

Automatic generation of 3D outdoor and indoor building scenes from a single image

Georgios Vouzounaras · Petros Daras ·
Michael G. Strintzis

© Springer Science+Business Media, LLC 2011

Abstract In this paper, a novel approach for creating 3D models of building scenes is presented. The proposed method is fully automated and fast, and accurately reconstructs both outdoor images of a building and indoor scenes, with perspective cues in real-time, using only one image. It combines the extracted line segments to identify the vanishing points of the image, the orientation, the different planes that are depicted in the image and concludes whether the image depicts indoor or outdoor scenes. In addition, the proposed method efficiently eliminates the perspective distortion and produces an accurate 3D model of the scene without any intervention from the user. The main innovation of the method is that it uses only one image for the 3D reconstruction, while other state-of-the-art methods rely on the processing of multiple images. A website and a database of 100 images were created to prove the efficiency of the proposed method in terms of time needed for the 3D reconstruction, its automation and 3D model accuracy and can be used by anyone so as to easily produce user-generated 3D content: http://3d-test.iti.gr:8080/3d-test/3D_recon/

Keywords Single-view 3D reconstruction · Projective correction · Vanishing point detection · Uncalibrated camera

G. Vouzounaras · M. G. Strintzis
Information Processing Laboratory, Department of Electrical and Computer,
Engineering Aristotle University of Thessaloniki, GR54124 Thessaloniki, Greece

G. Vouzounaras
e-mail: gvouzounaras@yahoo.gr

M. G. Strintzis
e-mail: strintzi@eng.auth.gr

P. Daras (✉) · M. G. Strintzis
Informatics and Telematics Institute, Centre for Research and Technology Hellas,
GR57001 Thessaloniki, Greece
e-mail: daras@iti.gr

1 Introduction

One of the most challenging topics in Computer Vision is to identify the way that computers can perceive 3D objects from 2D images. The process to recover the 3D information of an object from 2D images is called 3D reconstruction. When a human looks at a picture is really easy for him/her to understand how the scene is constructed by using his/her experience of the man-made world. On the other hand, computers and robots do not have such ability. They can only see the pixels of the image. Even when more intelligent algorithms are used they can find lines and planes but they are incapable of combining all these contents together and form a structure. As a scientific discipline, computer vision is concerned to deal with the theory of building artificial systems that obtain information from images. State-of-the-art methods that have been reported in the literature and use more than one images for achieving 3D reconstruction follow completely different concept from the proposed algorithm. These algorithms do not depend on criteria such as lines and planes extraction. In order to reconstruct a scene they take advantage of the position of specific points in the different views of the scene. Consequently, having multiple images of a scene the need to make assumptions about the construction of the scene is reduced or eliminated. Thus, these methods could reconstruct greater variety of pictures however they face two main disadvantages: firstly, they require greater amount of input data to perform the reconstruction, which are not always available, especially when dealing with real life cases and secondly, the time required for such a reconstruction is higher when compared to the real-time method proposed in this paper.

3D reconstruction from a single image is definitely a challenging problem and many researchers have dealt with it. The purpose of the 3D reconstruction from a single image is what humans can do to perceive 3D information. Both automated methods and methods that need user's interaction exist. The most well known commercial method, which is based on user interaction, is Google Sketch-Up [6]. Other algorithms that work with the assistance from the users are Façade [4], Tour into the Picture [10], Single View Metrology [3] and ATIP.

This paper focuses on fully automated methods, which can be classified into three main categories depending on different assumptions: depth reconstruction, geometrical projection and image content-based methods. The methods based on depth analyze the distance between each pixel and the camera, and the methods based on the image content try to find the similar area and combine the pixels which belong to the same object. However, simple observation cannot show the effectiveness of the methods to solve the problem. For example, some methods are more effective for urban images and not for natural scenes. "Automatic Photo Pop-Up" [8, 9] is the first fully automated method for 3D reconstruction. The method is based on training data and works for outdoor photos, both man-made and natural. The computational time of this method for an 800×600 image is about 1.5 min using unoptimized MATLAB code on a 2.13 GHz Athalon machine [9]. The main drawbacks of this algorithm is that it relies on statistics, thus wrong models may be produced and it cannot be used for real-time 3D reconstructions. One of the most well known algorithms is "Make 3D" by Saxena et al. [18–20], where 3D depth estimation from a single still image is automatically estimated. This method is based on a collection of a training set of monocular images of unconstructed indoor and outdoor environments, which include forests, sidewalks, trees, buildings, etc. Then, a supervised learning technique is applied to predict the value of the depthmap of the image. The distinct advantage of this method is that there are no any assumptions for any specific features. However, the

multiscale Markov Random Field (MRF) model used is complex and time consuming for real-time 3D reconstruction from a 2D image. Furthermore, in [13] geometric constraints are used to recover an indoor structure from a single image. More specifically, several assumptions are generated about the structure of the image by joining the line segments, that were previously extracted, and producing angles and joining angles to create the whole structure. An evaluation process then takes place in order to select the best representation of the indoor scene from the previous assumptions and the 3D model reconstruction follows. The algorithm produces good results even if the scene has many occluding objects but is functional only in indoor images. In the same line with this work, [11] deals with the problem of 3D reconstruction of indoor scenes with perspective cues. With this method an accurate model is produced but without resolving the problem of projective distortion and without using texture mapping at the 3D models. Another recent algorithm is presented in [5], where a dynamic Bayesian network model capable of recovering 3D information from many images is presented. The model assumes a “floorwall” geometry of the scene and is trained to recognize the floor-wall boundary in each column of the image. The method depends on the calibration of a camera and the parameters of the camera are not always available.

