Stereo image processing system for robot vision

Ferenc Tél, Béla Lantos

Budapest University of Technology and Economics
Departement of Control Engineering and Information Technology
Hungary
http://www.iit.bme.hu

Abstract. More and more applications (path planning, collision avoidance methods) require 3D description of the surround world. This paper describes a stereo vision system that uses 2D (grayscale or color) images to extract simple 2D geometric entities (points, lines) applying a low-level feature detector. The features are matched across views with a graph matching algorithm. During the projective reconstruction the 3D description of the scene is recovered. The developed system uses uncalibrated cameras, therefore only projective 3D structure can be detected defined up to a collineation. Using the Euclidean information about a known set of predefined objects stored in database and the results of the recognition algorithm, the description can be updated to a metric one.

1 Introduction

The paper describes a stereo vision system that is able to produce useful information to an intelligent robot control algorithm. The developed system consists of four main phase. During the low-level processing the 2D features are extracted from raw camera images. In the image matching phase the corresponding features are determined across images using a graph matching method. Because the cameras are not calibrated, in the third phase only the 3D projective structure could be recovered from the paired featues. In order to get Euclidean (metric) description of the scene, the required a priori information is extracted from a database, that contains the description of predefined objects. An indexing method using projective invariants was developed to recognize objects. The output of the system is the list of recognized objects and the relative pose (position and orientation) between them that could be determined using the different coordinate systems (a common projective frame and the metric frame attached to the objects). This information could be directly used in an intelligent robot control system.

2 Low level processing

The aim of a low level (early vision) image processing is to extract useful information for subsequent phases from raw camera images and to reduce the amount of the data, and give some parametric description of the image.

2.1 Feature extraction and edge segmentation

The developed system uses an enhanced version of Canny detector as proposed in [7] to extract edges that are usually the projections of important things in the 3D scene from raw camera images. The output of the detector is the linked list of edges and each edge point is described with its coordinates (u, v) at subpixel accuracy.

Because the Canny detector uses gray scale images (intensity information) only, therefore in order to handle color images two methods have been developed. The first method performs a preprocessing step to get a scalar valued description. After testing the different possibilities, the developed system uses the brightness channel of the HSB transformed image. The second method uses a modified edge detection where the derivatives are replaced with vector relations such as distance and angle between RGB vector, where the relative weight depens on the saturation.

Edges can be usually composed as a collection of straight line segments and pieces of second order curves. The boundaries (vertices) are mainly those edge points, at which the curvature is high.

The curvature is estimated fitting an ellipse onto the edge points in the neighbourhood of the given point. There is a tradeoff between the smoothing and accuary depending on the neighbourhood size. Global thresholding cannot be applied, because it does not detect two lines which are connected at obtuse angle while it does separate a circular arc with small radius (high curvature).

It turned out, that interesting parts of the curvature function has (splitting etc.) have special shapes. Therefore a template based edge segmentation was developed, applying correlation with templates containig the wanted shape. The used shapes is shown in Figure 1. Determining the peaks above threshold and applying a non-maxima suppression, the vertex points along the edge chain are determined.

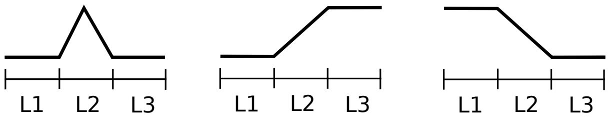

Fig. 1. Correlation function template profiles used in edge segmentation

2.2 Building the feature graph

Features are fitted onto the segmented edges. The feature set consists of line segments and second order curves. Because the fit is achieved onto continuous edge segments, therefore there are no outliers. The lines and conics are fitted using the methods in [1], [5].

After the initial classification of the edges and edge segments, a further refinement is also achieved. Initially point (junction, vertex) features are generated as endpoints of standalone segments or as the intersection of the connected segments. An iterative method is applied till the the structure (feature set) changes.

