MONOCULAR OBJECT LOCALIZATION BY SUPERQUADRICS CURVATURE REPROJECTION AND MATCHING

Enrico Zappia, Ilya Afanasyev, Nicolo' Biasi, Mattia Tavernini, Alberto Fornaser, Antonio Selmo and Mariolino De Cecco
Department of Mechanical and Structural Engineering (DIMS), University of Trento, via Mesiano, 77, Trento, Italy
enrico.zappia@gmail.com, {ilya.afanasyev, mariolino.dececco}@unitn.it

Abstract

This paper presents a new method for 3D object localization from a single image. It is known that single camera provide 2D image data, annihilating valuable 3D information about object and its localization in space. The main new idea is to match 2D image gradient to the reprojection of 3D curvature to retrieve objects position relative to the camera. The object parameters are a-priori known and modelled by SuperQuadrics (SQ) that enable the calculation of the analytical form of curvature. The image processing stage includes object detection and segmentation by the Histogram of Oriented Gradients (HOG) algorithm. The method proposed uses the dependencies between SQ curvature and image gradient also considering the illumination model and object contour embedded in a proper cost function. To manage local minima we propose the use of particle swarm optimization (PSO).

Keywords: *SuperQuadrics, single camera, 3D object localization, Histogram of Oriented Gradients (HOG), curvature matching, and particle swarm optimization (PSO).*

1. INTRODUCTION

Object localization is an important task of computer vision and robotics with many applications in the fields of autonomous-guided vehicles, robot picking and manipulation, augmented reality, non-contact measurement, etc. In recent years, thanks to the increasing interest in these fields, some different approaches have been proposed. This work presents a new and effective method of object localization by matching image properties acquired by a single CCD to object curvature in 3D whose model is obtained in analytical form by SuperQuadrics. There are some papers related to the pose estimation with SQ [1, 3, 14], but all of them match 3D range data (point cloud) to SQs models. On the one hand this approach becomes very efficient and robust to outliers [22], but on the other hand it requires 3D depth cameras, lasers rangefinders or multicamera setup. Other related papers focus on the pose estimation from a single image. Such papers can be divided into two main groups according to their main model descriptors[8]: the first uses the model edges (wireframes), the second spatial localized model features (regions). The main idea of the edge-based methods like [5, 9, 15] is to reproject a model contour on the gradient image. These methods are very time efficient and can be used also for object tracking, but the object's profile is often strongly varied along the edges due to e.g. clutter, shading, and texture. For these reasons, the edge detection is usually performed on the maximal image gradient. The region-based methods such as [24, 21] rely on the homogeneity of spatially localized features (e.g. RGB values, curvatures, etc.). The assumption is that the features of all pixels of a region are distributed with statistical independence according to the same probability density function. Often this assumption leads to incorrect results if e.g. the distributions of RGB values of foreground and background depend on the object location within the image. Fusing edge-based and region-based approaches gives a more effective and robust way for objects segmentation and matching [10]. Other works have been proposed to retrieve 3D information from set of images like [4]. In this case object identification and 3D parameters are obtained from SIFT features. The model is stored with a sample image that is correlated to the real object. This method gives good results with variegated texture.

SuperQuadrics are an extension of basic quadric surfaces, which were introduced in computer vision by Alan Barr [2]. These mathematical functions allow the represention of a pretty high number of elementary solids, e.g. sphere, box, cylinder, toroids. Advantages of this formulation are compactness and its closed-form mathematical expression. Furthermore, SQ can be roughly described as deformation of a sphere so they are continuous surfaces, even through edges.

Histogram oriented gradient (HOG) is able to retrieve objects in the picture. It is a stochastic algorithm that uses the distribution of intensity gradients for object detection. This method has been first introduced to solve the problem of pedestrian identification in static images [17, 18]. The algorithm focuses on finding the robust features descriptor of human model, maintaining invariance to a wide variety of articulated pose minimizing the influence of background and illumination. HOG processes a sample image comparing it with a inner object model. It is obtained through a training session with several different samples of the object acquired from different points of view. All data are collected for training support vector machine that compares the sample image with its model. The output is the positive detection status and a rough localization of the object in the image plane. We propose a novel method of object detection and localization that, exploiting SQ analytical formulation, implements curvature reprojection keeping into account contours, edges and region properties at once. Detection is obtained using state of the art HOG algorithm. With this approach we rely on a wider set of informations than just edges or key points as traditional model based approaches.