In this paper, 3D geometry is derived from line segments in one-point perspective indoor image that consists largely of orthogonal planes, and from the exterior of buildings. The aim of this work is to present a novel algorithm for real-time 3D scene generation from a single image able to be used for both indoor and outdoor images of buildings, to be fully automated, real-time, without any user interaction. The proposed method is based on detecting and fitting the line segments correctly so as to form the structure of the indoor or outdoor building scene. In addition, it deals with the problem of perspective distortion and by correcting it, it generates accurate 3D models. The work presented in this paper is fully aligned with the scope of this special issue since it provides an easy to use tool for automatic creation of user-generated 3D content using a single image (http://3d-test.iti.gr:8080/3d-test/3D_recon/).

The rest of this paper is organized as follows: In Section 2 all the distinct steps of the proposed algorithm are presented in detail along with innovations which substantially improve the current state-of-the-art. In Section 3, the relevant website is shown. Finally, conclusions are drawn in Section 4.

2 Proposed algorithm

The proposed algorithm consists of several steps. For the sake of completeness all these steps are explained in the following paragraphs along with our innovations.

2.1 Edge detection

The changes of brightness between neighboring regions in a binary image are called edges and are usually related to the different attributes of the three dimensional objects such as changes of texture, depth, limits of objects, different lighting and reflection. Thus, using edge detection algorithms is possible to obtain the attributes of the objects presented in the image.

The algorithm that was proposed by Canny [2] for edge detection is assumed to be the most accurate one with the presence of white noise and used in this paper.

2.2 Line extraction

After detecting the edges, the proposed algorithm tries to join the edges and form lines. Due to imperfections in either the image data or the edge detector there may be missing points or pixels on the desired line as well as spatial deviations between the ideal line and the noisy edge points. In order to deal with the latter, the well known Hough transform is used [12]. The Hough transform consists of parameterizing a description of a feature at any given location in the original image's space. A mesh in the space defined by these parameters is then generated, and at each mesh point a value is accumulated, indicating how well an object generated by the parameters fits the given image. The Hough transform uses parametric description of simple geometric shapes (curves) to reduce the computational complexity of search in a binary image. Examples are presented in the following figures (Figs. 1 and 2) after applying the canny edge detector and the line extractor using the Hough transform at a building and at an indoor scene, respectively.

2.3 Vanishing points estimation

In a perspective photo the parallel lines of the world, which are not parallel to the image plane appear to converge to a point, which is called vanishing point. Vanishing points can be either finite, which are real points inside or outside the image plain, or infinite, which are ideal points at infinity. Vanishing points which lie on the same plane in the scene define a line in the image, the so-called vanishing line.

The understanding and interpretation of an environment that is constructed by humans can be simplified by finding the vanishing points because such an environment is composed of many parallel lines and orthogonal edges. In an indoor environment the examples of shelves, doors, windows and corridor boundaries and in an outdoor environment the examples of streets, buildings and pavements satisfy this assumption. This means that vanishing points provide strong cues for inferring information about the 3D structure of a scene. If the camera geometry is known, each vanishing point corresponds to an orientation in the scene and vice versa [16].

Generally speaking, automatic vanishing point detection is achieved by clustering the extracted lines in separate sets depending on the point that they converge, which is considered as a vanishing point of the corresponding address space. Rother [16] chooses as accumulator cells the intersection points of all pairs of line segments. The contribution of all line segments is computed for each possible vanishing point with a "voting" process. The vote v of each accumulator cell, i.e. vanishing point, is computed by:

$$v = w_1 + \left(1 - \frac{a}{a_0}\right) + w_2 \frac{d_s}{d_{max}} \quad (1)$$

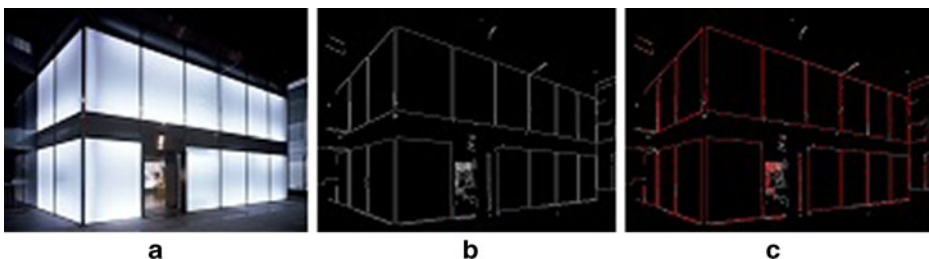


Fig. 1 **a** Input image, **b** edge detection (low threshold=100, high threshold=200), **c** line extraction

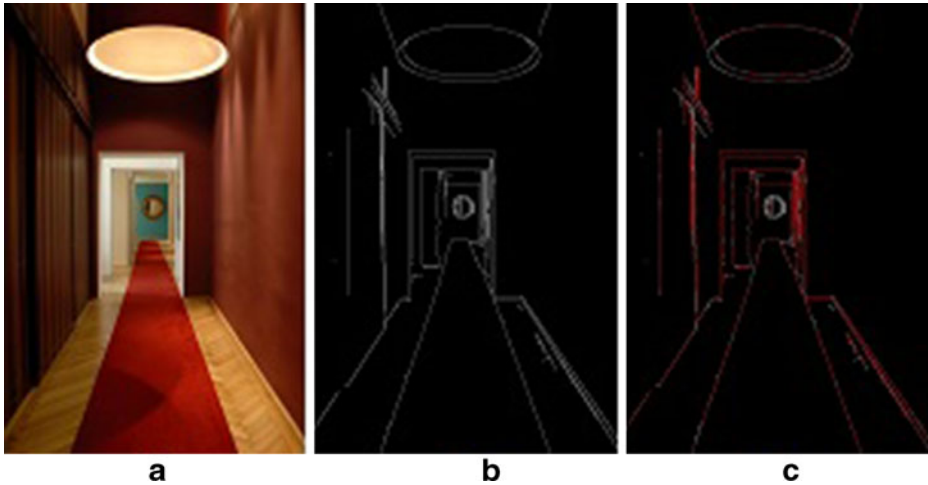
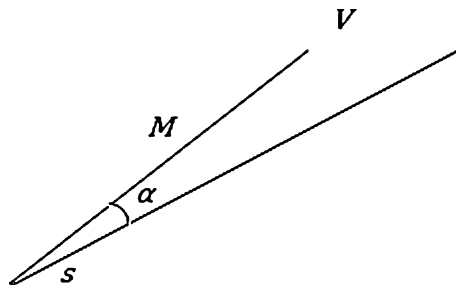


Fig. 2 **a** Input image, **b** edge detection (low threshold=70, high threshold=210), **c** line extraction

where a is the angle between the line segment s and the line that connects the midpoint M of the line segment s with the vanishing point V , a_0 is the threshold for the angles a , d_s is the length of s and d_{\max} is the longest line segment (Fig. 3). Every line segment with an angle $a > a_0$ does not participate in the voting process.

