International Journal of Geo-Information

MDPI

Article

An Indoor Scene Recognition-Based 3D Registration Mechanism for Real-Time AR-GIS Visualization in Mobile Applications

Wei Ma [1], Hanjiang Xiong [1,2,*], Xuefeng Dai [1,2,*], Xianwei Zheng [1,2] and Yan Zhou [1]

[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, 129 Luoyu Road, Wuhan 430079, China; maweiweiweiwei@163.com (W.M.); zhengxw@whu.edu.cn (X.Z.); 2009302590044@whu.edu.cn (Y.Z.)

[2] Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China

* Correspondence: xionghanjiang@163.com (H.X.); daixuefeng203@126.com (X.D.)

Received: 6 February 2018; Accepted: 14 March 2018; Published: 15 March 2018

Abstract: Mobile Augmented Reality (MAR) systems are becoming ideal platforms for visualization, permitting users to better comprehend and interact with spatial information. Subsequently, this technological development, in turn, has prompted efforts to enhance mechanisms for registering virtual objects in real world contexts. Most existing AR 3D Registration techniques lack the scene recognition capabilities needed to describe accurately the positioning of virtual objects in scenes representing reality. Moreover, the application of such registration methods in indoor AR-GIS systems is further impeded by the limited capacity of these systems to detect the geometry and semantic information in indoor environments. In this paper, we propose a novel method for fusing virtual objects and indoor scenes, based on indoor scene recognition technology. To accomplish scene fusion in AR-GIS, we first detect key points in reference images. Then, we perform interior layout extraction using a Fully Connected Networks (FCN) algorithm to acquire layout coordinate points for the tracking targets. We detect and recognize the target scene in a video frame image to track targets and estimate the camera pose. In this method, virtual 3D objects are fused precisely to a real scene, according to the camera pose and the previously extracted layout coordinate points. Our results demonstrate that this approach enables accurate fusion of virtual objects with representations of real world indoor environments. Based on this fusion technique, users can better grasp virtual three-dimensional representations on an AR-GIS platform.

Keywords: AR-GIS; FCN; mobile phone; pose tracking; scene fusing

1. Introduction

GIS technologies are becoming widely used in a growing number of application scenarios, thus more attention has focused on the display and visualization of spatial information. Traditional media for presenting spatial information, such as 2D or 3D maps, cannot meet growing user requirements for dynamic displays. Flexibility and realism in GIS visualizations are becoming ever more demanding, as the volume and complexity of this information expands; AR-GIS is a response to these challenges. High quality AR representations can help users better comprehend and interact with real world spatial information. In order to achieve realistic visual effects and coherent rendering, camera pose tracking techniques are necessary for accurate understanding of spatial relationships in AR-GIS. Precise tracking of the camera within an augmented environment is required to achieve proper alignment of the virtual objects to their real-world counterparts and create a rich user experience [1]. The existing tracking methods obtain only a 6DOF camera pose. AR-GIS renderings, however, require exact positions of objects appearing in target scenes, consistent with the real world; for example, desks must be on the floor, or pictures must hang on the wall. Thus, the AR system must not only

consider the correspondence between two images, but also recognize where the floor is located, and where a picture is superimposed on a wall. Therefore, the goal of this paper is to enable realistic augmented user experiences in 3D scenes through 3D scene understanding and indoor scene tracking that can properly integrate and deal with the limitations of AR-GIS visualization.

Considering the indoor limitations of GPS and other related positioning sensors, vision-based tracking is preferable for indoor AR. However, scene fusion after vision-based tracking of the camera pose has not received much attention. AR-GIS visualization considers proper scene fusion of spatial information in the real world. Incorrect fusion may cause problems such as illogical rendering of 3D model to the real scene. Most of the existing indoor vision-based tracking methods focus on the development of invariant and robust feature detectors, descriptors, and matching methods [2–7]. However, these tracking systems only consider the correspondence between two images, but cannot provide exact proper position for virtual objects in the real world. On the other hand, when establishing 2D to 3D registration, some scholars have proposed 3D model based tracking techniques that rely on Structure from Motion (SFM) approaches [8], Simultaneous Localization and Mapping (SLAM) [9,10], or the textured CAD model [11]. These kinds of methods need either professional data acquisition equipment or costly human intervention in order to bridge this crisis of representation between a dynamic reality and the static representation of spatial information; in this instance, 3D scene understanding technology [12] can be an effective solution for AR-GIS scene fusion.

In this paper, we propose a novel method for fusing virtual objects and a real indoor scene, based on natural feature tracking and 3D indoor scene understanding technologies. During the first, offline stage, features are extracted from the reference images. At the same time, we perform interior layout extraction on a captured image using Fully Convolutional Networks (FCN) [13]. Then, the calibration parameters and camera poses corresponding to image viewpoints are computed by Fast Retina Key (FREAK) [14] feature matching. Finally, we render the virtual objects onto corresponding positions in the real world, based on previously extracted scenes.

Our contribution can be summarized as follows:

(1) We combine the spatial layout estimation with pose tracking technology for rational and logical AR visualization, breaking through rendering limitations.
(2) We present a novel automatic method for fusing AR scenes in an indoor environment, which does not rely on the conventional depth detection or 3D modeling processes. No professional data acquisition equipment is needed in this approach, as it is more resilient to spatial alterations and more faithfully represents changing indoor scenes.

The organization if this paper is as follows. The related work is reviewed in Section 2. The main methods are presented in Section 3. Section 4 describes the experimental setup and discusses the experimental results. The conclusions and recommendations for future work are offered in Section 5.

2. Related Work

2.1. AR Tracking Technology

Pose tracking technology has made remarkable progress, but scene fusion, however, has not kept up. On the one hand, current tracking methods always focus only on the capabilities and efficiency of tracking camera poses, but ignore visual appearance [2–7]. On the other hand, natural feature tracking approaches that only employ reference detail correspondences are not accurate enough when it comes to rational expression of spatial information and seamless scene fusion in indoor environments.

A number of researchers have investigated the use of AR visualization with GIS data. Gary R. et al. [15] built an AR platform to visualize 3D data outdoors, but they used a tripod-based computer. Wei Huang et al. [16] developed an outdoor AR system that needs 3D GIS to improve the efficiency and accuracy of registration in outdoor environment. Pei-Jung Lin et al. [17] integrated GIS, LBS (location-based services), MAR (mobile augmented reality), and information related to corporate

mobile marketing to create an app for tourists using Android and iPhone systems. These approaches, however, only served for outdoor use, and specialized equipment was required.

