Paper accepted for publication in CVIU (Computer Vision and Image Understanding), july 2006

A Random Sampling Strategy For Piecewise Planar Scene Segmentation

Adrien Bartoli
Adrien.Bartoli@gmail.com

LASMEA – CNRS and Université Blaise Pascal
24, avenue des Landais
63177 Aubière cedex
France.

Abstract

We investigate the problem of automatically creating 3D models of man-made environments that we represent as collections of textured planes. A typical approach is to automatically reconstruct a sparse 3D model made of points, and to manually indicate their plane membership, as well as the delineation of the planes: this is the piecewise planar segmentation phase. Texture images are then extracted by merging perspectively corrected input images.

We propose an automatic approach to the piecewise planar segmentation phase, that detects the number of planes to approximate the scene surface to some extent, and the parameters of these planes, from a sparse 3D model made of points. Our segmentation method is inspired from the robust estimator RANSAC. It generates and scores plane hypotheses by random sampling of the 3D points. Our plane scoring function and our plane comparison function, required to prevent detecting the same plane twice, are designed to detect planes with large or small support. The plane scoring function recovers the plane delineation and quantifies the saliency of the plane hypothesis based on approximate photoconsistency.

We finally refine all the 3D model parameters, *i.e.* the planes and the points on these planes, as well as camera pose, by minimizing the reprojection error with respect to the measured image points, using bundle adjustment. The approach is validated on simulated and real data.

1 Introduction

The automatic 3D modeling of rigid scenes from images is one of the most challenging research areas in Computer Vision. This is a very important task that has many applications, for example virtual building editing in computer aided architecture and video augmentation in the film industry. There exist a wide variety of approaches to the image-based modeling problem, see *e.g.* [2, 6, 8, 11, 13, 16, 18, 19, 20, 21, 24, 25, 28, 30, 37]. The main difference between these methods is the representation of the scene they employ. For instance, Kutulakos and Seitz use voxels [19], Strecha *et al.* use a depth map [30], Gargallo and Sturm use multiple depth maps [13], Baillard and Zisserman use a set of planes [2], while Debevec *et al.* use a combination of those [6]. The most appropriate representation obviously depends on the type of scene that is to be reconstructed, and the foreseen application.

We consider the case of man-made environments. We choose to model those scenes by collections of textured planes. Figure 1 (a) shows an image of a building, overlaid by a piecewise planar segmentation, while figure 1 (b) shows a view of the corresponding recovered model. The piecewise planar model is

motivated by the following reasons. First, man-made environments are often composed of piecewise planar or nearly-planar primitives [2, 3, 6, 8] and are thus modeled as such to a reasonable degree of approximation. Second, this is a very constrained, compact representation that is thus very stable, and allows one to make the reconstruction process automatic. Third, this representation allows one to modify the reconstruction very easily, *i.e.* by adding, removing or augmenting objects.

Most of the existing systems are semi-automatic, based on a three-stage process, *e.g.* [6, 20, 31]. First, a sparse 3D reconstruction of features (points, lines, *etc.*) as well as cameras is performed automatically using Structure-from-Motion techniques, see *e.g.* [4, 26]. Two stages remain: choosing the scene model and estimating its parameters. The first stage is achieved by clustering reconstructed features into higher level geometric primitives such as cubes by *e.g.* marking edges in the input images. The second stage consists of optimizing the quality of the model parameters by *e.g.* minimizing the disparity between marked and predicted edges.

This approach has proven to give highly photorealistic results, but becomes effort-prone as the scene considered grows in complexity. In this paper, we devise a method that is automatic and takes into account

(a) (b)

Figure 1: Example of piecewise planar modeling: (a) shows an image overlaid with the automatically recovered piecewise planarity and (b) the model rendered from a different point of view. The reconstruction process for this example is detailled in §6.2.2.

both the photometric and point-based geometric information given by the input images.

In [24, 25], the scene surface is modeled as a set of triangles. The most likely triangulation with respect to the input images is computed using edge swaps from an initial solution obtained using a Delaunay triangulation. The process is however not guaranteed to converge to the global optimum. This means that some edges of the triangulation may cross the 'true' scene edges. Piecewise planarity is not taken into account, which reduces photorealism when rendering planar surfaces from oblique points of view.

Representing a scene as a collection of planes overcomes these problems. This reduces the complexity of the model computation as well as its rendering and yields more photorealistic view synthesis of planar and nearly-planar surfaces. The recovered structure and camera motion can be greatly enhanced by using the coplanarity information [3]. We follow the same three stages described above. Our modifications are two-fold. First, we fill the gap of interactivity by automatically computing a piecewise planar model of the scene – this is our main contribution. Second, we calculate the Maximum Likelihood estimate of the scene model parameters with respect to the observed image points.

Choosing the scene model is done by fitting planes to reconstructed 3D points. This is difficult because coplanar configurations that do not correspond to any world surface often arise in practice and have to be disambiguated using a physically meaningful criterion, *e.g.* based on photometric information. For that

purpose, we propose to use a random sampling technique to hypothesize multiple plane equations. We then select the most likely planes with respect to the input images while checking for them to be distinct from each other. Such an algorithm allows a point to lie on several planes.

We then refine the model parameters by nonlinearly minimizing the difference between the observed and predicted image points in a bundle adjustment manner. This is the Maximum Likelihood estimate. During this stage, we enforce the multi-coplanarity constraints of the 3D points using constrained bundle adjustment.

Our method depends on a point-based 3D reconstruction of the scene structure and cameras, either projective, affine and Euclidean. We note that Structure-from-Motion techniques that can be used to obtain such a reconstruction are well-established and effective, see *e.g.* [4, 26].

Organization of the paper. In §§2 and 3, we review existing work and give our scene model and notation. In §4, we formally derive our geometric and photometric criteria. We propose an algorithm to estimate the model of the scene in §5. Finally, we validate the method on simulated and real data in §6.

2 Existing Work

As briefly overviewed, there exists a huge body of work on 3D scene modeling from images, *e.g.* space carving [19], multiple view stereo [18] or consistent image triangulation [24, 25]. Below, we review existing work on piecewise planar scene segmentation.

Existing work can be divided into two subsets, whether the photometric information, *i.e.* dense greylevel data, is taken into account or not. In particular, purely geometric criteria are sometimes used, see *e.g.* the work by Alon and Sclaroff [1] and Berthilsson and Heyden [5] for segmenting a sparse point reconstruction, and by Faugeras and Lustman [9], Fornland and Schnörr [12], Sinclair and Blake [29] and Yang *et al.* [39], directly utilizing the image points without an explicit reconstruction. Such criteria do not allow to eliminate some coplanar point configurations that do not correspond to physical planes, or do not detect the planes that have a small support. Koch [17] proposes to segment a surface reconstructed using dense stereovision, based on local orientation. The drawback of this approach is the strong dependency on the reconstructed surface, that may be difficult to obtain reliably and automatically in the general case. Vestri and Devernay [37] use a robust method based on RANSAC to segment the scene. They detect a large number of planes by random sampling from a Digital Elevation Map, and merge the planes that are similar. Yang *et al.* [39] propose two methods for piecewise planar segmentation based on embedding the image point coordinates in a higher-dimensional real of complex plane. The advantage of their methods is that a closed-form solution is obtained. The drawback is that they are limited to two views.

Photometric information is taken into account by Baillard and Zisserman [2], Dick *et al.* [8] and Tarel and Vézien [32]. The method described in [2] follows an approach based on line matching to segment planes from aerial images of urban scenes. The results are very convincing, but the complexity is also very high. The method proposed by Dick *et al.* performs the segmentation using a collection of architectural primitives (pillars, doors, windows, etc.) with prior probabilities depending on the scene type. The approach given by Tarel and Vézien is specific to two views, and does not produce visually convincing models. Vidal and Ma [38] propose a solution to the segmentation of the optical flow between two images. A closed-form solution is obtained, similarly to the above mentioned work by Yang *et al.* [39], by embedding the measurements so that there is a single multibody motion model satisfied by all the pixels.

Most of these work, besides the methods of Fornland and Schnörr [12], Sinclair and Blake [29], Yang *et al.* [39] and Vidal and Ma [38], require a metric reconstruction of the scene, in other words, the knowledge or the on-line computation of camera internal calibration, while our method does not.

3 Notation and Scene Model

We denote a scene as $\mathcal{S} = \mathcal{S}^m \cup \mathcal{S}^p$. As said above, it is essential to make a clear distinction between the scene model $\mathcal{S}^m$ and its parameters $\mathcal{S}^p$. The scene model $\mathcal{S}^m = \{l, \theta_1^m, \ldots, \theta_l^m\}$ is made of discrete parameters, including the number of planes l and the plane models (the list of points supporting their polygonal delineation). The scene parameters $\mathcal{S}^p = \{\theta_1^p, \ldots, \theta_l^p, \mathcal{Q}, \mathcal{P}\}$ is made of continuous parameters, including the plane equations, the cloud of reconstructed 3D points $\mathcal{Q}$ and the projection operator $\mathcal{P}$, equivalent to reconstructed cameras.

Similarly, each scene plane π is modeled by $\theta = \theta^m \cup \theta^p$, *i.e.* it has a model $\theta^m = \{\mathcal{V}, \Pi, \Pi', \partial\}$ and an associated set of parameters $\theta^p = \{\boldsymbol{\pi}, \mathcal{T}\}$, both described in table 1. In more details, $\mathcal{V}$ designates the set of images where π is visible, Π and Π' are two sets of points lying on the plane with respect to two different criteria[1] described in §5 and ∂ is the polygonal delineation of the plane, expressed as a list of points.

model θ^m	$\mathcal{V}$	visibility images
	Π	individual geometric support
	Π'	global photometric support
	∂	polygonal delineation
parameters θ^p	$\boldsymbol{\pi}$	plane equation
	$\mathcal{T}$	texture map

Table 1: A plane π is modeled by a set $\theta = \theta^m \cup \theta^p$ where θ^m is the plane model and θ^p its parameters. Both the model and parameters are determined at the segmentation phase, while only the parameters are refined at the subsequent constrained bundle adjustment phase.

The set of n actual images is denoted $\mathcal{I}$. The projection operator $\mathcal{P}$ allows, given a scene $\mathcal{S}$, to predict its images $\hat{\mathcal{I}}$. Notation $\#(\Pi)$ and $\mathrm{conv}(\Pi)$ designate respectively the number of elements of a set and the convex hull of a set of points Π. The segmentation phase is the computation of the scene model $\mathcal{S}^m$ and some initialization for the scene parameters $\mathcal{S}^p$. Those are latter refined at the constrained bundle adjustment phase.

4 The Segmentation and Reconstruction Criteria

We describe the criteria we use at the segmentation phase and at the subsequent constrained bundle adjustment phase.

4.1 A Photometric Segmentation Criterion

We mention in the previous sections how important it is to use a criterion based on photometric quantities at the segmentation phase. The two prominent reasons to choose such a criterion over a purely geometric one is that *(1)* it is less sensitive to spurious planar configurations, *i.e.* points which are coplanar, but which do not lie on a real physical plane, and *(2)* it allows one to detect planes with small geometric support, *i.e.* planes onto which only a small number of points have been reconstructed.