2. ALGORITHM OUTLINE

The algorithm (1) starts with the HOG object detection. The result is a detection window on the acquired image containing the object. This defines a lower and an upper bounds for the optimizer research domain. Then a first pose guess is generated by a transformation matrix applied to the a-priori known SQ-model. From analytical SQ formulation the curvature, the normals and the lighting model are computed and reprojected onto the image to evaluate the level of matching with image features (gradients) by means of a properly defined cost function. Until the overall cost exceeds a certain threshold, the optimization process continues, when the threshold is reached, object localization is returned.

3. OBJECT DETECTION USING HISTOGRAMS OF ORIENTED GRADIENTS

This work focuses on the estimation of the object pose and location with respect to the camera. The first stage of image processing is the HOG object detection and identification of the Region Of Interest (the Detection Window). Then the resulting window is used to set the initial guess on object location for the optimizer initialization). HOG involves two main phases: features extraction and learning.

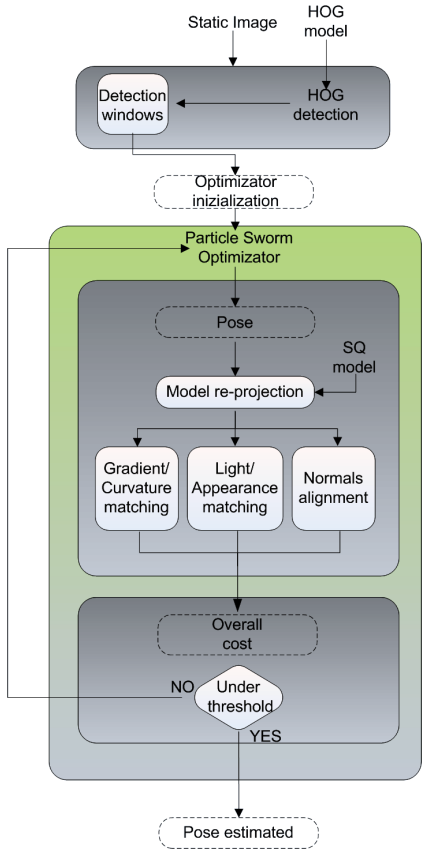


Figure 1: The flowchart of the object localization algorithm from static images.

3.1 Features extraction

The object images are normalized (e.g. by gamma normalization) and then their gradients magnitude and orientation evaluated. The detection window is divided into sub cells. The histogram of oriented gradients, weighted according to the magnitude of the gradients intensity, is then computed. Data are collected into Blocks with an additional normalization to provide better illumination invariance. To complete the dataset of positive images, also negative ones are processed (i.e. images without the target). After that HOG data are collected for all the detection windows and then feature vectors (both negative and positive images) are combined together to be processed by the support vector machine (SVM) for the Learning phase.

3.2 Learning phase

Images from the dataset created in the preview phase are encoded as spatial feature vectors. This set is processed into a binary classifier for object / non-object class identification. At this stage the detection/recognition is not robust as a high number of false positives can be obtained from the first experimental dataset, i.e. the non-object class in not properly acquired/described. To reduce the false positive detection the second sample dataset with only negative images is prepared and processed. In this way all positive results in this sequence are false positive, so they can be re-introduced in the classifier as the hard negative example to perform better non-object class description. A new classification is finally obtained using the two classes. False positives are now reduced by an order of magnitude.

3.3 HOG implementation

The HOG implementation is the same as described in [19]. Some of the improvements comprised the UoC-TTI LSVM-MDPM entry in the PASCAL VOC 2009 comp3 challenge [6]. The dataset chosen concerned simple objects like a box (219 x 120 x 230 mm) and a cylinder (190 x 90 mm). Training is performed with different images of the same object (80 positives image and 30 negatives for each model). As shown in figure 2 the final HOG result is a bounding box around the detected object. This information provides the segmentation of the region of interest from the background to initialize the optimizer.

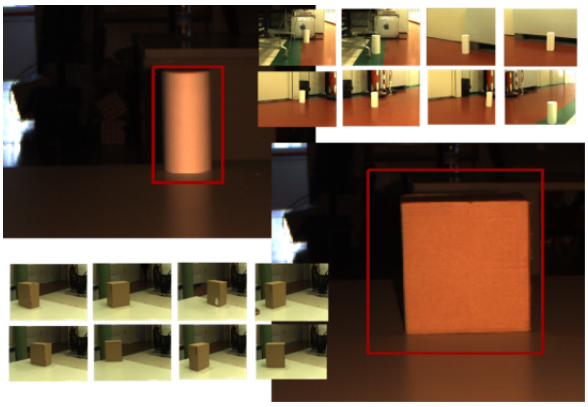


Figure 2: The set of some training images (for a cylinder and a box) and the final results of HOG detection (the red rectangle surrounding the objects for test images).