In the related computer vision literature, geometric features are usually considered in the estimation of camera pose. Ferrari et al. [18] recognized and textured by tracking of parallelogram-shaped and elliptical image regions, which extracted in affinely invariant regions. Thierry et al. [19] proposed a technique composed of cameras and projectors used to determine the pose of the object in the real world. The major handicap of this technique is the necessity of system calibration using grids. Both of their methods were limited to the presence of planar structures in an AR scene, thus restraining the applicability; other approaches are based on the 3D model overcome this problem.

Instead of learning the appearances of the edge sample points from the images, a textured CAD model was used in [11]. Skrypnyk and Lowe [20] developed a system which localizes object in the context of AR based on Scale-Invariant Feature Transform (SIFT) descriptor [21], the recognition of their method relies on the 3D model established by multi-view correspondences. Gabriele Bleser [22] argued that solution relies on a 3D model of the scene that is used to predict the appearances of the features by rendering the model based on data from the sensor fusion algorithm. Some recent visual–inertial SLAM (simultaneous localization and mapping) systems provide experimental results on realistic data, but within simple test environments [9,10]. Recently, a study investigated reaching and matching tasks in near-field distances [23]. Other approaches allow computing both camera displacement and the structure of the scene using on-the-fly techniques based on real-time structure from motion [8] or Virtual Visual Servoing [24]. However, since such 3D information is not easily available on mobile devices in certain circumstances, it is sometimes necessary to achieve the pose computation with less constraining knowledge on the viewed scene.

As shown in Table 1, most of the available tracking techniques can be divided into four classes: sensor-based, feature-based, marker-based, and model-based. Despite the great progress made in pose tracking, reasonable and precise scene fusion remains core concern in AR visualization. Most of the techniques developed to overcome those issues either ignore visual effect or require costly human or material means for pre-building 3D models.

Table 1. Augmented reality (AR) tracking techniques.

Tracking Technique	Advantage	Limitation
Sensor-based	No maintenance required and no range limit	Hard to apply in indoor environments
Feature-based	Flexible and without pre-building 3D model	No exact position
Marker-based	Simple to operate and easy to realize	Needs regular maintenance and suffers from limited range
Model-based	Provides exact position	Requires costly model processing

2.2. 3D Indoor Scene Understanding

Single image based spatial layout extraction is one of the most fundamental tasks in computer vision and image understanding. Recently, machine-learning algorithms have shown outstanding performance in fulfilling this task. These approaches can be divided into two major classes. In the first class, researchers employ only geometry-oriented techniques to estimate objects and layout candidates, determining the best matches using Structure Learning. With the advent of deep learning came the second type of layout extraction approaches that introduced FCN to add features to the process and enhance the extraction results.

Representative of the first type is the work in 2009 by Hedau et al. [12], who generated candidate box layouts based on vanishing points and line segments. The best layout is chosen by the structured learning framework [25]. In the same context, Lee et al. [26] improves the results by adding one more feature to the extraction process, an orientation map that labels three orthogonal surface directions based on line segments. In addition, Hedau et al. [27] advanced their earlier work by

extending the cuboid detector to a more general class of boxy objects and incorporated the spatial constraints. Similarly, Fouhey et al. [28] incorporated human pose estimation into indoor scene layout understanding. Choi, W. et al. [29] presented a 3D geometric phrase model that captures the semantic and geometric relationships between objects.

Some research has also been dedicated to improving efficiency when generating layout hypothesis. Wang et al. [30] proposed a discriminative learning method using latent variables and the prior knowledge to infer mutually 3D scenes and clutter. Schwing et al. [31] derived a branch and bound method that splits the label space according to 3D layout candidates, and bound the potentials in these sets, thus detecting objects in 3D boxes based on integral geometry.

With the rise of deep learning, more scholars are applying neural networks to solve indoor layout estimation problems. The second type of spatial layout estimation combines neural networks and structured learning. For instance, Mallya and Lazebnik [32] applied a FCN to learn the informative edge and geometric context jointly from an RGB image. The results of FCN training were used as new features in a maximum margin structured classifier to select the best-fitting layout. Ren et al. [33] proposed a Coarse-to-Fine Indoor Layout Estimation (CFILE) method; they adopted FCN to obtain a coarse-scale room layout estimation, which combines layout contour properties and surface properties. Then, they formulated an optimization framework that enforces several constraints for layout detail refinement. Instead of learning edges, Dasgupta et al. [34] employed a FCN to learn the semantic surface labels and optimized the spatial layout using vanishing lines.

3. Methods

3.1. Overview

In this study, our goal is to fuse, in real time, a virtual 3D geometric model onto a suitable section of the real world using a monocular image. The AR system consists of two parts, as illustrated in Figure 1, the client side and the server side.

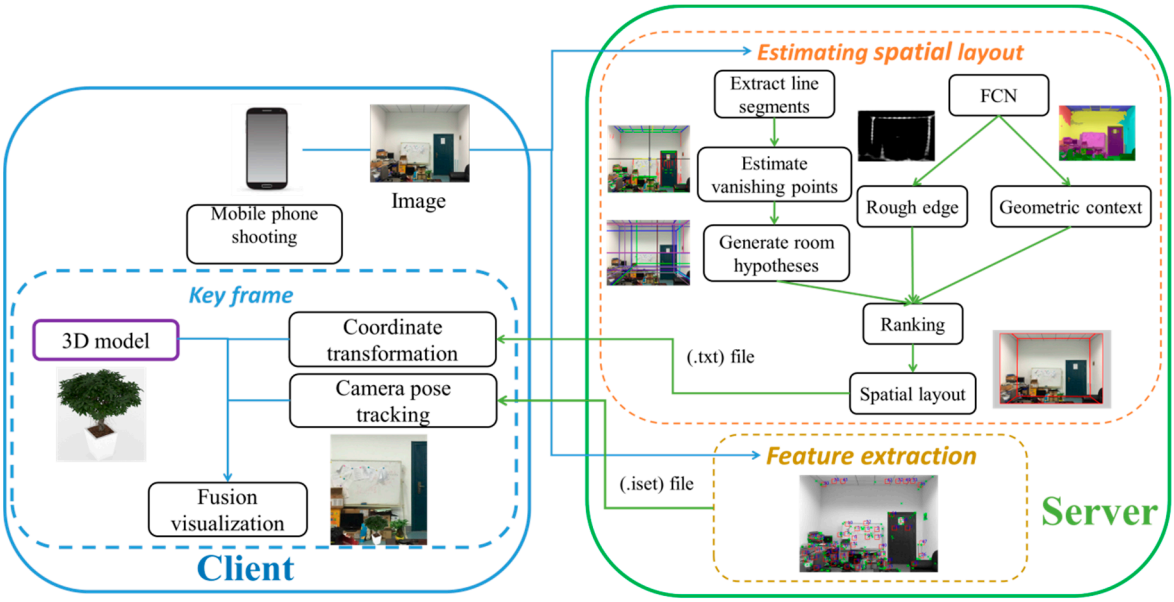


Figure 1. The process of indoor 3D scene fusion for AR-GIS visualization.

