# AUTOMATIC OBJECTS DETECTION FOR MODELING INDOOR ENVIRONMENTS

Marion Decrouez[1,2], Romain Dupont[1], François Gaspard[1] and James L. Crowley[2]
[1] *CEA, LIST, Content Engineering and Vision Lab., Point Courrier 94, F-91191 Gif-sur-Yvette, France*

Keywords: Visual SLAM, Bag of Words, Place Recognition, Objects Detection.

Abstract: In this paper we describe a new solution for constructing a model of a scene and its objects using various explorations of a single camera in an unknown environment. Object motion presents a difficult challenge to scene modeling. The proposed method combines metric localization and place recognition to detect and model objects without *a priori* knowledge and to incrementally extend a scene model by adding new places and objects. We demonstrate the quality of our approach with results from image sequences taken from two different scenes.

## 1 INTRODUCTION

Live processing of a video sequence taken from a single camera enables to model an *a priori* unknown 3D scene. Metrical SLAM (Simultaneous Localization and Mapping) algorithms track the camera pose while reconstructing a sparse map of the visual features of the 3D environment. Such approaches provide the geometrical foundation for many augmented reality applications (Mouragnon et al., 2006) in which informations and virtual objects are superimposed on live images captured by a camera. Improving such systems will enable in the future precise industrial applications such as guided-maintenance or guided-assembly in wide installations. A problem with current methods is the assumption that the environment is static. Indoor environments such as supermarket ailes and factory floors may contain numerous objects that are likely to be moved, disrupting a localization and mapping system. In this article, we explore a method for automatic detection and modeling of such objects. We define the scene as a static structure that may contain moving objects. Without any *a priori* knowledge, we define an object as a set of visual features that share a common motion compared to the static structure, as illustrated figure 1. We analyse multiple explorations of a camera in the same environment to extract as many informations as possible on the scene and its temporal evolution. The system presented in this article enables us to reconstruct an unknown environment in 3D, to locate the observing camera within the scene, to recognize previously visited areas and to model new objects. We start with a review of the current state of the art for localization and mapping section 2. We then describe a new technique for localization and mapping in section 3, followed by a description of methods for automatic detection of mobile objects in section 4. Results from experiments with this method are presented in section 5.

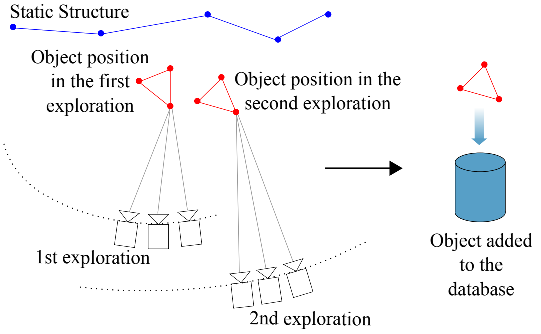
Static Structure
Object position in the first exploration
Object position in the second exploration
1st exploration
2nd exploration
Object added to the database

Figure 1: Automatic object detection. Comparing two video sequences of the same scene taken at different times highlight moved objects.

## 2 RELATED WORK

Vision-based methods for localization and mapping can be classified as metric or topological. Metric SLAM algorithms enable a visual sensor to explore an *a priori* unknown environment performing live mapping while simultaneously using the map to estimate the position and orientation of the camera. However, existing solutions (Mouragnon et al., 2006) are prone

Decrouez M., Dupont R., Gaspard F. and L. Crowley J..
AUTOMATIC OBJECTS DETECTION FOR MODELING INDOOR ENVIRONMENTS.
DOI: 10.5220/0003818205730578
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2012), pages 573-578
ISBN: 978-989-8565-03-7