This kind of segmentation provides information only into 2D domain, whereas pose and/or depth information are not estimated. Anyway this stage allows concentrating the optimization algorithm attention only on a limited zone of the image thus speeding up the subsequent steps.

4. SUPERQUADRICS FORMULATIONS

SQ surfaces can be obtained as spherical product of two parametric curves.
Given two parametric curves:

$$h(\omega) = \begin{bmatrix} h_1(\omega) \\ h_2(\omega) \end{bmatrix} \qquad \pi \le \omega \le \pi \tag{1}$$

$$m(\eta) = \begin{bmatrix} m_1(\eta) \\ m_2(\eta) \end{bmatrix} \qquad \frac{\pi}{2} \le \eta \le \frac{\pi}{2} \tag{2}$$

where ω and η are spherical coordinates respectively for the horizontal and vertical curves. Spherical product is defined as:

$$m(\eta) \otimes h(\omega) = \begin{bmatrix} m_1(\eta) \cdot h_1(\omega) \\ m_1(\eta) \cdot h_2(\omega) \\ m_2(\eta) \end{bmatrix} \tag{3}$$

For SQ-representation the known explicit formula [11] is used:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 \cos^{\varepsilon_1}\eta \cos^{\varepsilon_2}\omega \\ a_2 \cos^{\varepsilon_1}\eta \sin^{\varepsilon_2}\omega \\ a_3 \sin^{\varepsilon_1}\eta \end{bmatrix} \tag{4}$$

where
- x, y, z - SQ coordinate system
- a_1, a_2, a_3 - scale parameters of the object;
- $\varepsilon_1, \varepsilon_2$ - object shape parameters;
- ω, η - spherical coordinates;

The implicit SQ formulation is anyway more suitable for mathematical modeling:

$$\mathbf{F}(x,y,z) = \left(\left(\frac{x}{a_1}\right)^{2/\varepsilon_2} + \left(\frac{y}{a_2}\right)^{2/\varepsilon_2} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{2/\varepsilon_1} \tag{5}$$

4.1 Curvature estimation

As shown in (3), if the generative curves of the spherical product are continuous, the surface created is also continuous. This means that it is possible to use differential geometry to retrieve curvature information about the parametric surface. Definition (5) describes the different surfaces, so that, according to the scale and shape parameters, every point on the shape is analytically known. The leading idea of this work is to use the relation between the curvature and the object change of appearance. For this reason the accurate evaluation of the curvature is crucial. In this work we followed the approach of Ron Goldman's work on the calculation of surfaces curvature [7], especially focusing on mean curvature and normal directions.

4.1.1 Identification of the Normals

Normal can be easily calculated starting from the evaluation of the gradient of the parametric surfaces. Since the gradient of $\mathbf{F}(x, y, z)$ is perpendicular to the level curves $\mathbf{F}(x, y, z) = \text{const}$, the gradient ∇F is parallel to the normal of $\mathbf{F}(x, y, z) = 0$. Therefore we have the following formulas:

$$\mathbf{N}(x, y, z) = \frac{\nabla \mathbf{F}(x, y, z)}{|\nabla \mathbf{F}(x, y, z)|} \tag{6}$$

where $\mathbf{N}$ is a set of unitary vectors in normal direction.

4.1.2 Mean Curvature

Usually mean curvature is taken as the divergence of unit vector:

$$\mathbf{K_M}(x, y, z) = -\nabla \cdot \mathbf{N}(x, y, z) \tag{7}$$

This formulation is very compact, but it may be computationally hard to handle so therefor formulation is suggested:

$$\mathbf{K_M} = \frac{\nabla F \cdot H(F) \cdot \nabla F^T - |\nabla F|^2 \text{Trace}(H)}{2|\nabla F|^3} \tag{8}$$

4.2 SuperQuadrics Representation

The representation of SuperQuadrics is obtained from its explicit formula 4. According with the scaling and shape parameters it is possible to obtain the coordinates of each point by varying the spherical coordinates ω and η. Unfortunately for an equally spaced sampling of these coordinates does not correspond and equally spaced sampling of x, y, z. To take this problem we followed the approach described in [16]. The idea is to model the surfaces like *Superellipsoids* to obtain a linear arc-length parameterization in order to provide a regular sampling along the surface.
The following formulation regards to equally spaced samples along the transversal arc η:

