *correspondence matching* between left and right stereo images and *3D*

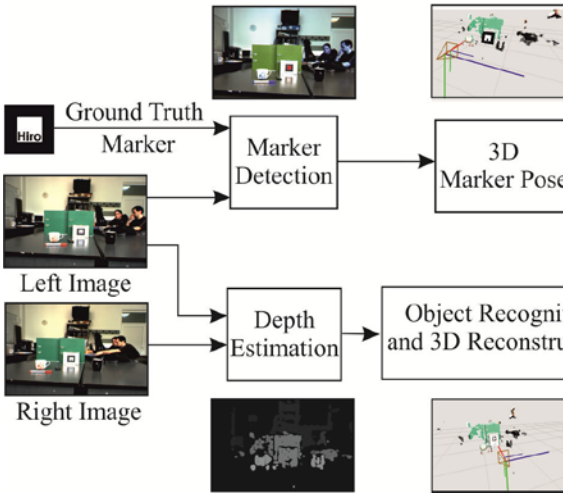

Fig. 1. Block diagram of the proposed real-time 3D object reconstruction system and the 3D pose evaluation approach.

reconstruction. Camera calibration is the process of finding the *intrinsic* (e.g. focal length, optical centre, etc.) and *extrinsic* (camera pose) parameters of the camera that produced a given image. Starting from a set of image correspondence points, $p_L \leftrightarrow p_R$, the calibration task aims to find the camera matrices Q_L and Q_R which describe the projection of a real world 3D point onto the 2D image planes:

$$p_L = Q_L \cdot P \qquad p_R = Q_R \cdot P \tag{1}$$

where X represent a point in the real world and p_L and p_R are the points in the left and right 2D image plane, respectively, as seen in Fig. 2.

2.1 Camera geometry

The accurate estimation of the camera's geometry is critical for relating the image information, expressed in pixels, to an external reference world coordinate system. The camera's geometry is determined during the calibration process. The first geometry gives the pose (position and orientation) of each camera while the internal geometry consists in a series of intrinsic parameters. The camera geometry, also known as projection matrix $Q \in \Re^{3x4}$, is described in Eq. (2),

$$Q = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = K \cdot [R \mid t] \tag{2}$$

where K is the intrinsic matrix, while R and t are the rotation and translation matrices of each sensor, respectively.

The intrinsic matrix stores values which are essential for representing the camera's internal parameters, such as focal length f, skew constant γ, or the optical centre $p_p(u_0, v_0)$. Within the projection matrix Q, the focal length can be computed also in terms of pixels instead of meters. Thus, α represents the focal distance in metric units computed based on the focal length and a scale factor m relating pixels to distance as follows:

$$\begin{aligned} \alpha_x &= f \cdot m_x, \\ \alpha_y &= f \cdot m_y. \end{aligned} \tag{3}$$

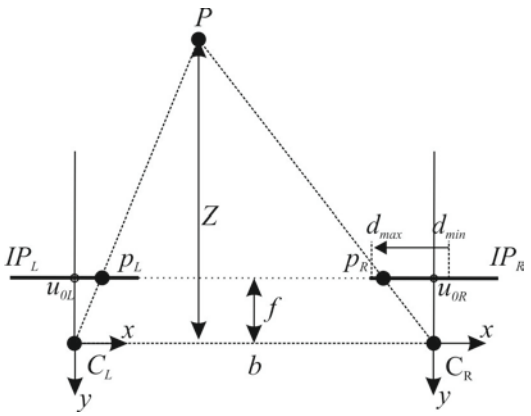

Fig. 2. Estimation principle of a real world point *P*.

Having computed all the matrices which geometrically describe the scene, we can focus the attention on computing the 3D coordinates of the information form the images. The most common method used is triangulation. In this sense, a set of correspondence features between the left $I_L(x, y)$ and right $I_R(x, y)$ images has to be found. Based on the projection of the real world point

$P = [X\ Y\ Z\ 1]^T$ onto each *principal plane*, the projected 2D image points of *P* are:

$$\begin{cases} p_L = \begin{bmatrix} x_L & y_L & 1 \end{bmatrix}^T \\ p_R = \begin{bmatrix} x_R & y_R & 1 \end{bmatrix}^T \end{cases} \quad (4)$$

The principle planes, in which the points p_L and p_R are projected, are positioned between the optical centre C_L and C_R of each camera sensor and point *P*. The image is formed at a distance equal to the focal distance from the optical centre. Each pixel from the image is determined by intersecting the principal plane with the line connecting the optical centre of the sensor and the corresponding point *P* from the real world.