to accumulation of numerical errors. In such a case, it becomes difficult to detect that a camera has returned to a previously visited position and such algorithms do not provide the capability to relocate the camera within the map in cases where the camera pose is lost. Several topological or appearance-based approaches have been described to address such challenges. Cummins and Newman (Cummins and Newman, 2009) define a probalistic model over the bag-of-words representation (Sivic and Zisserman, 2003). They determine when the camera is revisiting a previously mapped area on the basis of image similarity and do not require metric estimations. Their method has been found to be robust to perceptual aliasing (the fact that different places have similar appearances) by taking into account the co-occurence of the visual words in the appearance likelihood estimation. Unfortunately, it does not allow real-time processing and it has proven its effectiveness on panoramic images and we want to use medium-sized images of indoor environments. Recent efforts have been made to combine both approaches, in order to deal with longer trajectories while maintaining the 3D point map required for augmented reality applications. Castle et al. (Castle et al., 2008) propose an approach that works with several 3D maps. The system automatically switches between maps by relocalizing the current image relative to previously visited area. Other recent approaches attempt to extract additional 3D information from the video stream to extend the scene understanding and improve the SLAM results. Angeli et al. (Angeli and Davison, 2010) suggest grouping feature points into 3D clusters using similar appearance and 3D proximity information. Lastly, Kim et al. (Kim et al., 2010) describe a solution for modeling and tracking multiple 3D objects in unknown environments. An object database is built offline and the user can add a new object by selecting a region of the image. To our knowledge, there is no solution to automatically enrich the objects database using multiple explorations of the same environment with a mobile camera. This paper proposes a method to analyse and compare different explorations of the same indoor environment in order to detect displaced objects and add them to an object database. Our method is illustrated in the figure 2. We use a keyframe-based SLAM algoritm. Every new keyframe, we update the 3D map and search for the closest previous frame to compare the 3D reconstruction. The scene model and the object detection are described in the following sections.

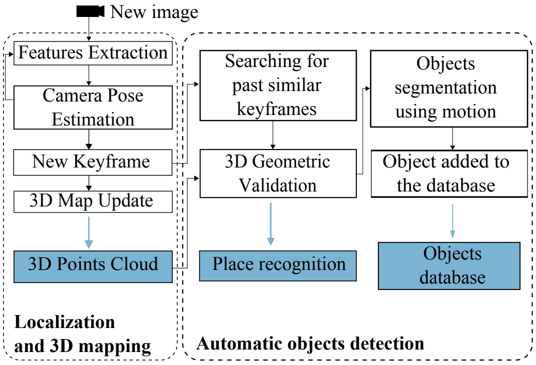
New image
Features Extraction
Camera Pose Estimation
New Keyframe
3D Map Update
3D Points Cloud
Localization and 3D mapping
Searching for past similar keyframes
3D Geometric Validation
Place recognition
Objects segmentation using motion
Object added to the database
Objects database
Automatic objects detection

Figure 2: Proposed method: 3D reconstruction and automatic objects detection.

## 3 MODELING THE SCENE

This section presents our proposed algorithms for metric SLAM and appearance-based place recognition. These two methods are combined in the 3D geometrical validation solution described section 3.3.

### 3.1 3D Reconstruction

We use the method described by Mouragnon et al. (Mouragnon et al., 2006) to estimate the points positions and the camera pose. Live processing of the video stream enables to build a sparse map of 3D points. The system detects Harris-Stephen points in the current image and extracts SURF descriptors (Bay et al., 2006). These interest points are matched with the projection of 3D points seen in the previous frame to compute the camera pose. Some keyframes are selected to compute the 3D coordinates of the observed points with the new camera pose. Thus, the map is updated and the system optimizes the scene using local bundle adjustment. This method constructs the 3D environment in real-time but it is prone to errors in the camera pose and scale drift. Such errors can disrupt detection of loop closure. To avoid such problems, we use a place recognition algorithm based on appearance to first check whether the current image comes from a previously visited location 3.2 and then detect displaced objects 4.

### 3.2 Place Recognition

We use a place recognition algorithm based on the bag-of-words image representation (Sivic and Zisserman, 2003) to detect previously visited areas. This method shows outstanding retrieval results on large-scale image databases. Local features are detected in the image and quantized in visual words with respect to a vocabulary. The vocabulary is learned beforehand by clustering all feature vectors from a set of training data (3000 random Flickr images) using k-means clustering and contains 10000 visual words. Each keyframe selected by the SLAM method is registered in the image database and is represented as a vector of visual words. We take advantage of the inverted index to find the most likely past keyframe that matchs the current keyframe. Each time a word is found, we update the similarity scores of the past images retrieved from the index by adding the term frequency - inverted document frequency (tf - idf) weighting term as in (Sivic and Zisserman, 2003). Thus, we measure the similarity between a pair of images and then assume that two images with high similarity score are taken from the same location. However, the current observation may come from a previously unknown place. A geometric post-verification stage, which tests the geometric consistency of the matched images, is required.