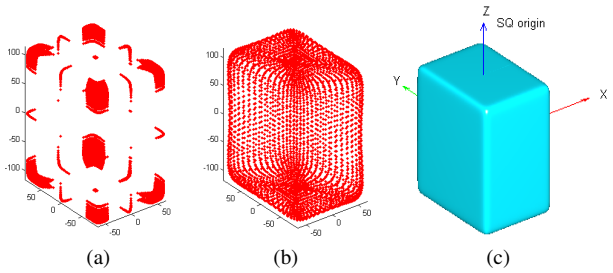
$$\begin{cases} \mathbf{x_s} = x\,(1 + \frac{k_1}{c^2}\,z^2)\,(1 + \frac{k_2}{b_T^2}\,y^2) \\ \mathbf{y_s} = y\,\frac{b\,a_T}{a\,b_T}\,(1 + \frac{k_1}{c^2}\,z^2)\,(1 + \frac{k_2}{a_T^2}\,x^2) \\ \mathbf{z_s} = z\,(1 + \frac{k_1}{a^2}\,a_T^2) \end{cases} \tag{9}$$

where:

$$\begin{cases} \theta_T &= \arcsin(\frac{z}{c}) \\ a_T &= a \cdot \cos(\theta_T) \\ b_T &= b \cdot \sin(\theta_T) \end{cases} \tag{10}$$

With this representation an equal distributed surface is available.

Referring to figure 3 the number of points for the first two representations is the same, but their distributions are much more homogeneous.



(a) (b) (c)

Figure 3: The Superquadric representations: (a) - mapping from the points calculated by explicit SQ form; (b) - mapping from a transformed ellipsoid with algebraic manipulations (9); (c) - the surface mapping.

5. MATCHING ALGORITHM

The image is elaborated by the object detection algorithm (par. 3.3) which detects the object on the image plane defining a confidence region by means of bounding box. This information is the starting point to retrieve the object position in 3D. By means of the bounding box dimensions is in fact possible to roughly estimate the distance.
Position and orientation of the object in 3D space relative to the camera is parametrized with a homogeneous transformation matrix containing the rotation matrix $\mathbf{R}$ and the displacement from the origin $\mathbf{T}$.

$$\begin{bmatrix} \mathrm{P}_{3\times1}^{W} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{3\times1}^{SQ} \\ 1 \end{bmatrix} \tag{11}$$

Using a pinhole camera model with lens distortion compensation [12], it is possible to re-map the model from 3D space to image coordinates. In other words, starting from the 3D points with their attributes (e.g. curvature, normals, and light appearance) it is possible to match with the corresponding candidate pixel that is a function of the coordinates transformation matrix. The goal is to find the homogeneous transformation that best matches the above attributes.

5.1 Curvature matching

It is well known that strong variations on images gradients are located along edges and corners. Methods that rely only on high image gradients along the edges while neglecting the small variations inside the surfaces, become instable if borders are partially occluded. As already said the SQ formulation allows to obtain a continuous surface, so that edges and corners are modelled as local high curvature. According to the edge based methods this high curvature can be related with high gradient intensity, on the other side also low curvature can be connected with low gradient magnitude regions. With our method is possible to use both informations. From equation (8) it is calculated the mean curvature for each sampled point (9). In the image domain it is possible to evaluate the gradient. The distance between gradient and curvature images is evaluated as vector norm of the matrix difference. The lower the distance, the better the matching.

$$\mathbf{E_f} = \frac{\sum_{i=1}^{n} ||G^N(i) - K^N(i)||^2}{n} + ||\mathbf{G}^N - \mathbf{K}^N||^\infty \tag{12}$$

G^N and K^N are the normalized values for the mean curvature and the gradient. The norms to infinity count the maximum displacement between curvature and gradient.

