Computers & Graphics (2018)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Recovering 3D existing-conditions of indoor structures from spherical images

Giovanni Pintore[a,*], Ruggero Pintus[a], Fabio Ganovelli[b], Roberto Scopigno[b], Enrico Gobbetti[a]

[a] *CRS4 Visual Computing, Via Ampere 2, Cagliari, Italy*
[b] *ISTI-CNR Visual Computing Group, Via G. Moruzzi 1, Pisa, Italy*

ARTICLE INFO

Article history:
Received 19 April 2018
Accepted 15 September 2018
Available online 22 September 2018

Keywords: Panoramic scene understanding, Omnidirectional images, Mobile capture, Indoor reconstruction, As-built models

ABSTRACT

We present a vision-based approach to automatically recover the 3D *existing-conditions* information of an indoor structure, starting from a small set of overlapping spherical images. The recovered 3D model includes the *as-built* 3D room layout with the position of important functional elements located on room boundaries. We first recover the underlying 3D structure as interconnected rooms bounded by walls. This is done by combining geometric reasoning under an Augmented Manhattan World model and Structure-from-Motion. Then, we create, from the original registered spherical images, 2D rectified and metrically scaled images of the room boundaries. Using those undistorted images and the associated 3D data, we automatically detect the 3D position and shape of relevant wall-, floor-, and ceiling-mounted objects, such as electric outlets, light switches, air-vents and light points. As a result, our system is able to quickly and automatically draft an as-built model coupled with its existing conditions using only commodity mobile devices. We demonstrate the effectiveness and performance of our approach on real-world indoor scenes and publicly available datasets.

1. Introduction

Acquiring and recovering *as-built* and *existing conditions* models [1] of an indoor environment is an important task for many real-world applications. *As-built* models have to be created during and after construction to document any deviations from the architect's original design, such as doors that were placed in a different location from the construction documents, or to determinate the actual size and shape of the building versus what was specified. Moreover, *existing conditions* surveys are usually created post construction on top of an as-built plan, and include more details regarding, e.g., locations of electrical and data outlets, duct work, and sprinkler lines. Such information is essential to obtain an effective Building Information Model (BIM). Methods and tools to quickly recover such information is therefore of major practical importance.

The wide availability of mobile cameras, e.g., on smartphones, is making image-based methods very appealing in this context, since the capturing process is fast, simple and extremely cost effective. This is particularly true when exploiting emerging 360° cameras, since omnidirectional coverage simplifies geometric reasoning, as capturing of the environment requires very few shots.

In order to generate plausible reconstructions from the acquired visual information, state-of-the-art approaches (see Sec. 2) cope with the major problems posed by common interiors, such as typical offices or apartments, using visibility reasoning methods and exploiting prior knowledge. Handled problems include occlusions or texture-poor walls, while typical exploited constraints include the presence of vertical walls and horizontal floors (and ceilings).

Based on these concepts, a number of 3D solutions have been introduced for indoor floor mapping, 3D clutter analysis [2] (e.g., furniture detection or objects for daily human use) or indoor content creation [3], leading to methods capable to recon-

*Corresponding author
e-mail: giovanni.pintore@crs4.it (Giovanni Pintore)

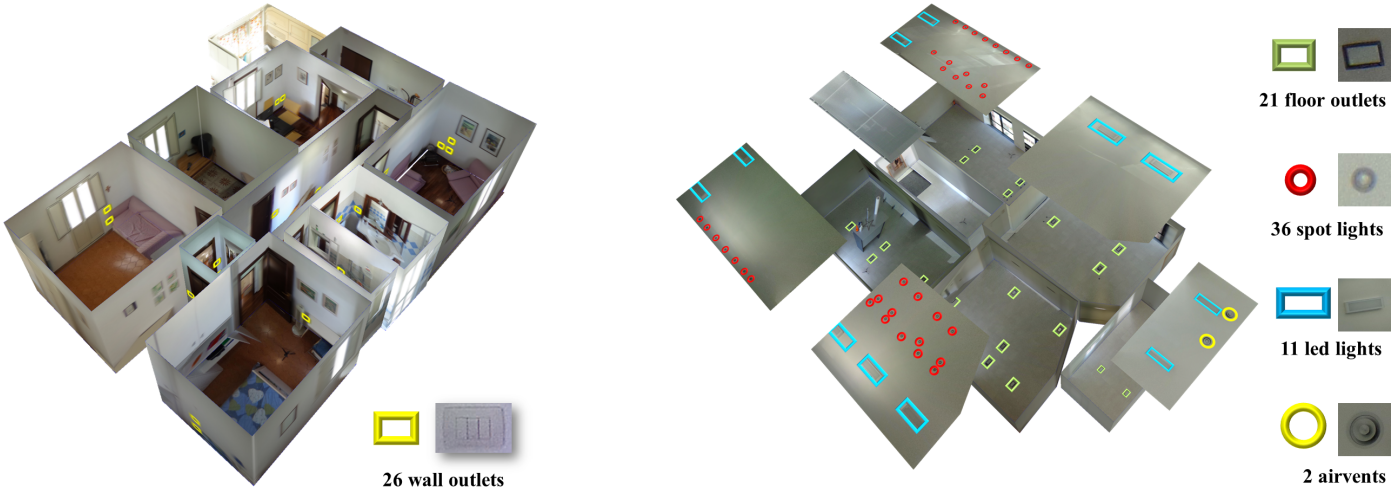


Fig. 1. Results preview. Multi-room models reconstructed by our method and the detected objects mapped on them (i.e.*Apartment* (left) and *office* (right)). Ceilings have been moved in the examples to enhance illustration. Beside the models we show the query images (i.e. patches extracted from the processed images), and the number of occurrences detected by our system. As illustrated in the Results section 6, we achieve solid performances both on real-world scenes and on publicly available datasets.

struct room shapes, as well as roughly determining furniture placement.

However, many fundamental objects defining *existing conditions* are approximately flat and placed on walls, ceilings or floors (e.g., outlets, air-vents, and a wide variety of integrated lighting fixtures). The shape, location, and placement of objects are an integral part of existing condition models, but, due to their placement and flatness, 3D solutions for object detection are generally ineffective in automatically identifying these elements. Therefore, specialized techniques for flat objects detection in the acquired images must be introduced to suitably augment the 3D models coming from a 3D indoor reconstruction pipeline.

The typical approach used when looking for approximately flat objects in sets of images is to apply a pure image-based method (see Sec. 2). Object detection in spherical images is, however, difficult, since angles and straight lines are not globally kept, objects appear variably stretched, and, as for any image-based approach, the missing metric information leads to the need for variable scale detectors, which increase the number of localization errors.

Several works deal with the general problem of object detection in spherical images in different ways, either by converting images with arbitrary projections to standard perspectives [4, 5, 6], or, more specifically, by modifying feature computation to work directly on catadioptric camera cases [7, 8]. We refer to Section 2 for a more detailed discussion. In this work, we avoid the complication of recognition and matching in spherical images by introducing a novel integrated approach tuned for our specific case of reconstructing an indoor environment and detecting functional elements in the scene. Our method exploits the 3D information recovered when extracting a 3D structured layout to locate the room boundaries, as well as to directly support image-based search for the elements located on them. Since functional elements are flat and typically stick on walls solely or on the ceilings solely, detection can thus exploit rectified projection on the reconstructed boundary.

1.1. Our approach

We first register a set of overlapping spherical images in a common reference frame using Structure-from-Motion. Then, we recover the underlying 3D structure of the environment, in terms of interconnected rooms bounded by walls, by analyzing the registered images. This is done using a novel approach that exploits the *Augmented Manhattan World* model (which assumes that walls are vertical but not necessarily orthogonal to each other). By exploiting the recovered 3D information we project the original spherical images, obtaining a set of *bi-dimensional* undistorted images re-sampled in a metric scale. Using those undistorted images and the associated 3D data, we can automatically detect, starting from examples, the 3D position and shape of relevant wall, floor, and ceiling-mounted objects. By exploiting the fact that we work on a rectified metric scale, we introduce, for detection, an approach based on sliding window and *Histogram of Oriented Gradients* (HOG) descriptors. As a result, the recovered 3D models, in addition to as-built room shapes, includes functional elements located on room boundaries (Fig. 1).

1.2. Contribution

Our approach combines and extends several state-of-the-art solutions in indoor reconstruction. The main contributions are the following:

- we introduce, to the best of our knowledge, the first purely image-based pipeline to automatically recover an existing-conditions model, which in addition to the 3D room layout, includes the position of important functional elements located on room boundaries; the pipeline is demonstrated on a set of real-world scenes, including publicly available datasets;

- we introduce a 3D layout recovery method which jointly exploits multiple spherical images for 3D layout extraction; the method is robust to clutter, and allows for a consistent 3D structure extraction for complex multi-room environments. Only few overlapping images are required, and, thus, the method is much less time-consuming than dense multi-view approaches;
- we introduce an approach to automatically detect, starting from examples, the 3D location of a large set of flat objects placed on room boundaries, important for existing conditions mapping and difficult to handle using geometric search; the proposed method exploits the recovered 3D information of the layout and the associated undistorted images to robustly perform an efficient image search.

1.3. Advantages

Our system enables even non-professional users to quickly draft an indoor as-built and existing conditions model requiring only commodity off the shelf devices.

By combining a small number of spherical views, we are able to fully recover the 3D layout of complex environments, overcoming the limitations of current single-view and multi-view approaches.

Even though spherical images are now very popular, thanks to their clear advantages in terms of easy capture and visual coverage, current state-of-the-art approaches for indoor reconstruction are generally related to single room/point-of-view scenes (see Sec. 2), basically by fitting simple box-like room models to the image [9] or by sketching the visible space through a combination of 3D planes [10]. Such approaches are very limiting in common indoor scenarios that usually include many occlusions, hidden corners, open spaces and multi-room environments, especially when finding far and small objects on room boundaries. On the other hand, dense multi-view approaches, which are usually effective in SLAM [11], need for a very small baseline to get enough 3D features on poorly textured environments [12], such as the indoor scenes, thus minimizing the advantage of capturing a scene with few spherical images and with minimal user interaction. Instead, our approach exploits geometric reasoning, and works with a small number of captured images. Furthermore, the adopted solution for 3D layout extraction covers a wider range of indoor environments compared to other general approaches (e.g.,[10, 13]), neither using additional tracking system for doors matching [13], nor externally calculated 3D data [14]. Our pipeline recovers, in addition to the 3D layout, the location of objects on room boundaries. In order to obtain a complete 3D building model, such a pipeline can be effectively integrated with existing orthogonal solutions for furniture detection (e.g., [2]).

