International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 4, No. 3, September, 2011

Location Assignment of Recognized Objects via a Multi-Camera System

Rigas Kouskouridas, Antonios Gasteratos,
Group of Robotics and Cognitive Systems,
Department of Production and Management Engineering,
Democritus University of Thrace
Vas. Sofias 12, Building I, 67100 Xanthi, Greece
Coresponding author: rkouskou@pme.duth.gr

Abstract

This paper aims at making a contribution to a typical recognition scheme by means of providing position information of the recognized objects. In the near future, robotics systems are expected to be able to provide services in humans' daily life which can only be achieved if they are designed with advanced intelligent vision systems. Highly motivated from that fact, this work presents a framework capable of recognizing objects and estimating their location in the 3D space. The method excels in simplicity and computational cost, whilst its database can be easily adopted to the needs of simultaneous multi-object recognition. In real-life scenarios, lighting conditions may alter drastically and, along with possible shadow effects, may affect directly the efficiency of the visual data encoding scheme. Towards this end, recently proposed image enhancement methods emphasize in reducing the effect of unfavorable illumination conditions and their consequences. The proposed 3D position estimation framework was assessed under several image-enhancement conditions by means of selecting the most appropriate pre-processing algorithm.

1 Introduction

The task of recognizing objects found in a particular scene attracted computer vision researchers' attention for several decades, whilst instances where recognition suffers from occlusions, noise, cluttered background or even poor lighting conditions, test the robustness of the respective algorithms. Local-appearance-based object detection frameworks dominate due to their ability to extract meaningful visual information efficiently and accurately. State-of-the-art recognition frameworks [32, 27, 22, 6] rely on two mechanisms, those of the detector and descriptor, respectively. The first one is responsible for extracting spots or areas of a scene that can be easily distinguished in their local setting. As a follow-up step, a descriptor is devoted to the transformation of information collected from the detector into high dimensional feature vectors. Consequently, an object is represented by these vectors. A method comprising of both detector and descriptor is referred in the literature as a two-part approach. SIFT [21] and SURF [9] are the two most popular ones adopted in almost any newly proposed recognition framework.

An aspect that has not received much attention in the literature, is how to exploit the data derived during recognition with a view to provide objects' spatial information. Apart from its identification, several other object-related characteristics, such as its distance to the camera or its pose, i.e.

orientation relative to the camera's plane, can be obtained [28, 24, 12, 14]. The problem of actively searching for a target in a 3D environment has lately received attention in [8], where both multi-view and single-view recognition and detection schemes are exhaustively examined. However, the results stand only in theory, whereas sophisticated vision systems require realistic and practical measurements of targets' location in the 3D space. On the other hand, it has been shown [15] that it is possible to estimate objects' location in a scene by combining 3D-based target models with information derived from a single 2D image. The final 6-Degree of Freedom (DoF) localization is obtained by accumulating the correspondences between extracted features and the 3D target on a Hough table. However, the main drawbacks of the method include both its deficiency to recognize and estimate the location in the 3D space of non-rigid objects and the unnecessary extraction of irrelevant low-level features. A framework with low computational demands that utilizes stereo cameras, a large amount of frame memory and several vision modules has been presented in [34]. The outcome of this algorithm is a real-time solution in terms of both object recognition and position estimation. However, its limitations stem from the constraints of objects' CAD-based model knowledge and the contour-based feature extraction. A recent algorithm for searching a scene for unknown objects in the 3D environment is shown to provide remarkable results [26]. That method was implemented in a mobile, four-wheeled robotics platform equipped with a Point Grey Research Bumblebee camera mounted on a pan-tilt mechanism. This particular method relies on the efficient estimation of a scene's depth obtained by embedded camera's modules and the target's detection accomplished via SIFT.

In the field of multi-camera systems, the Stanford Multi-Camera Array [29] has dominated, over a period of several years, as the ultimate multi-view framework. It consists of 128 cameras that can be arranged and used in a variety of ways. In this system, the final camera architecture corresponds to an arc of separate panels aiming at a point in the middle of the room. The multi-camera system, developed by the UD Graphics Lab [35], consists of 10 Flea2 1394b cameras, with a subset of them mounted on movable plates attached to ceiling racks and the remaining placed at different positions using tripods. The target applications of the aforementioned framework are multi-view surveillance, capturing of dynamic fluid surfaces or appearance modeling. In addition, the Mitsubishi Electric Research Laboratories (MERL) has designed a framework, where the object-wise semantics are extracted from a non-overlapping field-of-view multi-camera system [36]. Its main goal is the construction of an automatic object tracking and video summarization method via background subtraction and mean-shift analysis.

This paper intends to present a simple and easy to construct framework for location assignment of multiple objects found in a scene. It emphasizes in building a sophisticated scheme with low complexity and computational burden that could be adopted to systems with moderate processing capabilities. This is due to the fact that, the proposed algorithm estimates only the 3D position and not the orientation of any target in the workspace. Therefore, it avoids the laborious estimation of the full pose using planar homographies [14]. The proposed framework is based on a multi-camera system comprising of four Grasshopper cameras [1] manufactured by PointGrey Research. The algorithm exploits basic proportional geometrical attributes of the object in a scene, related to the camera. The whole process can be understood as a two stage procedure, where the first corresponds to the recognition task and the second to the 3D position estimation one. As far as recognition is concerned, the algorithm requires, but is not limited to, any two-part method (both a detector and a descriptor), whilst in the current implementation the classical SIFT is adopted. The training phase of the system is devoted to its database construction where objects-targets are registered. As a follow-up step, the location assignment process of the sought objects takes into account essential visual data derived during recognition. In particular, it exploits features distribution over the object's surface in

Figure 1: a) A scene containing several objects and its enhanced version when applying the b) CLAHE, c) FLOG, d) MSRCR, e) ORASIS and f) SAPONARA algorithms, respectively.

order to effectively estimate its location in the 3D space. The proposed method excels in simplicity and computational cost, whilst its main contribution is its ability to transform information from the image plane to the real world. Our claim that the distributions' similarity estimation can produce an accurate measurement about objects' position in the 3D space is demonstrated experimentally. Furthermore, acquired images may suffer from poor lighting conditions or inherent noise, leading to reduced viewing quality and, consequently, to limited extracted features. Towards this end, in the same paper, the impact of several image-enhancement methods is compared, when acting as a pre-processing step. The one leading to optimum recognition results is selected for our system.

