Skyline Matching: A robust registration method between Video and GIS

Shupeng Zhu, Muriel Pressigout, Myriam Servières, Luce Morin, Guillaume Moreau

Skyline Matching: A robust registration method between Video and GIS

S. Zhu[1], M. Pressigout[1], M. Servière[2], L. Morin[1] and G. Moreau[2]

[1]IETR, INSA Rennes, 35700 Rennes Cedex 7, France
[2]CERMA, Ecole Centrale de Nantes, 44321 Nantes Cedex 3, France

Abstract. Camera pose estimation from an image has long been a very active research for Video/GIS registration. Different from traditional methods which are based on direct building feature matching, this paper presents an alternative method which uses an overall Skyline feature to determine the camera's orientation. Through our skyline matching method we find out a way to transform an image/model matching problem into a curve matching problem which significantly reduced the computation complexity and make it a possible solution for real-time applications.

1 Introduction

With the development of digital techniques and information industry, the requirement for more detailed and updated geographic information has been growing rapidly in last two decades. Based on this demand Geographic Information System (GIS) technology is mushrooming and also widely used, especially in the fields of navigation, territory management and environment evaluation. As one result of this mutual development, new technologies like Computer Vision and Augmented Reality (AR) are integrated into GIS. At the same time 3D models are replacing traditional 2D data. Under this context, Video/GIS registration technique is proposed as a solution for both Augmented Reality Application [1] and 3D texture update problem [2].

According to [3], Video/GIS registration is a fusion between videos and geo-referenced building models which are extracted from GIS. The camera pose estimation problem is really the essential challenge of Video/GIS registration issue. If both the camera location in the world coordinate system and its orientation can be determined, then it is easy to find the relationship between an image pixel

and an object from the real world (such as building facades, edges, windows or some other particular features which can be used to identify such building in the image).

In the general case, a camera pose can be expressed by a projection matrix **P** between the 3D world coordinate system and the 2D pixel coordinate system. This projection matrix can be decomposed into three main components:

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{t}] \quad (1)$$

Where **K** is the camera intrinsic parameter matrix, **R** is the rotation matrix and **t** is the translation vector. In normal case, a rough translation vector which is the position of the camera can be easily acquired by GPS signal and the intrinsic parameters can also be determined by simple camera calibration technique [4]. So the most difficult part of the camera pose estimation is the calculation of the rotation, which is defined by the rotation matrix **R**.

One of the traditional ways to solve this problem is try to calculate the projection matrix directly through several corresponding features between an image and the GIS model. However these kinds of methods usually use iterative algorithms to establish the correspondence, which is quite computationally expensive and heavily depends on the quality of features database and on features extraction in the image. The method proposed here exploits the fact that most images taken in urban environments always contain a skyline which is a border between buildings and the sky and we assume that it is quite unique for different orientation of one known position. Moreover, it can be extracted from the image much more easily and robustly than features such as buildings edges. Our method aims to present a computationally inexpensive, simple and robust method for automatic registration.

The remainder of this paper is structured as follows: first, we summarize the related previous works. Next, section 3 describes the method of skyline extraction. Section 4 explains and applies skyline matching algorithm. Results are presented in section 5 followed by the conclusions and future work. Two hypotheses are used: 1. Horizontally rectified images; 2. Calibrated camera.

2 Related works

Video/GIS registration technique has been growing rapidly in recent years. The essential of Video/GIS registration is aligning each pixel in the image with 3D coordinates in the GIS model. In order to achieve this registration, we need to know the exact position and orientation of the camera (the camera pose) when the picture is shot.

The most common method consists in first trying to build up a feature database of searching areas then extract specific features from the real image then searching the database to find the best correspondence. The database may either be composed of geo-referenced images like [5] or of unique features extracted from images such as in the work of Arth et al. [6]. This kind of methods takes a lot of time and requires storing resources to establish such database and it still needs a remarkable computation power to conduct the searching and matching process. The advantage of this kind of methods is that they do not need the initial position or rough pose information.

Another type of methods is focused on directly using the geometry features of the GIS model to establish the correspondence. Such a method, proposed by Reitmayr & Drummond [7], uses a rough textured GIS model as matching database and applies an edge-tracking system to calculate the camera pose. An improved work was done by Sourimant et al. [2] where a RANSAC algorithm is applied to find out the best correspondence of line features between GIS model and image. Both their works are heavily dependent on a robust initial rough pose which is provided either through differential GPS or other additional sensors such as gyroscope. The work of Bioret et al. [8] provides a different point of view to solve the problem by only using the geometrical information reconstructed from the image such as the angle between the facades as a query key to find the best correspondence in the GIS database. However the robustness is not perfect because ambiguity still remains in most of the cases. Direct 3D reconstruction from image has also been considered as a solution of the Video/GIS registration problem. Some works such as [9] show the possibility to completely reconstruct a 3D building within the local camera referenced system but it needs more than one camera as input source.