As shown in the top left half of Figure 1, we first capture a reference image of the indoor environment that we want to track using a mobile phone. On the sever side, as shown on the right side of the figure, features are extracted, and the spatial layout structure estimated based on a FCN. The spatial layout divides an indoor scene into three separate parts: walls, ceilings, and floors. As shown in the down left

half of figure, the client side grabs a key frame from a real-time smartphone video stream. A set of input relevant configuration files are obtained from server-side. These include the feature descriptors (.iset) used for camera pose tracking, and corresponding spatial layout coordinates (.txt) for exact positioning virtual object. Camera-pose tracking is based on feature matching between reference and video frame image. After applying a coordinate transformation to the layout structure, the world coordinates of the indoor scene layout are calculated. Fusion visualization is achieved by rendering the 3D model onto the corresponding position of the real world based on the camera pose and the transformed layout coordinates.

3.2. Camera-Pose Tracking Based on Natural Features

In a camera-imaging model, the imaging process can be described as a process transforming the real-world coordinates in sequence to the camera, image, and pixel coordinates. As is shown in Equation (1), f is the camera focal length, and K is known as the camera intrinsic matrix.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = KT_{cw} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

where R and t of matrix T_{cw} are the position and pose of the camera, known as the camera extrinsic matrix. $X_wY_wZ_w$ is world coordinate, $X_cY_cZ_c$ is camera coordinate, and uv is pixel coordinate.

A tracking camera pose for a target scene is essential to establish a relationship between reference and video frame image. Consider two images of a plane shown in Figure 2.

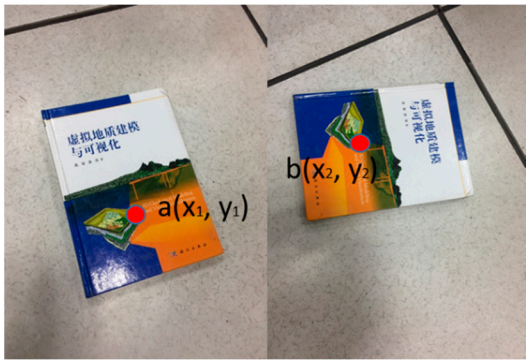


Figure 2. Two images of a 2D plane are related by a homography.

We captured two images from different perspective. The red dot represents the same physical point in the two images. With the homography matrix between two images, we can easily figure out the point b value according to point a.

Looking at points P_a in a plane, passing from the projection $P_b = (u_b, v_b, 1)$ of p in b to the projection $P_a = (u_a, v_a, 1)$ of p in a:

$$p_a = \frac{z_a}{z_b} K_a * H_{ba} * K_b^{-1} * p_b$$

where z_a and z_b are the z coordinates of P in each camera frame and homography matrix H_{ba} is $H_{ba} = R - \frac{tn^T}{d}$. R is the rotation matrix by which b is rotated in relation to a; t is the translation vector from a to b; n and d are the normal vector of the plane and the distance to the plane, respectively. K_a and K_b are the cameras' intrinsic parameter matrices. The homography matrix is a transformation (a 3×3 matrix) that maps the points in one image to the corresponding points in the other image. It has eight degrees of freedom.

We set the target plane Z = 0, the Equation (1) can be described as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda C T_{cw} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} = H \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{2}$$

where $\lambda = 1/Z_w$.

For calculating a homography between two images, we need to know at least four point correspondences between the two images. If there are more than four corresponding points, the Random Sample Consensus (RANSAC) approach can be used to iteratively calculate and obtain the most inliers as the optimal result. Iterative calculation counteracts the effect of noise, and thus can achieve a better result than using only four point correspondences. Thus, 3D registration based on natural planar features using homography is composed of the dot product of the camera intrinsic matrix and the camera extrinsic matrix. The former can be acquired by camera calibration, and the latter acquired by matrix decomposition, and an object can be easily rendered onto the position recognized and extracted from a reference image, according the homographic relationship.

Feature matching solves 3D registration based on planar natural features; this process can be divided into feature detection, feature description, and descriptor matching. There has been a lot research on this particular topic [14,35–37]; thus, we will not detail feature matching techniques. In AR fusion visualization, before drawing a 3D model into indoor environment, the real scene must be recognized, as homography only provides corresponding relationships between two images.

3.3. Spatial Layout Estimation

Achieving seamless scene fusion requires an exact position for a 3D model in a dynamic camera image. To solve this problem, we employ spatial layout estimation in a way similar to the method described by Mallya et al. [32]. The pipeline of spatial layout prediction is divided into two steps: learning to predict rough layout (Figure 3c,d) and ranking box layouts (Figure 3e). In the first step, we apply a FCN to learn the rough layout and geometric context from a RGB image captured by a smartphone (Figure 3c,d). In the second step, the straight lines and three mutually orthogonal vanishing points are estimated from the realistic image (Figure 3b). Then, the layout candidates (Figure 3e) are generated based on direction information of the vanishing points [12]. At this stage, the candidates of a room define the 3D parametric representation of the layout major surfaces in the scene. The best-fit layout is selected by ranking these layout candidates based on the FCN results. Layout estimation divides an indoor scene into three separate parts as walls, ceilings, and floors. Based on these results, we set the 3D model into a suitable position in a real-world scene as represented in a dynamic mobile phone device.

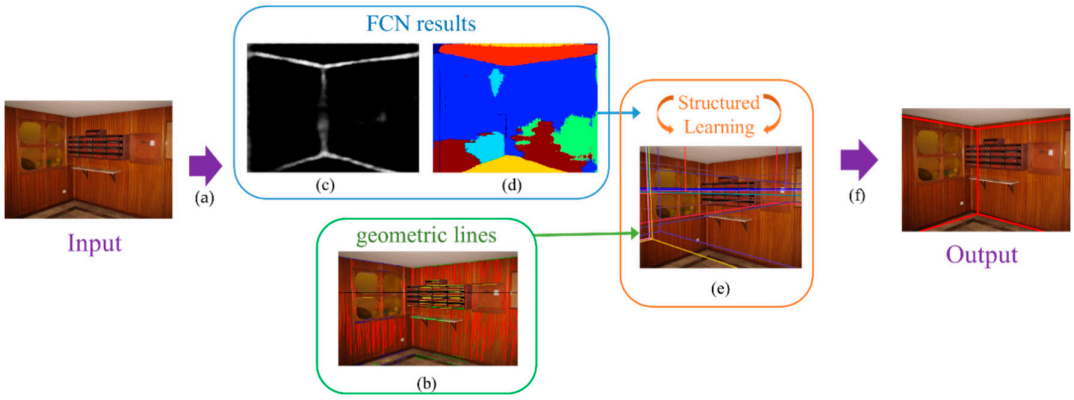


Figure 3. Indoor layout estimation by FCN. (**a**) FCN learning; (**b**) Line segments and vanishing points; (**c**) Rough layout; (**d**) Geometric context; (**e**) Layout hypotheses; (**f**) Ranking.