2 Image Enhancement Methods

During the past decade machine vision evolved sufficiently enough to provide sound solutions to numerous complex and demanding tasks [19, 10, 7, 11]. Moreover, images obtained by a vision sensor contain large amount of noise accompanied with poor illumination circumstances that result in poor image clarity. Several image enhancement algorithms try to solve this problem. [23, 20, 16, 30, 25]. In this paper a limited selection of a large number of contrast modification algorithms was assessed, particularly, those that provide the most promising results and, in more wide use in computer vision. More specifically, the methods utilized are the:

- Contrast Limited Adaptive Histogram Equalization (CLAHE) [23]
- Fused Logarithmic Transform (FLOG) [20]
- Multi-scale Retinex with Color Restoration (MSRCR) - PhotoFlair [16, 2]
- Fast Centre-surround Contrast Modification - ORASIS [3]
- Algorithmic and Architectural Design for Retinex Processing - SAPONARA [25]

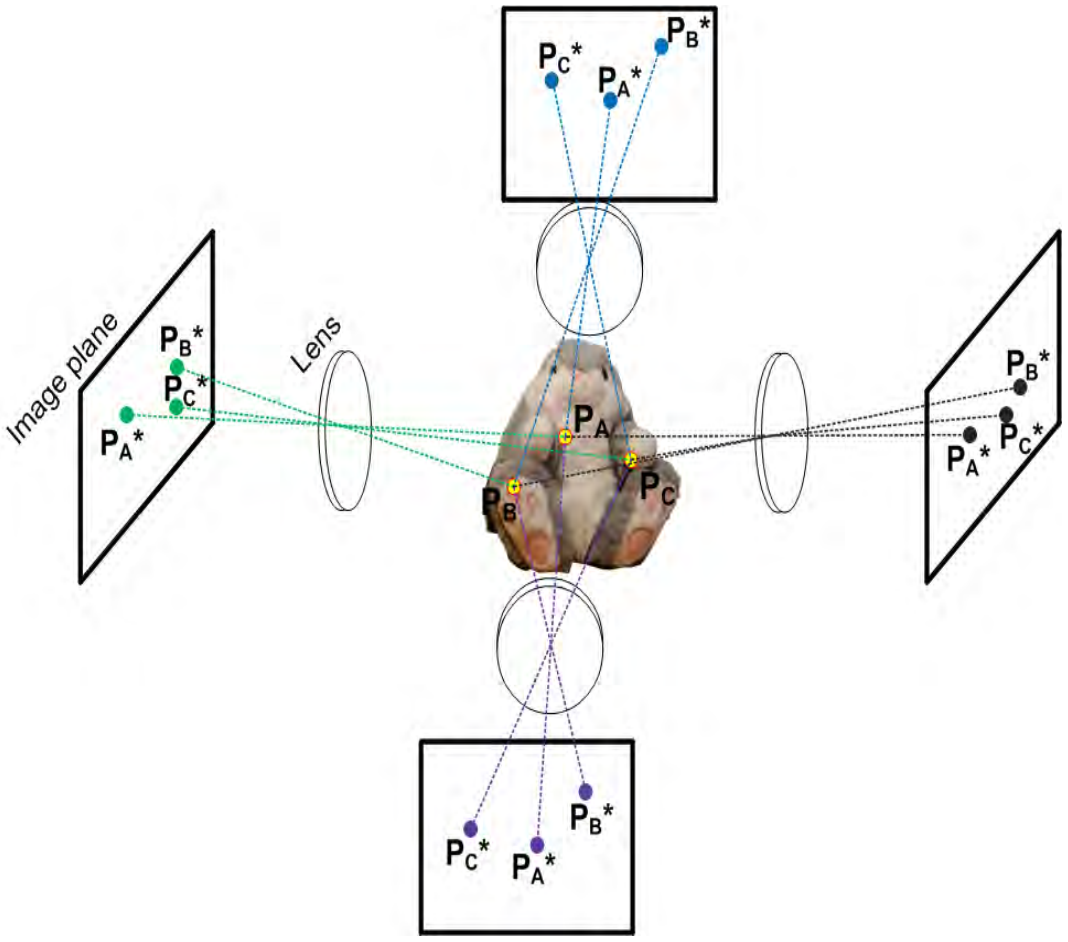


Figure 2: The conceptual idea of the proposed 3D position estimation method. An object is recognized and its position within the working volume, defined by the multi-camera system is estimated by taking into account features' distribution over the surface of the sought target.

The tenacious reader can refer to [31] for a more exhaustive comparison of the aforementioned image enhancement algorithms. The effect of each algorithm in a test scene is depicted in Figure 1.

3 3D Position Estimation

The goal of this paper is to exploit data derived through recognition, in a way that the absolute position in the three dimensional space of the sought objects can be easily estimated. The key idea underlying the proposed 3D position estimation method is presented in Figure 2. The multi-camera system consists of four sensors corresponding to image planes A, B, C and D respectively. During training, where the system remains off-line, several objects from different viewpoints and distances to each camera are captured and stored into the database. Features extracted via any two-part method are distributed over the surface of the objects whilst, in the particular example of Figure 2, object's feature points (P_A, P_B, P_C) are located on the respective image planes and represented by (P_A^*, P_B^*, P_C^*). Thus, given that one distribution of features corresponds to a known object's distance from the camera [17, 18], the goal is to estimate the latter in cases of new features' scattering. Thus, we are able to calculate the distance of objects from the cameras in any environment, which in turn,

could be labeled as a top-down depth estimation technique.

In addition, multi-camera calibration provides spatial data with respect to the cameras' arrangement in the 3D space in terms of absolute positions. In other words, due to the performed calibration, each camera's pose in the 3D environment, as well as the relative pose to each other, are precisely known. Since spatial information concerning both, the distance of the objects' from each camera and the geometrical alignment of the cameras in the 3D space is known, it is possible to assign an absolute position to the recognized objects. The most significant contribution of the proposed method is its ability to translate information, derived from even one single 2D image, into a registration of the objects in the 3D space.

The proposed algorithm is divided into two discrete phases: an off-line and an on-line one. Cameras' calibration techniques aim at estimating the intrinsic and extrinsic parameters of the respective sensors. In the proposed algorithm, the intrinsic attributes are extracted via the toolbox available in [4], while the extrinsic ones by adopting the method proposed in [33] for multi-camera calibration. As a result, information concerning the cameras' position during the capturing process and their coordinates in the 3D space are extracted. Notationally, the four cameras' coordinates in the real world correspond to $(X,Y,Z)^i$, while each sensor's intrinsic parameters are the focal length, f^i and the principal point, $C = (C_x, C_y)^i$ for *i=1,2,3,4*, respectively. During the off-line phase of the algorithm, the training session of the proposed method also takes place; where several objects are shot from different viewpoints and distances from the cameras. The distance of each object along with its identification number and the camera used for capturing are also stored into the database.