### 3.3 Merging Both Approaches with a 3D/2D Geometric Validation

We confirm the place recognition hypothesis with a 3D validation. Features extracted in the database image are matched to the projection of 3D points seen in the current image. We estimate the relative pose between these two similar images. We retain the match if there are enough points verifying the geometric constraint. Thus, we reject all errors due to perceptual aliasing. Besides, this method makes it possible to determine the static structure of the scene and to identify a set of inconsistent points. In the case where a scene is composed of multiple rigid objects moving relative to each other, we can detect possible objects as nearby sets of points that share similar movement. We present an overview of the state of the art for two-view multiple motion estimation in the section 4.1 and our method for object detection in section 4.2.

## 4 AUTOMATIC OBJECTS DETECTION

Comparing two views of the same 3D scene taken at different times highlights 3D points inconsistent with the static structure. We want to infer the presence of moved objects by clustering points according with their motion. The setting is the following: given the set of corresponding points in two similar images, we have to estimate the movement of the camera and the movement of an unknown number of moving objects. In this section we first review alternative methods for two-view multibody estimation and then describe our approach.

### 4.1 Two-view Multiple Structures Estimation

To simplify the problem, we consider only planar objects. We need to detect multiple planar homographies in image pairs. Zuliani et al. (Zuliani et al., 2005) describe the multiRANSAC algorithm but this method requires prior specification of the number of model. Toldo and Fusiello (Toldo and Fusiello, 2008) present a simple method for the robust detection of multiple structures in pairs of images. They generate multiple instances of a model from random sample sets of correspondences and then merge group points belonging to the same model using a agglomerative clustering method called *J-Linkage*. Our method is based in part on this algorithm. We combine planar detection with 3D reconstruction to detect only moving objects.

### 4.2 Identification of the Moving Objects in the 3D Scene

Our metrical SLAM algorithm constructs a sparse map of the environment. Inconsistent points retrieved at the 3D validation step aren't enough to estimate a model and define an object. To tackle this problem, we extract a large number of features in each image, match them and generate many local hypotheses of homographies. We then merge sets of points belonging to the same motion using a technique explained below and finally keep those with points associated with 3D inconsistent points. We use SURF features to describe interest points. Each feature is matched with its nearest neighbor in the similar image. Figure 3 illustrates our method.

#### 4.2.1 Preliminaries and Notation

Points in the 2D image plane of a camera are represented by homogeneous vectors $p$. $p_1$ and $p_2$ are two corresponding points detected in pair of similar images. These points are the projection of the same 3D points in different camera views. We have to detect perspective transformations (homographies) that map planar surfaces from one image to the other. To do so, we find the set of correspondences fitting the same homography H:

$$p_2 \sim Hp_1. \tag{1}$$

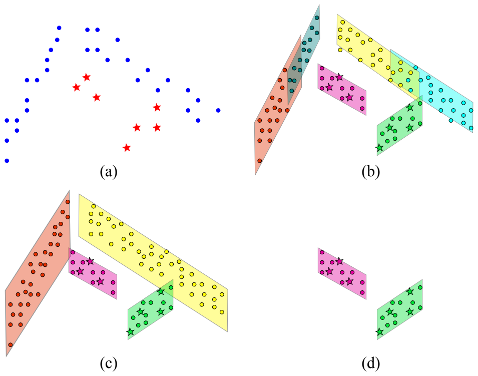

Figure 3: Object segmentation. **(a)** Estimation of the relative pose between pair of similar images highlights points of the static structure (blue circles) and inconsistents points (red stars). **(b)** Multiple planes detection: best homographies in each subregions of the image planes. **(c)** Sets of points belonging to the same model are merged. **(d)** Moved objects are detected as the sets of points containing inconsistents 3D points.