We give a statistical derivation of the photometric segmentation criterion that we use. We assume each pixel luminance to be corrupted by an *i.i.d.* centred Gaussian noise. If we assume our prior probability on

[1] Π is the *individual geometric support, i.e.* the set of points that geometrically lie on plane π, while Π' is the *global photometric support, i.e.* the set of points defined such that the triangulated surface they induce is photoconsistent in the set of input images.

the scene to be uniform, computing the most likely scene parameters $\mathcal{S}^p$ with respect to the inputs images, given the scene model $\mathcal{S}^m$, is achieved by finding [8, 24]:

$$\arg\max_{\mathcal{S}^p} \Pr(\mathcal{I}|\mathcal{S}),$$

where $\Pr(\mathcal{I}|\mathcal{S})$ denotes the likelihood of the scene parameters $\mathcal{S}^p$ for the scene model $\mathcal{S}^m$ and the image set $\mathcal{I}$. Using the same reasoning as in [24], we obtain that the most likely parameters correspond to the minimum squared difference between the actual and predicted images, respectively written $\mathcal{I}$ and $\hat{\mathcal{I}}$.

The problem is then to find $\arg\min_{\mathcal{S}^p} \mathcal{C}(\mathcal{S})$, where the cost function $\mathcal{C}$ is defined by:

$$\mathcal{C}(\mathcal{S}) = ||\mathcal{I} - \hat{\mathcal{I}}||^2. \tag{1}$$

This can be decomposed over the set of planes as:

$$\mathcal{C}(\mathcal{S}) = \sum_{\pi \in \mathcal{S}} \mathcal{C}(\mathcal{S}_\pi),$$

where $\mathcal{S}_\pi$ designates the scene restricted to plane π. Our photometric criterion for scene segmentation is derived from the individual plane likelihood.

4.2 A Geometric Refinement Criterion

Bundle adjustment is known to be a reliable reconstruction procedure, to which geometric constraints such as colinearity or coplanarity in the reconstructed structure can be incorporated, and is known to usually improve the accuracy of the final result, see *e.g.* [3].

Once the structure has been segmented, *i.e.* the scene model $\mathcal{S}^m$ has been established, it is thus natural to launch a bundle adjustment procedure in order to finely tune the scene parameters $\mathcal{S}^p$ while enforcing the detected geometric constraints. In other words, we minimize the point-based reprojection error over the plane equations, the set of 3D points, constrained so they lie exactly onto the planes that they have been assigned to during the segmentation phase, and the cameras.

The optimization is performed using the Levenberg-Marquardt algorithm. The piecewise planarity constraints are enforced by a minimal parameterization that we described in a previous paper [3]. The idea is basically, for each point, to eliminate as many of its coordinates as the number of planes it lies on. For instance, a point being assigned to one plane is represented by two parameters only, giving its position on the plane. Its 3D position is then recovered from these two parameters and the plane equation, and used to compute the reprojection error.

5 Piecewise Planar Segmentation

We present a method to compute a piecewise planar model of the scene, *i.e.* to choose $\mathcal{S}^m$, using random sampling of the 3D points and a photometric evaluation of the plane hypotheses.

5.1 Overview of the Algorithm

The algorithm inputs are the actual images as well as a cloud of reconstructed 3D points and camera matrices or equivalently the projection operator.

The segmentation task consists of assigning each reconstructed point to a certain number of planes. This is equivalent to determining the number of planes l and the support of each plane (the support of a plane is the set of points lying on it, up to a certain tolerance). We note that the planes allow to constrain image point matching in a very reliable manner.

Piecewise planar segmentation is done by iteratively selecting the most likely plane using a random sampling technique. We use a multiple hypotheses version of RANSAC [10] modified in two ways. First, as described in [35], we maximize the likelihood of the plane instead of its support. Second, we devise a segmentation scheme inspired from [1] that allows for overlapping data segmentation, which is important in the piecewise planar segmentation of real scenes since a great number of planes are defined by points lying on several other planes.

5.2 Segmentation by Random Sampling

The basic idea of our method is to estimate the dominant plane using a robust estimator, and iterate the robust estimation while taking care that the newly estimated plane has not already been detected. We review different robust estimators below, and more specifically RANSAC, and propose modifications so that it applies to disjoint and overlapping data segmentation.

5.2.1 Robust Estimation Methods

Robust methods are used in Computer Vision to solve various problems such as the estimation of the epipolar geometry [4, 33, 40] or the trifocal tensor [4, 34] from point correspondences containing blunders. The most popular methods are the M-estimators, least median of squares [23] and RANSAC [10]. Such estimators allow one to estimate the parameters of a given model, even when the data contain outliers. The main difference between these estimators is the maximum outlier ratio they can handle.

Segmenting data using a robust estimator as depicted above implies that the robust estimator must handle high outlier ratios. Among the aforementioned estimators, only RANSAC deals with more than 50% of outliers. We therefore use it for segmentation. A description is given below.

RANSAC (Random Sample Concensus) is a robust hypothesize and verify algorithm that proceeds by repeatedly generating solutions estimated from minimal sets of points drawn from the data. This is the random sampling phase. It then tests each solution for support from the complete set of data. The parameters that maximize the support, *i.e.* the number of inliers, is then selected.

The number of iterations is chosen as a function of an a priori given outlier ratio, the probability of success, and the minimal number of data required to make an hypothesis. In practice, see [15], we do not specify the outlier ratio, and dynamically recompute the number of iterations each time a better hypothesis is drawn. A function indicating the consistency of a data point with some model parameters is needed to discriminate the inliers and the outliers, and thus to score the hypotheses.

In the case of plane estimation, three points are randomly selected and a plane equation $\boldsymbol{\pi}$ is computed while testing for degeneracy. The set of points Π that geometrically lie on $\boldsymbol{\pi}$, up to a predefined tolerance, is then computed image-based [1, 29] and the operation repeated. It is very important to compute the individual point on plane error in the images in order to handle uncalibrated, *i.e.* projective or affine, reconstructions. In practice, we use the distance between the image points and the point on the plane which reprojects as close as possible to the image points. The threshold for assigning a point to a plane is the noise level returned by the initial, unconstrained bundle adjustment procedure. The dominant plane is the one that maximizes $\#(\Pi)$. The number of iterations is computed as indicated in *e.g.* [10]. This is not a satisfying solution since planes with small support are typically missed, and spurious coplanar configurations are idendified as physically valid planes, as illustrated by the experimental results of §6.2.3.

As proposed in [35] in the case of image points, it is possible to use another cost function instead of $\#(\Pi)$ such as the likelihood $\Pr(\mathcal{S}_\pi|\mathcal{I})$ of the plane hypothesis. We use this likelihood to score the plane hypotheses. Computing this likelihood from the plane equation is described in §5.3.

Once the dominant plane has been estimated, points can be classified whether they lie, or not on the plane. In the former case, they belong to Π and in the latter, they are outliers for this plane.

5.2.2 Disjoint Data Segmentation

In the case of data segmentation, outliers are interpreted as points that do not lie on the dominant plane. Consequently, it is possible to recursively apply RANSAC on the set of outliers to perform data segmentation. At the end of the process, the points which have not been assigned to any of the detected planes are classified as outliers for the complete scene model. This way of segmenting has proven to be efficient in the case of motion segmentation, see the work by Demirdjian and Horaud [7, 33] or for special planar grouping, see the work by Schaffalitsky and Zisserman [27]. Indeed, in these cases the clusters are disjoint, *e.g.* a point will not, in the general case, satisfy two different rigid motions.

In the case of piecewise planar segmentation, see for example the work by Dick *et al.* [8], we can not make such an assumption of disjoint clusters since a point can lie on several planes. The above described method does only allow one to recover the main planes of the scene, and only simple coplanarity relationships (*i.e.* a point is in the support of a single plane at most). Figure 2 illustrates the possible problems and the difference between disjoint and overlapping data segmentation.

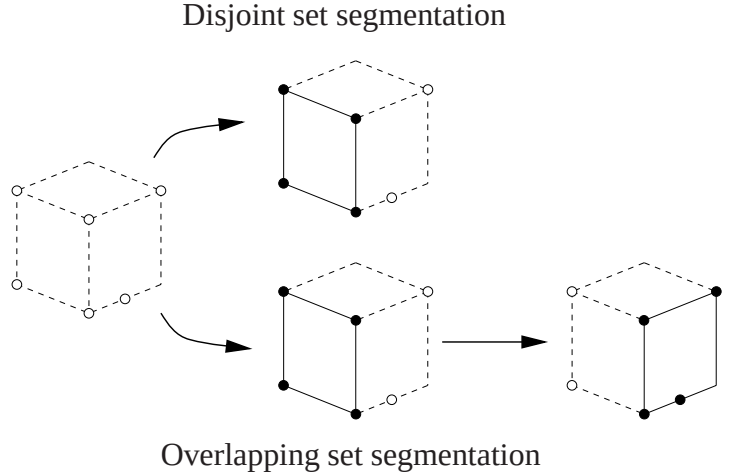


Figure 2: Illustration of the difference between algorithms segmenting in disjoint sets (top) and algorithms segmenting in overlapping sets (bottom). The disjoint set segmentation algorithms stop after one iteration since the unsegmented points (in white) do not contain enough cues to detect the second plane. The overlapping set segmentation algorithms find the second plane since they use both segmented and unsegmented points.

In the following paragraph, we present another scheme for data segmentation that allows for overlapping clusters.

5.2.3 Overlapping Data Segmentation

For overlapping data segmentation, we can not recursively apply RANSAC on the set of outliers for the reasons mentioned above. Instead, we modify it to formulate multiple hypotheses, as described in [1], so that a point can be selected as an inlier for more than one plane. The problem is that the same plane may be hypothesized more than once since inlying points are not removed from the point set before re-applying RANSAC. This can be solved by constraining the random sampling phase to select planes that have not been detected so far. We thus have to define a distance measure between two planes π_i and π_j given their support, Π_i and Π_j. This measure must be physically meaningful, *e.g.* comparing the equations of the two planes will not give satisfactory results since it is an algebraic measure. Comparing the normals of the planes is a geometric measure, but reflects only partially the closeness of the planes since parallel planes would not be disambiguated.

We rather compare the sets of inliers Π_i and Π_j of the two planes. The two planes are considered identical if $\mathcal{D}(i,j) > \gamma$ where:

$$\mathcal{D}(i,j) = \frac{2 \cdot \#(\Pi_i \cap \Pi_j)}{\#(\Pi_i) + \#(\Pi_j)}.$$

In other words, $\mathcal{D}(i,j)$ is the ratio of the number of common inliers to the total number of inliers. For example, if the two planes have exactly the same support, $\mathcal{D}(i,j) = 1$, and if they do not have any common point, then $\mathcal{D}(i,j) = 0$. We choose the predefined rate γ as $\gamma = 0.5$ in our experiments. Another solution for comparing the two planes would be to assess each support with the other plane equation.