The second phase of the proposed 3D position estimation method consists of an on-line system that requires visual input from the four cameras, i.e. four monocular images, and provides an accurate location estimation of the recognized objects. We estimate the distance of objects from a single camera first by extracting feature points positions using SIFT in both the scene's (streamed by the camera) and the object's (retrieved from the database) image. We have chosen SIFT (from a large deposit of available two-part approaches) due to the fact that it produces larger features' distributions over the objects' surfaces. In [18], we have evaluated the most popular two-part approaches, namely SIFT and SURF, in objects' depth estimation tasks. SIFT has proved to be more reliable due to the fact that the extracted features appear more scattered than the ones extracted by SURF. Furthermore, by adopting the matching sub-procedure of SIFT we obtain the N features that match in the two images. We define as $(X_{S_j}, Y_{S_j})^i$, where *j=1,2,...,N* and *i=1,...,4* the positions of the N features on the scene's image plane as captured by camera i and (X_{O_j}, Y_{O_j}), where *j=1,2,...,N* those in the object's one. Moreover, the features' centers of mass for both images are represented as $(X_{S_C}, Y_{S_C})^i$ and (X_{O_C}, Y_{O_C}), respectively, and are calculated via the following expressions:

$$\begin{aligned} (X_{S_C})^i = (\frac{1}{N}\sum_{j=1}^{N} X_{S_j})^i \text{ and } (Y_{S_C})^i = (\frac{1}{N}\sum_{j=1}^{N} Y_{S_j})^i \\ X_{O_C} = \frac{1}{N}\sum_{j=1}^{N} X_{O_j} \text{ and } Y_{O_C} = \frac{1}{N}\sum_{j=1}^{N} Y_{O_j} \end{aligned} \tag{1}$$

As a following step, the mean Euclidean distance of each feature from the corresponding center of mass is computed. $(E_S)^i$ and E_O, are those distances in the scene's and object's image plane,

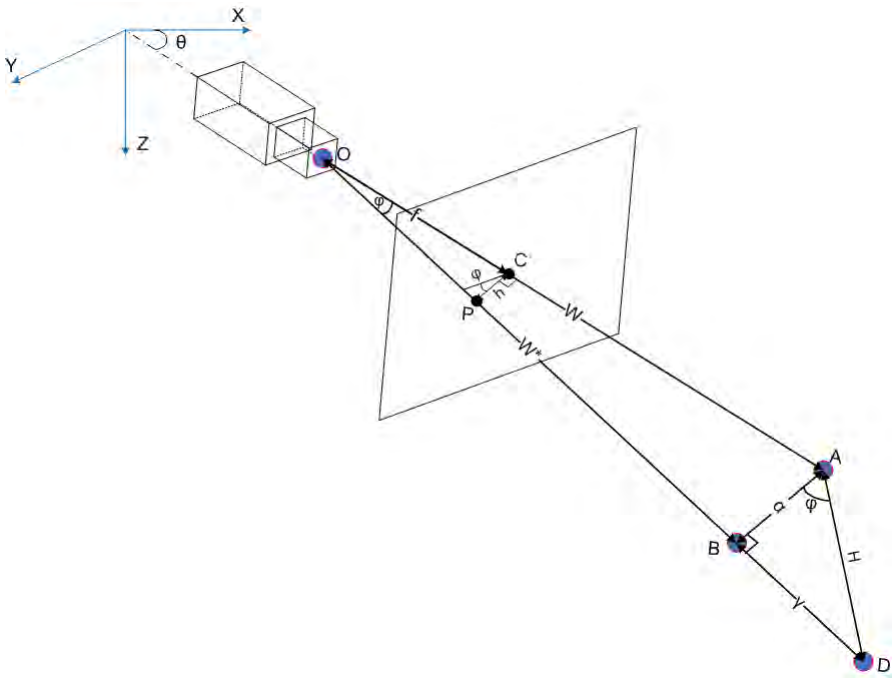


Figure 3: The projection of the actual object's distance from the camera (W^*) on the camera's optical axis.

respectively and are estimated as:

$$(E_S)^i = \left(\frac{1}{N}\sum_{j=1}^{N}\sqrt{(X_{S_j}-X_{S_C})^2+(Y_{S_j}-Y_{S_C})^2}\right)^i$$
$$E_O = \frac{1}{N}\sum_{j=1}^{N}\sqrt{(X_{O_j}-X_{O_C})^2+(Y_{O_j}-Y_{O_C})^2} \tag{2}$$

In the next stage, we estimate the ratio d_S of the two mean distances $(E_S)^i$ and E_O, respectively. The initial object's distance W from the camera i is obtained by multiplying this ratio with the respective distance from the sensor (d_O) as measured during training.

$$W = d_O \cdot d_S = d_O \cdot \frac{E_O}{(E_S)^i} \tag{3}$$

Although, the aforementioned process provides a relatively accurate estimate for W, the realistic and more representative estimation is derived by taking into account the attributes of the pinhole camera model as it is presented in Figure 3, where W corresponds to the object's distance from the camera only in the cases where the latter lies on point *(A)* belonging to the optical axis, i.e. the line passing through the sensor's principal point C. In any other case, the line starting from point D crosses the image plane at point P and ends at point O, forming an angle $\hat{\varphi}$. Point P on the image plane corresponds to the features' center of mass of the recognized object. Moreover, the triangles $\overset{\triangle}{OCP}$ and $\overset{\triangle}{OAD}$ shown in Figure 3 are similar, therefore:

$$\frac{OC}{CP} = \frac{OA}{AD} \Rightarrow \frac{f}{h} = \frac{W}{H} \Rightarrow H = \frac{W \cdot h}{f} \tag{4}$$

In turn, angle $\hat{\varphi}$ is estimated on the image plane taking into account the coordinates of both the principal point and the features' center of mass. Thus, $\hat{\varphi} = atan2(y,x)$. Moreover, from triangle $\overset{\triangle}{BAD}$ we obtain that $\alpha = cos(\varphi) \cdot H$ and $\gamma = tan(\varphi) \cdot \alpha$. In addition, the triangles $\overset{\triangle}{OAD}$ and $\overset{\triangle}{BOA}$ are similar resulting in:

$$\begin{aligned} &\frac{OD}{AD} = \frac{OA}{AB} \Rightarrow \frac{W^* + \gamma}{H} = \frac{W}{a} \Rightarrow \\ &(W^* + tan(\varphi) \cdot \alpha) \cdot cos(\varphi) = W \Rightarrow \\ &W^* = \frac{W \cdot (f - tan(\varphi) \cdot cos^2(\varphi) \cdot h)}{cos(\varphi) \cdot f} \end{aligned} \tag{5}$$

which corresponds to the actual object's distance from the camera projected on the image plane. It is apparent that, for the above estimation a single view is required, to extract angle $\hat{\varphi}$ and distance h. Thus, from a single 2D view we may obtain reduced information regarding the 3D positioning of the object.