3.3.1. Learning to Predict Rough Layout

For estimating the layout structure, a rough layout is defined as the edges of the projected 3D box that fits the room. There are three types of such edges: those between two wall faces, between walls and the ceiling, and between walls and the floor. Ground truth edge maps are generated from the original ground truth format of [12], which consists of polygons corresponding to the different room faces. All of the pixels in the resulting mask are considered positive examples of a rough layout, even when the actual edges are occluded by clutter, such as furniture, and all other pixels are considered as part of the background or negative class.

The FCN is based on VGG-16 structure using Caffe [38]. It is trained with 32-pixel prediction stride to accomplish two tasks simultaneously: rough layout extraction and geometric context labeling. The FCN structure is illustrated in Figure 4.

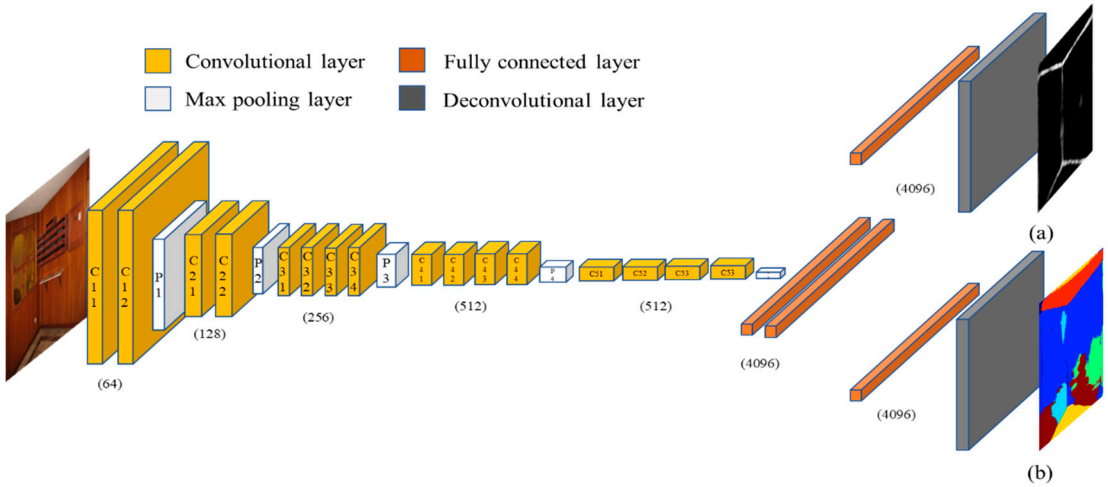


Figure 4. Structure of the FCN-VGG16 with two output branches. (**a**) Rough layout; (**b**) Geometric context.

We perform joint training by sharing all layers of the FCN except for the de-convolutional layers that produce the softmax probability maps for the respective types of output. The total loss of the network is the sum of the two cross-entropy classification losses: one for rough layout prediction, and one for geometric context label prediction. Joint loss optimization helps to improve the accuracy of the edge maps.

3.3.2. Ranking Box Layouts

Given an input image, we use the approach presented in [12] to estimate the vanishing points of the scene. This approach exploits edge-based votes for vanishing points using the Canny edge detector [39]. Once the vanishing points are acquired, layout candidates are generated by combining the rays of vanishing points from two directions.

For selecting the best-fit layout, we train a function, $f(x, y; w)$, to assign a score to the automatically generated candidate layouts for an image. The mapping gives a higher value for the correct combinations of input images and layouts. Given a set of training images $\{x_1, x_2, \ldots x_n\}$ and their layouts $\{y_1, y_2, \ldots y_n\}$, for an image x_n with a best-fit box y_n, the function f will assign a higher score to layout y, as it is most similar to y_n. We give a new test image to the function, then the best-fit layout is chosen by

$$y^* = \arg\max_y f(x, y; w) \tag{3}$$

This structured regression problem is solved by using a max-margin framework [25], in which the function $f(x, y) = w^T \Psi(x, y)$, where $\Psi(x, y)$ is the set of features [12]. In our experiments, we use two types of input features: rough layout and geometric context; both are FCN results.

3.4. Fusion of Real Indoor Scene and Virtual 3D Object

For fusing a virtual 3D model and a real scene, the key operation is correcting the projection in a reference image until it appears suitably rendered. We render the 3D model onto reference images based on previous feature matching results. At the same time, the exact position of wall, ceiling, and floors in indoor scene is determined according to the spatial layout estimation result. The spatial layout coordinate points and the feature points in reference images are saved together. The complete fusion workflow on the client-side is illustrated in Figure 5.

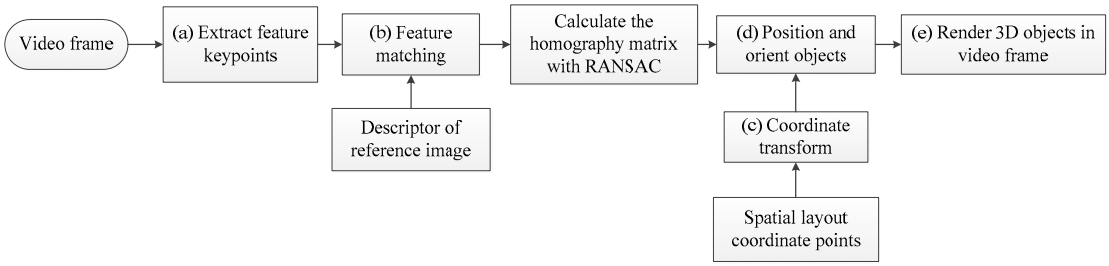


Figure 5. The complete fusion workflow in client-side.