2.2. Depth perception

In order to navigate through a scene without colliding with different objects, a robot must be able to sense the environment. Through stereo geometry this can be achieved by computing the so-called *disparity map* which represents an image having its pixel intensities correlated with the distance between the camera and the scene. Based on this map and on the projection matrix obtained during the calibration process, the visualised scene can be rendered in a virtual 3D space.

One important issue is to perfectly align the coordinates of the camera and world. Thus the image plane will be aligned with the world plane assuring that the depth perception of a point *P* will be the same in both images.

The depth is determined by finding correspondences between the two stereo images and computes the difference of the point coordinates from the left and the right image as in Eq. (5)

$$Z = \frac{f \cdot b}{d}, \quad (5)$$

$$d = x_L - x_R \ . \quad (6)$$

The computed depth is inversely proportional with the disparity *d* calculated in Eq. (6), where the x_L and x_R are the coordinates of the same world point *P* projected onto the image planes of the stereo camera. The disparity is computed by a method called *feature matching* which search the corresponding point from the left image into the right image. This is called *correspondence matching*. More precise, the algorithm tries to match a window of pixel in the left image with a corresponding sized window on the right one as can be seen in Fig. 3.

The time needed to compute the search increase substantially for high resolution images, because the matching window has to be moved, in search of a match, through the entire image domain. This issue can be simplified by using rectified images. Image rectification is a transformation process used to align the two images to a common axis.

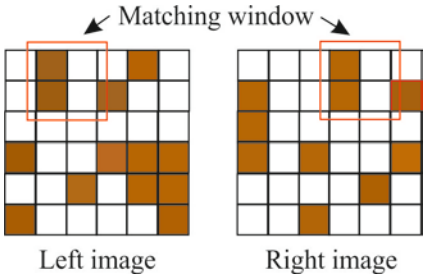

Fig. 3. Stereo matching correspondence search.

Rectification corrects a distorted image by transforming the image into a standard coordinate system (Oram, 2001). The basis of rectification is the *epipolar geometry*. This geometry aims to reduce the search of a correspondence point to a search line. This line is called epipolar line *L*, and is generated by intersecting the epipolar plane with the image plane. The epipolar plane is the plane that contains the reference line and the epipolar line.

The idea of epipolar geometry is that a point from the left image generates an epipolar line in the right image, thus the corresponding point from the right image lies on the respective epipolar line. This substantially reduces the time during the search of correspondence (Hartley and Zisserman, 2004).

The correspondence computation can be tackled in different ways, as in (Brown et al., 2003), although the most popular correspondence matching algorithm used in robotics is *Block Matching* (BM).

The BM method uses wither of the two main correlation functions named: *Sum of Squared Differences* (SSD) (see Eq. (7)) and *Sum of Absolute Differences* (SAD) (see Eq. (8)). Both functions are applied on small sliding windows in the image domain. The SSD function is commonly used as the similarity measure. Because of the power compitation involved, the SSD method suffers from the windowing problem and computational cost (Trucco and Vierri, 1998).

$$SSD(x) = \sum_{x}^{x+d} \left[I_L(x,y) - I_R(x+d,y) \right]^2 \quad (7)$$

$$SAD(x) = \sum_{x}^{x+d} \left| I_L(x,y) - I_R(x+d,y) \right| \quad (8)$$

For these two reasons the SSD method cannot be used in real-time application for autonomous robots because the system reaction is to slow to avoid an obstacle. The second approach is more feasible and simple. He simply searches the corresponding SAD value in the right image using a shifting window. The window moves along an interval *H* called *horopter*

described in Eq. (9). *H* is defined as the area covered by the search range of BM.

$$H = \lfloor d_{\min} \ , d_{\max} \rfloor \tag{9}$$

Having computed all the disparities, a disparity map can be obtained. This is a grey scale map where the intensity represents depth. The lighter shades (greater disparities) represent regions with less depth as opposed to the darker regions which are further away from the camera.