The four cameras of the system are positioned at the respective corners of a rectangle room with a known geometry, whilst all the sensors lay at the same level of the vertical axis (Z). Furthermore, the global reference system noted with $(X,Y,Z) = (0,0,0)$ is assigned to camera 1 and, therefore, cameras 2,3 and 4 are set at $(\delta,0,0), (\delta,\kappa,0)$ and $(0,\kappa,0)$, respectively. By applying the aforementioned steps in the proposed algorithm we are able to estimate the distance of the object from each camera. Its final position in the 3D space is calculated by taking into account information derived from all the four viewpoints. Practically, an object lies on the surface of a sphere centered on the camera's frame with a radius equal to the respective object's distance from the sensor. It is apparent that the sought object's location is established at the intersection of any two spheres corresponding to the respective sensors. To this end, the spheres' equations for all cameras are represented by the following formulae:

$$x^2 + y^2 + z^2 = Q_1^2 \text{ for camera 1,} \tag{6}$$

$$(\delta - x)^2 + y^2 + z^2 = Q_2^2 \text{ for camera 2,} \tag{7}$$

$$(\delta - x)^2 + (\kappa - y)^2 + z^2 = Q_3^2 \text{ for camera 3 and} \tag{8}$$

$$x^2 + (\kappa - y)^2 + z^2 = Q_4^2 \text{ for camera 4} \tag{9}$$

where Q_1, Q_2, Q_3 and Q_4 correspond to the object's distances from the cameras 1,2,3 and 4, respectively. Let us consider the intersection of the two spheres corresponding to cameras 1 and 2:

$$x = \frac{\delta^2 + Q_1^2 - Q_2^2}{2 \cdot \delta} \tag{10}$$

The intersection of the two spheres corresponds to a curve lying in a plane parallel to the YZ plane. By substituting Eq. (10) in Eq. (6) we obtain:

$$\begin{aligned} y^2 + z^2 &= Q_1^2 - \left(\frac{\delta^2 + Q_1^2 - Q_2^2}{2 \cdot \delta}\right) \\ &= \frac{4 \cdot \delta^2 \cdot Q_1^2 - (\delta^2 + Q_1^2 - Q_2^2)^2}{4 \cdot \delta^2} \end{aligned} \tag{11}$$

Table 1: The errors obtained after calibrating the four cameras.

	Camera 1	**Camera 2**
Focal length (fc)	[376.27 , 375.56]	[376.83 , 376.14]
Focal length uncertainty (fc error)	[1.1336 , 1.1270]	[0.9999 , 0.9922]
Principal point (cc)	[316.0181 , 258.6816]	[324.9401 , 233.3931]
Principal point uncertainty (cc error)	[0.8764 , 0.6881]	[1.1726 , 0.8014]
Distortion coefficients (kc)	[-0.2354 , 0.0651 , -0.0011 , -0.0007 , 0]	[-0.2403 , 0.0771 , -0.0007 , -0.00001 , 0]
Distortion coefficients uncertainty (kc error)	[0.0023 , 0.0025 , 0.0002 , 0.0003 , 0]	[0.0025 , 0.0025 , 0.0002 , 0.0004 , 0]
	Camera 3	**Camera 4**
	[376.43 , 375.65]	[373.33 , 372.68]
	[1.5235 , 1.5481]	[1.3906 , 1.3921]
	[325.9704 , 245.8028]	[324.0573 , 256.7374]
	[1.3855 , 1.0438]	[0.8714 , 0.6902]
	[-0.2324 , 0.0641 , -0.0014 , -0.0005 , 0]	[-0.2422 , 0.0820 , 0.0007 , 0.0006 , 0]
	[0.0042 , 0.0061 , 0.0004 , 0.0003 , 0]	[0.0027 , 0.0033 , 0.0002 , 0.0002 , 0]

which represents a circle with radius:

$$\rho = \frac{1}{2\cdot\delta}[(-\delta+Q_2-Q_1)\cdot(-\delta-Q_2+Q_1)\cdot (-\delta+Q_2+Q_1)\cdot(\delta+Q_2+Q_1)]^{\frac{1}{2}} \tag{12}$$

Through calibration we obtain cameras' orientation relative to the multi-sensor system, in terms of global translations and rotation angles $\hat{\theta}$ with respect to Z axis measured on the X axis. An object's location with respect to the Y axis is estimated by taking into account the extracted object's distance from the camera and the angle $\hat{\theta}$, such as $y = W^* \cdot sin(\theta)$. By substituting the latter into Eq. (11) we obtain a first estimation of object's position in the 3D space, i.e (X,Y,Z) relative to the respective cameras' pair. The final position of the sought target is calculated by the mean value of the available spheres' intersections.

4 Experimental Results

The proposed objects' 3D position estimation method is evaluated through extended tests containing several scenes. The tests were executed on a typical PC with a core2duo 2.2 GHz processor, 2 GB RAM and Windows XP operating system. Furthermore, the cameras used in the experiments are able to capture images up to 1280x960 pixels resolution and are connected to the PC via a firewire port. The lenses utilized are the C418DX manufactured by Pentax with 4.8mm focal length. The data transmission is accomplished by using IEEE 1394b transfer protocol. Moreover, in order to provide more realistic results we tested the proposed method in large scale environments, where an autonomous robotic platform operates [13]. In addition, we used 20 different objects in over 30 experimental setups. Moreover, in Table 1 the focal length, principal point and distortion coefficients along with the respective errors, are presented. In the following section we seek to present both qualitative and quantitative results.

4.1 Database Construction

The database is constructed while the system is off-line. Several objects from different viewpoints and distances to each camera are captured and stored into it. The groundtruth measurements are the distances of the objects as provided by a Bosch DLR165K Laser Distance Measuring Device, which offers $\pm$ 1/16 inch accuracy. The goal of this process is to extract features that are