As is shown by the discussion below, there is a quite wide range of admissible values for γ giving the same segmentation result. The role of γ is to prevent the Overlapping Data Segmentation algorithms to detect the same plane more than once. If γ is chosen too tight, *i.e.* close to 0, then once a plane is detected, all the other planes having points in common with the former one are rejected. If γ is chosen too loose, *i.e.* close to 1, then the same physical plane may be detected more than once, with a slightly different support. For two distinct planes, we can interpret $\mathcal{D}$ as the ratio of the number of intersection points to the total number of points on the two planes to be compared. We must choose γ loose enough so that a high number of points at the intersection of the two planes is tolerated. Indeed, for low textured planes, *i.e.* planes with a homogeneous texture, most interest points are likely to lie at the plane boundary, and thus also lie on another plane. In practice choosing γ between 0.4 and 0.7 gives the results obtained in the experiments reported in §6. We therefore recommend to choose $\gamma = 0.5$.

Note that this approach is different from the work by Vestri and Devernay [37] since they are detecting a large number of planes, possibly very close to each other, and apply a plane merging step based on the geometric support of each plane.

We have shown how to formulate plane equation hypotheses with respect to possibly overlapping clusters. In the following section, we examine how to estimate the likelihood of an hypothesis.

5.3 Retrieving Plane Parameters

So far, we have an hypothesis of a plane equation $\boldsymbol{\pi}$ as well as its geometric support Π. In order to evaluate the likelihood of this hypothesis, we have to determine its model θ^m and parameters θ^p described in table 1.

A criterion which is often used to score a plane hypothesis is the number of points lying on the plane. This is the criterion used in RANSAC. One problem with this criterion is that some points can form a false planar configuration. This criterion is thus not appropriate to score the planes. Figure 3 illustrates this problem, and compares this criterion with a photometric one, which performs better since those false coplanar configurations do typically not induce photoconsistent surfaces.

We evaluate the saliency of a plane hypothesis using a four-step method, illustated on figure 4. The goal of the method is, assuming that the plane surface is represented by a triangulated mesh, to count the number of triangles forming this mesh. This number is used as a score for ranking plane hypotheses. Counting the number of triangles requires to first compute the delineation of the plane – this is done by testing the photoconsistency of a 'larger' mesh, induced by the points in Π, *i.e.* the set of points which are geometrically consistent with the plane. In more details, the algorithm takes the following steps:

- **Step 1: Computing the visibility images $\mathcal{V}$.** Denote Π' the set of points determined to lie on the plane after the delineation process explained below has been conducted, *i.e.* the set of points forming the photometrically consistent surface. Obviously, we have $\Pi' \subset \Pi$ since any point in Π' defining the photoconsistent surface is also in Π in that it lies on the plane in the geometric sense. Consequently, we use for $\mathcal{V}$ the images where all points of Π are visible.

- **Step 2: Computing the polygonal delineation ∂.** The process of determining the delineation, *i.e.* the polygonal boundary of the plane, in other words the set of points in Π which also lie on the boundary of the plane, is illustrated on figure 4 and explained below.

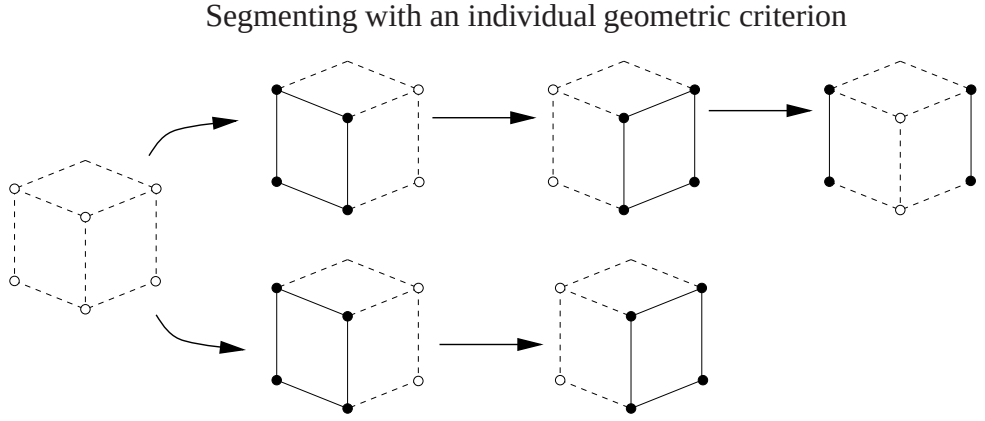


Figure 3: Illustration of the difference of segmentation between an individual geometric criterion (top) and a dense photometric criterion (bottom). The third plane detected with the individual geometric criterion does not correspond to a real plane. It is not detected when the dense photometric criterion is used.

It is straightforward to show that:

$$(\Pi' \subset \Pi) \Rightarrow (\partial \subset \mathrm{conv}(\Pi)).$$

Therefore, we compute $\mathrm{conv}(\Pi)$ and iteratively 'cut' it, by removing non-photoconsistent triangles, to reach ∂.

For that purpose, we first warp each image of the plane with delineation $\mathrm{conv}(\Pi)$, so that image points are aligned, and perform a Delaunay triangulation. The warping is done using inter-image plane homographies and bicubic interpolation. Removing triangles from $\mathrm{conv}(\Pi)$ leads to a slightly 'smaller' delineation ∂ than the ideal one. If we assume that the triangulation does not cross the edges of the polygonal boundary, then ∂ is in theory the ideal one. In any case, there is no chance that the final delineation ∂ crosses the edges of the ideal delineation.

Each triangle is then registered using an r-consistency check, following the idea in [18]. This check is a kind of multiple view correlation measure which allows for some error in the alignment, *i.e.* in the geometrical position and orientation of the surface. The *dispersion radius* r, expressed in pixels, controls the maximum image error of a reprojected point, allowing for non-perfectly planar surfaces and noisy images to be handled. The r-consistency check determines whether aligned triangles arising from different images represent the projection of the same planar surface, up to a tolerance modeled by r. The check fails if the greylevels of all pixels of all triangles are not shared by the other triangles in a disk of radius r around the pixel considered.

Such an algorithm allows for non convex and non-simply connected plane delineation recovery, provided an appropriate data structure in the latter case. The set of points that lie inside ∂ is the global photometric support Π' of the plane. Finally, the texture $\mathcal{T}$ is evaluated by warping each image of the plane onto a virtual fronto-parallel plane [22] and by taking the average of these values. The median can also be used to discard possible artefacts due to *e.g.* specularities.

The whole procedure is summarized in table 2. Mathematical details are given below.

Image warping. All images in $\mathcal{V}$ are warped onto a reference one with index i_0. This is done by using the homographic warps $\mathcal{H}_{i_0 i,}$ induced by the reprojected points $\hat{\mathbf{q}}_{i,j} = \mathcal{P}_i(\mathbf{Q}_j)$. The warped images $\tilde{\mathcal{I}}_i$ are obtained as:

$$\tilde{\mathcal{I}}_i[\mathbf{q}] \leftarrow \mathcal{I}_i[\mathcal{H}_{i_0,i}(\mathbf{q})],$$

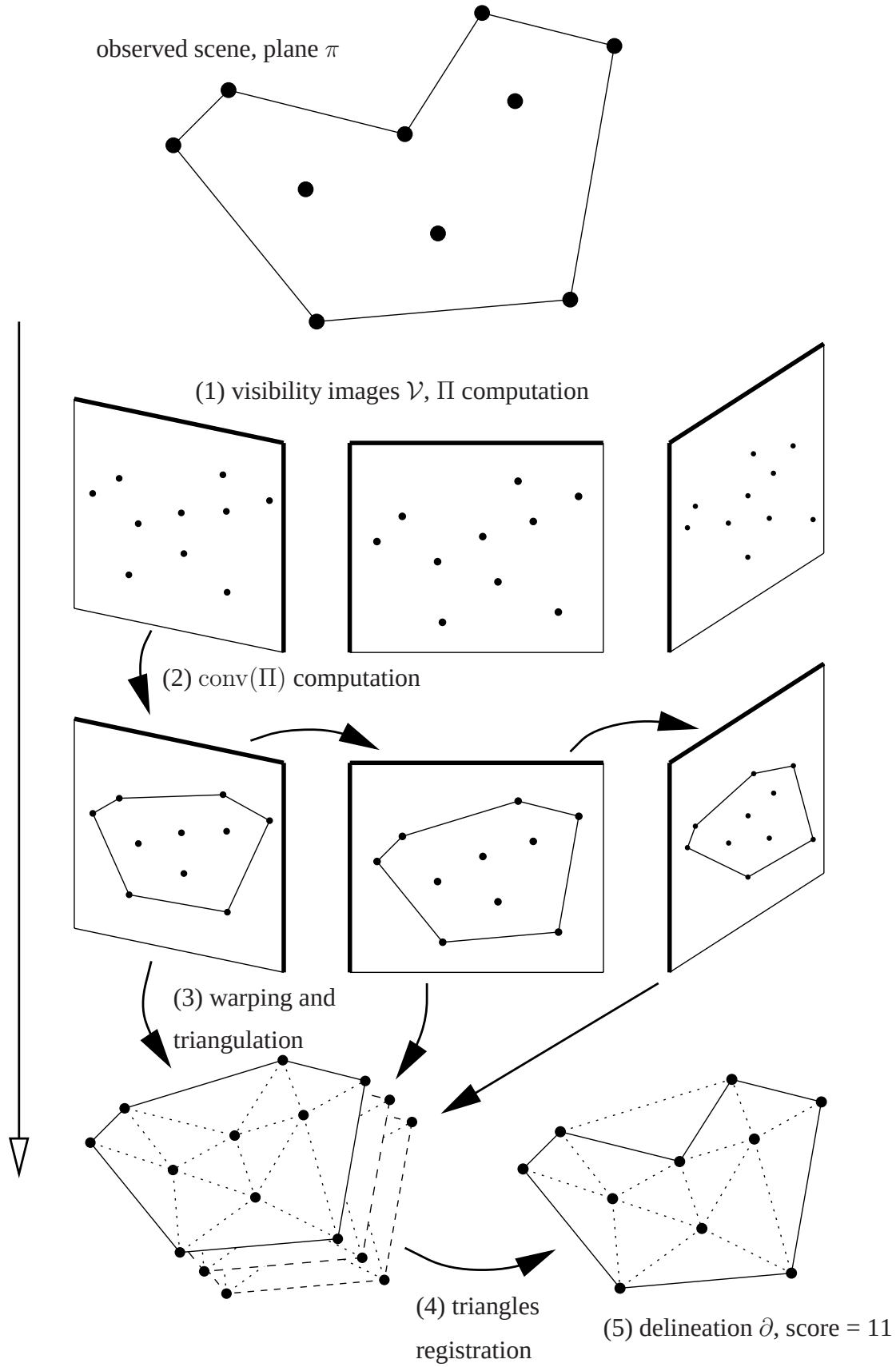


Figure 4: Evaluating the delineation ∂ of a plane and its score based on a dense photometric criterion. (1) The images $\mathcal{V}$ where all points lying on the plane are visible are determined. (2) and (3) The convex hull of those points is then triangulated. (4) Each triangle is kept or removed by testing its photoconsistency in the visibility images. (5) The final score is given by counting the number of remaining triangles.