5.2 Light appearance matching

Directions of normals are known for each point from equation (6). If it is also supposed known the position of the light source that makes the object illumination appearance available. The known light position provides the information about bright and dark sides. We used the light model of Phong [20]. It provides a spot light source and includes a combination of diffuse and specular reflection.

$$\begin{aligned} I_p &= k_a i_a + (k_d(\mathbf{L}\cdot\mathbf{N})i_d + k_s(\mathbf{R}\cdot\mathbf{V})^n i_s) \\ \mathbf{R} &= 2(\mathbf{L}\cdot\mathbf{N})\mathbf{N} - \mathbf{L} \end{aligned} \tag{13}$$

k_a, k_d, k_s are the ambiance, diffuse and specular constants; i_a, i_d and i_s are the respective light intensities. $\mathbf{L}$ is the light direction vector, $\mathbf{R}$ is the specular reflection vector, $\mathbf{V}$ is the viewer direction, and $\mathbf{N}$ is the surface normals. With this information is possible to compare the expected illumination appearance with the pixel intensities. This time the key idea is to compute the convolution directly among the images. Illumination model (13) is built so as illumination values $\mathbf{I_p}$ span from 0 to 1, same as for the gray levels $\mathbf{L_g}$ of the normalized image. To compute the convolution the mean value is subtracted in order to have a zero-centred distribution. The value for the *normalized* convolution is expressed by the formula:

$$\mathbf{C_f} = \sum_{i=1}^{n} \frac{I_p^N(i)\cdot L_g^N(i)}{||I_p^N||^2 \cdot ||L_g^N||^2}; \tag{14}$$

The higher the convolution the better the matching.

5.3 Normal and gradient alignment

Another method to improve the matching between the static image and the re-projected model is to check the alignment of normals onto image plane with the gradient pixels orientation.
It is well known that along edges the gradient magnitude is higher and its orientation is orthogonal to the border. Normal directions are 3D data; anyway if we project, as in the case of the sampled 3D SQ model, we obtain normals on the image plane. Normals will have the same orientation as the image's gradient.
With this assumption the matching is computed by the angle between silhouette normal $\mathbf{V_n}$ vectors on the image and the gradients orientation $\mathbf{V_g}$. If the scalar product is considered, the cosine of $\theta_{\mathbf{ng}}$ is close to 1:

$$\mathbf{V_f} = \frac{1}{n}\sum_{i=1}^{n} \mathtt{If}(|\cos(\theta_{\mathbf{ng}}(i))| > 0.9, 1) \tag{15}$$

Quality of matching is estimated counting the number of approximately parallel vectors (15).

6. OPTIMIZATION

All the contributions (12 - 14 - 15) are mixed into cost function (18). Practically we recognized that the problem is multi-modal so square regression algorithm and other gradient based techniques are not well suited as far as their result is dependent on the initial conditions that easily leads to missing the global minimum. That is why we implemented PSO (Particle Swarm Algorithm) to randomize the starting point [13].
The algorithm starts form a set of agents, call particles, that move along the trust region. This region is initialized by the centroid of HOG's detection window $\mathbf{X_n}$ and its area A. From this data we can infer a guess position $\mathbf{X_w}$ for the object, with the formula:

$$\mathbf{X_w} = \frac{k}{A}\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \tag{16}$$

where k is a fixed proportional factor. The reaching of minimum is leaded by force $\mathbf{F}_k^{(s)}$ that can be divided into two contributions. The first is relative to *cognitive behavior* $\mathbf{F}_k^{(s)}(\mathbf{p}_k^{(s)})$ focused on the single particle experience (personal best $\mathbf{p}^{(s)}$). The second is the *social behavior*, $\mathbf{F}_k^{(s)}(\mathbf{g_k})$ focused on the swarm attitude (global best $\mathbf{g_k}$). The update of particle's position is obtained through a velocity with the following expression:

$$\mathbf{v}_{k+1}^{(s)} = \omega\cdot\mathbf{v}_k^{(s)} + \underbrace{C_1\, r_1(\mathbf{p}_k^{(s)} - \mathbf{x}_k^{(s)})}_{\mathbf{F}_k^{(s)}(\mathbf{p}_k^{(s)})\ \text{Cognitive Term}} + \underbrace{C_2\, r_2(\mathbf{g}_k - \mathbf{x}_k^{(s)})}_{\mathbf{F}_k^{(s)}(\mathbf{g}_k)\ \text{Social Term}} \tag{17}$$

Where C_1, C_2 are the acceleration coefficients, ω is the inertial weight, and r_1, r_2 are random variables; k is the iteration step and (s) is particle's index.
The goal is to achieve the transformation matrix $\mathbf{M}^\star$. This represents the transformation from camera to object frames. Cost contributions depend on the transformation of the SQ model onto the image plane. Minimizing the cost function $\mathbf{Fc}$ we find iteratively the matrix $\mathbf{M}$ with the best matching.

$$M^\star = \min_M \Longrightarrow \frac{\mathbf{E_f}(M)}{\mathbf{C_f}(M)\cdot\mathbf{V_f}(M)} \tag{18}$$

The localization of the object is supposed on a plane known with a non negligible level of accuracy. The camera position relative to the plane is known by the calibration process by placing one of the reference target acquisitions on the plane. Localization is performed in xy direction and orientation around z-axis. Because of plane's uncertainty along z the search is allowed also along that direction but for a limited displacement. The parameters that the optimizer tries to retrieve are displayed in table 1, with the trust region upper and lower bound settings.
As already stated, multi-modality introduces local minima issues. The use of normals alignment (15) and light (14) together with curvature decrease the failure rate. Anyway there are still situations in which the optimizer is "trapped" into a local minima.