By performing object recognition in the rectified space, we avoid distortion and scaling problems, outperforming query-by-example methods applied directly to the original images (see Sec. 6). Specifically, we integrate in the pipeline an effective detection method based on sliding-window and HOG descriptors, whose applications has already been proved particularly reliable in other unrelated bi-dimensional domains, such as text retrieval [15]. As the rectified images produced by our integrated system have the same metric scale, and this scale is globally kept in all the undistorted room images, the proper HOG cell size depends only by the query image size, without the need for any external tuning or pre-calibration [16], thus improving in terms of automation compared to the methods mentioned above.

2. Related work

Our approach relies on results obtained in several areas of computer graphics and compute vision. Here, we only discuss the methods that are most closely related to our technique.

2.1. Indoor reconstruction from spherical images

In recent years, efforts have been focused on simple and fast approaches for indoor reconstruction from panoramic images, regardless of a special hardware (i.e., using the most common format of equirectangular image). Yang et al. [10] propose an efficient method to recover the 3D appearance of a single room based on a constraint graph encoding the spatial configurations of line segments and super-pixels of a single panoramic image. Although effective in many indoor layouts, this approach is limited only to single room environment where all the corners are visible from a common point-of-view. Cabral and Furukawa [14] propose a system to reconstruct piecewise planar floorplans from images, however the reconstruction is strengthened by an externally calculated 3D point cloud. Similarly to Yang et al. [10], Pintore et al. [13] integrate the super-pixel labeling through the analysis of the image's edgemap, extending the result for the single room to multi-room environments by doors matching [17], with the aid of motion sensors embedded in a mobile device. As for [10], their approach works only when all the structural features of the room are visible from a single position. In this work, we improve over previous solutions by integrating an approach based on the *Augmented Manhattan World* assumption and Structure-from-Motion camera alignment, focusing more on boundaries detection. Our new approach returns a structured reconstruction even when single-view approaches fail [10, 13], thus requiring less capturing effort (see Sec. 6). A similar approach is taken by Pintore et al. [18]. Here, however, we target a very sparse sample, while that work requires the use of multiple-view features and relative 3D points to produce a result.

2.2. Flat object localization on spherical images

Object localization is a very active area of research, and we refer the reader to a recent survey for a wide coverage [19]. Specific works recover 3D foreground objects from a spherical indoor environment, such as the method of Zhang et al. [9], but these approaches are not well tuned for flat/2D objects on room boundaries. This is because they target recognition of fully 3D object shapes using 3D cuboid hypotheses, which are not compatible with the flatness of our target objects. The usual approach for 2D objects is instead to perform detection in image space, and to take into account deformation in a consistency check. For standard narrow-FOV perspectives, this is done by

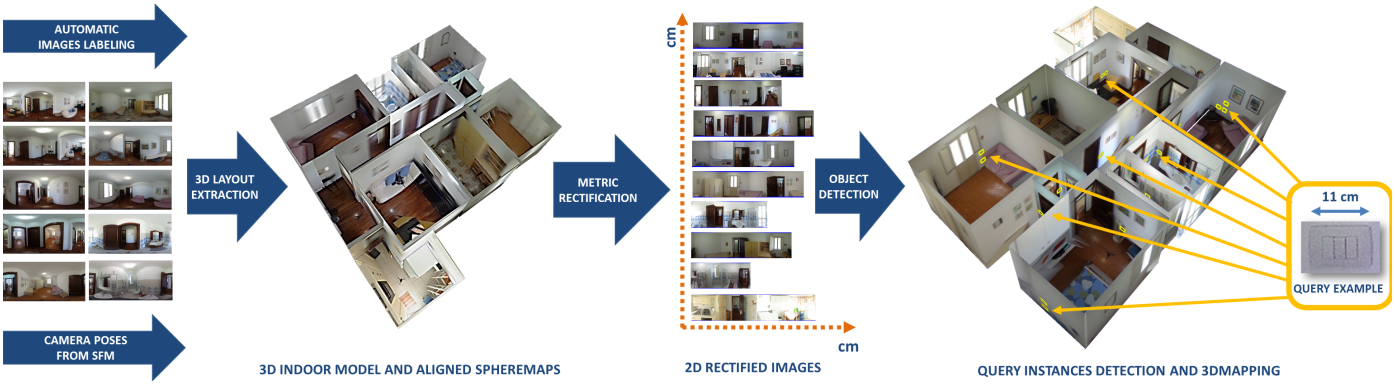


Fig. 2. System overview. We capture a set of overlapping *equirectangular* images, covering the indoor scene and the potentially hidden parts of the walls. We register the spherical images in a common reference frame using Structure-from-Motion and we automatically recover the underlying 3D layout by analyzing the registered images through an *Augmented Manhattan World* model. We project each spheremap on the recovered structure to generate uniformly sampled *bi-dimensional* images. Exploiting those undistorted images we are able to introduce an effective instance detection system starting from examples. The recovered instances are also automatically mapped on the 3D structure.

first detecting *scale-invariant features* (e.g., SIFT) and checking the consistency of the geometric relations of detected 2D points with respect to their supposed 3D position [20]. In the case of an indoor environment captured using equirectangular images, however, these approaches are not particularly effective, since the distortion is too wide. Many object detection methods, instead, use various types of re-projections prior to object recognition, in order to simplify the detection step by performing it in a suitable image space. For instance, face recognition often exploits locally conformal projection [21]. Similarly, panoramic projections are used for identifying objects from omnidirectional images [22] or videos [4]. Lines, however, appear bent, and objects that are not centered in the foreground are widely deformed, making it difficult to exploit detectors for regular shapes, such as rectangles or lines. Multiple perspective projections (e.g., cubemaps) are often exploited in order to preserve straight lines (e.g., [23]), with the drawback of discontinuity and object distortion at wide FOVs.

Our approach instead adapts the projection to the content of the scene, exploiting, instead of a generic arbitrary projection, the boundary reconstruction to detect objects in a fully undistorted and re-sampled space.

This approach allows the use of an effective *sliding window* strategy [24, 25], with the additional benefit of having to recognize small variations of a known form of a known scale. Having a known scale is fundamental to the success of such a sliding window approach because it allows to define the size of the window and the sliding step consistently with the size of the searched object, especially when dealing with small objects, since a small feature size causes the loss of contextual information and a large size results in capturing too much irrelevant information.

Moreover, combined with sliding window, HOG descriptors have proven to be very effective[26, 27, 28]. Dalal et al. [25] experimentally show how HOG descriptors significantly outperform other feature sets, such as SIFT, for human detection, while Almazan et al. [15] and Pintus et al. [29] successfully employed HOGs for word-spotting, using an empirical cell size adjustment. Cinaroglu and Bastanlar [7, 8] also HOG descriptors and a sliding window approach on a catadioptric image obtained by a photograph of a reflecting sphere, with the goal of detecting humans and objects in an outdoor scenario. Such an approach, instead of re-mapping the images, deforms the window and changes the gradient computation in order to work directly in the input catadioptric image space. Since our approach must, anyway, reconstruct wall surfaces in order to compute the 3D layout, and we look for flat objects placed on walls and ceilings, we can perform detection directly on the rectified images associated to room boundaries. This way, we can directly use the HOG and sliding window without the need for modifications.

3. Overview

Our pipeline, depicted in Fig. 2, extracts a simplified and structured 3D layout from a small set of overlapping *equirectangular* images (Sec. 4) (i.e., spherical images that have 360° longitude and 180° latitude field of view), with the goal of detecting and mapping the objects of interest located on the room boundaries (Sec. 5).

We assume that the input images are already aligned to the gravity vector (i.e., vertical lines in the image are vertical structures in the real scene) and rectified accordingly. These conditions are usually satisfied by spheremaps generated with the aid of sensor fusion (for example to compensate hand-held camera or a support not perfectly parallel to the ground), as in the case of the adopted mobile spherical camera.

To extract the 3D layout, consisting of interconnected rooms bounded by walls, we start performing a super-pixels segmentation on the single images (Sec. 4.1) in order to find a simplified geometric context labeling (i.e. ceiling, wall, floor). Then, under an *Augmented Manhattan World* model (e.g., vertical walls and horizontal floor/ceilings), we apply on the segmented boundaries (i.e. ceiling-wall and floor-wall contours) a *3D mapping function* and optimization scheme to infer, in Cartesian

Coordinates, the 3D room structure visible from each point-of-view/image (Sec. 4.2). Afterward we exploit the multi-view clues to find the 3D position of each spherical pose, (Sec. 4.3), thus merging in the same 3D model the contribution of all point-of-views.

Once the 3D model has been extracted, we generate *rectified images* by projecting the contents of each spherical image on the detected floor, ceiling and wall planes (Sec. 5.1). Exploiting the rectified images, we address the problem of flat object recognition as a *query-by-example* task: given a query object image, we identify and retrieve regions of the rectified images where the query object may be present, using an approach based on HOG (*Histogram of Oriented Gradient*) descriptors combined with an *Exemplar Support Vector Machine* (E-SVM) (Sec. 5.2). The recovered object instances are also automatically mapped on the 3D structure.

We show in Sec. 6 how our system provides solid performances on a variety of indoor scenes, such as real-world multi-room environments and various scenes from the publicly available *SUN360* dataset [30].

4. 3D layout extraction

4.1. Image labeling

The pipeline starts by performing, on each image, a super-pixel segmentation [31] and geometric context labeling, which assigns regions of the image to *ceiling*, *wall*, *floor*, leaving undecided areas labeled as *unknown*.