OBJECTIVE

Compute a photometric score for the plane of equation $\boldsymbol{\pi}$ with geometric support Π through the recovery of the plane model θ^m and parameters θ^p. Other inputs are the cloud of reconstructed 3D points $\mathbf{Q}_j$, cameras $\mathcal{P}_i$, point visibility indicators $v_{ij} \in \{\text{false}, \text{true}\}$ and the dispersion radius r (in pixels).

ALGORITHM

- **Step 1: Compute the visibility images $\mathcal{V}$.**

$$\mathcal{V} \leftarrow \{i \mid \forall j \in \Pi,\ v_{ij} \text{ is true}\}$$

- **Step 2: Compute the polygonal delineation ∂.**
  - Reproject the geometric support:

$$\text{For } i \in \mathcal{V} \text{ and } j \in \Pi, \text{ set } \hat{\mathbf{q}}_{i,j} \leftarrow \mathcal{P}_i(\mathbf{Q}_j).$$

  - Choose the reference image i_0:

$$i_0 \leftarrow \arg\max_{i \in \mathcal{V}} \left(\text{area}(\text{conv}(\hat{\mathbf{q}}_{i,j},\ j \in \Pi))\right).$$

  - Warp all images $\mathcal{I}_i$ for $i \in \mathcal{V}$ and $i \neq i_0$ to the reference one:

$$\tilde{\mathcal{I}}_i[\mathbf{q}] \leftarrow \mathcal{I}_i[\mathcal{H}_{i_0,i}(\mathbf{q})] \qquad \text{for } \mathbf{q} \in \text{conv}(\hat{\mathbf{q}}_{i_0,j},\ j \in \Pi),$$

    where $\mathcal{H}_{i_0,i}$ is the homographic warp induced by the point correspondences $\hat{\mathbf{q}}_{i_0,j} \leftrightarrow \hat{\mathbf{q}}_{i,j}$.
  - Compute the Delaunay triangulation $\mathcal{R}$ of $\text{conv}(\hat{\mathbf{q}}_{i_0,j},\ j \in \Pi)$.
  - Remove each triangle $\mathcal{A} \in \mathcal{R}$ if the r-consistency check fails, *i.e.* if:

$$\kappa(\mathcal{A}) > \varepsilon,$$

    where the r-consistency $\kappa(\mathcal{A})$ is given by:

$$\kappa^2(\mathcal{A}) = \frac{1}{\#(\mathcal{A})(\#(\mathcal{V})-1)} \sum_{\mathbf{q} \in \mathcal{A}} \left(\sum_{i \in \mathcal{V}, i \neq i_0} \left(\min_{\mathbf{q}', \|\mathbf{q}'-\mathbf{q}\| \leq r} |\mathcal{I}_{i_0}[\mathbf{q}] - \tilde{\mathcal{I}}_i[\mathbf{q}']| \right)^2 \right).$$

    and the threshold ε is a parameter (see the main text).
  - Extract to Π' the set of points supported by at least one triangle in $\mathcal{R}$: this is the photometric plane support.
  - Extract the polygonal delineation ∂ from $\mathcal{R}$.
  - Extract the texture map $\mathcal{T}$ of the plane as:

$$\mathcal{T}[\mathbf{q}] \leftarrow \underset{i \in \mathcal{V}}{\text{mean}}(\tilde{\mathcal{I}}_i[\mathbf{q}]) \qquad \text{for } \mathbf{q} \in \mathcal{R}.$$

Table 2: The photometric score computation for a plane equation hypothesis $\boldsymbol{\pi}$ with geometric support Π.

where the pixel $\mathbf{q}$ lies in the convex hull of the reprojected points, *i.e.* $\mathbf{q} \in \mathrm{conv}(\hat{\mathbf{q}}_{i_0,j},\ j \in \Pi)$. The reference image is chosen among the images in $\mathcal{V}$ as the one for which the area of the convex hull is maximum, so as to select the one with as less perspective distortion as possible.

r-consistency check. The r-consistency check we use is strongly inspired by the one proposed in [18]. The idea is, for all pixel of a triangle $\mathcal{A}$ in the reference image, to look at the intensities around the corresponding pixels in the warped images, and check if these intensities are consistent. In practice, the candidate triangles are obtained by computing a Delaunay triangulation $\mathcal{R}$ of the points in Π. Each triangle $\mathcal{A} \in \mathcal{R}$ is kept if its r-consistency $\kappa(\mathcal{A})$ is below some threshold ε. How to compute $\kappa(\mathcal{A})$ is explained below.

Ideally, *i.e.* in the absence of noise and with perfect image warping, the photoconsistency score vanishes since:

$$\mathcal{I}_{i_0}[\mathbf{q}] = \tilde{\mathcal{I}}_i[\mathbf{q}] \qquad \forall \mathbf{q} \in \mathcal{A},\ \forall i \in \mathcal{V}.$$

Classically, photoconsistency is thus a measure of deviation from this perfect equality, *e.g.* the Root Mean of Squares $\rho(\mathcal{A})$ over the photometric error of the reference to the warped images:

$$\rho^2(\mathcal{A}) = \frac{1}{\#(\mathcal{A})(\#(\mathcal{V})-1)} \sum_{\mathbf{q}\in\mathcal{A}} \left(\sum_{i\in\mathcal{V}, i\neq i_0} \left(\mathcal{I}_{i_0}[\mathbf{q}] - \tilde{\mathcal{I}}_i[\mathbf{q}] \right)^2 \right).$$

This measure could be used for our procedure if the images were perfectly warped, *i.e.* in the absence of geometric misalignment. This is however not the case in practice since the 3D model is reconstructed from the actual images, and is thus contaminated by noise, causing the warped images to be misaligned.

We thus use the r-consistency trick of Kutulakos [18], consisting, for each pixel $\mathbf{q}$, to search the intensity of the other pixels around its position in each of the warped images to find the best match, giving:

$$\kappa^2(\mathcal{A}) = \frac{1}{\#(\mathcal{A})(\#(\mathcal{V})-1)} \sum_{\mathbf{q}\in\mathcal{A}} \left(\sum_{i\in\mathcal{V}, i\neq i_0} \left(\min_{\mathbf{q}', \|\mathbf{q}'-\mathbf{q}\|\leq r} |\mathcal{I}_{i_0}[\mathbf{q}] - \tilde{\mathcal{I}}_i[\mathbf{q}']| \right)^2 \right).$$

The threshold ε is chosen between 5% and 10% of the maximum intensity value, since it must be greater than the noise on the pixel intensities. We observed similar results for various thresholds in this range in our experiments.

6 Experimental Results

We present the results we obtained by applying the algorithm described in this paper on both simulated data and real images.

6.1 Simulated Data

We compare our method to existing ones, notably to those consisting of using a disjoint data segmentation [8] or a purely geometric criterion [1, 29, 39].

The test bench consists of a cube of one meter side length observed by a set of cameras. Points are generated on the cube, possibly offset from their planes in order to simulate non-perfect coplanarity, and projected onto the images. Each face of the cube is texture mapped using random values. The texture is also projected onto the images and normalized to lie between 0 and 1. Up to 30, 5 and 1 points are generated on respectively each face, edge and vertex of the cube. Two cameras with a focal length of 1000 (expressed in number of pixels) and a 1 meter baseline are situated at a distance of 10 meters from the cube, such that 3 of its faces are observed. The intrinsic parameters are not supposed to be known which yields projective reconstructions.

In the sequel, we vary independently each of these parameters to compare the different approaches under various conditions. We measure the quality of segmentation using the absolute difference between the number of recovered planes and the number of simulated planes, *i.e.* 3. We use the median value over 100 trials. Matched image points are supposed to be known up to a Gaussian centred noise of 1 pixel. The cloud of 3D points as well as camera matrices are obtained by minimizing reprojection errors in a bundle adjustment manner, using the ground truth as a starting point. The estimators compared are:

- DDS-GEOM: Uses disjoint data segmentation and a purely geometric criterion, *i.e.* $\#(\Pi)$, see §5.2.
- DDS-PHOTO: *idem* but uses the photometric likelihood $\Pr(\mathcal{S}_\pi|\mathcal{I})$ criterion given in §5.3.
- ODS-GEOM: Uses overlapping data segmentation described in §5.2 and a purely geometric criterion.
- ODS-PHOTO: Uses both the overlapping data segmentation and the photometric likelihood criterion.

Let us describe the different experimental situations when varying a scene parameter and the simulation results we have obtained:

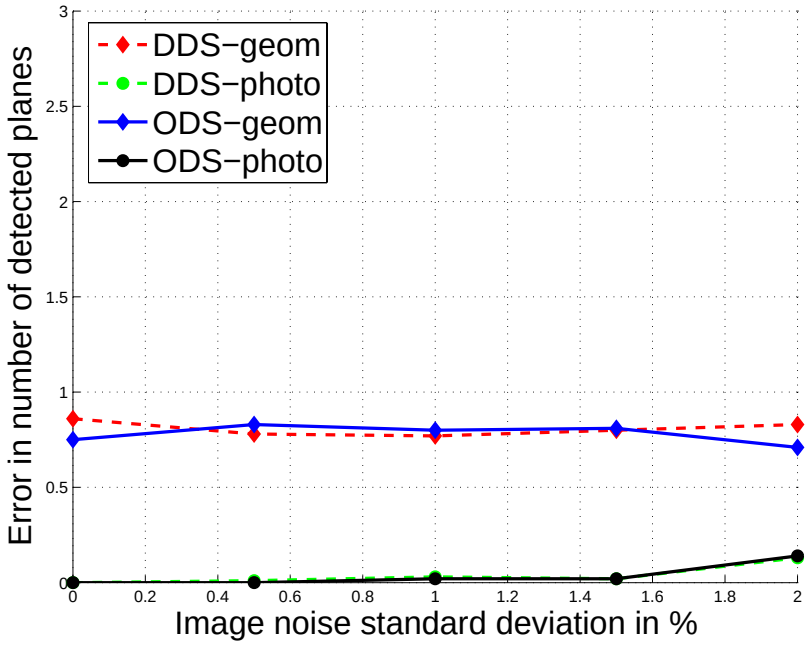


Figure 5: Comparison of the error in the number of detected planes for varying image noise. Note that DDS-PHOTO and ODS-PHOTO give indistinguishable results.

- *Added image noise*, figure 5: A Gaussian centred noise with standard deviation between 0 and 0.02 (*i.e.* 2% of the maximum) is added to the greylevel of each pixel.
- *Number of points*, figure 6: The number of points is varied from 0 to 40 for each face of the cube. We also project each vertex but no points are simulated on the edges.
- *Plane unflatness*, figure 7: 3D points are offset from the plane they lie on by distances drawn from a normal distribution with standard deviation between 0 and 0.1 meters.

We observe that in the general case, methods *-PHOTO, based on photometric information perform better than methods *-GEOM, based on a purely geometric criterion. In particular, the method ODS-PHOTO performs better than the others in all cases.