- Try to merge segments at vertices in topology preserving way, namely the merge is achieved if only two features are connected at a vertex. The fit is successfull if the error of the merged segments remains below a threshold.
- Junction coordinates are updated to the closest point to the involved curves.

The output of the feature refinement is used to generate a feature graph $G = \{\mathbf{V}(\mathbf{A}), \mathbf{E}(\mathbf{A})\}$, where $\mathbf{V}$ is the set of nodes (vertices) containing features and its internal parameters (type, length), $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$ denotes the edges containing 2D relative affine transformation between features, $\mathbf{A}$ represents the attributes. The application of this feature graph prevents to define of artifical higher level features (such as T-junctions, parallel lines).

3 Image matching

In order to build a 3D structure of the scene from 2D imaging information, generally a triangulation like process is required. This means that the corresponding entities must be determined in the participating images. This is a *correspondence problem* and the solution of this problem is difficult, because some entities have no correspondent in the other image caused by different viewpoints, occlusions.

There are two main classes of methods used to solve the image matching problem. The first class is based on the correlation information based on image intensities. The other class uses the comparison (similarity) of features extracted by a low level algorithm. The developed method tries to combine the advantages both classes. Our algorithm is based on graph matching and uses the feature graphs generated by low-level image processing, but for each node uses the image window around the feature to achieve a correlation based comparison. The pairing of the features can be described with a match matrix $\mathbf{M}$.

Without a priori information, initially every feature with similar attributes is candidate for pairing. Therefore it is not enough to initialize the match matrix with 1's and 0's only. The problem should be transformed from a discrete (binary) one to a continuous formulation. Hence the elements of the match matrix are represented by continuous real values $0 \leq \mathbf{M}(a,i) \leq 1$ during the minimization. The cost function of the matching can be written as:

$$E(\mathbf{M}) = -\frac{1}{2}\sum_{a,i,b,j} \mathbf{M}(a,i)\mathbf{M}(b,j)\sum_{r} C_{(r)}(a,i,b,j) + \gamma \sum_{a,i} \mathbf{M}(a,i) \sum_{s} C_{(s)}(a,i) + \\ + \sum_{a,i} F\left(\mathbf{M}(a,i), \kappa, \mu_a, \nu_i\right)$$

where a, b and i, j are indices of vertices of G_α and G_β. The similarity of the elements are involved by $C(\mathbf{.})$ cost functions. The first and second term compares the edges and vertices. The last term $F(\mathbf{.})$ represents the constraints for the elements of the match matrix, $\nu_i \sum_i \mathbf{M}(a,i) \rightarrow 1$ and $\mu_a \sum_a \mathbf{M}(a,i) \rightarrow 1$ enforces the row and column normality, the μ_a, ν_i are Lagrange parameters and $\frac{1}{\kappa} \sum_{a,i} \mathbf{M}(a,i) \log \mathbf{M}(a,i)$ prevents $\mathbf{M} \rightarrow \mathbf{0}$. The minimization is achieved by an application of a *deterministic annealing* method as in [4].

4 Projective reconstruction

The aim of the projective reconstruction methods is to extract 3D information (namely the structure of the surround scene) from corresponding data detected in 2D camera images. Two methods have been developed, the first one uses point features only, the second one is able to reconstruct 3D line features too.

4.1 Points only method

Using the pinhole camera modell, the basic projection equation for points has the form $\rho_{ij}\mathbf{q}_{ij} = \mathbf{P}_i\mathbf{Q}_j$, where $\mathbf{Q}_j$, $\mathbf{q}_{ij}$ is the homogeneous coordinate vector of the 3D point and its 2D projections, respectively. The $\mathbf{P}_i$ is the 3×4 projection matrix. If there are m cameras and n_P points in the scene, then the number of projected image points (and scale factors) are $m \times n_P$. But only $m + n_P$ are independent amongst them, therefore the scale factors should be decomposed into camera dependent and feature dependent parts, such that $\rho_{ij} = \pi_i\gamma_j$. Using this decomposition, the projection of a point is described by $\pi_i\gamma_j\mathbf{q}_{ij} = \mathbf{P}_i\mathbf{Q}_j$, where the unknowns are π_i, γ_j, $\mathbf{P}_i$, $\mathbf{Q}_j$. This decomposition has some advantages: i) the system is described with the minimum number of parameters, therefore the parameterization is consistent, ii) the number of unknowns is greatly reduced.