As for other indoor reconstruction approaches [13, 14], our method adopts a simplified classification based on only three labeled zones. Then, algorithmically, not all super-pixels are labeled in this stage of the pipeline. Moreover, our method does not need to recover the full geometric context for all of them (i.e., in the presence of clutter). For our neeeds, this partial classification is more than acceptable, as we are interested only in reconstructing the boundary structure of each room and in mapping solely the objects located on this boundary. The goal

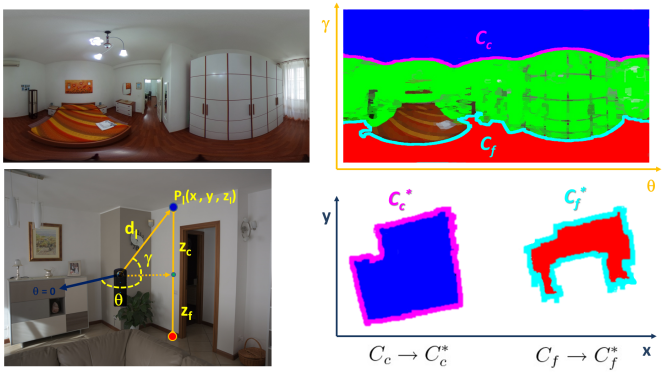


Fig. 3. Image analysis and Cartesian mapping. Each image (top left) is automatically labeled as *ceiling*, *wall*, *floor*, *unknown*, with the goal of finding contours between the labeled zones (top right). Contours are transformed through a mapping function from spherical to Cartesian space (bottom right).

of this classification is to find the contours C_c and C_f, containing the areas respectively classified as ceiling and floor (Fig. 3 top-right), labeling only those super-pixels that can be unambiguously assigned to floor, walls, and ceilings. Classification of all the super-pixels into ceiling, floor and walls will be made complete after having obtained the shape of the room (Sec. 4.2). A comparison between the error propagation at this stage and after the shape recovery is provided in Sec. 6.3.

In terms of segmentation and labeling, we improve over previous work [13, 14], which adopt a canonical color distance [26], by extending labeling propagation (Eq. 1) so that it takes into account not only a simple color distance, but also the *spatial distance* between super-pixels and the *statistical distribution* of the color [32]. We experienced that this approach better preserves geometric clues in indoor scenes, resulting in a better detection of ceiling/floor contours.

For each unlabeled super-pixel ν_i, the classification problem consists of finding the nearest labeled neighbor ν_j, according with the distance function:

$$D_p(\nu_i, \nu_j) = \frac{\|c_i - c_j\|^2}{(1 + \Omega(c_i, \sigma_i, c_j, \sigma_j)) \cdot (1 + \|p_i - p_j\|^2)} \quad (1)$$

where p, c and σ are respectively the *position*, the *mean color* and the *color variance* values of a super-pixel, and $\Omega(c_i, \sigma_i, c_j, \sigma_j)$ is an estimation of the volume intersection of the color space portions inside the super-pixels represented by the *Gaussians* $\mathcal{N}(c_i, \sigma_i)$ and $\mathcal{N}(c_j, \sigma_j)$. After having set all the super-pixels in the image as *unknown*, we start from the poles of the spherical image (defined, respectively, as *ceiling* and *floor*) and the horizon (*wall*), and we iteratively propagate the labeling of labeled super-pixel to its neighbors, until a distance threshold is reached. We tune the threshold (i.e. *th=0.5* in our tests) so as to maximize the *ceiling* coverage.

Once the propagation is complete (at least 60% of the whole image), we identify the edges (contours) C_c and C_f, respectively for the ceilings and for the floor boundaries. We exploit these contours to detect the room shape in world coordinates, as described in the following section.

4.2. Rooms boundary detection

The contours C_c and C_f are the boundaries of the ceiling and the floor found in the equirectangular image. Assuming the equirectangular image defines a direct mapping between pixels and angles of the spherical coordinates, the next step is to express such angles/pixels in Cartesian coordinates.

We employ a *3D mapping function*, based on the following spherical coordinates parametrization (illustrated at the bottom-left of Fig. 3):

$$P_l = \begin{bmatrix} d_l \cos\gamma \cos\theta \\ d_l \cos\gamma \sin\theta \\ d_l \sin\gamma \end{bmatrix}$$

where $l = (\theta, \gamma)$ is any pixel of the image. Obviously, from the image alone we cannot tell the depth d_l of each pixel, but from the assumption that the ceiling and the floor are horizontal we can write:

$$z_c = d_l \sin\gamma \Rightarrow d_l \cos\gamma = z_c / \tan\gamma \quad \forall l \in ceiling$$
$$z_f = d_l \sin\gamma \Rightarrow d_l \cos\gamma = z_f / \tan\gamma \quad \forall l \in floor$$

where z_c and z_f are the (unknown) heights of ceiling and floor, respectively. Therefore, we can express the contour of the ceiling and floor in 2D as a function of their height:

$$\begin{aligned} C_c{}^* &= \{(\tfrac{z_c}{\tan\gamma}\cos\theta, \tfrac{z_c}{\tan\gamma}\sin\theta) \;\; \forall(\theta,\gamma) \in C_c\} \\ C_f{}^* &= \{(\tfrac{z_f}{\tan\gamma}\cos\theta, \tfrac{z_f}{\tan\gamma}\sin\theta) \;\; \forall(\theta,\gamma) \in C_f\} \end{aligned} \tag{2}$$

An example of this mapping is illustrated at the bottom-right of

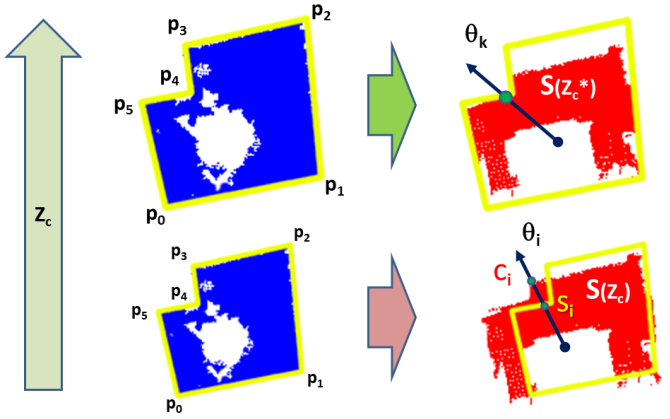


Fig. 4. Contours scaling and matching. To recover the 2D room shape from the transformed contours $C^*{}_c$ (up-to-scale) and $C^*{}_f$ (target metric scale) of eq. 2, we fit the polygon $S(z_c)$ (yellow - polygonal approximation of $C^*{}_c$) on $C^*{}_f$, varying z_c (red arrow), until the best scaling (green arrow) has been found.

Fig. 3, where labeled pixels of ceiling (blue) and floor (red) are represented in 2D Cartesian space, coupled with their boundaries (respectively magenta and cyan contours). As it can be easily verified looking at Equation 2, the 2D shape of the contours $C_c{}^*$ and $C_f{}^*$ is fully determined by C_c and C_f, while z_c and z_f are merely scaling factors. By using the assumption that walls are verticals we could state that $C_c{}^* \simeq C_f{}^*$, that is, the the two contours are *similar* shapes in the geometric sense and, furthermore, that we need only one of them to draw the room boundaries up-to-scale. Since that in practical situations, due to the furniture and clutter laying on the ground, the segmentation of the floor is way more prone to have missing parts than that of the ceiling, the boundary $C_c{}^*$ is usually more reliable. On the other hand, it is easy to know the actual value of z_f by measuring the height of the camera lens from the ground (i.e. *cm*), while is less so for the distance of the camera lens from the ceiling, which would give z_c.

Assigning to z_f a real world value has the important advantage of scaling the whole reconstruction in real-world metric scale (i.e. *cm*).

Therefore we adopt the following strategy. We set z_f to minus the height of the camera lens and build the contour $C_f{}^*$, than we find the value of z_c that minimizes the difference between $C_c{}^*$ and $C_f{}^*$. In other words we take the actual metric scale from the floor and the actual shape from the ceiling. This is carried out by the following steps. We apply an iterative end-point fit algorithm [33] to the ceiling contour $C^*{}_c(z_c)$ to recover the 2D shape of the room as a closed polygon $S(z_c)$ (Fig. 4, yellow shape). To find the right z_c^*, and consequently the shape of the room scaled in real-world dimension, we minimize the distance between points $s \in S(z_c)$ and points $c \in C^*{}_f(z_f)$ among all directions θ around the room center (Fig. 4):

$$z_c^* = \underset{\mathbf{z_c}}{argmin} \sum_{i=0}^{2\pi} \|s_i - c_i\|^2 \tag{3}$$

thus identifying the scaled polygon $S(z_c^*)$ enclosing the floor contour $C^*{}_f(z_f)$. From the 2D corners $\{p_0, \ldots, p_N\}$ of the polygon $S(z_c^*)$ we obtain the respective $2N$ 3D points by assigning them a z component (z_f for the floor and z_c for the ceiling points). The resulting 3D model $M_k \in R^3$ represents the simplified geometric structure visible from the k spheremap.

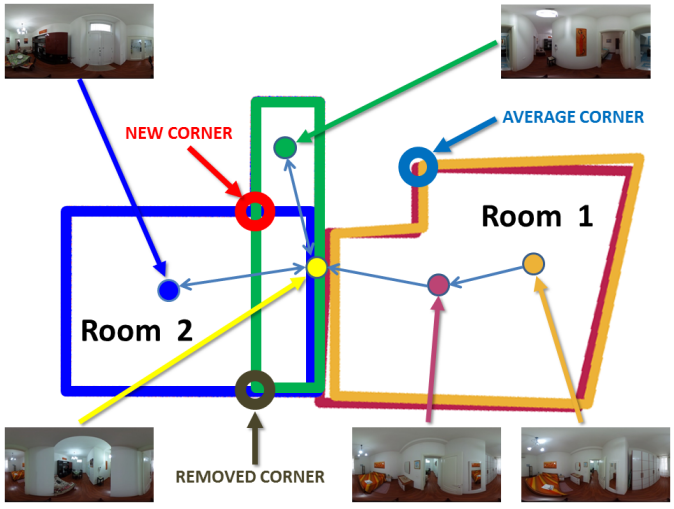


Fig. 5. Multiple poses merging. Single reconstruction are merged exploiting multi-view alignment between overlapping images. More than one reconstruction per room are exploited to refine the same room shape (Room1, orange and magenta), or joined to cover hidden corners (Room2 is fully covered by the green and blue reconstruction). Intermediate poses can connect different rooms (yellow).