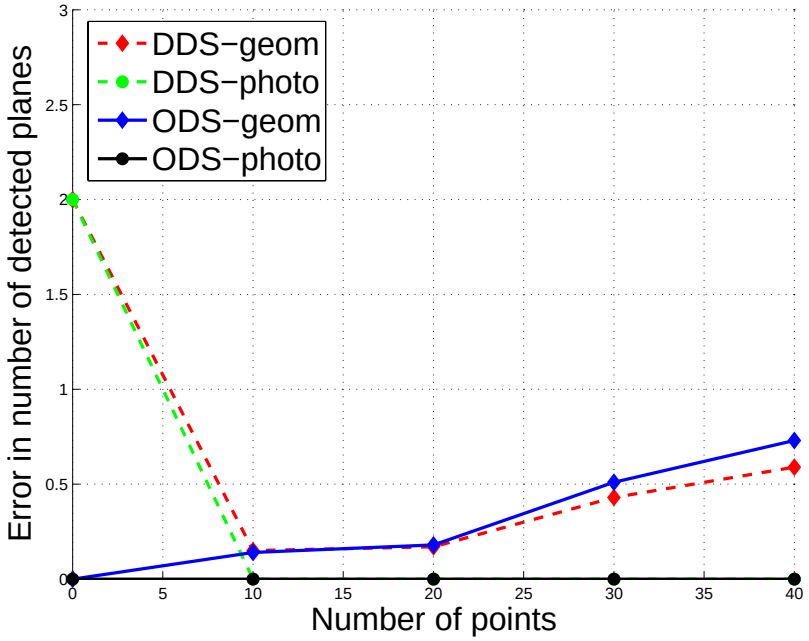


Figure 6: Comparison of the error in the number of detected planes for varying number of points.

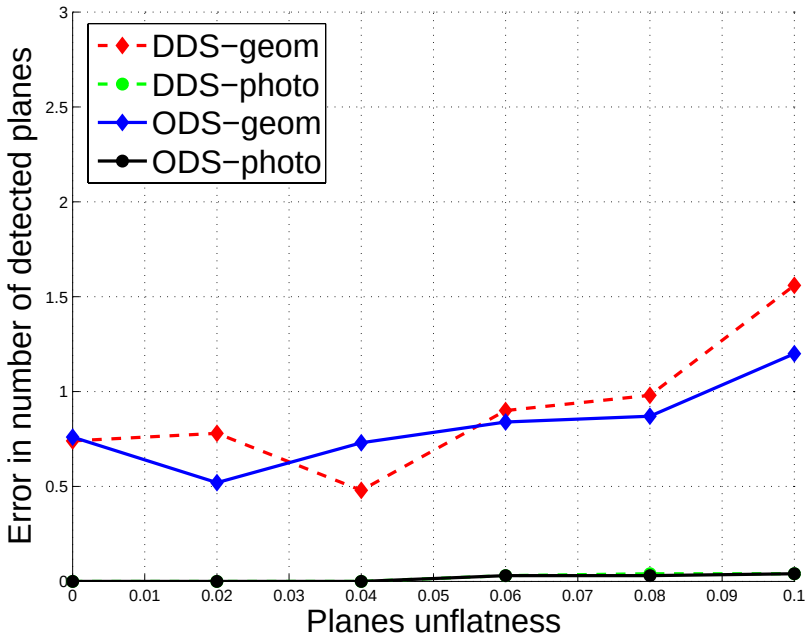


Figure 7: Comparison of the error in the number of detected planes for varying plane flatness. Note that DDS-PHOTO and ODS-PHOTO give indistinguishable results.

In the case of added image noise, we observe that methods *-GEOM perform worse than methods *-PHOTO. This can be explained by the fact that purely geometric criteria can not differentiate real planes from coplanar configurations that do not correspond to any simulated plane and as the number of simulated points is relatively low, such configurations often appear.

When varying the number of points, we observe that for low numbers (only the vertices of the cubes are used at the begining of the graph), methods DDS-* perform worse. This is due to the fact that planes can be detected only when taking into account multi-coplanarity, *i.e.* the fact that a point can lie on several planes. DDS-* methods eliminate points once they have been selected on one of the planes. The other random sampling trials are thus not able to detect the other planes with the remaining data. When the number of points increases, methods *-GEOM perform worse for the same reason as that given in the case of image noise. Handling planes supported by a low number of points is important since such planes are frequently encountered in practice.

In the case of increasing plane unflatness, methods *-GEOM perform worse since the probability of coplanar configurations that do not correspond to any simulated plane increases. The performance of methods *-PHOTO is not dramatically affected.

6.2 Real Images

We present the modeling results we obtained on three sets of images. In the first dataset, the planes are very well separated, and were very easy to extract, while in the second dataset, they were more difficult to obtain, since some of them are thin and have a small support. For both datasets, the camera were internally off-line calibrated using a calibration pattern. However, it is clear that this information is only exploited for visualizing the results and does not affect the behaviour of the scene modeling. The third dataset is more challenging since the scene surface is not completely piecewise planar. For this dataset, Euclidean reconstruction was obtained using Structure-from-Motion including self-calibration.

6.2.1 The BOX Dataset

The two images of the BOX dataset are shown on figure 8. Table 3 gives the reprojection errors we obtained for the different steps of the reconstruction process.

Figure 8: The two images of the BOX dataset.

Reconstructing points and the cameras. We detected interest points using the Harris and Stephen detector [14], shown on figure 9 (a) and (b). We matched them between the two views while robustly estimating

Approach	Step	Reprojection error (pixels)	# iterations
Unconstrained	Initialization	1.68	-
	Bundle Adjustment	0.15	5
Constrained	Initialization	0.85	-
	Bundle Adjustment	0.36	3

Table 3: Reprojection errors obtained at different steps, and number of iterations for the bundle adjustments.

the epipolar geometry using the method by Zhang [40]. We then used bundle adjustment, see *e.g.* [36] to reconstruct the points and the cameras. The metric 3D model we obtained is shown on figure 9 (c) and (d). The reprojection errors were 1.68 pixels and 0.15 pixels prior and after bundle adjustment.

Piecewise planar segmentation. Our algorithm successively detected three planes, shown on figure 10. The three main planes of the scene were detected. Note that the points on edges, *i.e.* at the meet of two planes, are correctly associated with those planes, and that the corner of the box is affected to the three planes. We used a dispersion radius $r = 2$ pixels The segmentation was stopped when the algorithm was not able to detect more than two photoconsistent triangles. Note that the dispersion radius has a small influence on the segmentation result since the 3 planes are very well distinct.

Joint constrained reconstruction of planes, points and cameras. The cloud of 3D points was corrected in order to satisfy the geometric multi-coplanarity constraints that were detected. The reprojection error then grew to 0.85 pixels. In order to get an optimal positioning, we minimized the reprojection error under the multi-coplanarity constraints using the algorithm proposed in [3], a bundle adjustment incorporating the constraints. The final reprojection error we obtained was 0.36 pixels, which is very reasonnable. The constrained reconstruction is shown on figure 11. Texture images were extracted and perspectively corrected. The textured model is shown on figure 12. More precisly, a texture image was computed for each plane by warping each image onto a fronto-parallel plane, and by computing the mean color for each pixel. This is justified by the Lambertian surface hypothesis.

Note that the fact that the reprojection error is higher for the constrained bundle adjustment than for the unconstrained one was expected since they both fit the same data, but the former one tunes less parameters than the latter one (or, equivalently, is subject to more constraints). Constrained bundle adjustment is known to increase 3D accuracy, see *e.g.* [3].

6.2.2 The BUILDING Dataset

We used six images of a building captured by a digital camera. Some images are shown on figure 13.

We insist on the difficulty to automatically build a piecewise planar model from these images. Indeed, windows undergo a significant parallax form their facade and can not be completely modeled using the given points which lie only on the facades. Moreover, their appearance changes across viewpoints due to specular effects. Several planes are also very thin.

We describe the different steps followed to perform a complete reconstruction, from the images to the 3D textured model.

Scene structure and camera motion initialization. Due to the wide baseline between the different images, points correspondences were indicated by hand. They are shown on figure 14. We performed a partial reconstruction from two images and incrementally add the others in turn to obtain the complete structure and motion. The reprojection error was 3.82 pixels. We then run a bundle adjustment to minimize the