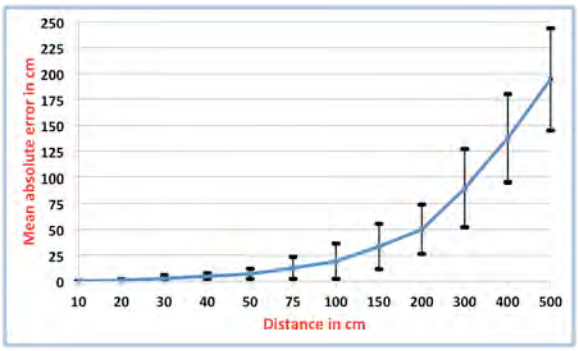
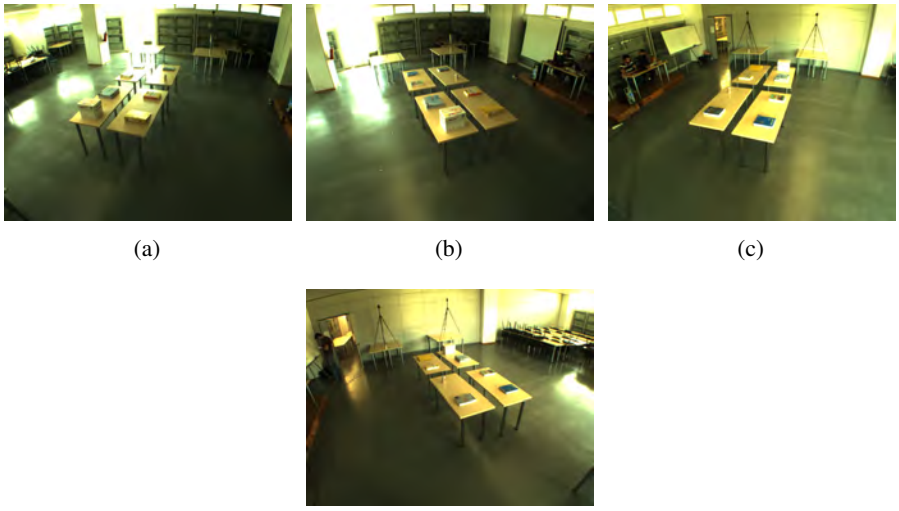


Figure 4: The absolute error (in cm) increases as the object's distance from the camera grows, in instances where we hold only one training image per object.



(a) (b) (c)

(d)

Figure 5: The scene 1 containing eight different objects is captured from 4 different viewpoints corresponding to camera 1 (5(a)), camera 2 (5(b)), camera 3 (5(c)) and camera 4 (5(d)). The objects - targets are positioned on four desks.

scattered over the surface of the objects-targets. The constructed database contains specific feature-distributions over the targets' surface, which tally with the respective objects' distances from the cameras. Each object - target was captured at six different distances from the camera, namely 100 cm, 150 cm, 200cm, 300 cm, 400cm and 500cm. The view point capturing angles where 0°, 90°, 180° and 270°, respectively resulting in 24 distinct instances for each object. Figure 4 illustrates the gradation of the absolute error measured in cm, whilst the x-axis corresponds to the difference between the trained and the current distance. Generally, this error increases parabolically with the distance whilst it remains very low for distances below 100cm.

4.2 3D Position Estimation

In Figure 5, scene 1, which comprises of eight different objects captured by the multi-camera system, is presented. The cameras are mounted onto four tripods and placed at predefined positions

Table 2: The eight objects comprising the scene depicted in Figure 5 and their estimated distance from the cameras (a,b,c and d) along with the respective ground-truth measurement.

		(a)	**Camera 1**		**(b)**	**Camera 2**	
	Object	**groundtruth**	**measured**	**accuracy**	**groundtruth**	**measured**	**accuracy**
desk 1	**klaser**	266	283.9	93.27%	294	267.6	91.02%
	book-a	328	304.2	92.74%	350	322.9	92.26%
desk 2	**paper**	282	303.1	92.52%	240	248.1	96.63%
	asus	348	323.5	92.95%	305	312.5	97.54%
desk 3	**book-c**	499	559.4	87.90%	468	414.2	88.50%
	book-b	434	489.2	87.28%	396	352.9	89.12%
desk 4	**book-d**	474	408.3	86.14%	486	398.9	82.08%
	coke	402	328.7	81.77%	420	514.2	77.57%
		(c)	**Camera 3**		**(d)**	**Camera 4**	
		groundtruth	**measured**	**accuracy**	**groundtruth**	**measured**	**accuracy**
desk 1	**klaser**	487	434.2	89.16%	480	539.6	87.58%
	book-a	422	468.0	89.10%	414	489.2	81.84%
desk 2	**paper**	479	523.1	90.79%	507	462.9	91.30%
	asus	416	382.4	91.92%	451	412.9	91.55%
desk 3	**book-c**	256	274.1	92.93%	321	341.6	93.58%
	book-b	320	292.9	91.53%	376	339.7	90.35%
desk 4	**book-d**	287	306.4	93.24%	278	261.9	94.21%
	coke	351	397.3	86.81%	332	297.0	89.46%

forming, a rectangle with $\delta = 250$ cm and $\kappa = 650$ cm. Moreover, all the sensors are laid at the same vertical height (250 cm) and all the necessary spatial information was obtained during callibration. The targets were laid on four desks as follows: "book-a" and "klaser" on desk 1; "asus" and "paper" on desk 2; "book-b" and "book-c" on desk 3 and "book-d" and "coke" on desk 4, respectively. Then, the online search engine took place and the results are presented in Table 2. Eq. 13 represents the accuracy metric used.

$$accuracy(\%) = \left(1 - \frac{\parallel groundtruth - measured \parallel}{groundtruth}\right) \times 100 \tag{13}$$

The overall accuracy for estimating objects' distances relatively to each camera has been remarkable, giving a mean value of 89.83% with standard deviation (σ) of 0.04. It is apparent that the accuracy of the method is directly related to the actual object's distance from the camera. More specifically, the closer the object to the camera the better the estimation. By observing Table 2 one can verify the aforementioned fact, since best results correspond to objects closer to the sensor (e.g. the distance of "book-a" from the camera is more accurately estimated from camera 1 than from camera 4). Another important aspect constitutes the fact that, both the size and the texture of the sought object play an essential role in the calculation of its spatial information. For example, location assignments for the target "paper", which is the largest object on desk 2, and "asus", which is the most textured object placed on desk 2, are more effective due to the fact that their attributes provide more features to extract. As a result, the greater the number of features the better their distribution over the surface of the object. Since our method is mainly based on the key-points' spread, enhanced distribution leads to more qualitative and quantitative results.

By now we have effectively estimated objects' distance from the cameras, which in turn, delimit several spheres centered at the sensor having a radius equal to the distance already calculated. It is

Table 3: The proposed method aims at estimating the position of "book-a" in the 3D space relative to each camera's frame.