In order to minimize a physically meaningfull quantity, the reprojection error is used in the cost function as

$$E_P\left(\centerdot\right) = \sum_{i=1}^{m}\sum_{j=1}^{n_P} \omega_{ij}^2 \left\|\pi_i\gamma_j\mathbf{q}_{ij} - \mathbf{P}_i\mathbf{Q}_j\right\|^2 \tag{1}$$

It can be seen, that $E_P\left(\centerdot\right)$ is nonlinear in the unknowns. Instead of using the Levenberg-Marquardt method and general initial values to directly minimize this cost function, the parameters to be estimated can be separated into different groups, because they are “independent” from each other. This resection-intersection method holds every group of parameters fixed, except those, that are currently minimized. In each iteration step closed form solution exists.

4.2 Points and lines method

The reprojection error function cannot be easily extended to handle line features, because for example the mapping between elements of the point and the line projection matrix is a non-linear function. Therefore the cost function is modified and the scale factors are eliminated.

In the *intersection* step the $\mathbf{P}_i$ (hence the line projection matrices $\mathbf{G}_i$) are held fixed and the 3D features can be treated as independent from each other:

$$E_I(\boldsymbol{.}) = \sum_{i=1}^{m}\sum_{j=1}^{n_P}\omega^2_{Q,ij}\|\mathbf{q}_{ij} \times (\mathbf{P}_i\mathbf{Q}_j)\|^2 + \sum_{i=1}^{m}\sum_{k=1}^{n_L}\omega^2_{\Lambda,ik}\|f(\boldsymbol{\Lambda}_k, \mathbf{P}_i, \mathbf{l}_{ik})\|^2 \qquad (2)$$

where $\boldsymbol{\Lambda}_k$ is the Plücker representation of the 3D line, $\mathbf{l}_{ij}$ represents the 2D projected line. The estimation for the jth feature can be calculated by making the derivative of $E_I(\boldsymbol{.})$ by $\mathbf{Q}_j$ and $\boldsymbol{\Lambda}_k$ to zero. Then the solution for each $\mathbf{Q}_i$ and $\boldsymbol{\Lambda}_k$ can be found in closed form.

In the *resection step* the $\mathbf{Q}_j$ and $\boldsymbol{\Lambda}_k$ entries are held fixed, the cameras are independent from each other:

$$E_R(\boldsymbol{.}) = \sum_{i=1}^{m}\sum_{j=1}^{n_P}\omega^2_{Q,ij}\|\mathbf{q}_{ij} \times \mathbf{P}_i\mathbf{Q}_j\|^2 + \sum_{i=1}^{m}\sum_{k=1}^{n_L}\omega^2_{\Lambda,ik}\left(\mathbf{l}_{ik}\mathbf{P}_i\mathbf{Q}_k(\Lambda, t)\right)^2 \qquad (3)$$

The estimation for the ith camera can be calculated in closed form by making the derivative of $E_R(\boldsymbol{.})$ by $\mathbf{P}_i$ to zero. Note, that in this case the error function contains only the "point-form" $\mathbf{P}$ of the projection matrices.

4.3 Minimization remarks

The two developed algorithms have some common properties.

- The ω_{ij} weights can be used to eliminate outliers.
- Handling of missing data (features having no projection on the given view) during the minimization is simple, the algorithms skip those entries in the error function.
- The initialization is achieved applying a rank 4 decomposition of the $3m' \times n_{P'}$ measurement matrix $\mathbf{Q} = [\mathbf{q}_{i'j'}]$ built from the subset of the point features visible in all of the images in the m' subset. The remaining cameras and features are initialized from these values, recursively.