$H$ is a (3 x 3) matrix, with eight degrees of freedom. This matrix can be determined from four correspondences. In our case, the system is overdetermined as we have to estimate $H$ by taking into account all the correspondences that may verify the relation 1. Since the image point measurements are corrupted by noise, a correspondance will not lie exactly on the homography, but will differ from it by a residual $e$. To quantify the residual, we use the approximation of the geometric error called *Sampson-distance* (Hartley and Zisserman, 2000). Relation 1 can be reordered as an equation system $Ah = 0$, such as $h$ contains the nine unknown entries of $H$. *Sampson-distance* with respect to a homography is thus given by

$$e^2_{Sampson} = h^T A^T \left(JJ^T\right)^{-1} Ah, \qquad (2)$$

where $J = \frac{\partial(Ah)}{\partial(\bar{p})}$ is the Jacobian of the linear equation system. We consider that a correspondance is inlier if its residual is below a threshold $\varepsilon$ (we take $\varepsilon = 1.5$). Using the form given in (Hartley and Zisserman, 2000) : $\varepsilon = \sqrt{5.99} * \sigma$. $\sigma$ is the scale of the noise of the data measurements, here $\sigma = 0.6$.

#### 4.2.2 Iterative Ransac Procedure

We want to use a simple method to find the best homographies between a pair of images. We use a sequential RANSAC procedure: we sequentially apply RANSAC and remove the inliers from the data set as each model instance is detected. At each iteration, the model that fits the larger number of points is the best model. The procedure is repeated if the number of remaining points is sufficient.

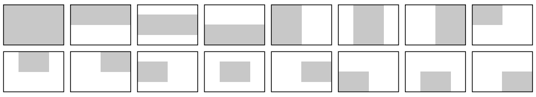

Figure 4: Local sampling used to generate RANSAC hypothesis. Samples are drawn in subregions of the image plane.

#### 4.2.3 Local Sampling

The scene may contain multiple plans and the data is corrupted by noise and matching errors. Under these conditions, only a small fraction of all correspondences belongs to each model. The problem may become intractable: using the standard formula for RANSAC and considering that an object covers 10 percent of the entire image we need $\frac{log(1-0.99)}{log(1-0.1^4)} \approx$ 46000 iterations. To overcome this problem, we exploit the spatial coherence of points belonging to the same object (clustered in a region of the image) and generate RANSAC hypotheses using a local sampling like Schindler et al. (Schindler and Suter, 2006). The image plane is subdivided into three overlapping rows and three overlapping columns and the samples are drawn from the entire image, each column, each row, and each of the nine regions defined by a row-column intersection 4. This heuristic takes advantage of the spatial coherence and reduces the required RANSAC sample number. We assume that an object covers at least 50 percent of one region, we also need to generate $\approx$ 100 samples per regions.

#### 4.2.4 Merging

Our algorithm generates groups of points $X_1, ..., X_n$ belonging to planar regions. Since the initial hypotheses are generated with local sampling, a large planar surface in the scene may result in several planes (figure 3(b)). We need to merge these groups of points. We first merge groups with more than 80 percent of points in common. Then, then for each pair of sets $X_1$ and $X_2$ we estimate a homography $\hat{H}$ with the group $X_1 \cup X_2$ using least-square minimisation estimation. $X_1$ and $X_2$ are merged if the mean of the error for the new model is below $\varepsilon$:

$$\frac{1}{|X_1 \cup X_2|} \sum_{c \in X_1 \cup X_2} e_{\hat{H}}(c) < \varepsilon \qquad (3)$$

#### 4.2.5 Objects Detection

The steps of the algorithm described above provide sets of points from planar regions in the scene (figure 3(c)). To detect an object, we must detect planar regions with motion that is different from the rest of