2.3. Stereo object detection

The principle of object detection and pose estimation is to find a unique set of features (shape, colour, area, texture, etc.) that characterise that object. Based on these features we can eliminate all the unwanted elements and keep only the object that fit to the profile. In this paper we used for evaluation a method based on colour segmentation.

The method consists in three main stages which are *image enhancement*, *colour segmentation* and *object detection*. In the image enhancement step, a spatial filter is applied on all 3 images belonging to the RGB (*Red*, *Green*, and *Blue*) format. The filter computes for each pixel a mean of the intensities of neighbourhood pixels which ensure a smooth colour translation between pixels intensities in an image (Gonzalez and Woods, 2002).

The second stage aims to separate the object of interest from other objects or environment. Thus, the colour segmentation is based on the *HSI* (*Hue*, *Saturation* and *Intensity*) colour model. These three elements can be extracted separately from the *RGB* images. Through this model we aim to obtain the colour information from only one plane, which is the *hue* plane, instead of three different planes. This simplifies the entire detection mechanism. The hue image $I_H(x, y)$, is segmented in order to separate the object of interest from the environment.

The segmentation process implies the elimination of all the colour information that do not belong to a threshold interval $t_H(x, y)$. The output image is a binary image which respects the following equation:

$$t_H(x,y) = \begin{cases} 1, \text{ if } I_H(x,y) \in [T_1, T_2] \\ 0, \text{ if } I_H(x,y) \notin [T_1, T_2] \end{cases} \tag{10}$$

The two threshold values are dynamically computed, as in (Grigorescu, 2010), in order to obtain trustworthy object identification. After the segmentation, the process continues with detection of the object contour in the 2D image plane using the chain-code-border method (Bradski and Kaehler, 2008). There are little chances that the contour of the segmented object to be similar to the contour of an unwanted shape.

The next step is to compute the moments of binary image. The moments represent a certain particular weighted average of the image pixel's intensities, defined as (Bradski, 2008):

$$M_{i,j} = \sum_{k=1}^{n} I_I(x,y) x^i y^j \tag{11}$$

where, $I_I(x, y)$ represent the intensity image and $M_{i,j}$ is the moment for a specific position *i*, *j*. Next, Hu moments are computed in in order to calculate a set of object recognition coefficients which are invariant to rotation, scaling and translation. Based on this uniqueness, the centre of gravity of the object of interest can be computed. Having this scenario implemented on the two stereo images, the 3D pose of the object can be computed based on the same principle as in the triangulation algorithm. In this case the projection points p_L and p_R are equivalent with the centres of gravity c_{gL} and c_{gR}.

3. Marker-based 3D reconstruction evaluation

In order to evaluate the accuracy of the estimated pose of the detected object we used a technique highly involved in the *Augmented Reality* (AR) domain. Such methods are intensely applied on mobile navigation, active tracking, computer tomography surgeries (Rosen and Laub, 1996) and military training (Urban, 1995). Thanks to the low processing cost and reasonable performance, accuracy and robustness, the method gained more and more popularity in complex robotic applications.

3.1 Ground truth

The usage of the term *ground truth* lays on complex domains like metrology, aerial photography and other remote sensing techniques. It refers to a process which compares the sensing information with the real information in order to analyse the accuracy of the content.

In our approach, the ARToolKit pose estimation is used as a ground truth in order to analyse and evaluate the pose estimation of the detected object.

We take as an absolute measurement the pose of the marker to which we relate the object pose. Base on this observation we determine the accuracy of the 3D reconstruction method.

3.2 ARToolKit marker detection

ARToolKit is a complex software library which enables a real-time computation of the marker pose together with accurate view point estimation.