As shown in Figure 5, the fusion procedure in client-side begins with a key frame from a real time smartphone video stream. The homography matrix was calculated based on the feature matching between reference and video frame image, while the RANSAC was used to remove outliers to improve the matching accuracy of feature points. The world coordinates of the indoor scene layout were calculated by applying a coordinate transformation to the layout structure coordinates extracted previously. Fusion visualization is achieved by rendering the 3D model onto the corresponding position of the real world based on the camera pose and the transformed layout coordinates.

(a) Feature extraction

Freak features are extracted from the current frame of the video sequence

(b) Feature matching

Feature matching occurs between a loaded dataset and a set of key points extracted from an input image. The new features are matched to the reference image features using the ICP algorithm, resulting in a set of 2D-to-2D correspondences.

(c) Spatial layout coordinate-points transformation

The geometric relationships between the real world, virtual objects, and the camera are defined in the same generic units. However, the spatial layout is based on pixel coordinates. To overcome this issue, we transform the spatial layout coordinates to the world coordinates according to the pixel dimensions and dots per inch (DPI) of the reference image.

(d) Position and orient object

For positioning and orienting the 3D model, we create a matrix suitable for passing to OpenGL to set the viewing transformation of the virtual camera. A matrix is formed so that the origin point of the reference image is registered to a corresponding point in the world coordinate system, with the image lying in the x-y plane. The positive x axis points to the right of the reference image, the positive y axis to the top of the image, and the positive z axis out of the image. This is a right-hand coordinate system in the common standard for OpenGL graphics.

(e) Render 3D objects in video frame

After the coordinate and viewing transformation, we can easily define the object coordinate values in AR viewport. Based on the perception of the indoor scene, the exact 3D model position was given according to the logical relationship between object and real scene, such as a desk on the floor, or picture on the wall. Finally, AR scene fusion is implemented by rendering the virtual object using OpenGL.

4. Experiments

In order to verify the proposed AR indoor 3D scene fusion method, several experiments were conducted to collect a target scene using smartphones in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS) of Wuhan University. In our experiment, a 3D tracking method based on the structural features of indoor scenes and scene recognition based on 3D indoor scene understanding is used, to investigate appropriate ways of combining them in AR.

4.1. Experiments Setup

Our prototype was operated on a PC server and a client mobile device. The server configuration included a 2.60 GHz Intel(R)Core(TM) i7-6700HQ CPU and 8 GB RAM. The client consisted of an android system mobile phone of Lenovo PB2-690N (operation system: Android 6.0.1, CPU: MSM8976, CPU frequency: 4×1.8 GHz + 4×1.4 GHz, RAM capacity: 4 GB).

The quickest way to develop an AR application is to use the AR Development Kit. Currently, the common AR Development Kit on the Android platform includes metaio AR, Vuforia, EasyAR and ARToolKit [40]. In our experiments, we established an AR fusion system using the ARToolKit5 in Visual studio 2013. In contrast to other widely used development kits, ARToolKit is open source, making it more suitable for theoretical research on 3D registration algorithms, improvements, and development of depth customization in AR applications.

4.2. Experimental Process

4.2.1. Offline Training Stage

During this stage, we trained the reference images and estimated spatial layout structure on the server side.

(a) Image training

In the first step, we decided on the image set resolutions. The source image was resampled at multiple resolutions, generating an image set (.iset) file. This file contained the raw image data loaded into the application at runtime for tracking. Features are extracted at three or more resolutions, since dots will appear in the image at different resolutions depending on how close or far away the camera is from the image. We subsequently extract the FREAK feature and generate a feature dataset from the reference images. The dataset is rendered in Figure 6, and the counts of feature points in each image are illustrated in caption.

(b) Estimating spatial layout

The indoor scene images were captured in different rooms at LIESMARS. Like in previous work [12], the structured regressor is trained on the Hedau train set that has 308 images. For training the FCN, the training dataset used in our method is Hedau+ [32], in which the number of training images are 284. We augmented the training set by 16 times using standard transformations such as cropping, mild rotation and scaling. The models were tested on a recently introduced LSUN dataset, including 1000 test images. The prediction error on the LSUN test set was 16.71%. This result demonstrates that the model generalizes

effectively, despite not being retrained. As shown in Figure 7, the scenes contain floors, walls, and ceilings. The extraction results are shown in Figure 7. As shown, the spatial layout divides an indoor scene into five parts: left wall, center wall, right wall, ceiling, and floor.

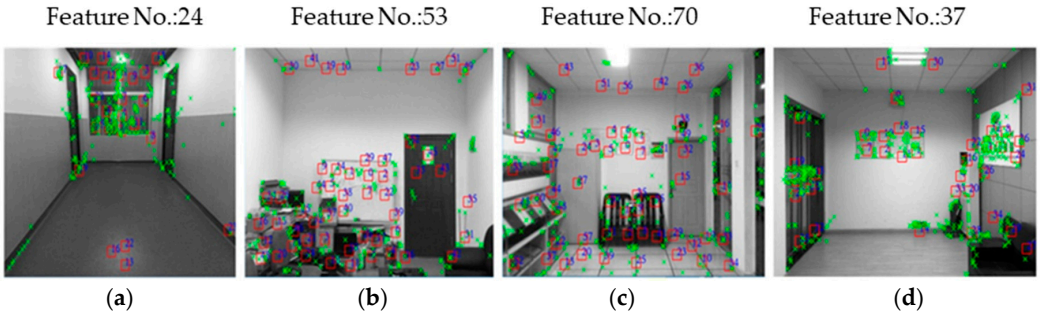


Figure 6. The pixel size of each reference image is 650 × 500 and the DPI value of each is 100. The features used in continuous tracking are outlined by red boxes, and the features used for identifying the pages and initializing tracking are marked by green crosses. (**a**) 24 feature points; (**b**) 53 feature points; (**c**) 70 feature points; (**d**) 37 feature points.