The total vote of an accumulator cell a is given by the sum of the Eq. 1 for all the line segments. The weights w_1 and w_2 were empirically defined in [17]. After the first step, which is called the accumulation step, the search step follows. At this step the algorithm seeks for the three mutual orthogonal vanishing points. These points have to satisfy the following three criteria: orthogonal criterion, camera criterion and vanishing line criterion. The first criterion, which was also applied in [21], is based on the fact that two vanishing points correspond to two perpendicular directions on space, only if the two vectors that connect the projection center with these two points are perpendicular to each other. In the case of three finite vanishing points the orthocenter of the triangle that is formed by the three vanishing points is the principal point of the image and the first criterion is fulfilled if the triangle is acute-angled. Respectively, the orthogonal criterion can be configured for the cases of one or two infinite vanishing points. The camera criterion is fulfilled when the principle point and the focal length are inside a certain range in the case they are calculable. Finally, the vanishing line criterion checks that a line segment which votes for two vanishing points is close to the vanishing line of these points. The trinity that satisfies the above criteria and has obtained the most votes addresses the three orthogonal vanishing points and also the three primary directions in the image.

Fig. 3 The angle α which is formed by the line segment s and the line segment VM , M is the midpoint of s and V is the candidate vanishing point



Nowadays the algorithm proposed by Rother is one of the most reliable ones and it is widely used for the localization of the vanishing points in non real-time applications, like the work presented in [1]. However, the proposed algorithm focuses on real-time applications. Consequently, the following alterations were made in Rother's method: Firstly, the accumulation step remains the same and (1) is used. Then, the accumulator cell with the highest vote is traced and is chosen to be the first vanishing point. Next, the line segments that vote for this accumulator cell are omitted and the procedure is repeated. Again, the cell with the highest vote is taken and this is the second vanishing point. Finally, the process is repeated for the last time and the third vanishing point is extracted. The proposed algorithm is efficient when there are lines to all orthogonal directions, which is actually obligatory for 3D reconstructions. Figure 4a depicts the three vanishing points of an outdoor building scene, while Fig. 4b shows the result of applying the proposed algorithm for vanishing point detection (Fig. 5).

2.4 Automatic detection of image orientation

The aim of this step is the automatic recognition and characterization of images as indoor or outdoor in order to follow the corresponding steps below. To fulfill this step the vanishing points and some reasonable assumptions are used. It has been observed that for the exterior scenes the vanishing point algorithm detects two vanishing points in the finite space and one infinite, as was also shown in Fig. 4, whereas for the indoor building scenes with perspective cues, it detects one finite vanishing point. Therefore, with a plain examination of the three vanishing points the algorithm could predict the orientation of the image.

2.5 Automatic planes detection—outdoor building image

In the case of the outdoor building scene, at least two planes exist (three if the ground is also depicted). Those planes are found using geometric constraints: Initially, the connection type of the two facades is found (Fig. 6a). Then, the purpose of the algorithm is to find the line that connects the two facades (Fig. 6b, Algorithm 2).

Since the entire crop line is not always visible in the image, some assumptions are introduced: Firstly, the algorithm seeks for the highest point in the image that lies on the

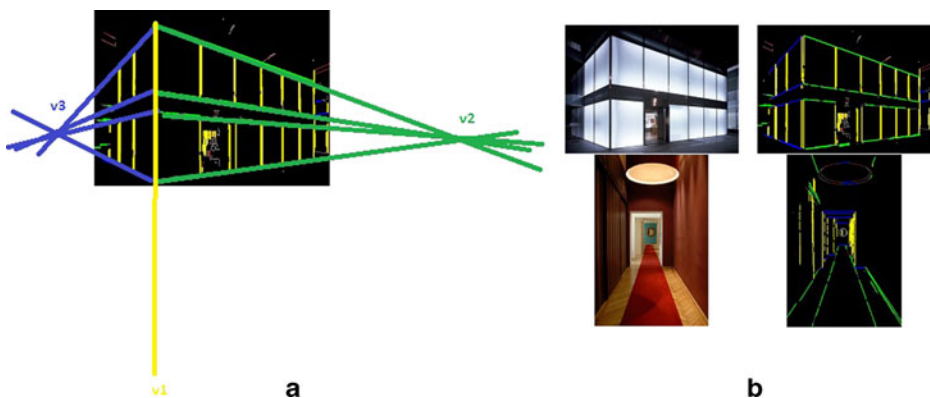


Fig. 4 **a** Three orthogonal vanishing points of an outdoor building scene. **b** Detection of the lines' orientation (red lines do not belong in any of the three principal directions)

Algorithm 1 Vanishing point estimation

```

for  $i=1$  to 3
    Find the intersection points of all non-collinear
    line segments.
    for all intersection points
        for all accepted line segments compute  $\text{vote}(c_i)$  (1)
        end for
    end for
    if  $i = 1$ 
        find the maximum vote, set  $v_1 = \max(\text{vote}(c_i))$ 
        omit the lines that vote for this cell and mark them as
        yellow
    else if  $i = 2$ 
        find the maximum vote, set  $v_2 = \max(\text{vote}(c_i))$ 
        omit the lines that vote for this cell and mark them as
        green
    else
        find maximum vote, set  $v_3 = \max(\text{vote}(c_i))$ 
        omit the lines that vote for this cell and mark them as
        blue
    end if
end for
    
```