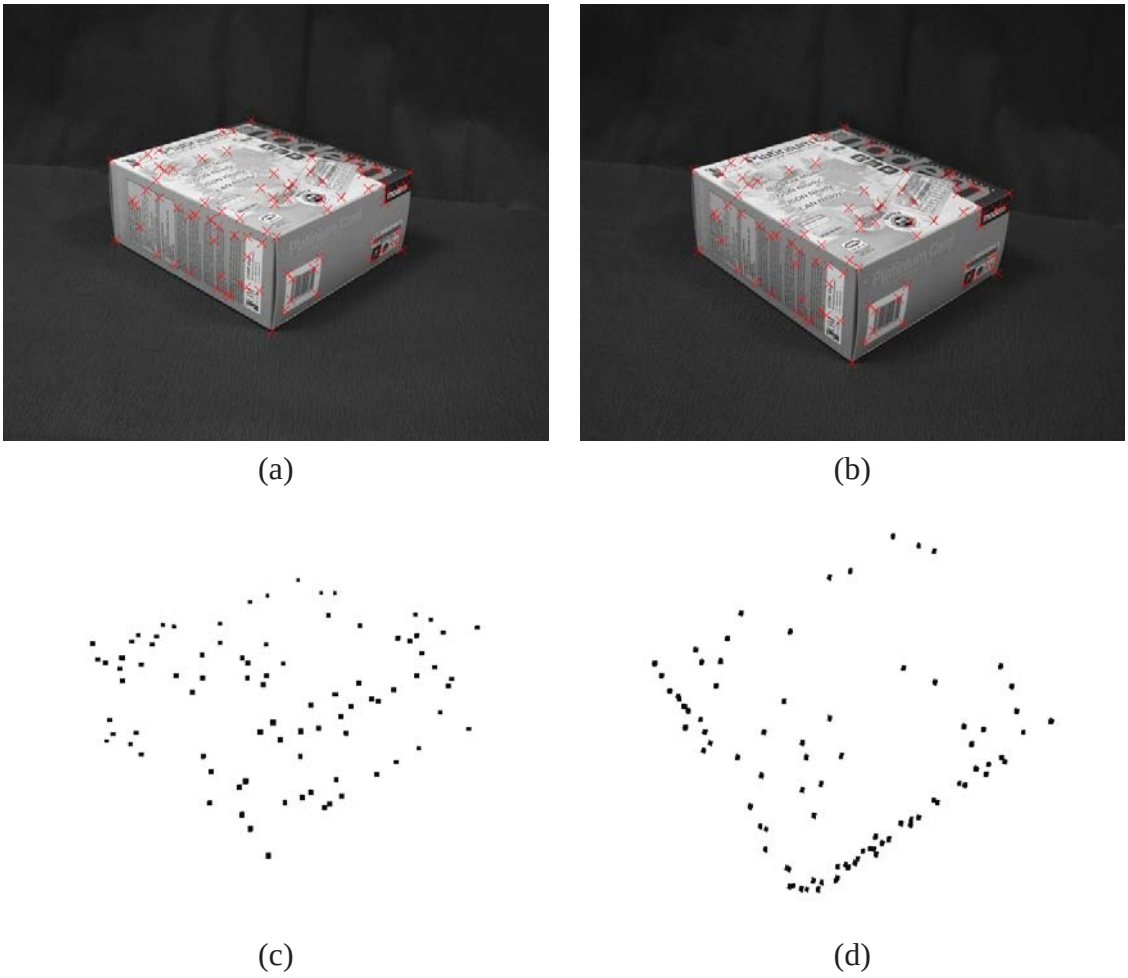
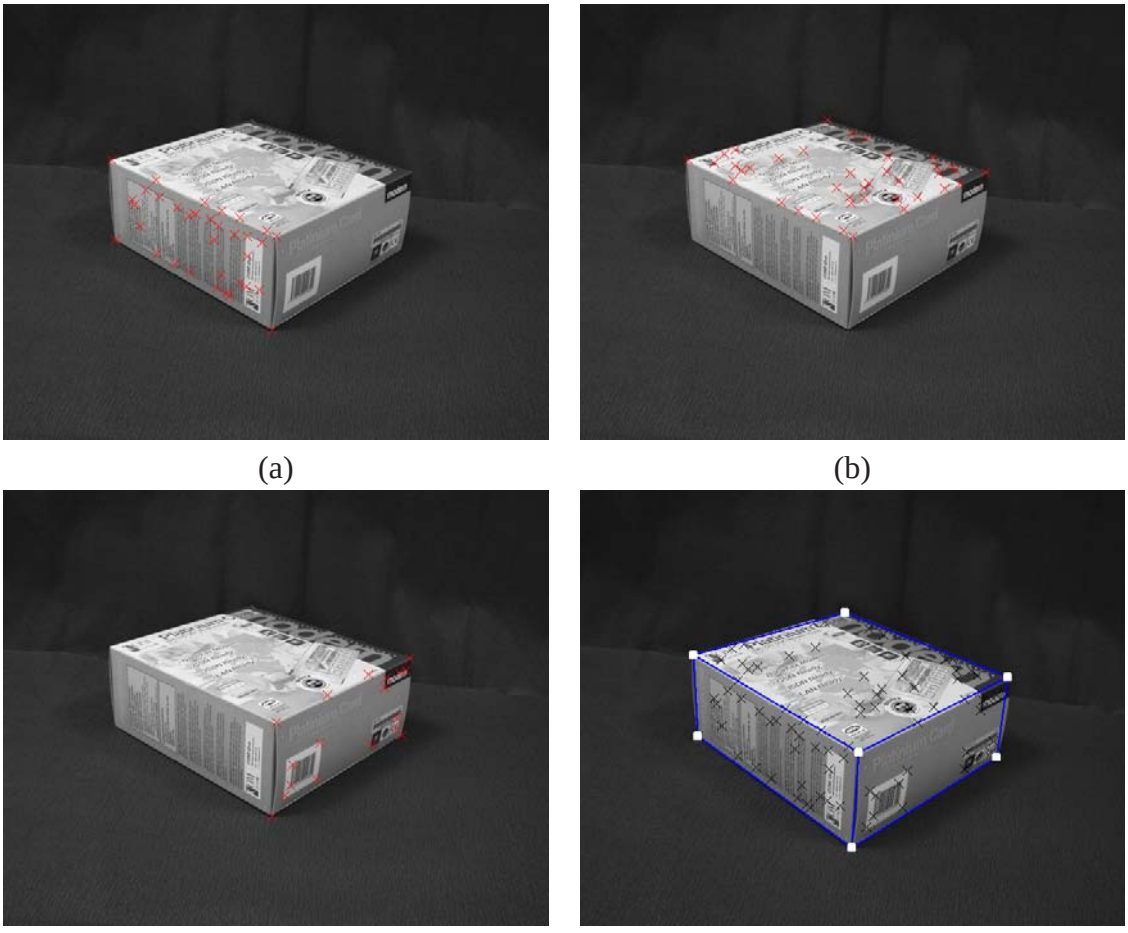


Figure 9: The 90 interest points used in the BOX dataset (a) and (b), and the 3D reconstruction we obtained, oblique view (c) and top view (d). We remark, on (d) particularly, that point coplanarity is very approximative.



(a) (b)

(c) (d)

Figure 10: Automatic piecewise planar segmentation obtained on the BOX reconstruction: (a), (b) and (c) show the points lying on the three detected planes, and (d) shows the reprojected 3D model in the first image with the plane delineations.

(a) (b)

Figure 11: Piecewise planar model obtained for the BOX dataset. (a) shows the sparse point reconstruction constrained by the planes, and (b) shows the model consisting of three planes and two cameras. We observe on (a) that the reconstructed points are perfectly coplanar after constrained bundle adjustment.

Figure 12: Some renderings of the textured model obtained for the BOX dataset.

Figure 13: Three out of the six images of the BUILDING dataset. Note the significant parallax of windows relatively to the wall.

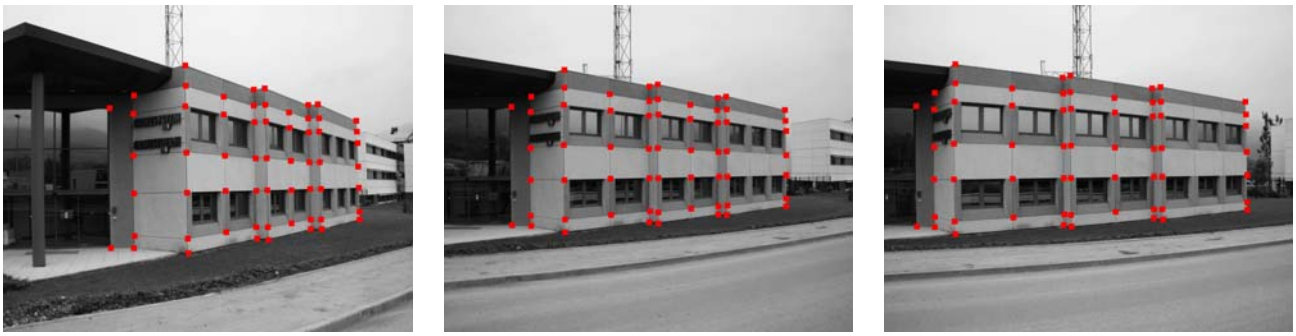


Figure 14: Three out of the six images of the BUILDING dataset overlaid with corresponding point features.

reprojection error, which was reduced to 1.03 pixels. Note that a radial distortion term is included in the minimization. The reconstructed 3D point model is shown on figure 15.

Figure 15: Some views of the points reconstructed using unconstrained bundle adjustment for the BUILDING dataset.

Piecewise planarity segmentation. We used the algorithm described in this paper to recover the piecewise planarity of the scene. We tried a large dispersion radius [18], see §5.3, of 2% of the image size, *i.e.* about 10 pixels. This tolerated too high errors and resulted in merging planes that could have been modeled separately, figure 16 (c). The obtained model is thus very imprecise.

For a low dispersion radius, 0.2% of the image size, which roughly corresponds to 1 pixel, the recovered piecewise planarity is incomplete, due to the parallax of windows relatively to the wall, figure 16 (a). The obtained model has holes and does not look very convincing.

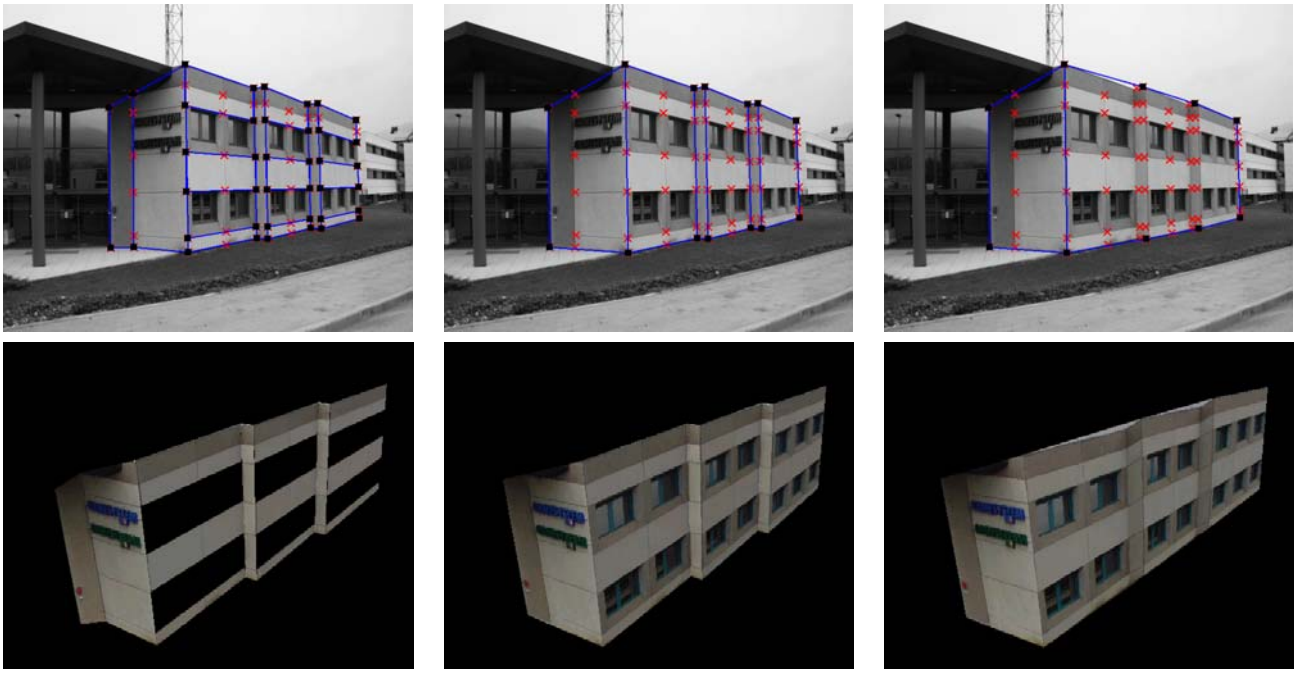
An in-between dispersion radius of 1% of the image size, *i.e.* about 5 pixels, gave a complete and photorealistic-looking piecewise planar model, figure 16 (b).

The results we obtained are therefore satisfactory. The method tolerates approximately planar surfaces and is able to detect and model small planes, provided that the dispersion radius is appropriately chosen.

Joint constrained reconstruction of planes, points and cameras. We used the technique in [3] to minimally parameterize the structure while enforcing previously recovered piecewise planarity. The optimization process is conducted as described in §4 using the scene model $\mathcal{S}^m$ obtained with r=5 pixels (figure 16 (b)). The sparse feature reconstruction is exactly piecewise planar after constrained optimization. Figure 17 shows texture-mapped renderings of the recovered model from various points of view different from the original ones. The texture maps for windows were extracted from one image each, to avoid the blur effect caused by the parallax. Another possibility to render such approximate planes is to use the View-Dependent Texture-Mapping described in [6].