The robust marker detection starts with the segmentation of the input image. The algorithm

needs only one image from the stereo image pairs, left or right, to fulfil the task. The enhancement consists in a binarization of the image based on a light threshold value. This step eliminates most of the unwanted information from the scene. Then, the detecting scenario continues with a search of all the squares from the image. Because the number of squares found is high, ARToolKit eliminates all the figures which have a low index of confidence. In the kept squares is fitted a specific pattern in order to uniquely identify a marker. If the pre-trained pattern fits perfectly on the pattern inside the square, the detection process receives a confidence note in the range 0 to 1. The average confidence mark during the tests was ≈0.879. ARToolKit then uses the known square size and pattern orientation to calculate the position of the real video camera relative to the physical marker. A 3x4 matrix *M* is filled in with the video camera real world coordinates relative to the card, see Eq. (12):

$$M_{Cam \to Marker} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = [R \mid t]. \quad (12)$$

The rotation matrix *R* is transformed into a rotation vector in order to set the position of the real camera coordinates. The translation matrix can be used to evaluate the distance between the camera and the marker. The bloc diagram of the real-time marker detection algorithm is described in Fig. 4.

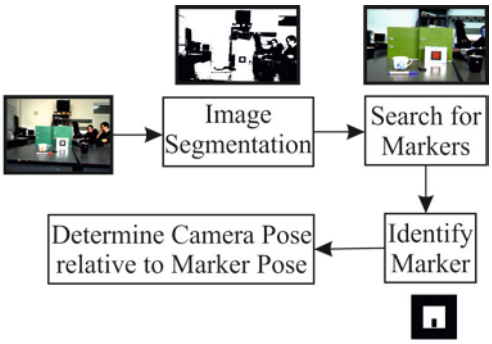

Fig. 4. Bloc-diagram of the real time marker detection algorithm.

Thanks to the a-priori information about the size of the square, the accuracy of the depth estimation is very high, the error ranking around 0.02% (see Tab. 1 below) (Malbezin et al., 2002).

Tab. 1. Error values for different distances.

Distance (m)	1	1.5	2	2.5
Error (mm)	± 14	± 18	± 22	± 27

4. Experimental results and analysis

Through this section, we study the performance of a 3D pose and scene understanding technique. First, the performance of pose estimation is exanimated when the camera pose is changed. Secondly, we made a preliminary pose comparison between the object detection and 3d reconstruction techniques and the pose of the marker detected using ARToolKit® library.

The real 3D position and orientation of the objects were manually determined based on the visual marker planted in the scene. The marker was installed near the object of interest in order to make easy measurement of the object pose relative to the marker.

A total number of 50 images containing 12 object of interest, such as balls, bottles, mugs or books, were involved in the evaluation process. All the experiments were conducted in an indoor environment with a constant illumination. The images were captured using a Point Grey Bumblebee® stereo camera.

In order to get a reliable and stable measurement, for both methods, the estimations were calculated 20 for each frame and averaged the values. In this way any noise values are eliminated. The pose of the robot has been varied during the scene crossing. Relative to the robot pose, the detected object pose with respect to the ground truth marker is illustrated in Fig. 5. As can be seen from the diagrams, the method used for estimate the pose of an unknown object shows values very closed to the real coordinates along the Cartesian axes. In Tab. 2 are shown statistical measures of achieved errors for the objects position estimation experiments. The block-diagram of the proposed scenario is described in Fig. 1.

Thanks to the good results obtain during pose estimation and object detection, the system under test represent a proper method that can be applied in usual robotic activities like identifying or grasping.

Tab. 2. Statistical results of the position error for the estimated object.

Axis	X_e (mm)	Y_e (mm)	Z_e (mm)
Max. error	122.2	39	69.7
Mean	44.61	9.66	67.24

5. Conclusions

In this paper, we have tested the performance of an object detection and 3D reconstruction system for real-time scene understanding. The measurement technique used showed consistent results in