Fig. 5 Detection of the three vanishing points, $v_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1, 2, 3$

Fig. 6 a The two types of junction, **b** planes detection algorithm



Λ junction

V junction

a

Algorithm 2 Planes detection

```

if  $v_2, v_3$  out of the image plane
    (like Figure 7) crop line is
    the longest of the vertical
    lines
else if  $v_2, v_3$  in the image plane
    crop line is the shortest of
    the vertical lines
end if
    
```

b

vertical lines, i.e. point 1 (Fig. 7). This point is difficult to be occluded thus, it is picked as the first crop point. Secondly, the algorithm seeks for the highest point that lies on the other lines (green and blue) in order to check if point 1 is truly the highest point. If this is not the case, then the highest point from the latter search is picked as point 1. Next, the algorithm seeks for the lower point, i.e. point 2 (Fig. 7). Point 2 is more likely to be occluded by many objects, e.g. cars, humans, trees, etc. Therefore, this point will not always lie on the vertical lines. Thus, the procedure continues by searching the lower point of all lines and then it is projected to the vertical line that passes through the higher point (point1). The equivalent procedure is being followed for building with Λ junction.

2.6 Automatic planes detection—indoor building scene

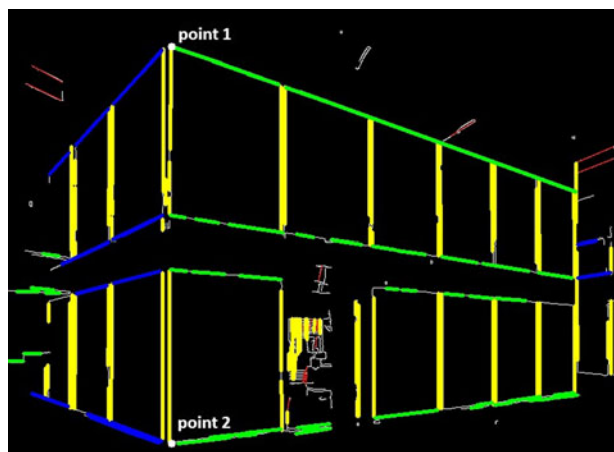
In the case that the image has been recognized as an indoor one, the algorithm identifies the structure of the indoor scene, the ceiling, the floor, the walls, etc. In this section some assumptions introduced in [11] are used.

2.6.1 Floor detection

The first step is to find the boundaries of the floor. Before searching for the floor, it is useful to set three labels to the lines, “vertical”, “horizontal” and “converging”. “Converging” lines are those which converge to the finite vanishing point (green lines in Fig. 7). It is obvious that the floor will belong to the horizontal lines. The horizontal line that is closest to the horizon is picked as the floor line. The horizon is the line that passes through the finite vanishing point. In the case that the floor has not been detected the algorithm searches the floor line indirectly. More specifically, the vertical lines which are closest to the finite vanishing point are searched and the floor is defined as the line which connects the lower points of these vertical lines.

The second step is to find the left and the right boundaries of the floor. Firstly, the “converging” lines that are below the horizon are collected and they are further divided into the lines left and right from the finite vanishing point. Next, all the intersection points of these lines with the estimated floor are found along with the corresponding line equations. From the intersection points the rightmost of the ones that are in the left part, the leftmost of those that are in the right part and the corresponding line equations are selected. Thus, the

Fig. 7 Detection of the two facades of the building with V junction

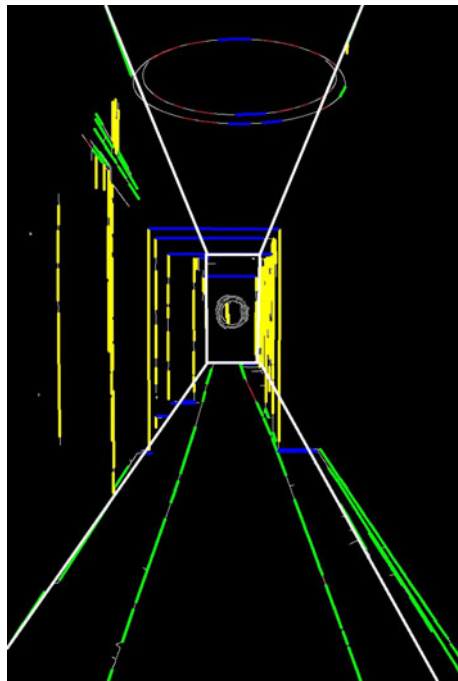


floor is the line that connects the latter points and the boundaries are found by the line equations of the corresponding lines. If the algorithm does not find any of these boundaries, the left and the right corner of the image are selected, respectively.

2.6.2 Ceiling detection

The following step of the plane detection algorithm describes the ceiling detection. The ceiling is found with respect to the floor position. It does not simply choose the horizontal line that lies closest above the horizon as the ceiling, because there may be windows, door frames, and wall decorations. First of all, the angles between the finite vanishing point and the upper corners of the image are obtained, which define a range of lines that may be located on the ceiling. Next, the intersection points between the vertical lines that pass through the endpoints of the floor and the orthogonal lines above the horizon are found. The intersection points that lie on a “converging” line, which is between the previously defined range, are selected. Finally, the line which is closer to the horizon and passes through the latter intersection points is chosen. Furthermore, the line equations of the “converging” lines are stored in order to find the endpoints of the ceiling. When there are not any orthogonal lines above the horizon, the intersection points of the “converging” lines with the vertical lines that pass through the endpoints of the floor are collected and those which are closest to the horizon are selected as the endpoints of the ceiling. In the case that no “converging” line exists inside the defined range it is assumed that the ceiling is not depicted in the image. Having found the floor line, the ceiling line and their boundaries, the two vertical walls are automatically found. Figure 8 depicts an image after the application of the plane detection algorithm. The planes are separated by the white lines.