5 Object recognition and euclidean reconstruction

During the recognition process two sets of entities are used. The first one is the feature sets of the object as stored in the object database. The second one is the features of the recovered scene. Some elements (a subset) represent the same entity in different context (e.g. two representation of the geometric primitives in different coordinate systems). In order to determine the pairing of the two representations of the same entities the process requires the usage of those properties which are not changing (invariant) between representations which are related by a most general homogeneous transformation (collineation).

5.1 Projection and permutation invariants

In order to use in the indexing method within the database, the invariant values must be preserved applying a collineation. The developed system uses (generalized) cross ratio built from simple feature configurations (six points, two lines and three points, five lines), that can be written as the ratio of product of determinants. The simplest generalization of the cross ratio [3] requires at least $N+3$ points in an N-dimensional space, as

$$I(\mathbf{Q}_1, \ldots \mathbf{Q}_N, \mathbf{Q}_{N+1}, \mathbf{Q}_{N+2}, \mathbf{Q}_{N+3}) = \frac{|\mathbf{Q}_1 \ldots \mathbf{Q}_N \mathbf{Q}_{N+2}|\,|\mathbf{Q}_1 \ldots \mathbf{Q}_{N+1} \mathbf{Q}_{N+3}|}{|\mathbf{Q}_1 \ldots \mathbf{Q}_N \mathbf{Q}_{N+3}|\,|\mathbf{Q}_1 \ldots \mathbf{Q}_{N+1} \mathbf{Q}_{N+2}|}$$

It can be seen, that changing of the labelling of the first $N-1$ points leaves the value of the invariant intact (the sign changes of the four determinants cancel each other), the permutation of the last four points yields the six different values. Putting together, the projective and permutation invariants must fullfill two requirements

- *Problem 1*: Eliminate the effect of the six possible value of the cross ratio. This is accomplished using stereographic permutation invariants.
- *Problem 2*: Eliminate the effect of interchanging between first $N-1$ and the last four elements.

First a stereographic projection of the calculated cross ratio is achieved such that the values $\{0.5, 1, 2, \infty, -1, 0\}$ are mapped onto the angles $\left(0, \frac{pi}{3}, \frac{2pi}{3}, \pi, \frac{4pi}{3}, \frac{5pi}{3}, \right)$ and 0 is at the south pole. Then applying a periodic function $(J_p(I_{st}) = J_p(I_{st} + k\frac{\pi}{6}), k = 0, \ldots, 5)$ the ouput gives a solution to the *Problem 1*. The reson of such indexing is that the values are mapped into $[0, 1]$ interval. In order to apply a simple distance function during the indexing, a nonlinear transformation was defined so, that the output density is close to the uniform one. Amongst the severeal tested possibilities, the best choice was the $J_p = 0.57(J_{pb}(I) - 6J_{pb}(I_{st} - \frac{\pi}{6})) + 0.86$, where $J_{pb} = \frac{1}{\pi} \arcsin\left(\sqrt{\frac{1}{\pi}\,|\arcsin(\sin(3I_{st}))|}\right)$.

To solve the *Problem 2*, the vector valued projection and permutation invariants are defined as proposed in [6]. A method has been developed that is able to extract feature pairing from the result of the indexing.

5.2 Query into the database and verification

The query process extracts those elements from the database that are closest to the querying element within a tolerance (kNN problem). Because the invariants are higher dimensional vector valued entities, therefore the standard R-tree algorithm is very inefficient, due to the curse of dimensionality. Hence the developed method uses X-tree [2].

Because of the query eliminates only the false positives (those configurations, that are surely do not yield a valid answer to the query), the remaining candidates are post-processed with a verification process. Initially the ouput of the query is an ordered (matched) feature set (corresponding configurations), containing only as many (minimum number) of features as required by indexing. During the verification process a 3D homogeneous transformation (collineation) is calculated, that maps projective coordinates of the scene features into the Euclidean space of the candidate object. Checking those remaining object features that are not yet on the candidate list, corresponding scene features are searched (closest mapped scene feature, within a given distance threshold). If a sufficient pair is found, it is appended onto the support list of the given configuration. If the number of supports is above a limit, the whole transformation is stored for final consolidation processing, that is a multidimensional clustering process.