Camera 1		
(X,Y,Z) groundtruth	**(X,Y,Z) measured**	**(X,Y,Z) accuracy**
(95, 295, 146)	(101.47, 316.41, 135.99)	(93.19%, 92.74%, 93.14%)
Camera 2		
(X,Y,Z) groundtruth	**(X,Y,Z) measured**	**(X,Y,Z) accuracy**
(155, 295, 146)	(138.53, 328.51, 129.06)	(89.37%, 88.64%, 88.40%)
Camera 3		
(X,Y,Z) groundtruth	**(X,Y,Z) measured**	**(X,Y,Z) accuracy**
(155, 355, 146)	(174.67, 306.89, 168.48)	(87.31%, 86.45%, 84.60%)
Camera 4		
(X,Y,Z) groundtruth	**(X,Y,Z) measured**	**(X,Y,Z) accuracy**
(95, 355, 146)	(82.47, 284.99, 119.97)	(86.81%, 80.28%, 82.17%)

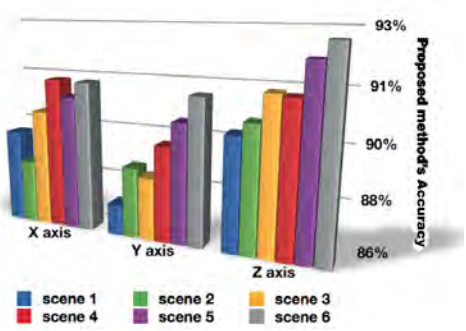
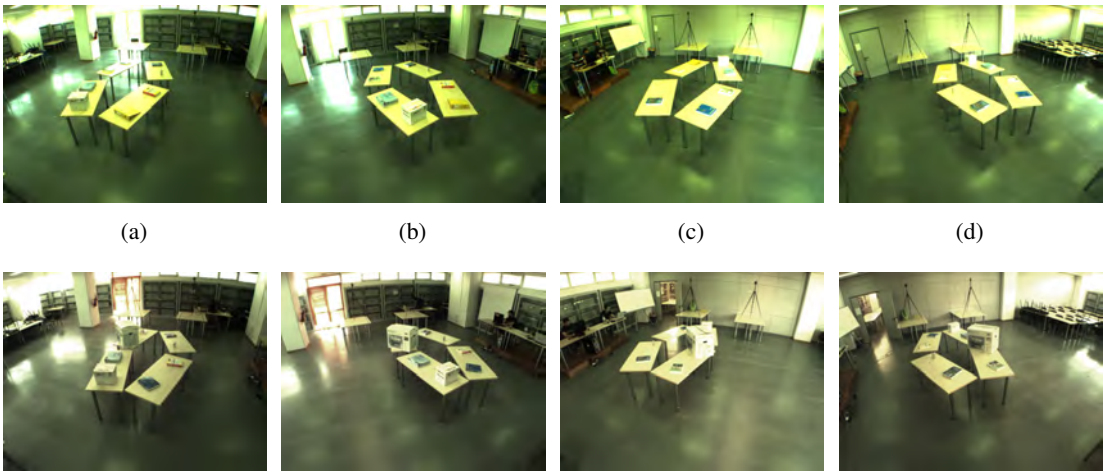


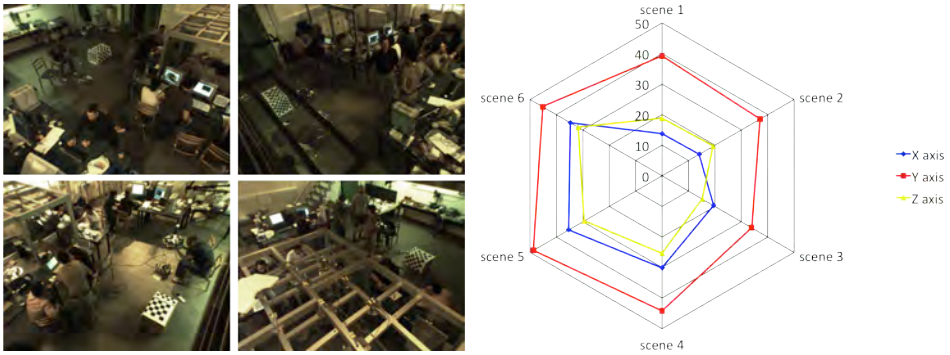
Figure 6: The overall accuracy of the proposed method over the 3D dimensions $(X, Y$ and $Z)$ accumulating all utilized objects and through all the experiments.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: 7(a) - 7(d) The second scene comprises of the same objects scattered at different positions in the 3D space. 7(e) - 7(h) Scene 3 that is almost identical to scene 1 (Figure 5) contains one additional object "hp-box", whilst in different geometrical constrains.



(a) (b)

Figure 8: a) This scene (4) shows the integration environment of the ACROBOTER project. b) The mean estimation error measured in cm in $(X, Y$ and $Z)$ for the first scene of ACROBOTER's environment.

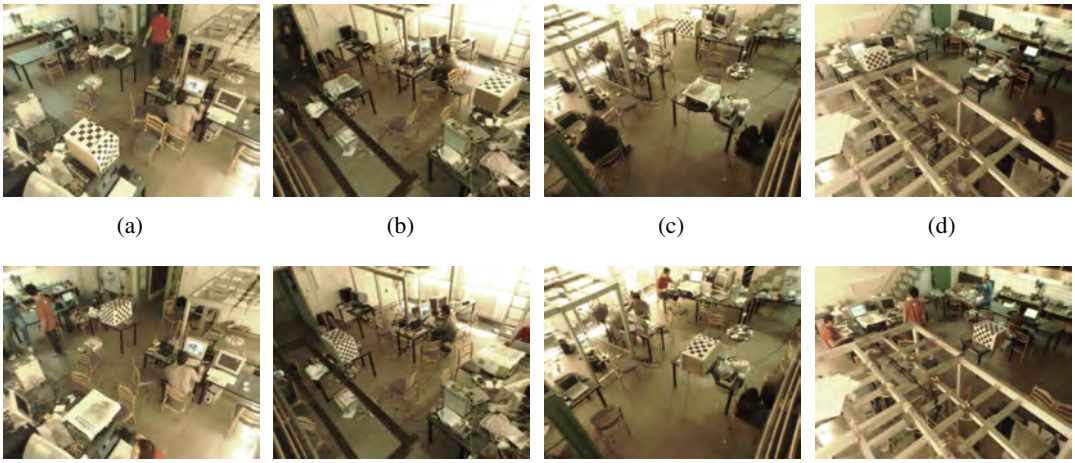
Table 4: The accuracy of the object's distance estimation for the purposes of the ACROBOTER project.

Camera 1			**Camera 2**		
groundtruth	**measured**	**accuracy**	**groundtruth**	**measured**	**accuracy**
635	618.4	97.38%	510	485.6	95.21%
Camera 3			**Camera 4**		
groundtruth	**measured**	**accuracy**	**groundtruth**	**measured**	**accuracy**
350	365.9	95.65%	530	455.7	85.98%

apparent that the aforementioned sphere surface represents the geometrical locus in the 3D working environment, in which the object is positioned. Consequently, after applying the mathematical formulae corresponding to spheres' intersections that are presented in Section 3, we are able to estimate any recognized object's location in the 3D space in terms of absolute *X,Y,Z* positions. The most important issue that arises is that the latter represent absolute measurements relative to each camera's coordinate system. As an example, we present the accuracy of the proposed method in the particular experiment of scene 1 (Figure 5) concerning the object "book-a" in Table 3. Once again, the accuracy of the method remains remarkable, since its mean value is 87.76%, which corresponds to a mean error around 26 cm (maximum 70 cm and minimum 6.47 cm measured at Y and X axis, respectively).