Fig. 8 Planes detection algorithm of an indoor building scene



2.7 Image rectification

According to [14, 15], the determination of the vanishing line of one plane allows for the recovery of the affine properties from an image. Two vanishing points of a plane are sufficient to define the vanishing line (\mathbf{l}_∞) of that plane. The projective transformation that connects the coordinates between the image and the plane is called homography and is given by the following equation [7]:

$$H \cdot x = X \text{ or } \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ 1 \end{pmatrix} \quad (2)$$

\mathbf{H} is a 3×3 matrix with 8 elements, $(x_1, x_2, 1)^T$ is a point in the image and $(X_1, X_2, 1)^T$ the corresponding point in the real world. The vector $(h_7, h_8, 1)^T$ describes the vanishing line \mathbf{l}_∞ of the plane and it can be calculated by the cross product of two vanishing points. Once the image line at infinity is identified, it is then possible to make affine measurements on the original plane. If, in addition, a vanishing point for a direction not parallel to the plane is identified, then affine properties can be computed for the 3-space of the perspectively imaged scene. Specifically, according to [14, 15] every homography can be considered to consist of two separate transformations, a “metric” (\mathbf{M}) and a non-metric (\mathbf{N}):

$$\mathbf{H} = \mathbf{M} \cdot \mathbf{N} \quad (3)$$

The transformation matrix \mathbf{M} is a similarity transformation, which can be a reflection of the form

$$\mathbf{M} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \quad (4)$$

where \mathbf{R} is a rotation matrix, \mathbf{t} a translation vector and s an isotropic scaling. There are four degrees of freedom in \mathbf{M} .

The second non-metric part \mathbf{N} of the homography \mathbf{H} is:

$$\mathbf{N} = \begin{pmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & 1 \end{pmatrix} \quad (5)$$

which has also four degrees of freedom.

The metric properties of the plane, such as angle and relative length, are invariant to \mathbf{M} since it is a similarity transformation. The complete metric rectification is thus known when \mathbf{N} is computed. The transformation matrix \mathbf{N} can be directly determined by the circular points \mathbf{I} and \mathbf{J} , which lie on the line at infinity and have coordinates $(1, \pm i, 0)^T$. Their relation with the homography \mathbf{H} is:

$$\begin{aligned} \mathbf{I} &= \mathbf{H}^{-1}(1, i, 0)^T = (\alpha - i\beta, 1, -l_2 - \alpha l_1 + il_1\beta)^T \\ \mathbf{J} &= \text{conj}(\mathbf{I}) \end{aligned} \quad (6)$$

Obviously, the display of the circular points depends only on the non-metric part of the homography. Therefore, by identifying the points \mathbf{I} and \mathbf{J} of the image, the parameters α, β, l_1, l_2 , i.e. the matrix \mathbf{N} is determined. Alternatively, the matrix \mathbf{N} can be determined in two

stages by analyzing it in two further matrices **A** (affine transformation) and **P** (projective transformation):

$$\mathbf{N} = \mathbf{A} \cdot \mathbf{P} \tag{7}$$

where

$$\mathbf{A} = \begin{pmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & 1 \end{pmatrix} \tag{8}$$

The vector $\mathbf{l}_\infty=(l_1, l_2, 1)^T$ is the line at infinity of the plane and can be calculated by the cross product of the two vanishing points of that plane $\mathbf{l}_\infty=\mathbf{v}_1 \times \mathbf{v}_2$. The line equation is homogeneous and has two degrees of freedom that contain the whole perspective distortion of the plane. Thus, initially, the affine properties of the image have been recovered. Then, the matrix **A** must be computed so as to recover the angles and the length ratios of non-parallel lines. In the case of a pair of orthogonal directions and the aspect ratio in these directions, it is possible to further stratify the rectification by decomposing **A** as:

$$\mathbf{A} = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{R} \tag{9}$$

The three stages of decomposition, as mentioned in [15], have the purpose of:

- **R**: Rotate a direction to the horizontal axis.
- **A₁**: Transform a second direction to the vertical axis without changing the orientation of the horizontal axis.
- **A₂**: Set the aspect ratio of vertical and horizontal directions without changing the direction of either.

After the implementation of the projective matrix **P**, a rotation through the matrix **R** is applied in order to align one of the addresses to the horizontal axis. Multiplying a vanishing point with the matrix **P** it transforms to the form $\mathbf{v}_{1A} = (x, x, 0)^T$, where it defines direction. The vector \mathbf{v}_{1A} can be written as a unit norm direction vector $\mathbf{v}_{1A} = (\cos(\varphi), \sin(\varphi), 0)^T$ where φ is the angel that \mathbf{v}_{1A} makes with the horizontal axis. Thus, **R** is a matrix which rotates \mathbf{v}_{1A} to the horizontal axis: $\mathbf{R} \cdot \mathbf{v}_{1A} = (1, 0, 0)^T$:

$$\mathbf{R} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

If the angle between \mathbf{v}_{1A} and \mathbf{v}_{2A} is θ , \mathbf{v}_{2A} now makes an angle of $(\pi-\theta)$ with the vertical axis and the transformation

$$\mathbf{A}_2 = \begin{pmatrix} 1 & -\cot(\theta) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{11}$$

converts a second direction to the vertical axis without changing anything to the horizontal axis. The final transformation converts **A₁**:

$$\mathbf{A}_1 = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

where λ and μ are constants that scale the image in the horizontal and vertical axis, respectively. The line that crops the image into facades is measured after the application of the matrices \mathbf{P} , \mathbf{R} , \mathbf{A}_2 and thus, the scale factor μ is calculated. The other scale factor λ is similarly calculated with respect to the horizontal axis.