The end-point fit algorithm returns a coarse approximation for the polygon, since we expect a room is represented by a limited number of corners and linear segments as walls. If there are many occluders on the ceiling, the projection of the super-pixels segmentation may lead to a very jagged polygon for which is hard to find a meaningful threshold for the end-point fit approximation. We conservatively re-run the end-point fit algorithm with an increasing threshold, until we obtain a shape with at least 4-corners. This way, a shape is always reconstructed for each view, considering also the fact that the final reconstruction, in particular for the most structured cases, will be given by the union of several views, as we will see in Sec. 4.3.

4.3. Connection of multiple views

As most of the indoor reconstruction methods from spherical images rely on a single-view (see Sec. 2), very few approaches bring together many rooms, basically exploiting external tools [13] or manual user interaction [17]. In our system instead we propose a pure vision-based approach, which exploits only the original images given as input. In-fact, once

a 3D model M_k is reconstructed in metric Cartesian coordinates, merging models obtained from different panoramic images reduces to finding the relative position and orientation of the cameras corresponding to those images. To do this we exploit a Structure-from-Motion approach to register each camera. As proven in many SLAM techniques for large baseline motion (e.g., [11]), omnidirectional images allow excellent performances when tracking 3D features to recover the camera path, although the recovered 3D features are too sparse for a dense 3D point-based reconstruction in an indoor environment. We exploit thus a multi-view registration method [34] to extract each spherical camera orientation $[R]$ and pose $[T]$. The resulting 3D tracking features, instead, give enough information to easily estimate a *ground* plan z, that is the floor level. We exploit this information to scale the camera trajectory according to the same metric z_f value of the local reconstructions (see Sec. 4.2), thus obtaining consistent dimensions for each reference frame. In order to account for measurement inaccuracies, we take as final z_c and z_f values the median of the estimates obtained for the different views. Then, through the transformation $[R_k T_k]$ acting on R^3 and associated to the *k-image*, we transform the coordinates of the corresponding room model M_k to the common reference frame.

Algorithm 1 Multiple rooms in a single model

1: S_n the set of 2D polygons/footprints
2: $G_m \leftarrow$ ClusterOnOverlap(S_n, 0.2)
3: **for all** $g \in G_m$ **do**
4: **for all** $f \in g$ **do**
5: $u \leftarrow u$ + Union(f)
6: $u \leftarrow u$ + MergeNearbyCorners(f, 20 *cm*)
7: $u \leftarrow$ RemoveCollinearCorners(u)
8: $r \leftarrow$ createShape(u)
9: *floorplan* $\leftarrow$ *floorplan* + Make3D(r)

An immediate application of such a kind of joining is the possibility to automatically connect different environments, even in case of having only an image per room, and to avoid mapping errors due to wall thickness. However, there are situations where it is useful to have more than one view per room, such as when the scene structure cannot be identified by a single view, or, as in our specific application, when we need to map the objects of interest on the rooms boundary.

Such cases are illustrated in the scheme of Fig. 5, where intermediate poses (yellow) connect different rooms and more than one footprint can be available for same room/environment.

Pseudocode in algorithm 1 shows how multiple footprints are merged to reconstruct the scene. The input of the algorithm (*line 1*) is a set of n footprints S_n, recovered as described in Sec. 4.3. At line *2* these S_n footprints are clustered in m groups G_m (e.g. $m < n$), in such a way that elements of the same group overlap for more that 20% their areas (i.e., orange/red and green/blue in Fig. 5).

Then, for each group $g \in G_m$, we perform a geometric *Union* (*line 5*) to encompass the $f \in g$ footprints contained in the same group, followed by a *merging* of nearby corners, so obtaining a new shape u (line 6). By definition, the *Union* operation will create new corners (for example, the one circled in red in Fig. 5), whereas *MergeNearbyCorners* will merge corners closer than a metric threshold (i.e. 20 *cm* - azure circle in Fig. 5).

At *line 7* we remove corners that lie in the segment passing by their two neighbors (for example, the one circled in brown of Fig. 5), and we generate the final 2D shape r from the remaining corners (*line 8*).

Finally, as already described in Sec. 4.2, we exploit the z_c and z_f components associated to the corner to generate 3D edges (*Make3D*) for each corner of the resulting 2D shape r, and we add the new room to the whole floorplan.

5. Detection of boundary-mounted objects

We exploit the 3D layout recovered in the previous task 4 to enhance the information present in the original images, such as eliminating distortion and scaling issues (*images rectification* 5.1), then we perform the object recognition in the rectified and re-sampled space (*query object recognition* 5.2).

5.1. Images rectification

For each spherical panorama with a correspondent underlying geometry we generate three *rectified images* by decomposing the room into walls, ceiling and floor sides (Fig. 6).

To create the images we exploit the original equirectangular map as a *texture*, using the projection function of eq. 4, which assigns to each 3D point $p(x, y, z)$ of the underlying reconstructed model its u, v texture coordinates.

$$u = \frac{\arccos(\frac{x}{\sqrt{x^2+y^2}})}{2\pi} \quad v = \frac{\arctan(\frac{z}{\sqrt{x^2+y^2}})}{\pi} + \frac{1}{2} \tag{4}$$

This projection represents the decomposition of the room boundaries into *2D planar elements* (walls, ceiling, floor planes), under the assumption of vertical walls and horizontal ceilings. The representation removes the distortion on straight lines, room boundaries and objects. Furthermore, it projects the information contained on the original image to a dense grid of uniformly spaced cells, with a constant metric scale (e.g. 4 pixels per *cm* - see 6). As proof-of-concept in Fig. 6 we show an example of the original spheremap compared with its three rectified images. In particular, the ceiling projection shows how two of the four air ducts, practically unrecognizable in the original spheremap, are clearly highlighted in the rectified ceiling image.

5.2. Query object recognition

This step takes as input the rectified images obtained at task 5.1. Any image patch extracted from them can be a query object for the detector (examples of queries are showed in Fig. 1). This is actually the only part of the system where a minimal user intervention is required, limited to selecting what is the object (image patch) to be searched. The advantage of such query-by-example approach is that does not need long training by using external labeled data, as well as the search can be easily extended to arbitrary flat objects, such as in the example showed at Fig. 11(a).

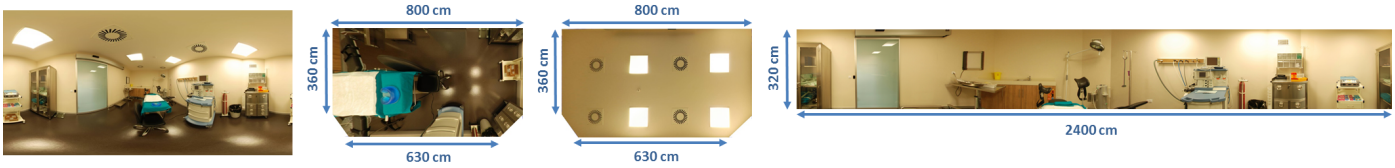


Fig. 6. Rectification. For each original equirectangular image we generate three bi-dimensional images in uniformed metric dimensions: floor, ceiling and walls images. The ceiling projection shows how two of the four air ducts, practically unrecognizable in the original image, are instead clearly highlighted in the rectified image.

Since the queries and the processed images are in bi-dimensional rectified space, we are able to simplifying searching by introducing a sliding-window approach based on HOG descriptors. HOGs, in-fact, combined with a sliding window approach, are particularly reliable for flat object detection [27, 28]. These descriptors are used to train a classifier for the query image in an *Exemplar Support Vector machine* (E-SVM) detector.

Such detector, compared to a standard nearest-neighbor scheme, can be trained in a high discriminative way [16]. Actually, instead of a single complex category detector, we exploit a large collection of simpler individual Exemplar-SVM detectors, each highly tuned to the single exemplar's appearance. In other words each of these Exemplar-SVMs is thus defined by a single positive instance and many of negatives.

Since we adopt a linear SVM, each classifier can be interpreted as a learned exemplar-specific HOG weight vector.

Given the HOGs of a set P of positive regions (relevant to the query), and a set N of negative regions (not relevant to the query), we train an E-SVM at query time to find the occurrences of the query across the dataset.

To evaluate the similarity between a region y and the query q, we consider the dot product $q^T y$ between their corresponding HOGs. We exploit this similarity definition to minimize this cost function:

$$\underset{w}{argmin}\frac{1}{2}\|w\|^2 + C_1 \sum_{y_p \in P} L\left(w^T y_p\right) + C_2 \sum_{y_n \in N} L\left(-w^T y_n\right) \qquad (5)$$

where $L(x) = max(0, 1 - x)$ is the hinge loss and C is a cost parameter (for a detailed description of cost parameters see the work of Malisiewicz et al. [16]). The weight vector w is the optimized, learned model giving the set of trained parameters for classification.

In our specific case, the vector w found is a weighting vector of the regions relevant to the query, and it can be seen itself as a new representation of the query. This vector is used during the retrieval process to compute the region similarities with the aforementioned dot product, and it will produce high positive scores for relevant regions, and high negative scores for regions not relevant to the original query.

Compared to standard two-class or one-class SVMs, our E-SVM produces a two-class classifier for each item in the positive set against all the negative regions, in order to cast votes that are calibrated and combined to obtain the final classification results. Positive/negative sets are built on-the-fly at query time. To this end, we construct the positive set by deforming the query and shifting the window around the query, and the negative set by randomly picking regions in the scene. Although different from a complete unsupervised approach, we perform training/learning at query time, thus increasing flexibility and applicability.