5.3 Euclidean update

The coordinates of the features of the recognized objects are known in two coordinate systems. The first is the common projective frame of the scene, in which every feature is described with its projective coordinates. The second is the Euclidean frame attached to the object as stored in the object database. This information is object dependent and contains metrical data. Using this twofold description of the recognized features makes it possible to determine the relative Euclidean transformation (position, orientation) between object frames as occurs in the scene. If one of the frames is absolute (known in world reference frame), then it is possible to describe the scene in the absolute Euclidean frame.

The candidate collineations between the common projective space of the scene and the local frame of the recognized object are already determined as the output of verification and clustering step of the object recognition, therefore this is the most accurate estimation that is available, no internal constraints applied. This means that the elements depend only on the data from which they are estimated, there is no inter-dependency between elements.

However the calculation of the Euclidean transformation between objects allows introducing additional constraints. Let us suppose, that the collineations describe the mapping from scene frame into Euclidean object frames. In this case the displacement describing the mapping from metric frame of the object A into metric frame of object B can be calculated as the product of collineations.

The resulted displacement matrix be factorized into a product of a scaling factor and the matrices are responsible for perspectivity, translation, rotation, mirroring and stretch (transformed shear). Using the decomposed form the constrains can be expressed as a limit on phisically meaningful quantities. Examining different decomposition algorithms, the system uses polar decomposition.

6 Results

An output of the image segmentation algorithm is shown in Figure 2 as cropped from a larger image. Note, only that edge is overlayed whose processing results is shown in the right side of the Figure. The determined breaking points is denoted by crosses. (All of the profiles was set to $L_1 = L_2 = L_3 = 60$).

The accuracy of the reconstruction algorithm was tested with simple simulated scenes where the ground truth of the data is known. In the reconstrcution part, the base of the evaluation criteria is the reprojection error. For points this is the distance between the true (original) and the reprojected 2D coordinates. In case of lines, the reprojection error consists of two parts, the angle between the original 2D line and the reprojected line and the maximum distances of the endpoints of the original line segments from reprojected lines. Result of the simulations shows that the accuracy of the reconstruction depends approximately linearly on the noise added to position of the detected 2D features. Checking the numerical results, the errors between the original (detected by a feature detector) and the reprojected (estimated) image features are in the range $0 \ldots 5$ pixels for points, and the angle error $0 \ldots 5$ degree for lines, see Figure 3.

The developed method was also tested with real data in order to demonstrate applicability, see Figure 4. The upper part shows the reconstructed 3D features reprojected and overlayed onto the image. The lower part shows the output of the recognition algorithm that can localize objects in the scene for use in robot applications.