For the correction of the perspective distortion of an image of an outdoor building scene, the matrices are applied two times, one for each façade of the building. As for the indoor building scene, the matrices are applied four times, because the different planes which need rectification are four: the ceiling, the floor, the left and the right wall. Examples are shown in the figures that follow (Figs. 9, 10).

2.8 3D reconstruction

After matrix \mathbf{N} is calculated, the algorithm is ready to proceed to the 3D reconstruction of the image. For the reconstruction VRML (Virtual Reality Modeling Language) is used. The 3D models are constructed depending on the orientation of the building scene (indoor or outdoor) and then the rectified images from the previous section are mapped. By having the matrix \mathbf{N} the 3D coordinates are also obtained by simply applying this matrix to a point in the input image.

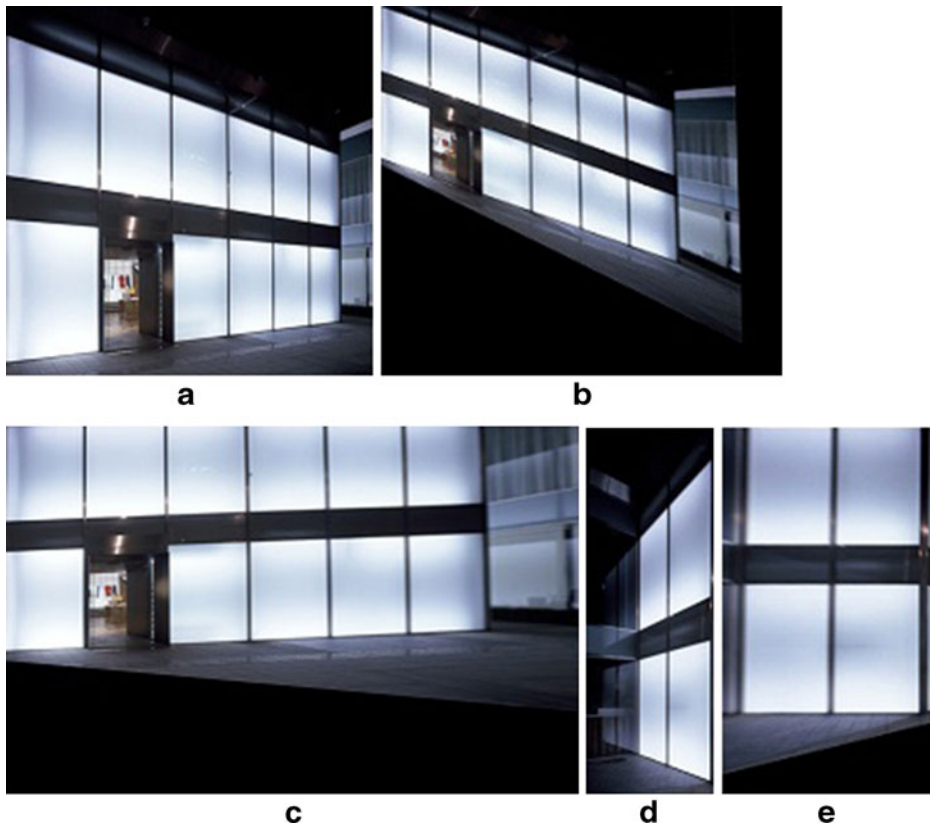


Fig. 9 **a** Cropped image with the left facade, **b** image after applying the matrix \mathbf{P} , **c** after applying the matrix $\mathbf{N} = \mathbf{A}\mathbf{P}$ (for the vanishing points $\mathbf{v}_1, \mathbf{v}_2$ as shown in Fig. 7), **d** Cropped image with the right facade, **e** after applying the matrix $\mathbf{N} = \mathbf{A}\mathbf{P}$ (for the vanishing points $\mathbf{v}_1, \mathbf{v}_3$ as shown in Fig. 4a)

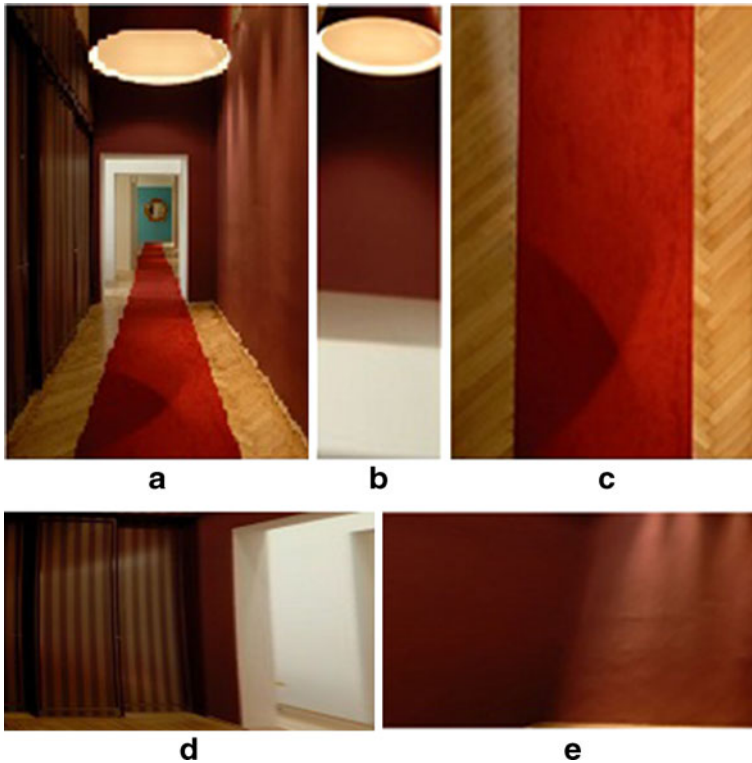


Fig. 10 a Input image, b ceiling, c floor, d left wall, e right wall after the rectification process

3 System development

In order to test the proposed algorithm, 50 indoor and 50 outdoor images were collected from the Internet. Some examples are depicted in Fig. 11.