Scene 2 depicted in Figures 7(a) - 7(d) is similar to scene 1, as it contains the same objects, but the latter placed at different locations. Moreover, the almost same objects ("klaser" was substituted by "hp-box") are utilized for the shake of the next scene presented in Figures 7(e) - 7(h). The yield of the proposed method remains high: In Figure 6 we present the mean accuracy on each axis (X, Y, Z) for all the objects used throughout the series ofl experiments.

The work presented in this paper was thoroughly tested during the final integration and review meeting of the ACROBOTER research project [5]. The ACROBOTER project developed a radically new locomotion technology, which can be used in a home and/or in a workplace environment for manipulating small objects autonomously. Scenes 4 (Figure 8(a)), 5 (Figures 9(a) - 9(d)) and 6 (Figures 9(e) to 9(h)) respectively, represent the working space of the ACROBOTER captured by the corresponding cameras. Here, the cameras were placed on the ceiling grid forming a rectangular



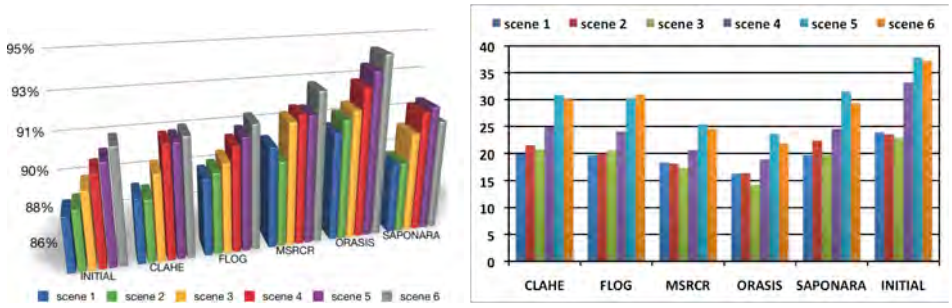
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 9: Scenes 5 (9(a) - 9(d)) and 6 (9(e) - 9(h)) contain the "fingerprint" and "chess box" objects positioned at different places.

grid with $\delta = 440$ cm, $\kappa = 620$ cm and height 320 cm. The pictures were captured during the aforementioned integration meeting with the illumination changing dramatically throughout the day. The objects presented in the following experiments correspond to a rectangular box covered with a chess pattern, used for calibration, and to a less rigid textured target looking like a large fingerprint (shown only in Figure 9). Our method was able to estimate objects' distance from the camera in scene 4. As far as the ACROBOTER's set of experiments are concerned, these are characterized by better system's accuracy compared to scenes 1, 2 and 3. One important issue that should be kept in mind is the fact that camera's 4 field of view represents a fully cluttered environment with a large trussing in front. This results in a slightly decreased accuracy (see Table 4 - camera 4). Nevertheless, despite of this, the overall accuracy of the proposed framework is not seriously affected. Finally, we introduce Figure 8(b) depicting the mean estimation error (in cm). It can be easily observed that the proposed method's error is found at very low levels, with the maximum error laying on the Y axis. The latter should be expected since the transpose on the Y axis is greater than the respective transposes on X and Z axes, due to the fact that it incorporates κ, which was significantly bigger than δ in the particular working environment.

All the aforementioned scenes suffer from several illumination discrepancies due to both the working environment's texture quality and the attributes of the lighting source. More specifically, almost all the captured images contain instances where reflectance phenomena produce over and under exposed regions. In order to increase the proposed method's robustness against illumination changes we appraised the image enhancement algorithms presented in Section 2. In our framework, these methods pre-process several input images to produce an enhanced one. Through the following experimental set, scenes 1 to 6 along with objects' training photos, are considered as input images, whilst the overall goal is to enhance our system's efficiency. In Figure 10(a), we present an analytical centralized chart that includes the actual influence of the aforementioned algorithms in the proposed 3D location estimation scheme, which is referred as INITIAL. In almost all instances, the pre-processing step does enhance our system's efficiency and by-passes difficulties in challenging illumination circumstances. ORASIS constitutes the most effective solution since its adoption as a pre-processing step results in proposed method's accuracy about 92%. A slightly reduced accuracy, compared to ORASIS, is provided by the MSRCR algorithm. The overall affect of each algorithm



(a) (b)

Figure 10: a) The accuracy of the 3D position estimation method increases using CLAHE, FLOG, MSRCR, ORASIS and the SAPONARA methods as a pre-processing step. b) Absolute error's (in cm) variation after adopting several image enhancement algorithms. INITIAL corresponds to the proposed 3D position estimation method without adopting any pre-processing technique for image enhancement.

over the proposed method can be apprehended by calculating the mean estimation error. The positive effect of the ORASIS algorithm is verified by the results depicted in Figure 10(b). Generally, all the image enhancement methods have a positive affect on our approach, since the absolute error is drastically reduced. The overall error of the proposed 3D position estimation method measured in cm is generally acceptable. The contribution of the ORASIS as a pre-processing step, enhances our algorithm's robustness against large local and global contrast modifications.

5 Conclusions

In this paper we have presented a novel method for estimating objects' location in the 3D space by exploiting spatial information derived through recognition and the utilization of a sophisticated multi-camera system. This approach excels in simplicity and computational cost, whilst its database can be easily modified for the needs of multiple-object recognition and location assignment. Experimental results provide credit to our claim that once the object is recognized its distance from each camera can be easily estimated through the calculation of the features' distribution over the sought target's surface. Moreover, the algorithm requires an object to be recognized by more than two different viewpoints in order to effectively estimate its location relatively to the multi-camera array. Occlusions do not affect the efficiency of the method, unless the visibility of an object is limited to one sensor only. The multi-camera system used is able to cover 97% of the working volume of the testing room. The proposed method was successfully tested in a realistic scenario. In conclusion, larger and more textured objects are more likely to be successfully recognized and spatially registered in the 3D space. On the other hand, the proposed method's accuracy drops in cases of small objects or even textureless targets, due to the fact that there is a limited features' spread over the objects' surface. Furthermore, due to the fact that illumination conditions affect directly the efficacy of the system, we tested several image-enhancement methods as a pre-processing step to the proposed 3D position estimation method. The ORASIS method, was favored due to the fact that it incorporates both global and local contrast modification, which lead to enhanced image quality. In addition, by adopting this method, we noted a remarkable decrease of the proposed method's estimation error.