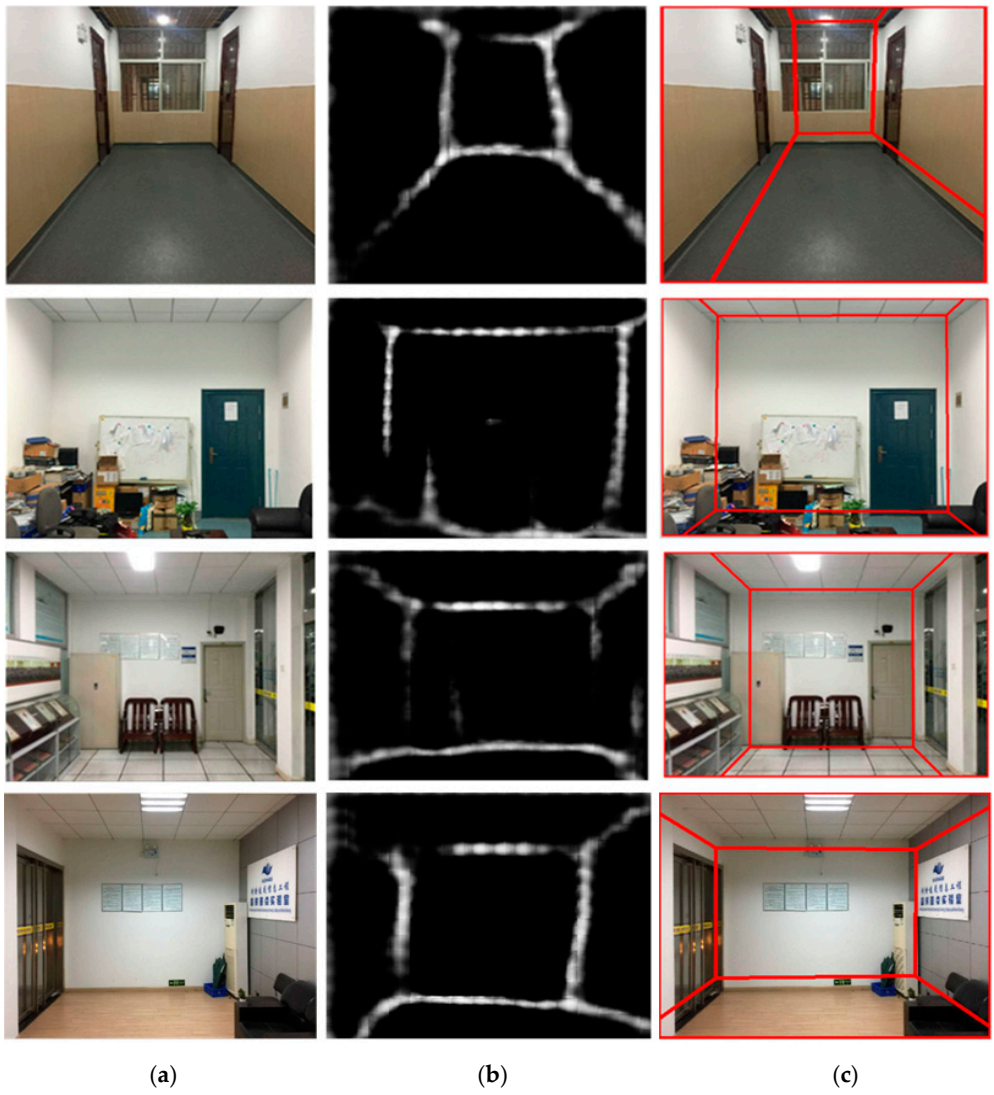


Figure 7. Visualization of spatial layout estimation. (**a**) Reference image; (**b**) Rough layout; (**c**) Estimation result.

4.2.2. Online Fusion Stage

In this implementation, we calculated the camera pose, and rendered the 3D object in the indoor scene based on the result of spatial layout estimation.

(a) Camera pose computing

ARToolKit5 provides the FREAK feature matching and iterative closest point (ICP) method to calculate the real camera position and orientation relative to square shapes or flat textured surfaces.

(b) Virtual object rendering

As discussed in Section 3.4, before rendering object in AR scene, we transform the spatial layout points to the world coordinate system. This relates to the required resolution, commonly expressed as pixels or Dots Per Inch (DPI).

According to the transformed spatial layout, we input the object position where we wanted and rendered it in the real world.

4.3. Experimental Results and Discussion

Our experiment required only reference images, and not manual initialization of the tracker. We took photos at different positions in LIESMARS using a mobile phone. For fusing the indoor AR scene, we run the system on a Lenovo smartphone and set the camera image size is 640 × 360.

We selected four indoor scenes and trained their reference images in the server side. The AR-GIS visualization results are illustrated in Figure 8. In our example, we tracked the target scene and rendered it as a 3D model using two AR systems. The first column figures only use the ARToolKit system, which lacks 3D indoor scene understanding (Figure 8a). Unlike Figure 8a, the second column 3D models in Figure 8b were rendered in more rational positions, using the proposed AR system that does incorporate 3D indoor scene understanding.

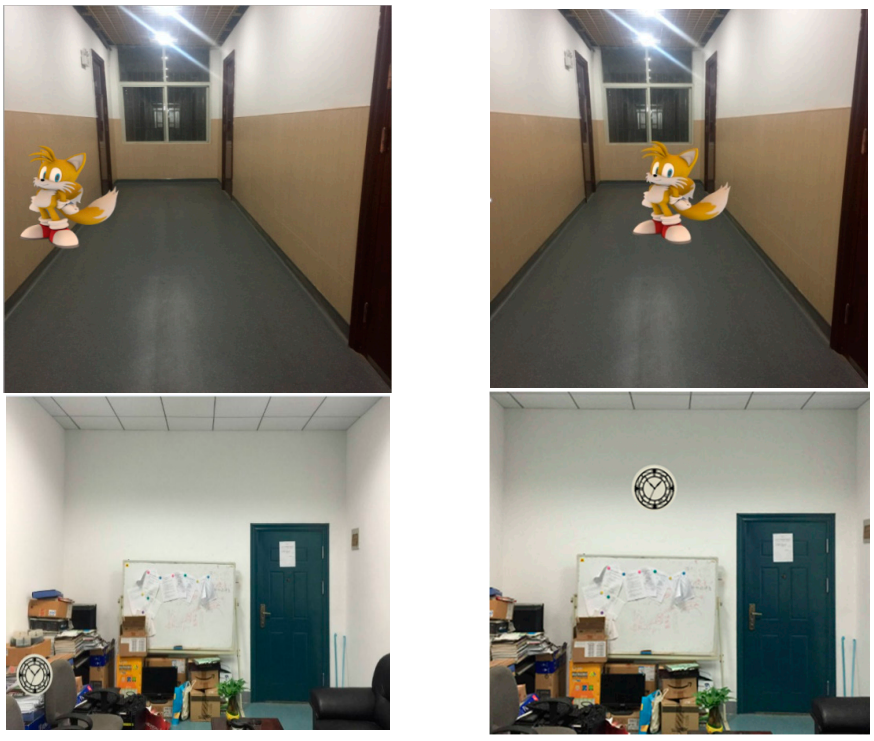
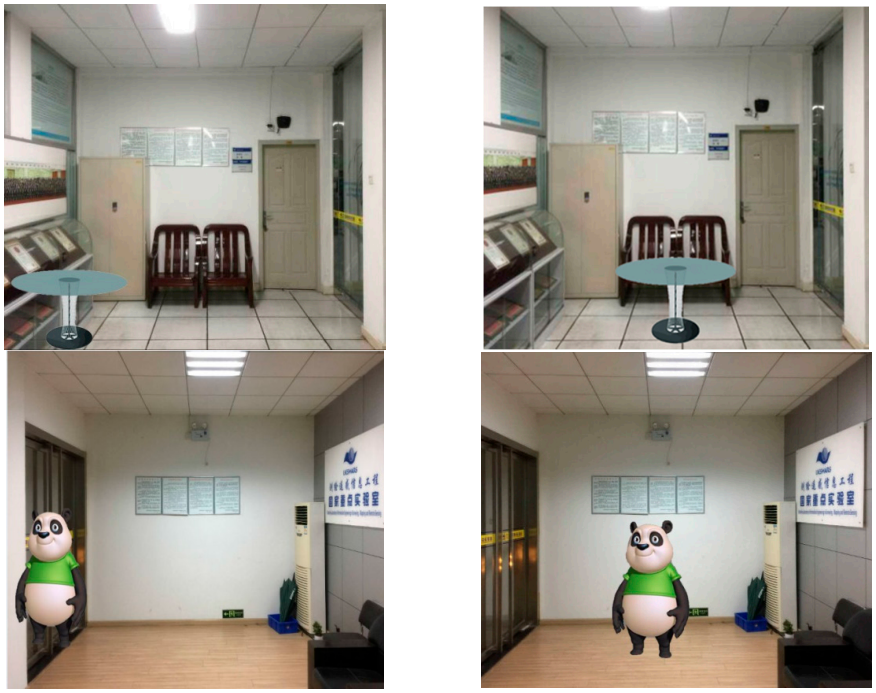