θ_z	X_w	Y_w	Z_w
$-\pi/2$	$X_w - \lvert X_w\rvert\cdot 0.3$	$Y_w - \lvert Y_w\rvert\cdot 0.3$	$Z_w - 0.02$
$\pi/2$	$X_w + \lvert X_w\rvert\cdot 0.3$	$Y_w + \lvert Y_w\rvert\cdot 0.3$	$Z_w + 0.02$

Table 1: Optimization parameters initialization (first row is lower bound, second row is upper bound). Angles are expressed in radiants and translations - in meters.

7. EXPERIMENTAL RESULTS

In this paragraph some results are shown with the aim to prove detection correctness and to provide a quantitative estimation of the localization accuracy in different conditions. We also propose a comparison between our method and state of the art algorithms that use RGB-d camera. For this we chose Microsoft Kinect sensor for its high quality to cost ratio.

The setup was assembled with an RGB camera (1280 x 960) and a Kinect sensor (ref. to figure 4). The camera was equipped with a known position spot light. Both sensors were calibrated in reference to the chessboard (origin on the top left corner). The localization algorithm was limited to the plane estimated with the calibration process.
A properly designed grid has been printed and used as a reference for calibration. This grid is arranged in 25 cells organized with a fixed displacement and variable attitude. For each cell we place the objects with a relative position accuracy of about 1 mm. We also know the absolute position with respect to the camera and the Kinect thanks to the calibration. The localization's range is spanned from 2 to 3 m.

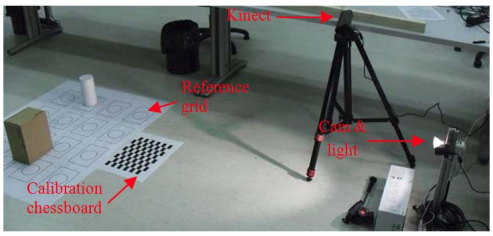


Figure 4: Experiment setup layout, data are acquired both from the camera and Kinect. Chessboard is used to estimate camera and Kinect positions in order to provide the reference frame for the localization.

Kinect uses IR structured light to provide full 3D data information. From each snapshot it retrieves a corresponding points cloud with colour information. The idea is to use the cloud of 3D points to fit the SQ model.
A very common approach is to use robust fitting algorithm (e.g., RANSAC), anyway the results of these depend upon the number of inliers and outliers allowed and its related parameters, this particularly when the number of outliers is large. To overcome this is preferable to cluster the complete cloud of points to segment only those corresponding to the object and then use a regression algorithm to estimate a more accurate fitting over the selected points.
First the initial cloud of points is processed with RANSAC to

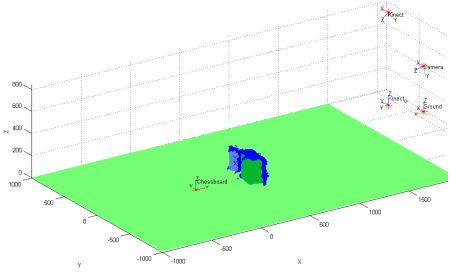


Figure 5: Example of SuperQuadrics fitting with depth data.