However one of the most critical parameter for an effective HOG computation is the cell size, which requires prior knowledge of the *scale* of the most relevant features in the query object. Differently from what done in other domains (e.g. [15, 29]), where the size of the cell and the desired scale are empirically established, our approach improves over previous applications since enables HOG use without manual intervention, as rectified images (including the query object) have a single known (metric) scale and thus even the objects of interest have a known size. It should be noted that for a single image our algorithm could, theoretically, work in the same way even without a metric scale. However, in most application scenarios, such as architectural surveys, it is mandatory to have a metric scale, and we can thus exploit it in the recognition.

In Sec. 6 we show that our technique outperforms detection methods applied directly to the original images. Furthermore the objects found on such a kind of rectified images implicitly have a 3D position in the underlying model.

6. Results

We have implemented a pipeline that, starting from a collection of spherical images, automatically produces a 3D room layout augmented with the locations of boundary-mounted objects.

6.1. Implementation

We developed the 3D layout extraction and rectification tools (Sec. 4 and Sec. 5.1) through C++ on top of OpenCV, for a better CPU/GPU optimization. To facilitate comparison with other approaches, the detection methods have been instead implemented with *Matlab*. To obtain camera poses we developed a tool based on the approach of Kangni and Laganiere [34]. Other available tools, such as *PhotoScan*[1], are equally valid

[1] http://www.agisoft.com/

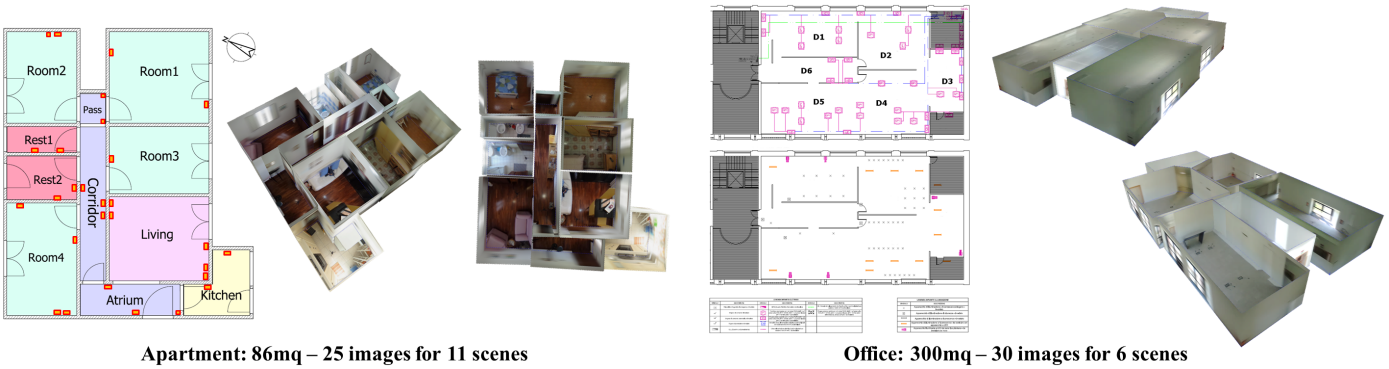


Fig. 7. Ground truth drawings and our as-built reconstruction. We show our 3D reconstruction of all the rooms beloging to *Apartment* (left) and *office* (right) environments, compared with their as-built drawings. For the *Apartment* rooms we manually recover the as-built situation and the position of functional elements by performing on-site inspection aided by laser measures. For the *Office* rooms we exploit instead the available existing conditions plans.

for the same purpose. The rectified images are generated by a GPU *shader* implementing Eq. 4, through the rendering of the original equirectangular images and the reconstructed underlying model on frame buffer objects. For each input spheremap, we create three rectified images, re-sampled with a number of pixels per *cm* chosen to make the most of the original resolution of the equirectangular image (e.g. 14Mpixel). Before detection, we convert rectified images to a gray-scale signal through a decolorization technique to increase the effectiveness of gradient-based descriptors [35].

6.2. Data collection

We tested our system on a variety of indoor scenes, including real-world residential and commercial buildings and various scenes from the public SUN360 dataset [30]. Since the goal of the system is to recover a real situation that can be different from blue prints or available schematics, the main priority has been to collect data where a ground truth was available. Although some indoor public datasets provide 360 images [30, 36], as they are mostly targeted to support other applications, they do not provide as-built and existing conditions information. Moreover, very few examples of private residential building are available, even in popular datasets as the SUN360 [30] (i.e. actually almost all images of SUN360 *bedroom* category are hotel rooms). Therefore we considered to carry out a specific acquisition campaign, creating ground truth data from on-site inspection aided by laser measures and comparing these reliefs to available blue prints (Fig. 7). We collected many scenes including objects of interest, which have been exploited both as single scenes or grouped in the multi-room structures from which they originate. We make such data available to allow further studies and comparisons [2]. Furthermore, we present detection results in Tab. 3 on the *SUN360* dataset [30], performed after manually labeling visible flat objects of interest, even if geometric ground truth and existing conditions are not available for this data. To collect the real-world examples (Fig. 7) we capture equirectangular images, covering a full viewport of 360° longitude and 180° latitude, at the resolution of 5376x2688, by using a commodity mobile *Ricoh Theta S* spherical camera [3]. To avoid unpleasant occlusion on the bottom hemisphere we mount the camera on a tripod, also using a fixed distance of 170cm (i.e. $-z_f = 170cm$ - see Sec. 4.2) from the ground floor, thus exploiting this information to obtain final models in real-world metric dimensions. To recover the camera positions and orientations we acquire the images so they always have some overlap between them, approximatively capturing at least two images for large environments to assure a full visual coverage of the room boundaries.

6.3. 3D layout extraction and mapping

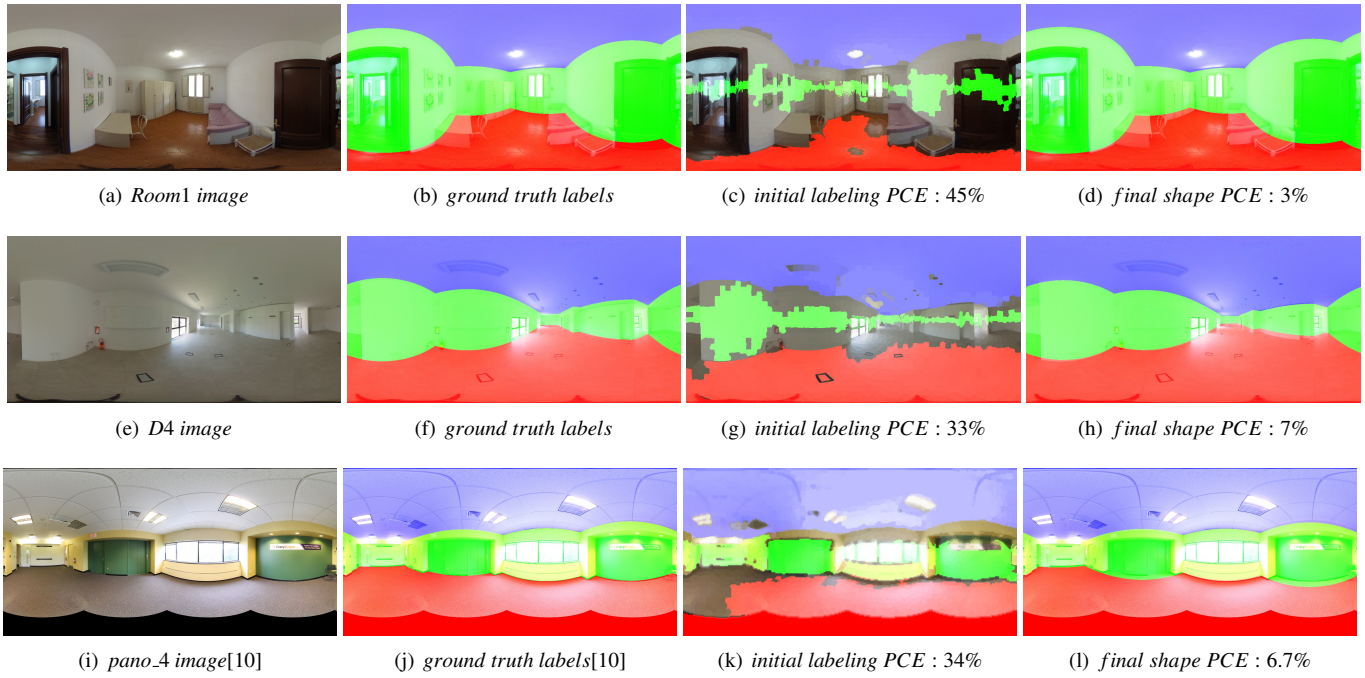
In Fig. 7 we show the 3D reconstruction of the multi-room environments *Apartment* (left) and *office* (right), beside their ground truth maps. Each view, once scaled by 4, requires *2.5 seconds* to be processed on an Intel i7-4700HQ processor with 16GB RAM. This time includes super-pixels segmentation, labeling and shape inference. Recovering the whole *SfM* trajectory (i.e., time to align about 30 full-size poses) costs less than one minute, on the same hardware. Once the trajectory has been recovered the merging of different views/shapes is almost immediate.

Results in Tab. 1 shows detailed performances for each scene contained in the *Apartment* and in the *Office* blocks. We compare with the method of Pintore et al. [13], which is the most relevant to ours in terms of multi-room metric reconstruction from spherical images. To do the tests we implemented their method for the single rooms reconstruction through *OpenCV*, while for rooms assembly by doors-matching we adopt an equivalent, in terms of accuracy, manual alignment approach [17].

We focus our tests to highlight the aspects which directly affect object detection and mapping: *PCE* (*Pixel Classification Error*), *Geometric error* and *Mapping Error*. The geometry to which we refer for our method is the final connected layout

[2] http://vic.crs4.it/download/datasets/

[3] https://theta360.com/en/about/theta/



(a) *Room*1 *image* (b) *ground truth labels* (c) *initial labeling PCE* : 45% (d) *final shape PCE* : 3%

(e) *D*4 *image* (f) *ground truth labels* (g) *initial labeling PCE* : 33% (h) *final shape PCE* : 7%

(i) *pano_*4 *image*[10] (j) *ground truth labels*[10] (k) *initial labeling PCE* : 34% (l) *final shape PCE* : 6.7%

Fig. 8. Intermediate labeling and final reconstruction coverage. We show some intermediate and final results of our method with respect to pixel classification error. We show case from the presented dataset (Room1 and D4) and, for comparison, with the data used by a generic single view method [10]. All ground truth images have calculated according with our simplified model (ceiling, floor and wall).