The user can automatically generate his/her own 3D models via the website that has been developed for this reason http://3d-test.iti.gr:8080/3d-test/3D_recon/. At this website the user can create his/her own 3D reconstructed models by simply uploading an image and setting the thresholds as explained or use the existing images. A screenshot of the website is shown in Fig. 12.

A widely used method, which is based on the same assumptions as the proposed algorithm, is Google Sketch-Up [6]. It is an easy tool for both two-dimensional and three-dimensional drawings, which makes easy the production of 3D models able to be uploaded and used in Google Earth. This tool has been designed to effortlessly fulfill the needs of architects, civil engineers, interior designers etc. Among the most interesting features of this software is its ability to make 3D models out of 2D images. For example, the user is able to insert an image of a building scene either outdoor or indoor and by following the necessary steps to construct the equivalent 3D model (Fig. 13). However, the user interaction is inevitable. The proposed algorithm can produce the same 3D models as Google Sketch-Up but fully automated.

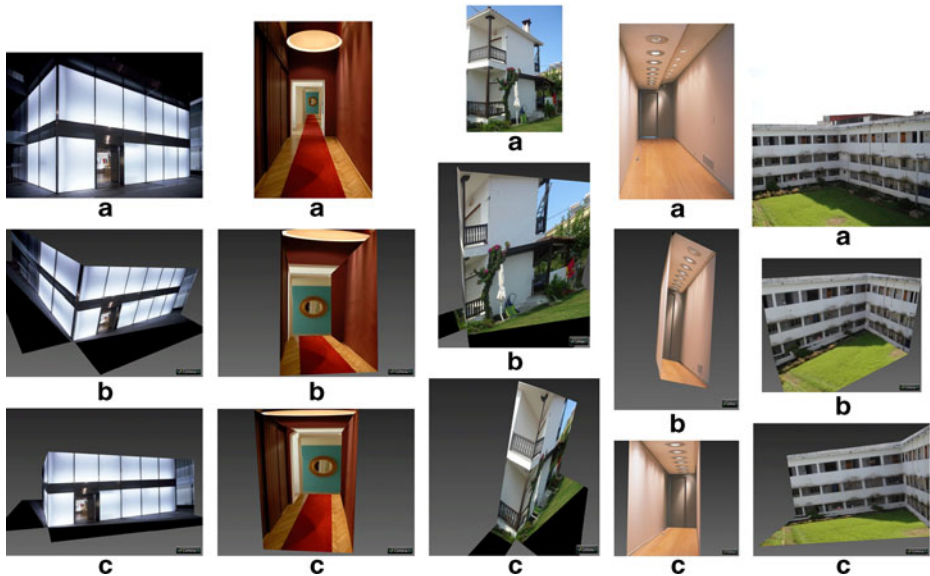


Fig. 11 a the first row depicts the Input images, b, c rows depict the corresponding 3D models

4 Conclusions

In this paper, a novel algorithm is proposed, able to create a 3D reconstruction of a building scene (indoor and outdoor), fully automated without any user intervention from a single image. The algorithm is free to be tested in the website http://3d-test.itigr:8080/3d-test/3D_recon/.

The proposed method manages to create 3D models in a fully automated manner. Furthermore, there is no need for camera calibration and the external and internal parameters of the camera are not necessary.

The problem that is attempted to be solved is new and challenging and different approaches have been proposed the past few years. Further research is needed in order to reconstruct all the possible indoor building scenes despite the presence of occluding objects. Moreover, the algorithm may be further extended to object or human recognition inside a scene, so as to fully reconstruct a scene using only a single image.

The purpose of a single image reconstruction method except for its clear research challenges, depicts the reality more accurately since by using such a method a building can be reconstructed with its real dimensions. Moreover, 3D models are widely used nowadays on the Internet either for navigation or for visualization and presentation of monuments and other cultural heritage sites from all over the world. For example, Google Earth uses extensively 3D models in order to make the navigation and presentation of such sites more fascinating. The single image reconstruction algorithms produce 3D models that have small size and can be distributed easily over the Internet.

3D Reconstruction Demo

Introduction Upload **Indoor examples** Outdoor examples Help Contact

Indoor image results

Select an image to display

Click on the selected image to display the 3D model (VRML)

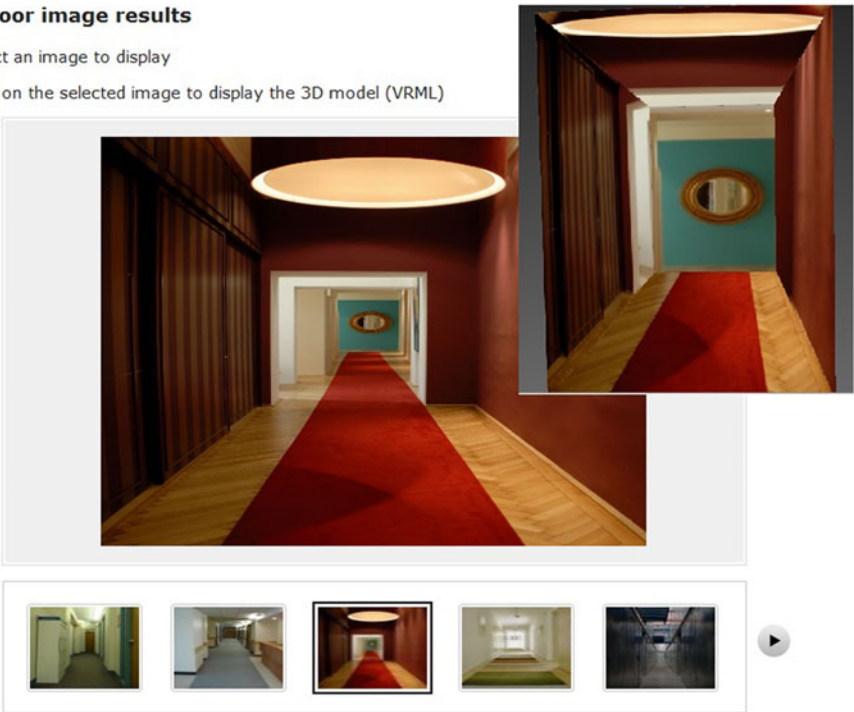


Fig. 12 The website for the 3D reconstructions



Fig. 13 a Presentation of Google Sketch-Up. b The produced 3D model

Generally speaking, 3D reconstruction is a necessary procedure for the architects. Nowadays, most of the projects of the architects are presented not only with 2D images and technical drawings but also with 3D models in order to capture the interest of the client and to visualize the buildings with more details.