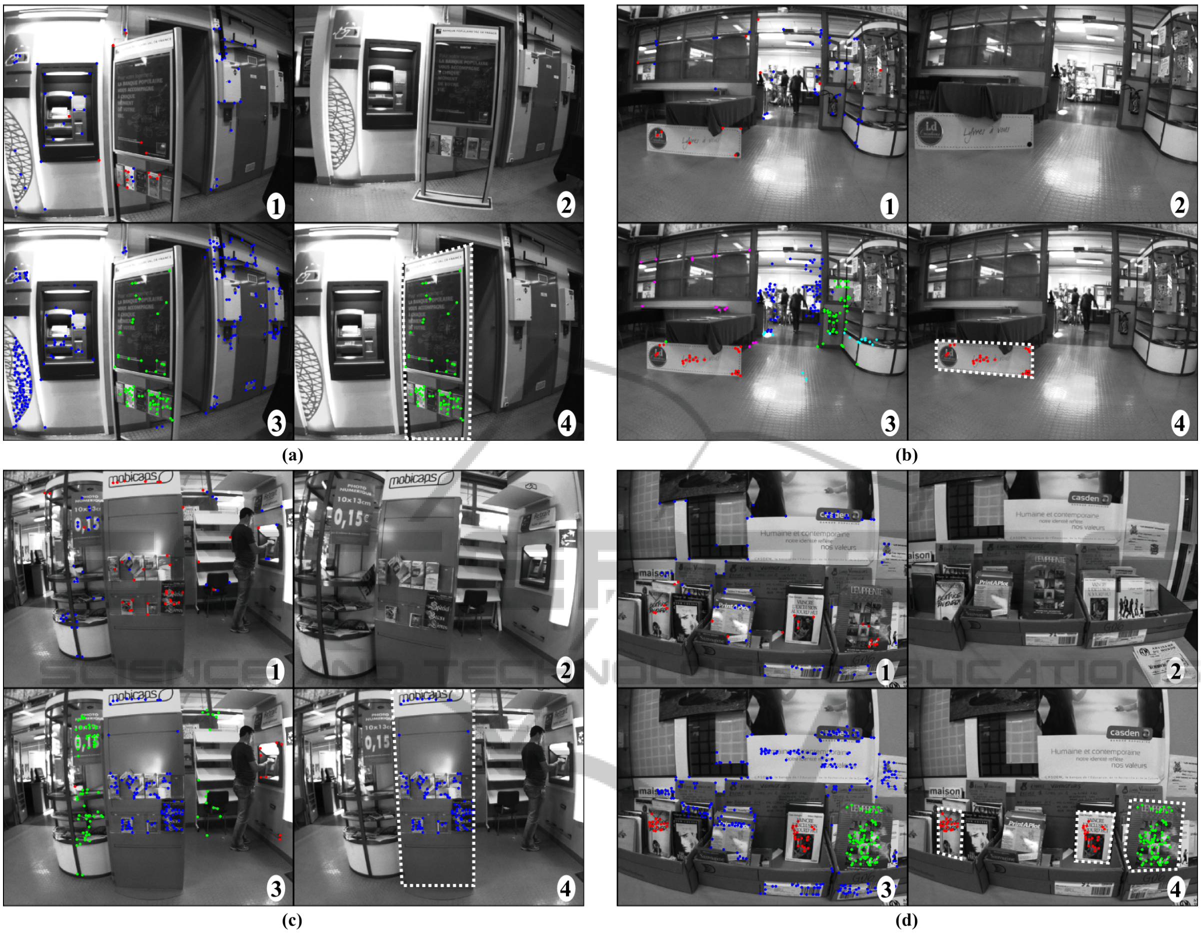

Figure 5: Objects detection on four place recognition cases. 1 : Current keyframe, blue circles denotes points of the static structure, red circles are inconsistent points. 2 : Similar database image. 3 : Homographies detected in image pairs. 4 : Edges of actually being moved objects are dashed, set of circles of different colors denotes the objects detected by our algorithm.

the scene. We use information from 3D reconstruction and geometric validation: moving objects are sets of points containing inconsistents 3D points (figure 3(d)).

# 5 EXPERIMENTAL RESULTS AND DISCUSSIONS

## 5.1 Experimental Validation

We have validated our algorithm on real data. We use a sequence of 2035 frames taken inside a building. Figure 5 presents our results for the object detection on four cases of place recognition. The first and second views are two images taken from the same location. The second view shows the projection of 3D points on the image plane: blue points are from the static structure and red stars are inconsistent points. The third view presents planar surfaces of the scene and the fourth view shows objects that have been actually moved (dashed) and points belonging to the same objet (circles of the same color). In figures 7(a), 7(b), 7(c) the moved objects are correctly detected. The disappearance of the person in figure 7(c) does not disrupt the recognition and the pose estimation. Indeed, the person occupy a small part of the image plane and the points detected on the person are not matched in the similar image. In figure 7(d), three objects have actually moved. Only two of them are detected by our algorithm as the two books in the left have a very similar motion.