Scene			PCE			Geometric error						Mapping error	
			Our		Other	Our			Other			Our	Other
Name	Type	V.	label [%]	final [%]	final [%]	c [deg]	l [%]	a [%]	c [deg]	l [%]	a [%]	dist [cm]	dist [cm]
Living	Res	3	42	4	11	0.5	4	6	0.5	10	10	0	0
Atrium	Res	2	46	5	12	4.5	8	10	5.0	9	12	10	18
Corridor	Res	5	48	3	10	3.0	7	9	9.0	16	20	22	32
Pass	Res	1	35	4	4	0.0	5	5	0.0	5	5	24	42
Room1	Res	2	45	3	6	0.5	9	10	0.5	12	12	18	65
Room2	Res	2	39	3	-	1.5	8	8	-	-	-	20	-
Room3	Res	2	35	5	12	0.5	4	4	1.0	5	6	25	56
Room4	Res	3	38	6	14	1.5	5	6	1.5	10	12	36	64
Rest1	Res	1	39	4	5	5.5	3	4	6.0	4	4	28	68
Rest2	Res	1	40	5	6	2.5	10	12	2.0	10	12	40	70
Kitchen	Res	3	44	12	-	6.5	6	8	-	-	-	15	-
D1	Com	6	38	3	9	1.0	7	8	1.5	10	10	0	0
D2	Com	5	40	3	-	1.5	9	10	-	-	-	23	-
D3	Com	7	21	9	-	0.0	5	6	-	-	-	26	-
D4	Com	4	33	7	-	1.5	10	12	-	-	-	12	-
D5	Com	5	41	4	-	1.5	10	10	-	-	-	16	-
D6	Com	3	33	6	12	0.5	8	10	0.5	10	10	8	-

Table 1. Reconstruction facts. We present performances of our method compared with ground truth and with the method of Pintore et al. [13] (indicated as *Other*). For each scene we indicate the typology (Residential or Commercial) and the number of views (V.) captured. PCE shows the *Pixel Classification Error* compared to the other approach, also detailing, for our method, the intermediate labeling performance (*label*). *Geometric error* shows the per-room performance in terms of absolute corner angle error (*c*), maximum percent wall length (*l*) and area error (*a*), compared with the alternative method. The *Mapping error* shows, instead, the error of each room in terms of absolute position in the floorplan.

from more views (column *Views*), projected to the equirectangular image which has the maximum coverage of the room, that is the image one used to compare the other approach [13]. We have chosen for our experiment mostly cases that highlight the need to have more views (i.e., clutter and hidden corners). In the simplest cases on only one image per room (i.e., Rest1, Rest2), our method basically performs similarly to the other method. Proposed PCE shows the capability to correctly map a pixel to the underlying room boundary (i.e. ceiling, floor and wall planes), that is a local error with respect to each environment. For the test the same equirectangular image have been projected on the ground truth room boundaries (from a CAD based on laser measures) and on the reconstructed as-built models for comparison.

As described in Sec. 4.2, such simplified classification is different from other models [10], and targeted mainly to identify areas that are *ceiling*, *floor* and *wall*, whereas the *wall* label means not only the actual walls but also everything that is not clearly floor or ceiling, as much of the clutter in the room. Thus, according with this labeling, our ground truth room model consists of vertical walls bordered by horizontal floor and ceilings. To better illustrated this aspect, we show in Fig. 8 *image labeling* evaluation and intermediate results of our reconstruction pipeline. We show original images from our dataset (e.g. Fig.8(a), Fig.8(e)) and from data presented in the work of Yang et al. [10] (e.g. Fig.8(i)). We show in Fig. 8(b), Fig. 8(f) and Fig. 8(j) the ground truth labeling (N.B. according with our simplified model), the intermediate super-pixels labeling (Sec. 4.2) in Fig 8(c), Fig. 8(g), Fig. 8(k), and the final labeling using the recovered 3D shape in Fig. 8(d), Fig. 8(h) and Fig. 8(l).

It should be noted that PCE is significant in the intermediate labeling stage (i.e., 34% in Fig. 8(k)), compared, for example, to PCE values of Yang et al. [10] on the same image (i.e., 26%), but also how this value rapidly decreases in the final stage (i.e., 6.7%). We detail these quantitative differences for each room in Tab. 1.

Geometric Error analysis shows 2D error for corner angles, wall length and room area, compared to the method of Pintore et al. [13]. As we expected, such error is similar for both methods when the room has little clutter and its structural parts are

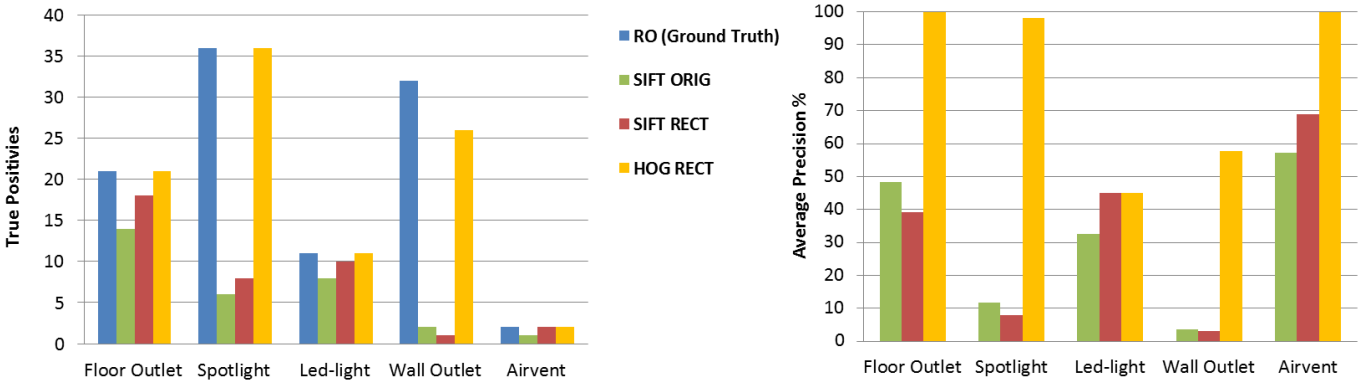


Fig. 9. Multi-room retrieval summary. We compare the performance our method (HOG RECT) with a common SIFT approach, both on the original images (SIFT ORIG) and on the rectified images (SIFT RECT). We can see how our approach retrieves almost all the *Relevant Objects* (RO - ground truth) as an output set of *True Positives* (TP). Average precision is computed on the ranked list of spotted instances of the queries showed in Fig. 1. Actually, the *Led-light* query is not a real flat object. In this case our method perform similarly to SIFT.

evident even from a single point of view (*Pass, Room3, Rest2*). Instead, while our method continues to maintain similar reliable performance, the compared single-view approach tends to considerable errors for long corridors (*Corridor*) and moderately cluttered rooms (*Living, Room4*), up to the point of failure the reconstruction in case of more clutter (*Room2,Kitchen*) or large open spaces (*D2,D3,D4,D5*).

The *Mapping Error* is instead related to the capability of mapping an object in a multi-room floor plan. This is an important measure because a functional map of a building should connect elements distributed among several rooms. We indicate this value as the Euclidean distance between the real position of a room (i.e. its relative center), with respect to the absolute center of the floor plan (i.e. the center of the first acquired room), and its estimation by a reconstruction method. Results on *Mapping error* highlight, beside the limitations already seen, one the major weakness of the currently alternative approaches [13, 17], namely the capability to manage multi-room environments. Such methods, in-fact, are based on very strong assumptions, i.e. that the rooms are practically closed, and that rooms are connected to each other through well-defined doors. Furthermore, even under the assumptions described, these approaches does not take into account the thickness of the walls. It is also noticeable how the failure of a single room reconstruction in the doors-matching methods affects the alignment of any other connected room, such as room *D6*. It should also be noted that the other method is not capable to return results in several difficult cases (cells containing "-" in the table). This is because many acquisitions do not have all corners visible from a single point of view, and because the edgemaps of the images are often noisy and do not allow that method to complete. By contrast our new solution exploits multi-view information, and is therefore more general and robust.

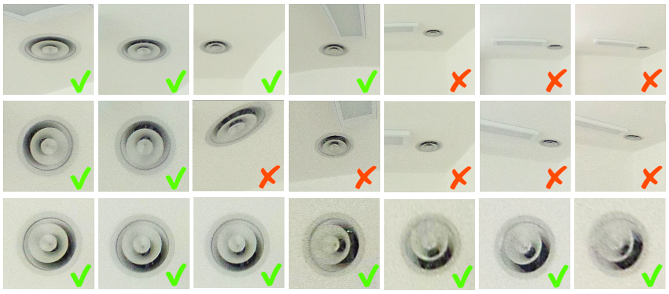


Fig. 10. Spherical vs Cubemaps vs Rectified. We compare object recognition of an almost circular shape on the original equirectangular images (first row), using cubemaps (second row) and with our rectified images (third row). Successfully retrievals are marked with a green check.

	SIFT			HOG
	Spherical	Cubemaps	Rectified	Rectified
Size	4Mpixel		4pixels/cm	
#Image	7	7	21	21
Time	28s/im	7s/im	12s/im	7s/im
TP/RO	**4 / 7**	**2 / 7**	**7 / 7**	**7 / 7**
FP	2	0	10	85
FN	3	5	0	0
AP	**57.1%**	**28.6%**	**68.9%**	**100%**

Table 2. Retrieval comparison. Retrieval details for room D3 (Fig. 7) and the query *airvent* (Fig. 1). We report the number of *True Positives* (TP) over *Relevant Objects* (RO), *False Positives* (FP), *False Negatives* (FN) and the resulting *Average Precision* (AP). We also give an average time-per-image spent to retrieve the query. our rectified images improve detection performance even using a standard detector based on scale invariant SIFTs and leads to near perfect results, in terms of average precision, with our full HOG/E-SVM approach.