References

[1] http://www.ptgrey.com/products/grasshopper/index.asp.

[2] http://www.truview.com/.

[3] http://sites.google.com/site/vonikakis/software.

[4] http://www.vision.caltech.edu/bouguetj/calib_doc/.

[5] http://www.acroboter-project.org/.

[6] A. Alahi, P. Vandergheynst, M. Bierlaire, and M. Kunt. Cascade of descriptors to detect and track objects across any network of cameras. *Computer Vision and Image Understanding*, 114(6):624 – 640, 2010.

[7] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1685 – 1699, 2009.

[8] A. Andreopoulos and J.K. Tsotsos. A theory of active object localization. *International Conference on Computer Vision*, 2009.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008.

[10] B. Bennett, D.R. Magee, A.G. Cohn, and D.C. Hogg. Enhanced tracking and recognition of moving objects by reasoning about spatio-temporal continuity. *Image and Vision Computing*, 26(1):67 – 81, 2008.

[11] D. Damen and D. Hogg. Detecting carried objects in short video sequences. *European Conference on Computer Vision*, pages 154 – 167, 2008.

[12] S. Ekvall, D. Kragic, and F. Hoffmann. Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. *Image and Vision Computing*, 23(11):943 – 955, 2005.

[13] S. Gabor, A. Toth, G. Nikoleris, A. Gasteratos, N. Kyriakoulis, D. Chrysostomou, and R. Kouskouridas. Acroboter: a ceiling based crawling, hoisting and swinging service robot platform. *Proceedings of British Human Computer Interaction Workshop*, 2009.

[14] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] G. Hüsler and G. Ritter. Feature-based object recognition and localization in 3d-space, using a single video image,. *Computer Vision and Image Understanding*, 73(1):64 – 81, 1999.

[16] D.J. Jobson, Z. Rahman, and G.A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6(7):965 – 976, 1997.

[17] R. Kouskouridas, E. Badekas, and A. Gasteratos. Simultaneous visual object recognition and position estimation using SIFT. *Intranational Conference on Intelligent Robotics and Applications*, pages 866 – 875, 2009.

[18] R. Kouskouridas, A. Gasteratos, and E. Badekas. Evaluation of two-parts algorithms for objects' depth estimation. *IET Computer Vision*, page Accepted on March 2011, 2009.

[19] M. LaCascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3 d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322 – 336, 2000.

[20] H.S. Le and H. Li. Fused logarithmic transform for contrast enhancement. *Electronics Letters*, 44:19 – 20, 2008.

[21] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, 2004.

[22] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. *European Conference on Computer Vision*, pages 434 – 445, 2008.

[23] A.M. Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *The Journal of VLSI Signal Processing*, 38(1):35 – 44, 2004.

[24] R. Sandhu, S. Dambreville, A. Yezzi, and S. Tannenbaum. Non-rigid 2d-3d pose estimation and 2d image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 786 – 793, 2009.

[25] S. Saponara, L. Fanucci, S. Marsi, and G. Ramponi. Algorithmic and architectural design for real-time and power-efficient retinex image/video processing. *Journal of Real-Time Image Processing*, 1(4):267 – 283, 2007.

[26] K Shubina and J.K. Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535 – 547, 2010.

[27] C. Sminchisescu, A. Kanaujia, and DN Metaxas. Bm^3e: Discriminative density propagation for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):2030 – 2044, 2007.

[28] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Van Gool. Shape-from-recognition: Recognition enables meta-data transfer. *Computer Vision and Image Understanding*, 113(12):1222 – 1234, 2009.

[29] V. Vaish, M. Levoy, R. Szeliski, CL Zitnick, and S.B. Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[30] V. Vonikakis, I. Andreadis, and A. Gasteratos. Fast centre-surround contrast modification. *IET Image Processing*, 2(1):19 – 34, 2008.

[31] V. Vonikakis, R. Kouskouridas, and A. Gasteratos. A comparison framework for the evaluation of illumination compensation algorithms. *Pattern Recognition,*, Submitted for acceptance on October 2010, 2010.

[32] D. Walther, U. Rutishauser, C. Koch, and P. Perona. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding*, 100(1-2):41 – 63, 2005.

[33] L.L. Wang and W.H. Tsai. Camera calibration by vanishing lines for 3-d computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370 – 376, 1991.

[34] S. Yasushi, I. Yutaka, and T. Fumiaki. Robot-vision architecture for real-time 6-dof object localization. *Computer Vision and Image Understanding*, 105(3):218 – 230, 2007.

[35] J. Zhang, L. McMillan, J. Yu, and U.N.C.C. Hill. Robust tracking and stereo matching under variable illumination. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[36] M. Zwicker, A. Vetro, S. Yea, W. Matusik, H. Pfister, and F. Durand. Resampling, antialiasing, and compression in multiview 3-d displays. *IEEE Signal Processing Magazine*, 24(6):88 – 96, 2007.

Authors

Rigas Kouskouridas received the Diploma degree from the Democritus University of Thrace, Xanthi, Greece, in 2006. He is currently working towards the Ph.D. degree with the Group of Robotics and Cognitive Systems, Laboratory of Robotics and Automation, Department of Production and Management Engineering, Democritus University of Thrace. His areas of interest include pattern recognition, machine learning, multi camera vision systems and robotics. He is involved in several international (European) and national (Greek) research projects in the field of machine vision systems. Mr. Kouskouridas is a member of the IEEE, euCognition II, the Technical Chamber of Greece (TEE), and the National Union of Production and Management Engineers.

Antonios Gasteratos is an Assistant Professor of Mechatronics and Artificial Vision at the DPME. He teaches the courses of Robotics, Automatic Control Systems, Measurements Technology and Electronics. He holds a Diploma and a Ph.D. from the Department of Electrical and Computer Engineering, DUTH, Greece, 1994 and 1999, respectively. During 1999-2000 he was a Post-Doc Fellow at the Laboratory of Integrated Advanced Robotics (LIRA-Lab), DIST, University of Genoa, Italy. He has served as a reviewer to numerous of scientific journals and international conferences. He is the Greek Associate High Level Group (HLG) Delegate at EUREKA initiative. His research interests are mainly in mechatronics and in robot vision. He has published one textbook, 3 book chapters and more than 90 scientific papers. He is a member of the IEEE, IAPR, ECCAI, EURASIP and the Technical Chamber of Greece (TEE). Dr. Gasteratos is a member of EURON, euCognition and I*PROMS European networks. He organized the International Conference on Computer Vision Systems (ICVS 2008).