Quality assessment. We have performed several measures on the structure obtained before and after the constrained bundle adjustment for the piecewise planarity corresponding to a 1 pixel dispersion radius (figure 16 (a)). Two kinds of quantity are significant: length ratios and angles. Table 4 shows measures of such quantities. In this table, σ_1 and σ_2 are the variances of the length of respectively the 6 vertical edges and the 6 horizontal edges of equal length, whereas μ is the mean of $\left|1 - \frac{2\alpha_i}{\pi}\right|$ where the α_i are the measures of right angles. The measured quantities are illustrated on figure 18.

Table 5 shows similar measures when varying the dispersion radius. As one could expected, these show that the lower the dispersion radius is, the better the reconstruction is. This is due to the fact that



(a) – $r \approx 1$ pixel (b) – $r \approx 5$ pixels (c) – $r \approx 10$ pixels

Figure 16: Recovered piecewise planarity and rendering of the corresponding model using successively 0.2, 1 and 2% of the image size (roughly 1, 5 and 10 pixels) for the dispersion radius r. In the first case, (a), planes are accurately modeled, see *e.g.* the two left planes. The holes correspond to windows that can not be entirely modeled and are not approximated by a plane when using such a low tolerance. For the second case, (b), the tolerance is sufficiently high, so that windows can be approximated by a plane. Note that the two left planes of (a) are now merged. In the last case, (c), two planes suffice to completely approximate the scene surface, which yields several artefacts on the rendering.

Figure 17: The texture-mapped model obtained for the BUILDING dataset.

Figure 18: Metric quality measures on the reconstruction for the BUILDING dataset. The segments with same color should have equal length and the angles should be $\pi/2$.

	σ_1	σ_2	μ
point-based	0.0138	0.0419	0.0700
plane-based	0.0129	0.0370	0.0633

Table 4: Metric measures on the initial reconstruction (point-based) and on that obtained after the constrained bundle adjustment (plane-based). The lower σ_1, σ_2 and μ (see text) are, the better the reconstruction is.

Dispersion radius r	σ_1	σ_2	μ
$r = 1$	0.0093	0.0284	0.0576
$r = 5$	0.0129	0.0370	0.0633
$r = 10$	0.0365	0.0511	0.1052

Table 5: Metric measures on the reconstruction obtained by constrained bundle adjustment for the planes detected with different values for the dispersion radius, see figure 16. The lower σ_1, σ_2 and μ are, the more accurate the 3D model is.

high dispersion radii tend to merge close planes (see *e.g.* the two left planes of figure 16 (a), merged in figures 16 (b) and 16 (c)). For that reason, the individual position of each plane is less accurate and the reconstruction quality is lower. Consequently, there is a trade-off between the reconstruction quality and the surface approximation.

The values given in tables 4 and 5 show that the metric reconstruction obtained with the method given in this paper is of superior quality than the one obtained with a traditionnal method based only on points.

6.2.3 The BOOK Dataset

This indoor dataset consists of 100 frames taken with a hand-held Canon camcorder. Some frames are shown in figure 19.

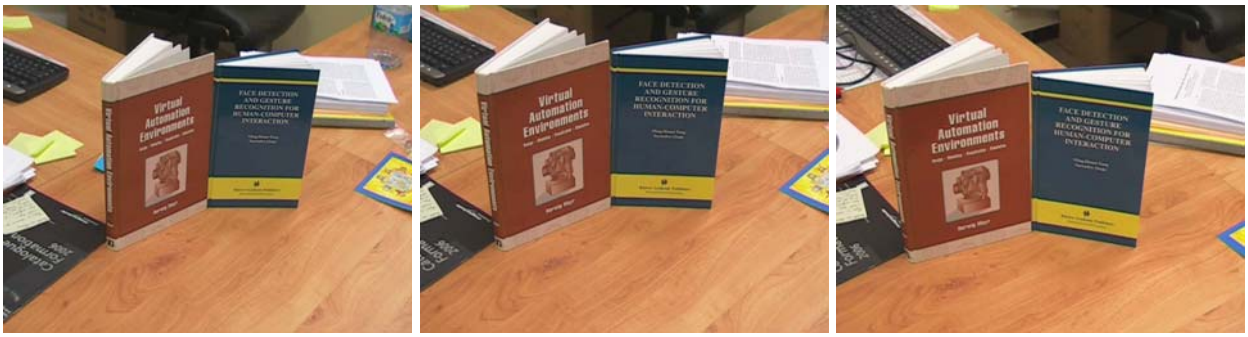


Figure 19: Some images extracted from the 100-frame BOOK dataset.

Point and camera tracking. We used a standard approach to Structure-from-Motion to track interest points while self-calibrating the camera [26]. We obtained a cloud of 254 3D points, with a reprojection error of 0.36 pixels, after unconstrained bundle adjustment. Figure 20 shows the 3D structure that was estimated.

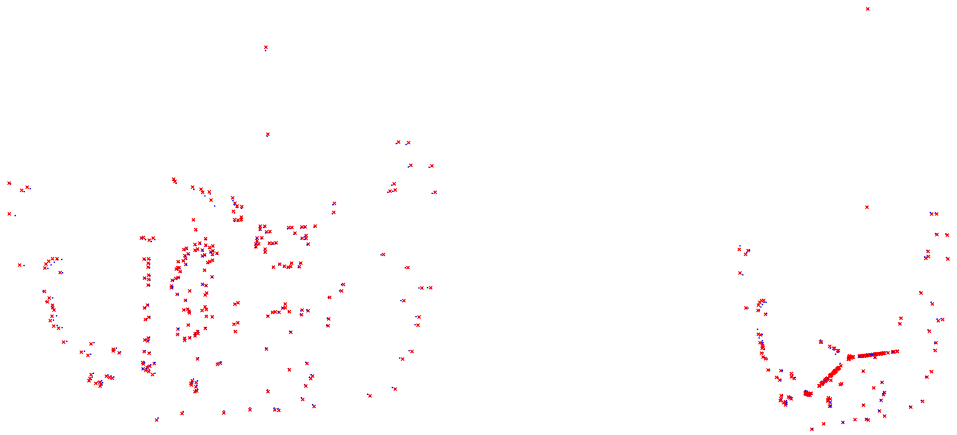


Figure 20: Two views of the 254 points reconstructed for the BOOK dataset using Structure-from-Motion. Crosses are points reconstructed with unconstrained bundle adjustment while dots are points reconstructed with piecewise planar constrained bundle adjustment. A top view is shown on the right.

Piecewise planar segmentation. We tried three values for the dispersion radius corresponding to roughly 1 pixel, 5 pixels and 10 pixels. They are also dubbed tight, intermediate and loose. We also tried ODS-GEOM, described in §6.1.

The following table summarizes the results we obtained:

Dispersion radius	tight	intermediate	loose	ODS-GEOM
Number of planes	6	5	5	7
Number of points off all planes	41	34	23	1
Number of points on a single plane	210	217	220	228
Number of points on two planes	2	2	9	22
Number of points on three planes	1	1	2	3

The tighter the dispersion radius, the higher the number of planes since it tends to split a single plane into several smallest ones. Making tighter the dispersion radius also increases the number of outlying points, *i.e.* points lying off all the modeled planes, while it decreases the number of points on three planes. ODS-GEOM includes almost all points in the model, since only one is an outlier.

The segmentation we found is shown overlaid on one of the images in figure 21.

We clearly see that while some planes are very stable, *e.g.* the one in the top right hand corner, some others strongly depend on the dispersion radius. In particular, the widest one at the bottom in the intermediate and loose dispersion radius cases is splitted into two smaller planes for the tight dispersion radius.

We observe that ODS-GEOM, which do not take photometric information into account, detects coplanar point configurations for which there is no real 3D plane.

Joint constrained reconstruction of planes, points and cameras. We ran the piecewise planar constrained bundle adjustment and obtained the following residuals (in pixels):

Dispersion Radius	tight	intermediate	loose	ODS-GEOM
Initialization	0.79	0.98	1.38	2.15
Bundle Adjustment	0.39	0.67	0.78	0.85

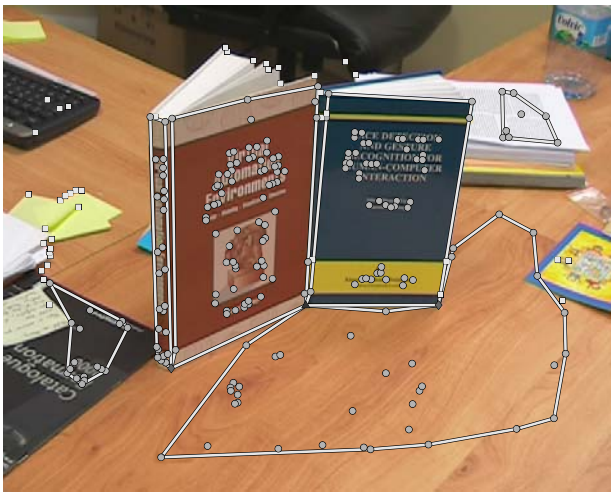
Rendering virtual views. Figure 22 shows virtual images rendered from the reconstructed piecewise planar model. The first row in this figure shows the rendering with a virtual camera aligned with the first camera of the dataset. The ODS-GEOM result is clearly better in that case. However, considering novel virtual cameras clearly shows that the planes detected by this algorithm do not all correspond to real planes, by introducing several artefacts.

7 Conclusion

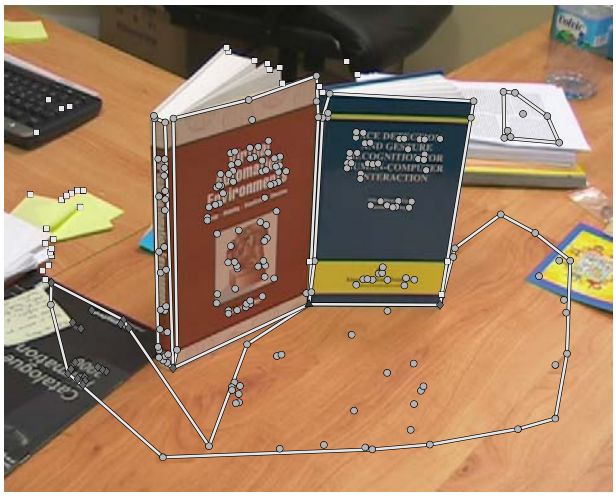
We investigated the automatic modeling of scenes using planes. The method is based on segmenting a cloud of reconstructed 3D points into multi-coplanar groups. We use random sampling to generate multiple plane hypotheses and select the most likely planes with respect to actual images. The use of a photometric measure allows us to distinguish coplanar point configurations that do not correspond to any world plane from real planes. The algorithm allows for overlapping data segmentation which permits to detect the majority of scene planes.

The Maximum Likelihood estimate of the model with respect to observed image points is then computed using an appropriate structure parameterization that enforces multi-coplanarity constraints.