6.4. Flat object detection

Since we are looking for flat objects placed on walls and ceilings, which are reconstructed by our pipeline, we can perform recognition directly in rectified space, searching for objects on the textures associated to each room boundary. This is a much more constrained situation than general object recogni-

Scene		TP/RO		FP		FN		AP [%]		3D model
Name	Query	Other	Our	Other	Our	Other	Our	Other	Our	PCE
Childroom *aclzqydjlssfry*	Lights	1/4	4/4	0	25	3	0	25	100	6.7%
Childroom *azzjipzfnrdhxv*	F. tiles	1/10	10/10	0	102	9	0	10	57	6.9%
Classroom *azeguairoehroh*	Outlets	1/6	6/6	0	256	5	0	17	86	4.8%
Classroom *azeguairoehroh*	Speakers	0/8	8/8	0	97	8	0	0	93	4.8%
Hospital *arziggvjlludoe*	Lights	1/3	3/3	0	216	2	0	33	100	7.2%
Hospital *arziggvjlludoe*	Outlets	0/5	5/5	0	495	5	0	0	100	7.2%
Hospital *ayuhcqtaohwvwn*	Air ducts	1/4	4/4	1	15	3	0	25	100	5.2%
Office *acpizmtldontna*	Lights	1/16	16/16	0	63	15	0	6	73	9.7%
Bedroom *abglgjxkigldd p*	Sprinklers	1/3	3/3	0	192	2	0	33	100	5.8%
Living *acroxxginqzgaw*	Lights	1/4	4/4	6	13	3	0	25	95	6.2%
Living *acroxxginqzgaw*	Sensors	1/2	2/2	0	247	1	0	50	100	6.2%

Table 3. Detection performances on the SUN360 dataset. In the first columns we indicate the scene naming (category and string) in compliance with the SUN360 database and the query object. We report the number of *True Positives* (TP) over *Relevant Objects* (RO), *False Positives* (FP), *False Negatives* (FN) and the resulting *Average Precision* (AP). We test both our method and a standard detector, indicated as *Other*, based on scale-invariant SIFTs [20] for comparison. In the last column we show a screenshot of the underlying 3D model and its PCE.

tion in spherical images, which is typically targeted through reprojection to remove distortions or through direct computation of features and neighborhoods in spherical images (see Sec. 2). In order to qualitatively show how important spherical distortions are in our case, we present detection results on room *D*3 (*Office* scenario illustrated in Fig. 7). We track the same object (*airvent*) over 7 different views, showing its deformation under different projections in Fig. 10. In the first row we show the deformation of the object in the original *equirectangular projection*, in the second row the deformation with a *cubemap projection* and in the third one our *rectification* result. Please note that with our approach the object is always recognizable.

In Tab. 2 we show the quantitative statistics and comparisons. Assuming as *Relevant Objects* (RO) the 7 occurrences of the same object in the different images, we report the number of *True Positives* (TP) over *Relevant Objects* (RO), *False Positives* (FP), *False Negatives* (FN) and the resulting *Average Precision* (AP), as well as the processing time to detect each image occurrences. Since we recover a ranked list of occurrences taking into account the order in which the elements are returned, we can adopt *Average precision* instead single-value metrics, such as Precision or Recall, which instead do not consider the order of the elements and are less suitable in this case.

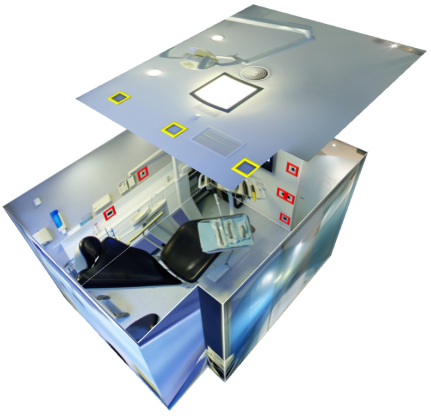
Results demonstrate how our rectified images improve detection performance (100% of true positives, 68.9% average precision) even using a standard detector based on scale invariant SIFTs [20], and leads to near perfect results (i.e. 100%), in terms of average precision, used in our full HOG/E-SVM approach. We have also verified how spherical projections work better than commonly used cubemaps (i.e. 57.1% vs. 28.6%), for circular shapes, that is the air vent shape, as theoretically expected (see Sec. 2). Successfully retrievals are marked with a green check in Fig. 10.

In Fig. 9, we summarize the quantitative detection results obtained on the 17 scenes enlisted in Tab. 1. Such rooms are the elements of multi-room datasets of which existing conditions are available (i.e. *Apartment* and *Office*). Final results are mapped on the whole multi-room models and illustrated in Fig. 1.

We search the occurrences in all the rooms of 5 relevant objects (wall outlet, airvent, led light, spot light, floor outlet), de-

(a) azzjipzfnrdhxv

(b) azeguairoehroh



(c) arziggvjlludoe

(d) ayuhcqtaohwvwn

(e) abglgjxkiglddp

(f) acroxxginqzgaw

Fig. 11. Detection details on SUN360 dataset. We present qualitative details about the SUN360 cases proposed in Tab. 3. In the first illustration (a) we show the detected particular tiles (i.e. butterfly-like tiles in a childroom), which have instead a very high deformation in the original image, highlighting the successful detections with red rectangles. In the classroom case (b), we show both speakers (red) and floor outlets (purple) detections. (d) case illustrate air ducts detection for the rectification prrof-of-concept case presented at Fig. 6. In the illustrations (d), (f) we show lights detections (yellow and red), wall outlets ((c) red) and fire sprinklers ((f) yellow).

(a) *aapcyahhkcfvkc* (b) *abouthfwaqoeew* (c) *abouthfwaqoeew* our (d) *abouthfwaqoeew* ground truth

Fig. 12. Failure cases. We show several failure cases on images from the publicly available *SUN360* dataset [30]. In 12(a) we present a case of sloped ceiling room, that is an example of non compatible case with our method. The second case 12(b) is instead formally under our assumptions, as well as the quality of the image should allow detection of structure. Instead, the strong discontinuities in the ceiling and floor zones lead to a wrong reconstruction of the room (Fig. 12(c)), where the whole part behind the first ceiling beam is not reconstructed,as well as the lights beyond are not detected (Fig. 12(d) shows ground truth).

fined by the query images showed in Fig. 1. Indeed the same procedure can be applied to any other custom object or image patch in the scene. We compare the performance of our method (HOG descriptor + rectification), SIFT [20] on the original equirectangular images and SIFT on our rectified images.

Our approach (HOG RECT) retrieves almost all the occurrences (RO-Ground truth), outperforming the standard detector based on scale-invariant SIFTs [20] both in terms of *True Positives* than in terms of *Average Precision*. Our rectification also positively affects other methods in a minor way (SIFT RECT). In the case on the *Led-Light* query our method (HOG RECT) performs as SIFTs on rectified images concerning the average precision, although it is still better in terms of occurrences found (100%). Actually the *Led-Light* query is not a real flat object but occupies a significant volume in the scene (thickness of 10cm). As a further proof of the effectiveness of our system we discovered, during our tests, that the provided existing-conditions plans for the *Office* floor (Fig. 7) had many inaccurate light and wall positions.

In Tab. 3 and Fig. 11 we show results on additional scene types from the public *SUN360* dataset [30]. Since no real ground truth is available for *SUN360* data, reconstruction is necessary up-to-scale. Query objects have been instead manually labeled on the images to provide ground truth for detection. We choose different target objects to stress the system, ranging between small sprinklers, smoke sensors, speakers, air ducts and even particular floor decals with notably deformations (i.e. butterfly-like tiles in a childroom). For an easy comparison we indicate for each scene the original SUN360 naming. Also in these use-cases our method achieves solid performances, even compared to competing methods. The number of retrieved occurrences in Tab. 3, no matter they are true or false positives, depends on two parameters: a matching threshold and a maximum number of retrieved occurrences, arbitrarily set to 500 for this test. Since we are looking to retrieve an ordered list of matches, we set these two numbers to very conservative values, in order to avoid missing true positives. For this reason we measure the quality of the algorithm with the Average Precision (*AP*), and not by using Precision/Recall statistics on the final set. The high values of *AP* proves that the true positives are among the first retrieved items; for instance, for *Hospital arziggvjlludoe* dataset, although we have 495 *False Positives*, all the *True Positives* are the first 5 elements of the retrieved list, producing an *AP* of 100%.

6.5. Limitations and failure cases

Based on the *Augmented Manhattan World* assumption [37], our approach works only on scenes described by 3D vertical and horizontal planes (but differently from *Manhattan World*, we do not require vertical planes to be orthogonal with respect to each other).

Since many indoor scenes meet this assumption [38, 39] this limitation is generally acceptable. Fig. 12(a) shows an example of environment not compatible with our method, due to the sloped ceiling (i.e. *aapcyahhkcfvkc* from *SUN360* dataset).

However, even under these limitations there are cases where the method may not work. As highlighted in Sec. 4.2 in-fact, a large fraction of the ceiling edges and also of the floor must be visible and recognizable. Various factors can limit the detection of the boundaries, such as the lighting conditions and the quality of the image. Anyway, even in the presence of recognizable structures in the image there may be cases where the method could fail. The case in Fig. 12(b), form example, should be formally under our assumptions, as well as the quality of the image clearly allows the detection of structural features. Instead, the strong discontinuities in the ceiling and floor zones lead to a wrong reconstruction of the room (Fig. 12(c)), where the first beam in the ceiling is recognized by our system as a room boundary (Fig. 12(d) shows ground truth), as well as the lights beyond can not be detected and mapped. Although our reconstruction is limited by non-negligible assumptions, the resulting approximation can provide an effective geometric context to enhance the detection of the boundary-mounted objects, even when only one image is available for the environment.

7. Conclusions

We have presented a novel and practical approach to automatically retrieve an augmented indoor representation, starting from a small set of overlapping spherical images that can be quickly captured with commodity cameras.

Our main advances are in two principal areas. First of all, we expand over purely single view by exploiting multiple views to automatically reconstruct global and consistent multi-room environments through geometric reasoning. The method removes the limitations of single-view-per-room approaches, which typically require manual stitching or auxiliary data to align partial reconstructions, while requiring much less acquisition burden than full multi-view approaches requiring dense coverage.