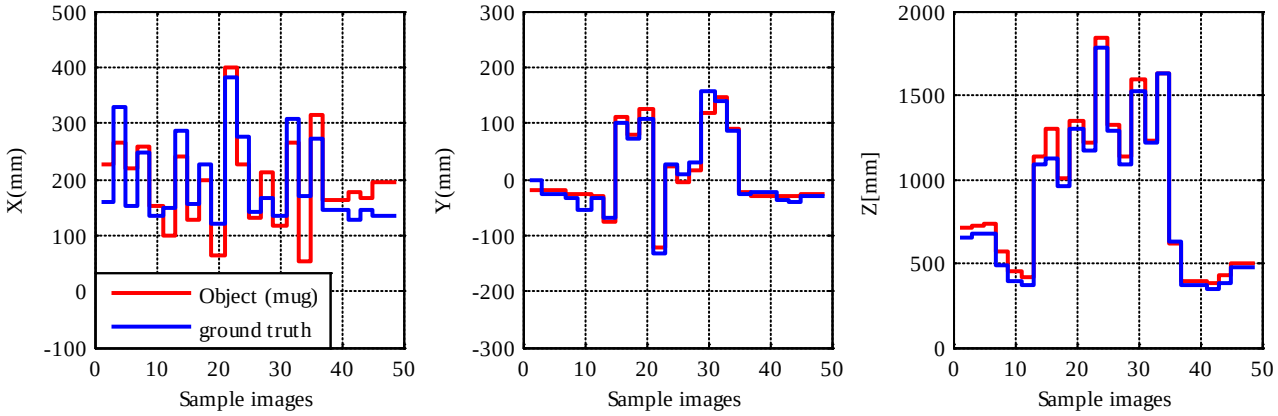

Fig. 5. Comparison of distances between ground truth and estimated object position along all 3 axes.

estimating the performance of the pose estimation system. As future work, the authors consider the extension of different test strategies as a key element in quality assurance measurements.

Acknowledgments

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOPHRD), financed from the European Social Fund and by the Romanian Government under the contracts numbers, POSDRU/107/1.5/S/76945, POSDRU/89/ 1.5/S/59323.

References

Ahmed, E .and A. Farag. 2005. On the Performance Evaluation of 3D reconstruction Techniques from a Sequence of Images. In: *EURASIP Journal on Applied Signal Processing*, January, pp. 1948-1955

Bradski, G. and A. Kaehler. 2008. *Learning OpenCV*, O'Reilly Media.

Brown, M., Burschka D. and G. Hager. 2003. Advances in computational stereo, *IEEE Trans. on Pattern Recognition and Machine Intelligence*, Vol.**25**(8), pp.993–1008.

Gonzalez, R. and E.R. Woods. 2002. *Digital Image Processing*, Prentice Hall.

Grigorescu, S.M. 2010. *Robust machine Vision for Service Robotics.* PhD thesis, Bremen University, Institute of Automation, Bremen (Germany), June.

Hartley, R. and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision*, Cambridge University Press.

Hussmann, S. and T. Liepert. 2007. Robot Vision System Based on a 3d-tof Camera. In: *Instrumentation and Measurement Technology Conference-IMTC 2007* , Warsaw (Poland).

Malbezin, P., Piekarski, W. and B. H. Thomas. 2002. Measuring artoolkit accuracy in long distance tracking experiments, In: *1st Int. Augmented Reality Toolkit Workshop*, Darmstadt (Germany), September.

May, S., Werner B., Surmann, H. and P. Kai. 2006. 3D Time-of-Flight Cameras for Mobile Robotics. In: *IEEE International Conference on Intelligent Robots and Systems*, Beijing (China), 9-15 October, pp. 790-795.

Oram D. 2001. Rectification for Any Epipolar Geometry. In: *Proceedings of the 12th British Machine Vision Conference(BMVC)* , pp.653-662.

Rosen, J. M. and D. R. Laub. 1996. Virtual Reality and Medicine: From Training Systems to Performing Machines. In: *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, Santa Clara (USA), pp.5-13.

Surmann, H., Nüchter, A. and J. Hertzberg. 2003. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments, *Robotic and Autonomus Systems*, Vol. **35**, pp. 181-198.

Szeliski, R. 1999. Prediction error as a quality metric for motion and stereo. In: *Proc. 7th IEEE International Conference on Computer Vision (ICCV '99)*, Vol. **2**, pp. 781–788.

Szeliski, R. and R. Zabih. 1999. An experimental comparison of stereo algorithms. In: *Proc. International Workshop on Vision Algorithms*, Corfu (Greece), September, pp. 1–19.

Trucco, E. and A. Verri. 1998. *Introductory Techniques for 3-D Computer Vision*, Prentice Hall.

Urban, E. C. 1995. The Information Warrior, *IEEE Spectrum*,Vol.**32** (11), pp.66-70.