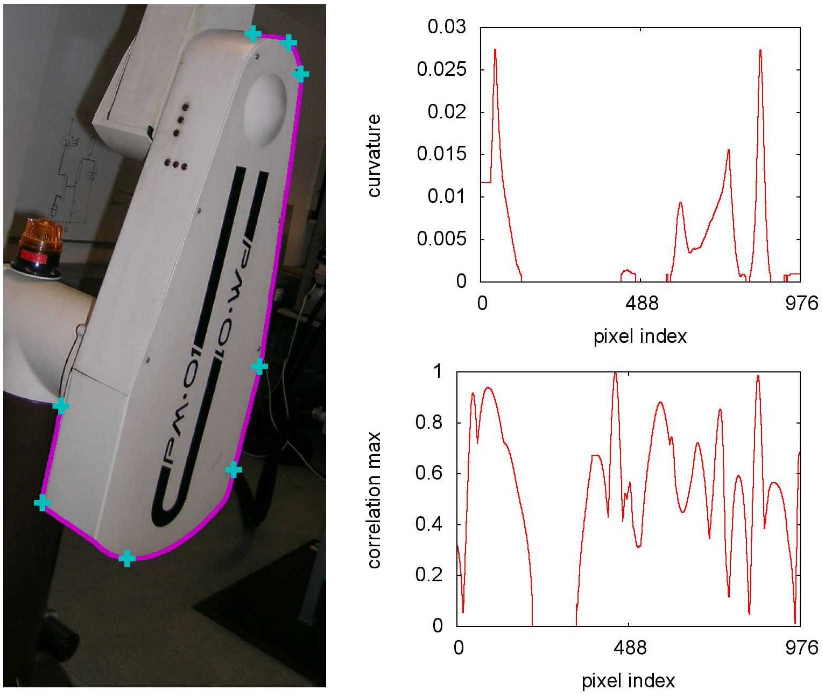

Fig. 2. Segmentation of a real edge

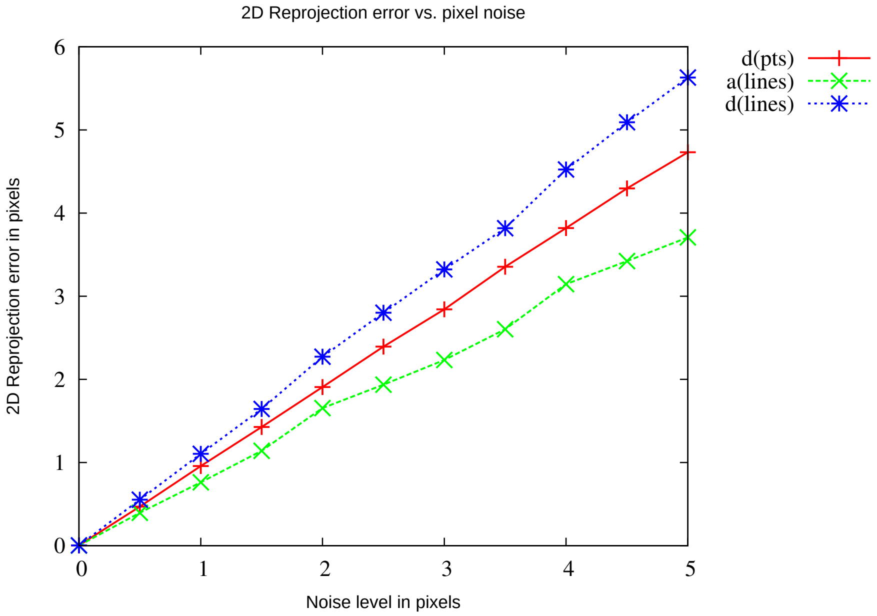

Fig. 3. Noise tolerance of the projective reconstruction algorithm for points and lines

Acknowledgments. Support for the research was provided by the Hungarian National Research Programs under grant No. OTKA K 71762 and by the Control Systems Research Group of the Hungarian Academy of Sciences.

References

1. Agin, G.J. Fitting ellipses and general second-order curves, Technical Report CMU-RI-TR-81-5. The Robotics Institute, Carnegie-Mellon University, Pittsburgh, USA, (1981)
2. Berchtold, S., Keim, D.A., Kriegel, H, The X-tree: An Index Structure for High-Dimensional Data, in VLDB96, Bombay, India, pp. 28–39, (1996)
3. Csurka, G., Faugeras, O.D., Algebraic and Geometric Tools to Compute Projective and Permutation Invariants, in IEEE Trans. on Pat. Anal. and Machine Intel., Vol. 21, No. 1., pp. 58–65, (1999)
4. Gold, S., Rangarajan, A., A graduated assignment algorithm for graph matching, in IEEE Trans. Pat. Anal. and Machine Intel., Vol. 18, No. 4, pp. 377–388, (1996)
5. Halir, R. and Flusser, J. Numerically Stable Direct Least Squares Fitting of Ellipses. Skala, V (ed.) Proc. Int. Conf. in Central Europe on Computer Graphics, Visualization and Interactive Digital Media. pp 125–132, (1998)
6. Meer, P., Lenz, R., Ramakrishna, S. Efficinet Invariant Representations in Int. J. of Computer Vision, Vol. 26, Issue 2., pp. 137-152, (1998)
7. Rothwell, C.A., Mundy, J.L., Hoffman, W., Nguyen, V.D., Driving Vision By Topology, in IEEE Symposium on Computer Vision SCV95, pp. 395–400, (1995)