Figure 8. *Cont.*



(**a**) (**b**)

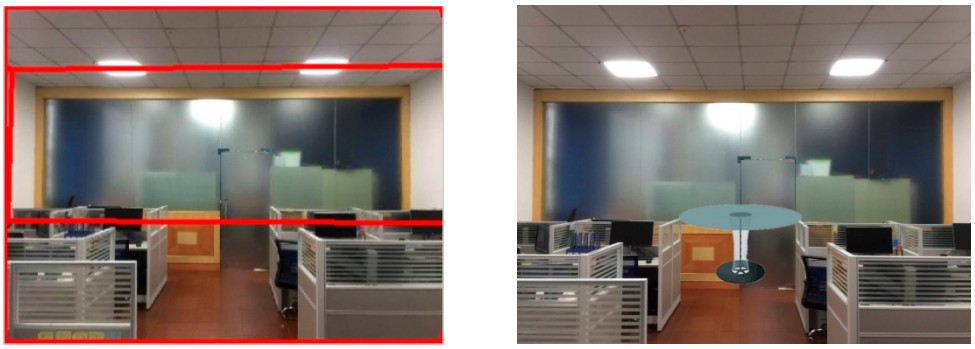
Figure 8. Experiment results. (**a**) A 3D model rendered by an AR system only using ARToolKit; (**b**) A 3D model rendered by an AR system based on 3D indoor scene understanding.

To demonstrate the real time capability of the proposed system, we have recorded its average computation time on a variety of scenes in Figure 8. As shown in Table 2, the average computation times were generally less than 20 ms.

Table 2. Average computation times.

	Average Computation Time
Test 1	17 ms
Test 2	18 ms
Test 3	21 ms
Test 4	17 ms

In this work, errors stem from the imprecise estimation of the spatial layout from a given image. The prediction error on LSUN test set and Hedau test set was 16.71%, and 12.83%, respectively. These results demonstrate that the layout estimation satisfies the requirements for AR visualization, in most cases. Errors may occur under conditions of extreme occlusion, when the edges of the predicted structural layout are largely occluded by indoor objects. As shown in Figure 9, in this indoor scene, for example, the wall–floor line is almost completely blocked by computer tables.



(**a**) (**b**)

Figure 9. (**a**) Spatial layout estimation failure; (**b**) AR fusion visualization failure.

From these experimental results, we clearly perceive that our AR visualization exhibits a close relationship with spatial information. The outcome presented in Figure 8 returns high quality fusion between virtual object objects and real indoor scenes. In fact, the fusion content manifests continuous and homogenous graphical and geometric details. Furthermore, these rendering and accuracy results conform to our claims about the relevance of our method not only for drawing bare 3D models, but also when detecting spatial information to fulfill GIS requirements in indoor environments.

5. Conclusions and Future Work

The work described in this paper aims to integrate an indoor scene recognition technique and 3D registration method in order to optimize the visual quality of AR-GIS in an indoor environment. We presented an innovative automatic approach to enhance the AR seamlessly alignment. The goal of this paper is to promote the AR expression rationally and logically in a way that enhances the user experience and solves rendering limitations stemming from imprecise positioning expression in AR-GIS visualization. We generate the correspondent relationship between reference images and 3D indoor scenes, without rebuilding the geometric construction. Before performing AR-GIS system, we consider the planar surfaces in an indoor environment and reference images obtained with a mobile phone. Then, the interior layout extraction performed on a captured image using Fully Connected Networks (FCN). One advantage of our system is that it efficiently allows the fusion of virtual objects and real-world visualization without any professional data acquisition equipment, such as a depth camera. This makes the AR-GIS technology easily accessible to a much broader audience. The fusion experiments with virtual objects in different sites found at LIESMARS at Wuhan University demonstrate that the proposed method can preserve a high accuracy fit between the virtual objects and real-scenes.

Regarding application fields, the proposed registration mechanism is not only advantageous for AR-GIS visualization, but it is also promising method for 3D navigation and environment monitoring. In the future work, we are expecting to introduce functionalities that will support varied GIS information and analysis result display, based on flexible and realistic AR scenes generated using the novel approach presented in this paper. As it is the case with any engineered system, we can also enhance the process, especially during the scene recognizing, using a more sophisticated and adaptable FCN.

Acknowledgments: The authors would like to thank the editors and anonymous reviewers for their careful reading and helpful suggestions. This research is supported by the National Key Research and Development Program of China [grant number 2016YFB0502203], the National Natural Science Project [41701445] and LIESMARS Special Research.