## 5.2 Objects Detection Improvements

Figure 6 illustrates our results for the first sequence: 3D reconstruction, place recognition and object detection. Six moving objects were detected. In practice

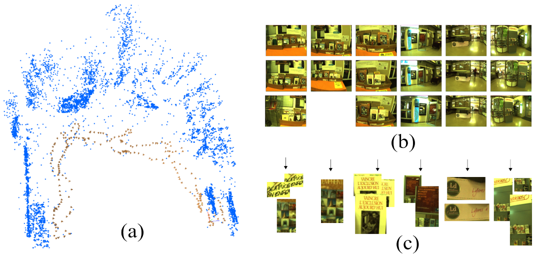

Figure 6: (a) 3D reconstruction of an indoor scene, (b) Matching images, (c) Objects detected in the scene.

we observe two limitations:

- outliers that do not belong to the object,
- falses object detection.

### 5.2.1 Outliers Management

Sets of points may contain outliers that do not belong to the object. We filter these outliers as follows: for each set of points associated with an object, we estimate the mean and standard deviation of the spatial coordinates and reject points outside $\bar{x}+2\sigma_x] \cap [\bar{y}-2\sigma_y, \bar{y}+2\sigma_y]$. $\sigma_x$ et $\sigma_y$ are the abscissa and ordinate standard deviations estimated with the *Median Absolut Deviation* estimator. $\bar{x}$ et $\bar{y}$ are the abscissa and ordinate medians.

### 5.2.2 Falses Detections

Our algorithm produces sets of points associated with objects in the scene. Nevertheless, we may observe falses positives in the detection (sets of static points considered as moved objets). There are a number of cases where the static structure of the scene is difficult to determine. For example, this can occur when much of the scene has moved between two explorations or when the moved object covers most of the image plane. For the first sequence, we count 10 falses positives on 70 detections. We filter these errors keeping only objects detected more than three times. In the future, we plan to avoid falses positives using *a priori* information from the analysis of previous frames, which can be used to precisely determine the static structure.

## 6 CONCLUSIONS

We have presented a scheme to automatically detect objects. Using several explorations of a camera in the same scene, we detect and model moved objects while reconstructing the environment. Experiments highlignt the performance of the method in a real case of localization in an unknown indoor environment. Our intention is to model a large environment subject to many changes, such as workshop factory or a shopping area, and to maintain a map of locations and frequently seen objects. Use cases for this work include providing context aware information for a user of a mobile device by providing location based information on the environment context and detecting objects of interest. We plan to improve the place recognition and metric localization results by taking into account non static hypothesis. Moreover, object and motion detection is useful for augmented reality applications.

## REFERENCES

Angeli, A. and Davison, A. (2010). Live feature clustering in video using appearance and 3D geometry. In *BMVC*.

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). SURF : Speeded Up Robust Features. *ECCV*.

Castle, R. O., Klein, G., and Murray, D. W. (2008). Video-rate localization in multiple maps for wearable augmented reality. In *SWC*.

Cummins, M. and Newman, P. (2009). Highly scalable appearance-only SLAM : Fab-map 2.0. In *RSS*.

Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Kim, K., Lepetit, V., and Woo, W. (2010). Keyframe-based Modeling and Tracking of Multiple 3D Objects. *ISMAR*.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2006). Monocular Vision Based SLAM for Mobile Robots. *ICPR*.

Schindler, K. and Suter, S. (2006). Two-view multibody structure-and-motion with outliers through model selection. *PAMI*.

Sivic, J. and Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. *ICCV*.

Toldo, R. and Fusiello, A. (2008). Robust multiple structures estimation with j-linkage. In *ECCV*.

Zuliani, M., Kenney, C. S., and Manjunath, B. S. (2005). The multiransac algorithm and its application to detect planar homographies. In *ICIP (3)*.