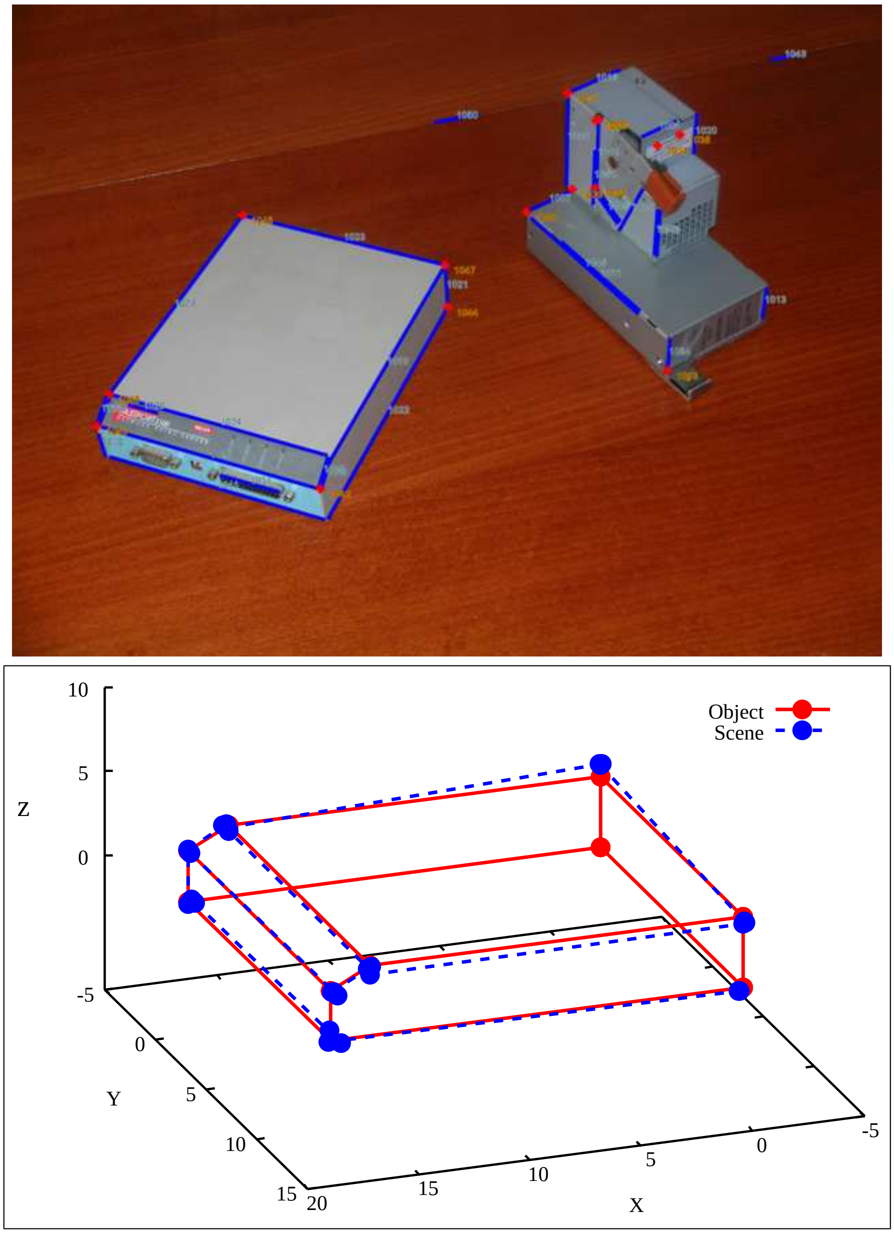

Fig. 4. Real scene example, detected features (above), recognition result (bellow)