Acknowledgments This work was supported by the EU FP7 project 3DLife Network of Excellence project, ICT-247688.

References

1. Aguilera DG, Gómez Lahoz J, Finat Codes J (2005) A new method for vanishing points detection in 3D reconstruction from a single view. International Workshop on 3D Virtual Reconstruction and Visualization of Complex Architectures (3D-ARCH '05), Venice
2. Canny J (1986) A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*
3. Criminisi IR, Zisserman A (2000) Single view metrology. *Int. J. Computer Vision*, 40(2):pp 123–148. <http://portal.acm.org/citation.cfm?id=365888>
4. Debevec PE (1996) Modeling and rendering architecture from photographs. PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA
5. Delage, HL, Ng AY (2006) A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image. *Computer Vision and Pattern Recognition*, IEEE Computer Society Conference on, 2: pp 2418–2428. http://www.stanford.edu/~hllee/cvpr06_3dReconIndoorScene.pdf
6. Google Sketch Up. <http://sketchup.google.com/>
7. Hartley R, Zisserman A (2003) *Multiple view geometry in computer vision*. Cambridge University Press, pp 47–57
8. Hoiem D (2005) Alexei A. Efros and Martial Hebert. Automatic Photo Pop-up. *ACM SIGGRAPH*
9. Hoiem D, Efros A, Hebert M (2005) Geometric context from a single image. In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*. http://www.cs.uiuc.edu/homes/dhoiem/publications/Hoiem_Geometric.pdf
10. Horry Y, Anjyo K-I, Arai K (1997) Tour into the picture: using a spidery mesh interface to make animation from a single image. *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp 225–232. [doi:10.1145/258734.258854]
11. Huang J, Cowan B (2009) Simple 3D Reconstruction of Single Indoor Image with Perspective Cues. *crv*, pp.140–147. *Canadian Conference on Computer and Robot Vision*. <http://www.computer.org/portal/web/csdl>, doi:10.1109/CRV.2009.33
12. Illingworth J, Kittler J (1988) A survey of the hough transform. *Comput Vis Graph Image Process* 44 (1):87–116
13. Lee DC, Hebert M, Kanade T (2009) Geometric reasoning for single image structure recovery. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June. www.cs.cmu.edu/~dclee/pub/cvpr09lee.pdf
14. Liebowitz D (1999) Antonio Criminisi, and Andrew Zisserman. 1999. Creating Architectural Models from Images. In *Proc. EuroGraphics*, vol 18. <http://www.robots.ox.ac.uk/~vgg/publications/papers/liebowitz99.pdf>.
15. Liebowitz D (2001) Camera calibration and reconstruction of geometry from images. Merton College Robotics Research Group Department of Engineering Science University of Oxford Trinity Term 2001. <http://www.robots.ox.ac.uk/~vgg/publications/papers/liebowitz01.pdf>
16. Rother (2000) A new approach for vanishing point detection in architectural environments. In *BMVC*, pages 382–391. www.bmva.org/bmvc/2000/papers/p39.pdf
17. Rother C (2002) A new approach to vanishing point detection in architectural environments. *Image Vis Comput* 20(9–10):647–655
18. Saxena A, Chung SH, AY Ng (2005) Learning depth from single monocular images. In *Neural Information Processing Systems (NIPS)*. [http://citeseerx.ist.psu.edu/viewdoc/summary?](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.8799), doi:10.1.1.72.8799

19. Saxena A, Sun M, Ng A (2007) Learning 3-D Scene Structure from a Single Still Image. In: Proc. of ICCV workshop on 3D representation for Recognition
20. Saxena A, Chung SH, Ng AY (2007) 3-d depth reconstruction from a single still image. IJCV
21. Van den Heuvel FA (1998) Vanishing point detection for architectural Photogrammetry. Int Arch Photogram Rem Sens 32(5):652–659



Georgios Vouzounaras was born in Thessaloniki, Greece in 1987. He received the Diploma degree in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Greece, in 2010. His main research interests include Computer Vision and he is involved in 3DLife Network of Excellence project.



Petros Daras was born in Athens, Greece in 1974 and he is a Senior Researcher at the Informatics and Telematics Institute. He received the Diploma degree in Electrical and Computer Engineering, the MSc degree in Medical Informatics and the Ph.D. degree in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Greece, in 1999, 2002 and 2005 respectively. His main research interests include Computer Vision, search and retrieval of 3D objects, the MPEG-4 standard, peer-to-peer technologies and medical informatics. He has been involved in more than 10 European and National research projects. Dr. Daras is a member of the Technical Chamber of Greece.



Michael Gerassimos Strintzis received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1967, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively. He then joined the Electrical Engineering Department at the University of Pittsburgh, Pittsburgh, PA., where he served as Assistant Professor (1970–1976) and Associate Professor (1976–1980). Since 1980, he has been Professor of electrical and computer engineering at the University of Thessaloniki, Thessaloniki, Greece, and, since 1999, Director of the Informatics and Telematics Research Institute, Thessaloniki. His current research interests include 2-D and 3-D image coding, image processing, biomedical signal and image processing, and DVD and Internet data authentication and copy protection.