Second, we exploit the reconstructed model to enhance image-based search for the elements located on room boundaries. This makes it possible to effectively detect the 3D position and shape of relevant wall, floor, and ceiling-mounted objects by looking for templates on undistorted images mapped on the room boundary polygons.

As a result, our system is an important step to quickly and automatically draft a rich as-built model coupled with existing conditions. The method can be easily combined with orthogonal solutions for 3D clutter detection [40] in order to create a complete furnished 3D model.

In our future work, we will explore how to integrate in our pipeline further detectors for generic object categories, such as, for example, the recent region-based fully convolutional networks [41] or deformable CNN [42], or to exploit spherically rendered views of the recovered scene to build a learning-based pipeline and do away with the semi-automatic object selection. We will also exploit how to efficiently use more images to remove some constraints on the methods (e.g., the *Manhattan World* assumption), expanding the approach to more complex buildings, such as ancient ones with curved walls and ceilings.

Given the increasing diffusion and performance of modern mobile devices, such as the mobile spherical cameras, we foresee in our future work to overcome these strict assumptions by taking advantage of many more panoramic images and their multi-view 3D clues, thus expanding scene recovery capability.

Acknowledgments. This work was partially supported by projects VIGEC and 3DCLOUDPRO. The authors also acknowledge the contribution of Sardinian Regional Authorities.

References

[1] American Building Calculations, . As-built vs. existing-conditions plans and drawings. "http://www.abcalc.biz/articles/as-built-existing-condition-plans/"; 2017.

[2] Fu, Q, Chen, X, Wang, X, Wen, S, Zhou, B, Fu, H. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. ACM Trans Graph 2017;36(6):201:1–201:13.

[3] Huang, J, Dai, A, Guibas, L, Niessner, M. 3dlite: Towards commodity 3d scanning for content creation. ACM Trans Graph 2017;36(6):203:1–203:14.

[4] Wang, ML, Lin, HY. Object recognition from omnidirectional visual sensing for mobile robot applications. In: 2009 IEEE International Conference on Systems, Man and Cybernetics. 2009, p. 1941–1946.

[5] Iraqui, A, Dupuis, Y, Boutteau, R, Ertaud, JY, Savatier, X. Fusion of omnidirectional and ptz cameras for face detection and tracking. In: 2010 International Conference on Emerging Security Technologies. 2010, p. 18–23. doi:10.1109/EST.2010.16.

[6] Kang, S, Roh, A, Nam, B, Hong, H. People detection method using graphics processing units for a mobile robot with an omnidirectional camera. Optical Engineering 2011;50:50 – 50 – 9. URL: https://doi.org/10.1117/1.3660573. doi:10.1117/1.3660573.

[7] Cinaroglu, I, Bastanlar, Y. A direct approach for human detection with catadioptric omnidirectional cameras. In: 2014 22nd Signal Processing and Communications Applications Conference (SIU). 2014, p. 2275–2279. doi:10.1109/SIU.2014.6830719.

[8] Cinaroglu, I, Bastanlar, Y. A direct approach for object detection with catadioptric omnidirectional cameras. Signal, Image and Video Processing 2016;10(2):413–420. URL: https://doi.org/10.1007/s11760-015-0768-2. doi:10.1007/s11760-015-0768-2.

[9] Zhang, Y, Song, S, Tan, P, Xiao, J. Panocontext: A whole-room 3d context model for panoramic scene understanding. In: Fleet, D, Pajdla, T, Schiele, B, Tuytelaars, T, editors. Computer Vision – ECCV 2014. Cham: Springer International Publishing; 2014, p. 668–686.

[10] Yang, H, Zhang, H. Efficient 3D room shape recovery from a single panorama. In: Proc. IEEE CVPR. 2016, p. 5422–5430.

[11] Zingg, S, Scaramuzza, D, Weiss, S, Siegwart, R. MAV navigation through indoor corridors using optical flow. In: Proc. IEEE IROS. 2010, p. 3361–3368.

[12] Im, S, Ha, H, Rameau, F, Jeon, HG, Choe, G, Kweon, IS. All-around depth from small motion with a spherical panoramic camera. In: Leibe, B, Matas, J, Sebe, N, Welling, M, editors. Computer Vision – ECCV 2016. Cham: Springer International Publishing. ISBN 978-3-319-46487-9; 2016, p. 156–172.

[13] Pintore, G, Garro, V, Ganovelli, F, Agus, M, Gobbetti, E. Omnidirectional image capture on mobile devices for fast automatic generation of 2.5D indoor maps. In: Proc. IEEE WACV. 2016, p. 1–9.

[14] Cabral, R, Furukawa, Y. Piecewise planar and compact floorplan reconstruction from images. In: Proc. CVPR. 2014, p. 628–635.

[15] Almazán, J, Gordo, A, Fornés, A, Valveny, E. Segmentation-free word spotting with exemplar svms. Pattern Recognition 2014;47(12):3967 – 3978.

[16] Malisiewicz, T, Gupta, A, Efros, AA. Ensemble of exemplar-svms for object detection and beyond. In: Proceedings of the 2011 International Conference on Computer Vision. ICCV '11; Washington, DC, USA: IEEE Computer Society; 2011, p. 89–96.

[17] Sankar, A, Seitz, S. Capturing indoor scenes with smartphones. In: Proc. ACM UIST. 2012, p. 403–412.

[18] Pintore, G, Ganovelli, F, Pintus, R, Scopigno, R, Gobbetti, E. Recovering 3d indoor floor plans by exploiting low-cost spherical photography. In: PG2018 Short Papers Proceedings. 2018,To appear.

[19] Zheng, L, Yang, Y, Tian, Q. Sift meets cnn: A decade survey of instance retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017;PP(99):1–1.

[20] Vedaldi, A, Zisserman, A. Object instance recognition. "http://www.robots.ox.ac.uk/~vgg/practicals/instance-recognition/index.html"; 2018.

[21] Carroll, R, Agrawal, M, Agarwala, A. Optimizing content-preserving projections for wide-angle images. ACM Trans Graph 2009;28(3):43:1–43:9.

[22] Karaimer, HC, Baçtanlar, Y. Car detection with omnidirectional cameras using haar-like features and cascaded boosting. In: 2014 22nd Signal Processing and Communications Applications Conference (SIU). 2014, p. 301–304.

[23] Gandhi, T, Trivedi, MM. Video based surround vehicle detection, classification and logging from moving platforms: Issues and approaches. In: 2007 IEEE Intelligent Vehicles Symposium. 2007, p. 1067–1071.

[24] Arteta, C, Lempitsky, V, Noble, JA, Zisserman, A. Learning to detect partially overlapping instances. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '13; Washington, DC, USA: IEEE Computer Society. ISBN 978-0-7695-4989-7; 2013, p. 3230–3237.

[25] Dalal, N, Triggs, B. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); vol. 1. 2005, p. 886–893 vol. 1.

[26] Felzenszwalb, PF, Huttenlocher, DP. Efficient graph-based image segmentation. Int J Comput Vision 2004;59(2):167–181.

[27] Crowley, E, Zisserman, A. The state of the art: Object retrieval in paintings using discriminative regions. In: Proceedings of the British Machine Vision Conference. BMVA Press; 2014,.

[28] Shrivastava, A, Malisiewicz, T, Gupta, A, Efros, AA. Data-driven visual similarity for cross-domain image matching. ACM Trans Graph 2011;30(6):154:1–154:10.

[29] Pintus, R, Yang, Y, Rushmeier, H, Gobbetti, E. An automatic word-spotting framework for medieval manuscripts. In: Proc. Digital Heritage. 2015, p. 5–12.

[30] Xiao, J, Ehinger, KA, Oliva, A, Torralba, A. Recognizing scene viewpoint using panoramic place representation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2012, p. 2695–2702.

[31] Achanta, R, Shaji, A, Smith, K, Lucchi, A, Fua, P, Susstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE TPAMI 2012;34(11):2274–2282.

[32] Agus, M, Jaspe Villanueva, A, Pintore, G, Gobbetti, E. PEEP: Perceptually enhanced exploration of pictures. In: Proc. 21st International Workshop on Vision, Modeling and Visualization (VMV). 2016, p. 93–100.

[33] Douglas, DH, Peucker, TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization 1973;10(2):112–122. doi:10.3138/FM57-6770-U75U-7727.

[34] Kangni, F, Laganiere, R. Orientation and pose recovery from spherical panoramas. In: 2007 IEEE 11th International Conference on Computer Vision. 2007, p. 1–8.

[35] Grundland, M, Dodgson, NA. Decolorize: Fast, contrast enhancing, color to grayscale conversion. Pattern Recognition 2007;40(11):2891 – 2896.

[36] Chang, A, Dai, A, Funkhouser, T, Halber, M, Niessner, M, Savva, M, et al. Matterport3d: Learning from rgb-d data in indoor environments. International Conference on 3D Vision (3DV) 2017;.

[37] Schindler, G, Dellaert, F. Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.; vol. 1. 2004, p. I–203–I–209 Vol.1.

[38] Schwing, AG, Urtasun, R. Efficient exact inference for 3d indoor scene understanding. In: Fitzgibbon, A, Lazebnik, S, Perona, P, Sato, Y, Schmid, C, editors. Computer Vision – ECCV 2012. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-33783-3; 2012, p. 299–313.

[39] Zeisl, B, Zach, C, Pollefeys, M. Stereo reconstruction of building interiors with a vertical structure prior. 2011.

[40] Xu, J, Stenger, B, Kerola, T, Tung, T. Pano2cad: Room layout from a single panorama image. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). 2017, p. 354–362.

[41] Dai, J, Li, Y, He, K, Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16; USA: Curran Associates Inc. ISBN 978-1-5108-3881-9; 2016, p. 379–387. URL: http://dl.acm.org/citation.cfm?id=3157096.3157139.

[42] Dai, J, Qi, H, Xiong, Y, Li, Y, Zhang, G, Hu, H, et al. Deformable convolutional networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). 2017, p. 764–773.