delete those lying on the floor plane (green in the picture). Then clustering of the 2 objects in the view is obtained with k-means. For each cluster is performed a fitting with Levenberg-Marquardt. Final results are shown in figure 5. Figure 6 presents some localization results of objects in general positions and also with occlusions. Figure 6 also presents different conditions, with known light source and without. In the same figure are also reported some local minima solutions. In those cases is possible to note an overlap of the figure but the orientation is completely wrong despite good normals alignment and curvature matching. Figure 7 shows the estimated uncertainty ellipses (green - for Kinect, blue - for our method). Each experimental sample is computed with respect to the reference positions of each SQ centre on the reference grid. Y-direction is aligned with the camera/Kinect axes. The ellipses are estimated using the k factor defined in [23] with a confidence level of 95%. To quantify uncertainty we report the eigenvalues along principal directions for each ellipse (values in mm) $\lambda_1^B = 18.85$ and $\lambda_2^B = 27.23$ (box) and $\lambda_1^C = 11.35$ and $\lambda_2^C = 13.55$ (cylinder) for our localization respect to $\lambda_1^B = 11.65$ and $\lambda_2^B = 40.56$ (box) and $\lambda_1^C = 13.83$ and $\lambda_2^C = 47.05$ (cylinder) . The figure 8 shows the dependence of the optimized cost function values on the percentage (relative) errors. Unfortunately we found no clear correlation between errors and cost function. This we believe is an important point to further analyze in order to find an estimate of the actual uncertainty directly from the final (optimized) value of the cost function.

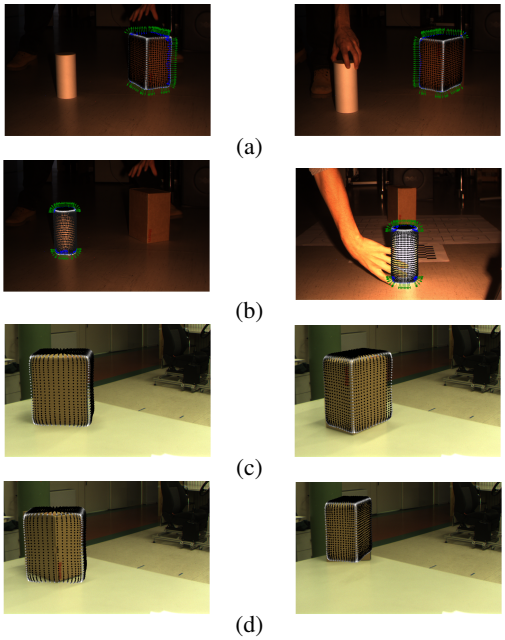


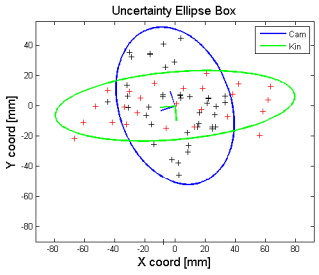
Figure 6: In (a) are presented some localization results in general locations; in (b) the same is proposed for cylinders. In the second picture a partially occluded scene is proposed while the localization remains correct. In pictures (c) and (d) localization is estimated without light informations. Obtained results are quite good in (c), but in (d) the localization faces local minima problems which stuck the optimization.

8. CONCLUSION

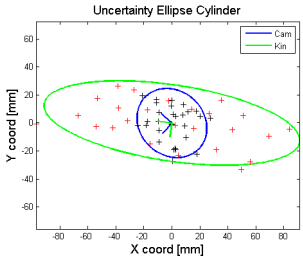
The article presents an innovative approach for monocular object localization. The object detection from the image has been made with HOG algorithm in a pre-processing stage. Then 3D localization is performed by a cost-function that considers: a) the matching of the object reprojected curvature with the image gradient; b) the convolution with object-Light appearance and the image gray levels; c) the quantification of the reprojected contours aligned with the image gradients. The 3D pose estimation accuracy has been quantified with a calibration grid giving the eigenvalues of the uncertainty ellipses identifying the dispersion of data along principal directions. These data are also compared to Microsoft Kinect. Our method gives an object localization accuracy comparable and in some cases even better using only a single camera. The algorithm is competitive in cluttered scenes as it relies on the whole object in contrast to only edges as in the case of edge/gradient methods. Figure 8 represent the limit of detection: for box the percentage error is around 2-2.5% and for cylinder is around 1.5%. Limitations are the knowledge of the light direction and the homogeneity of the surface texture. Nevertheless, in industrial fields for example, the method can cover a broad spectrum of applications. Also outdoor, where Kinect cannot be exploited, the use of a simple camera in combination to our method could represent a good alternative. Optimization is a tricky aspect. Especially the correct orientation retrieval generate many minimal local problems (ref. figure 6.d). It is important to point that the lowest values of the cost function always lead to correct localization while higher values are correlated to local minima. We are therefore confident that, automatizing the optimization process, it is possible to cope with multi-modal problems.

AKNOLEDGMENTS

The authors are very grateful to colleagues from Mechatronics dep. (University of Trento) and EU grants, incl. Marie Curie-COFUND-Trentino postdoc program, 2010-2013.