Author Contributions: Wei Ma conceived and designed the study, performed the experiments, and wrote the paper; Hanjiang Xiong, Xuefeng Dai, Xianwei Zheng and Yan Zhou supervised the work and revised the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Arth, C.; Klopschitz, M.; Reitmayr, G.; Schmalstieg, D. Real-time self-localization from panoramic images on mobile devices. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, 26–29 October 2011.
2. Wagner, D.; Reitmayr, G.; Mulloni, A.; Drummond, T.; Schmalstieg, D. Real-Time Detection and Tracking for Augmented Reality on Mobile Phones. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 355. [CrossRef] [PubMed]
3. Wagner, D.; Reitmayr, G.; Mulloni, A.; Drummond, T.; Schmalstieg, D. Pose tracking from natural features on mobile phones. In Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality, Cambridge, UK, 15–18 September 2008.
4. Lepetit, V.; Lagger, P.; Fua, P. Randomized trees for real-time keypoint recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005.
5. Chia, K.W.; Cheok, A.D.; Prince, S.J.D. Online 6 DOF Augmented Reality Registration from Natural Features. In Proceedings of the International Symposium on Mixed and Augmented Reality, ISMAR 2002, Darmstadt, Germany, 1 October 2002.
6. Genc, Y.; Riedel, S.; Souvannavong, F.; Akinlar, C.; Navab, N. Marker-less tracking for AR: A learning-based approach. In Proceedings of the International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, 1 October 2002; pp. 295–304.
7. Chen, P.; Peng, Z.; Li, D.; Yang, L. An improved augmented reality system based on AndAR. *J. Vis. Commun. Image Represent.* **2016**, *37*, 63–69. [CrossRef]
8. Nistér, D. Preemptive RANSAC for live structure and motion estimation. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003.
9. Pinies, P.; Lupton, T.; Sukkarieh, S.; Tardos, J.D. Inertial aiding of inverse depth SLAM using a monocular camera. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007.
10. Schon, T.; Karlsson, R.; Tornqvist, D.; Gustafsson, F. A framework for simultaneous localization and mapping utilizing model structure. In Proceedings of the International Conference on Information Fusion, Quebec, QC, Canada, 9–12 July 2007.
11. Reitmayr, G.; Drummond, T. Going out: Robust model-based tracking for outdoor augmented reality. In Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality, Santa Barbard, CA, USA, 22–25 October 2006.
12. Hedau, V.; Hoiem, D.; Forsyth, D. Recovering the spatial layout of cluttered rooms. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2010.
13. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the CVPR, Boston, MA, USA, 7–12 June 2015.
14. Vandergheynst, P.; Ortiz, R.; Alahi, A. FREAK: Fast Retina Keypoint. In Proceedings of the Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
15. King, G.R.; Piekarski, W.; Thomas, B.H. ARVino—Outdoor Augmented Reality Visualisation of Viticulture GIS Data. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality, Vienna, Austria, 5–8 October 2005.
16. Huang, W.; Sun, M.; Li, S. A 3D GIS-based interactive registration mechanism for outdoor augmented reality system. *Expert Syst. Appl.* **2016**, *55*, 48–58. [CrossRef]
17. Lin, P.J.; Kao, C.C.; Lam, K.H.; Tsai, I.C. *Design and Implementation of a Tourism System Using Mobile Augmented Reality and GIS Technologies*; Springer International Publishing: Cham, Switzerland, 2014; pp. 1093–1099.
18. Ferrai, V.; Tuytelaars, T.; van Gool, L. Markerless augmented reality with a real-time affine region tracker. In Proceedings of the IEEE and ACM International Symposium on Augmented Reality, New York, NY, USA, 29–30 October 2001; pp. 87–96.
19. Thierry, M.; Fofi, D.; Gorria, P.; Salvi, J. Automatic texture mapping on real 3D model. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, USA, 17–22 June 2007; pp. 1–2.

20. Skrypnyk, I.; Lowe, D.G. Scene modelling, recognition and tracking with invariant image features. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality, Arlington, VA, USA, 5 November 2004; pp. 110–119.
21. Lowe, D.G.; Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
22. Bleser, G.; Stricker, D. Advanced tracking through efficient image processing and visual-inertial sensor fusion. In Proceedings of the Virtual Reality Conference, Reno, NE, USA, 8–12 March 2008; pp. 1920–1925.
23. Singh, G.; Swan, J.E.; Jones, J.A.; Ellis, S.R. Depth judgments by reaching and matching in near-field augmented reality. In Proceedings of the IEEE Virtual Reality, Costa Mesa, CA, USA, 4–8 March 2012.
24. Comport, A.I.; Marchand, E.; Pressigout, M.; Chaumette, F. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 615–628. [CrossRef] [PubMed]
25. Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* **2005**, *6*, 1453–1484.
26. Gupta, A.; Hebert, M.; Kanade, T.; Blei, D.M. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–9 December 2010; pp. 1288–1296.
27. Hedau, V.; Hoiem, D.; Forsyth, D. Recovering free space of indoor scenes from a single image. In Proceedings of the Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
28. Fouhey, D.F.; Delaitre, V.; Gupta, A.; Efros, A.A.; Laptev, I.; Sivic, J. People watching: Human actions as a cue for single view geometry. *Int. J. Comput. Vis.* **2014**, *110*, 259–274. [CrossRef]
29. Choi, W.; Chao, Y.W.; Pantofaru, C.; Savarese, S. Understanding Indoor Scenes Using 3D Geometric Phrases. In Proceedings of the Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013.
30. Wang, H.; Gould, S.; Koller, D. Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding. *Commun. ACM* **2013**, *56*, 92–99. [CrossRef]
31. Schwing, A.G.; Urtasun, R. *Efficient Exact Inference for 3D Indoor Scene Understanding*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 299–313.
32. Mallya, A.; Lazebnik, S. Learning Informative Edge Maps for Indoor Scene Layout Prediction. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
33. Ren, Y.; Li, S.; Kuo, C.C.J. A Coarse-to-Fine Indoor Layout Estimation (CFILE) Method. *arXiv* **2016**.
34. Dasgupta, S.; Fang, K.; Chen, K.; Savarese, S. DeLay: Robust Spatial Layout Estimation for Cluttered Indoor Scenes. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
35. Herbert, B.; Tuytelaars, T.; Gool, L.V. Surf: Speeded up robust features. In Proceedings of the Computer Vision—ECCV 2006, Graz, Austria, 7–13 May 2006; Volume 3951, pp. 404–417.
36. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In Proceedings of the Computer Vision—ECCV, Heraklion, Greece, 5–11 September 2010; Volume 6314, pp. 778–792.
37. Rublee, E.; Rabaud, V.; Konolige, K.; Bradsk, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
38. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
39. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PANI-8*, 679–698. [CrossRef]
40. Artoolkit. Available online: https://archive.artoolkit.org/ (accessed on 13 March 2018).