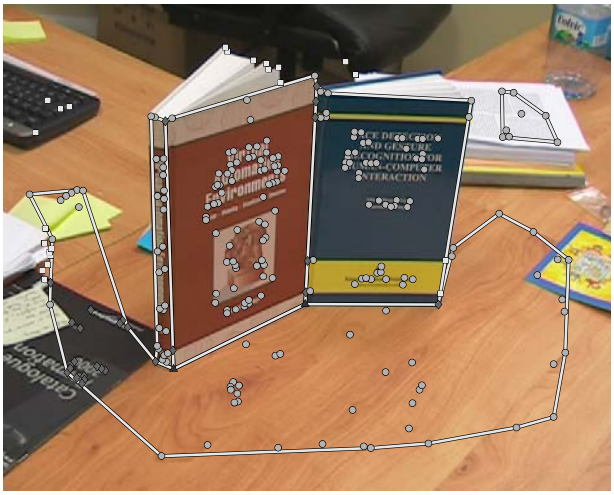
The use of simulated data showed that our method performs better, in terms of the number of detected planes, compared to those based on a purely geometric criterion or using a disjoint data segmentation scheme. We also validated the method using real images. This showed a good rendering quality and



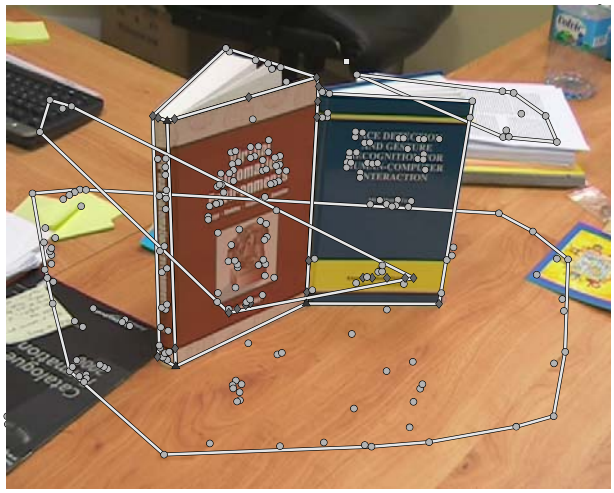
tight



intermediate



loose



ODS-GEOM

Figure 21: The first image of the BOOK dataset overlaid with the detected piecewise planar structure for different dispersion radii and the ODS-GEOM method. Points off all planes are drawn using squares, points on a single plane using circles, points on two planes using diamonds and points on three planes using triangles.

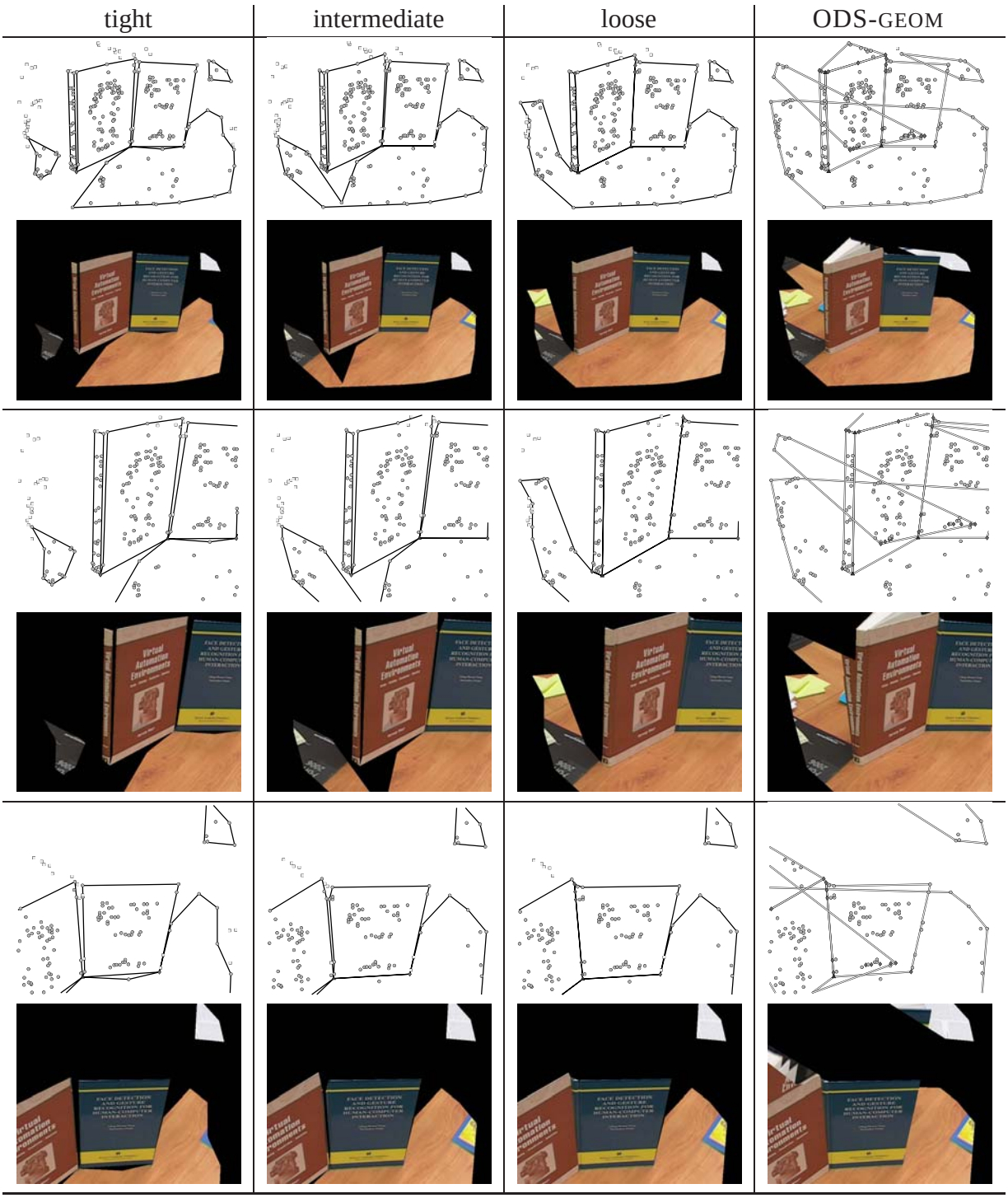


Figure 22: Some virtual views rendered from the reconstructed models for the BOOK dataset for the proposed method with three different dispersion radii and for method ODS-GEOM.

demonstrated how the difficulty of approximate planarity of real planes can be overcome using approximate photoconsistency, *i.e.* r-consistency.

However, there are still limitations and further research to pursue such as incorporating a model of lighting variation in the photoconsistency measure to specifically handle specular surfaces such as windows.

Finally, it would be interesting to try to replace the simple Delaunay triangulation by an image-consistent triangulation as described in [24, 25] in the plane delineation procedure. However, we do not expect a great improvement from this since the interest of the photometric score is mainly to get rid of spurious coplanar point configurations. In such cases, the score would remain low, while for the genuine coplanar configurations, we expect it to be slightly increased by photoconsistent triangulation, since more triangles would be photoconsistent and thus kept in the triangulation. Therefore, we expect the gap bewteen the scores of spurious and genuine coplanar configurations to be slightly more discriminative when photoconsistent triangulation is used.

References

[1] J. Alon and S. Sclaroff. Recursive estimation of motion and planar structure. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 550–556, 2000.

[2] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, pages 559–565. IEEE Computer Society Press, June 1999.

[3] A. Bartoli and P. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1):45–64, April 2003.

[4] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and R. Cipolla, editors, *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of *Lecture Notes in Computer Science*, pages 683–695. Springer-Verlag, April 1996.

[5] R. Berthilsson and A. Heyden. Recognition of planar point configurations using the density of affine shape. In *Proceedings of the International Conference on Computer Vision*, 1998.

[6] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach. In SIGGRAPH '96, *New Orleans*, August 1996.

[7] D. Demirdjian and R. Horaud. Motion-egomotion discrimination and motion segmentation from image-pair streams. *Computer Vision and Image Understanding*, 78(1):53–68, April 2000.

[8] A.R. Dick, P.H.S. Torr, S.F. Ruffle, and R. Cipolla. Combining single view recognition and multiple view stereo for architectural scenes. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, 2001.

[9] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, September 1988.

[10] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.

[11] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In *Proceedings of the SMILE Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, 1998.

[12] P. Fornland and C. Schnörr. A robust and convergent iterative approach for determining the dominant plane from two views without correspondence and calibration. In IEEE, editor, *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 508–513, 1997.

[13] P. Gargallo and P. Sturm. Bayesian 3D modeling from images using multiple depth maps. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

[16] H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 2005. To appear.

[17] R. Koch. Surface segmentation and modeling of 3D polygonal objects from stereoscopic image pairs. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria*, 1996.

[18] K. N. Kutulakos. Approximate N-view stereo. In *Proceedings of the European Conference on Computer Vision*, 2000.

[19] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 2000.

[20] F. Lang and W. Förstner. 3D-city modeling with a digital one-eye stereo system. In *Proceedings of the XVIII* ISPRS*-Congress, Vienna, Austria*, July 1996.

[21] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005.

[22] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, USA*, pages 482–488, June 1998.

[23] P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim. Robust regression methods for computer vision: a review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

[24] D.D. Morris and T. Kanade. Image-consistent surface triangulation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2000.

[25] A. Nakatsuji, Y. Sugaya, and K. Kanatani. Optimizing a triangular mesh for shape reconstruction from images. IEICE *Transactions on Information and Systems*, E88-D(10):2269–2276, 2006.

[26] M. Pollefeys, M. Vergauwen, and L. Van Gool. Automatic 3D modeling from image sequences. In *Proceedings of the XIX* ISPRS*-Congress, Amsterdam, Netherlands*, volume B5, pages 619–626, July 2000.

[27] F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18(9):647–658, 2000.

[28] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.

[29] D. Sinclair and A. Blake. Quantitative planar region detection. *International Journal of Computer Vision*, 18(1):77–91, 1996.

[30] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2004.

[31] A. Streilein and U. Hirschberg. Integration of digital photogrammetry and CAAD: Constraint-based modeling and semi-automatic measurement. In *Proceedings of the International* CAAD *Futures Conference, Singapore*, September 1995.

[32] J.-P. Tarel and J.-M. Vézien. A generic approach for planar patches stereo reconstruction. In *Proceedings of the 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden*, pages 1061–1070, August 1995.

[33] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In P.S. Schenker, editor, *Sensor Fusion VI*, pages 432–442, Boston, 1993. SPIE volume 2059.

[34] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In R.B. Fisher and E. Trucco, editors, *Proceedings of the seventh British Machine Vision Conference, Edinburgh, Scotland*, volume 2, pages 655–664. British Machine Vision Association, September 1996.

[35] P.H.S. Torr and A. Zisserman. Robust computation and parametrization of multiple view relation. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, 1998.

[36] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle ajustment — a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, 2000.

[37] C. Vestri and F. Devernay. Using robust methods for automatic extraction of buildings. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume I, pages 133–138, December 2001.

[38] R. Vidal and Y. Ma. A unified algebraic approach to 2-D and 3-D motion segmentation. In *Proceedings of the European Conference on Computer Vision*, 2004.

[39] A. Y. Yang, S. Rao, A. Wagner, and Y. Ma. Segmentation of a piece-wise planar scene from perspective images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[40] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998.