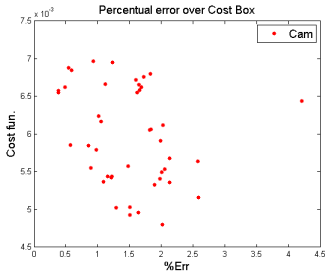


(a)

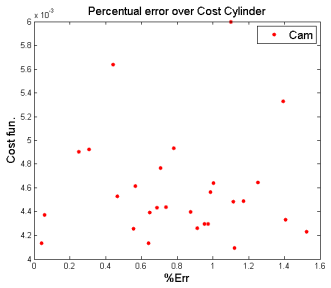


(b)

Figure 7: The comparison between the ellipses of uncertainty (green - for Kinect, blue - for our method) for the box (a) and the cylinder (b) localizations according to the reference (calibration) grid.



(a)



(b)

Figure 8: The dependence of the optimized cost function values on the errors percentage for the box (a) and the cylinder (b). Percentage error is the ratio between object distance and position errors. Good matching are considered under 0.006 value of the cost function.

9. REFERENCES

[1] Ilya Afanasyev, Massimo Lunardelli, Nicolò Biasi, Luca Baglivo, Mattia Tavernini, Francesco Setti, and Mariolino De Cecco. 3d human body pose estimation by superquadrics. In *VISAPP (2)*, pages 294–302, 2012.

[2] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:11–23, 1981.

[3] G. Biegelbauer and M. Vincze. Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation. In *2007 IEEE International Conference on Robotics and Automation,*, pages 1086 – 1091, april 2007.

[4] Alvaro Collet, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *ICRA*, pages 48–55. IEEE, 2009.

[5] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):932 –946, jul 2002.

[6] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[7] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des.*, 22(7):632–658, October 2005.

[8] Robert Hanek, Thorsten Schmitt, Sebastian Buck, and Michael Beetz. Fast image-based object localization in natural scenes. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2002*, pages 116–122, 2002.

[9] Ilic S. Sturm P. Navab N. Fua P. Hinterstoisser S., Cagniart C. and Lepetit V. Gradient response maps for real-time detection of textureless objects. In *IEEE Trans. on PAMI V.34(5). P. 876-888*, 2012.

[10] I. Huerta, A. Amato, J. Gonzàlez, and J. J. Villanueva. Fusing edge cues to handle colour problems in image segmentation. In *Proceedings of the 5th international conference on Articulated Motion and Deformable Objects*, AMDO '08, pages 279–288, Berlin, Heidelberg, 2008. Springer-Verlag.

[11] Aless Jaklivc, Alevs Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecth, 2000. ISBN 0-7923-6601-8.

[12] Fryer J.G. Lens distortion for close-range photogrammetry. *Photogrammetric engineering and remote sensing.*, pages P.30–37, 1986.

[13] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942 –1948 vol.4, nov/dec 1995.

[14] Jaka Krivic and Franc Solina. Superquadric-based object recognition. In *Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*, CAIP '01, pages 134–141, London, UK, UK, 2001. Springer-Verlag.

[15] Bouthemy P. Marchand E. and Chaumette F. A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing.*, V.19(13):P. 941–955, 2001.

[16] Eugenia Montiel, Alberto S. Aguado, and Ed Zaluska. Surface subdivision for generating superquadrics. *The Visual Computer*, 14(1):1–17, 1998.

[17] Bill Triggs Navneet Dalal. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, 2005.

[18] Bill Triggs Navneet Dalal. Object detection using object histogram of oriented gradiente. 2006.

[19] David McAllester Pedro F. Felzenszwalb, Ross B. Girshick and Deva Ramanan. Object detection with discriminatively trained part based models. *Computer Society Conference on Computer Vision and Pattern Recognition.*, V.32(9).:P. 1627–1645, 2010.

[20] Bui-Tuong Phong. Illumination for Computer Generated Pictures. 18(6):311–317, 1975.

[21] Christian Schmaltz, Bodo Rosenhahn, Thomas Brox, and Joachim Weickert. Region-based pose tracking with occlusions using 3d models. *Mach. Vis. Appl.*, 23(3):557–577, 2012.

[22] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Comput. Graph. Forum*, pages 214–226, 2007.

[23] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainly. *Int. J. Rob. Res.*, 5(4):56–68, December 1986.

[24] Fan-Tung Wei, Sheng-Ting Chou, and Chia-Wen Lin. A region-based object tracking scheme using adaboost-based feature selection. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2753 –2756, may 2008.