On account of all the limitations of previous research, we wish to develop a method which is computational inexpensive and can calculate the camera pose from a single image. Based on such requirements we need to find a feature which can represent the overall geometric character of the image and at the same time avoid direct building feature matching. Under such concern, the skyline feature becomes a perfect candidate for our method. The skyline is easier to extract and more robust for two reasons. First, there is almost no artificial object in the sky and the color of the sky is quite uniform that makes it suffers less noise during extraction. Secondly, in the general case, all occlusions happen under the location of the skyline, which makes it a feature more robust than others. Up to our knowledge, skyline features have not been employed for pose estimation from images. However, in the robotics and automation community the characteristics of the skyline have been investigated. Ramadingam et al. [10] use an upward pointing catadioptric (i.e., fisheye lens) camera and 3D urban models for precise localization without GPS using the skyline. A. Nuchter [11] uses skyline features in his registration work of 3D laser scans.

3 Skyline extraction

The proposed method of camera pose recognition through skyline matching is under two hypotheses: 1) the image is horizontally rectified; 2) there exists one significant sky region on the image. The idea is to extract the skyline from an image with known camera position. At the same time, a panoramic (360°) skyline around a point of view (which is the camera position of the real image) is extracted from the virtual world created from the GIS model. Then, we can compare these two skylines and find a correspondence between them, so that the partial skyline from the real image does match a sub-part of the panoramic skyline from the GIS model. The first step is the extraction of these two kinds of skyline.

3.1 Coordinate transformation

In order to describe the skyline feature, we need to define a coordinate system which can best represent such information. The basic idea is trying to transform the 3D coordinate of the skyline point into a 2D presentation and such presentation must be independent of the scale factor. Under such concern, we use only the Azimuth angle φ and the Elevation angle θ to describe the point P on a skyline (See Figure 1). It is indeed the so-called cylindrical coordinate system and it is a 2D coordinate system.

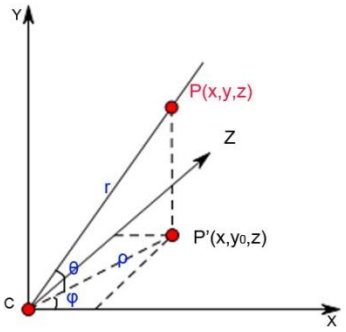

Fig. 1. The transformation between the Cartesian coordinate and the Cylindrical coordinate; C, the camera center, is the point of view and P is an arbitrary point in 3D space.

So for any point *P(x,y,z)* in the Cartesian coordinate system the transformation is as follows. First, we can get the distance **ρ** between *P'* and *C* where *P'* is the projection of point P in the horizontal plane which go through the camera center $C(x_0,y_0,z_0)$: $\rho = \sqrt{(x-x_0)^2 + (z-z_0)^2}$

Then we can got the Azimuth angle: $\varphi = \begin{cases} 0 & if\ x = x_0\ and\ z = z_0 \\ \arcsin\left(\frac{z-z_0}{\rho}\right) & if\ x \geq x_0 \\ -\arcsin\left(\frac{z-z_0}{\rho}\right) + \pi & if\ x < x_0 \end{cases}$ (2)

Then, we can calculate the elevation angle θ by using the distance between *P* and *C*:

$r = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}$ and their elevation differences: $\theta = \arcsin(\frac{y-y_0}{r})$ (3)

3.2 Panoramic skyline

In this section we will explain how the panoramic (360^o) skyline is generated around a known position from the GIS model. First four joint images were created (rendered) around such the same viewpoint by using OpenGL (see Figure 2).

Secondly the depth map of each image is calculated. Based on depth information of each point, the skyline points are extracted from the image together with their 3D coordinates. Finally the coordinate transformation is conducted as presented in section 3.1 and the panoramic view of the skyline is obtained (see Figure 4.B). The same method can also be used to extract the partial skyline from a synthetic image.

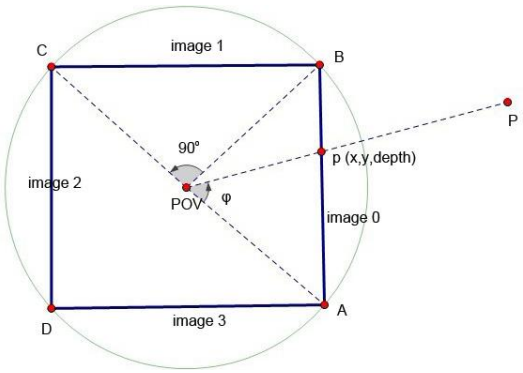

Fig.2. Create four joint images around one point of view

3.3 Partial skyline extraction from the image

The extraction of the skyline from a real image is much more complicated, because there is no depth information and the sky region is not as identical as in the virtual world. In theory, the skyline is characterized of a curve which: (1) is composed of edge points that separate the terrain objects and the sky, (2) locates at the upper parts of the image and extends from one side to the opposite side of the image boundary [12]. In order to extract the skyline we need to distinguish the sky region from other part of the image. Generally the intensity of the sky region is more uniform than in other parts of the image, which means the intensity gradient inside the sky region is small. This characteristic allows us to use a very fast and robust method, namely adaptive threshold, to find the boundary of the sky (see Figure 3.a).

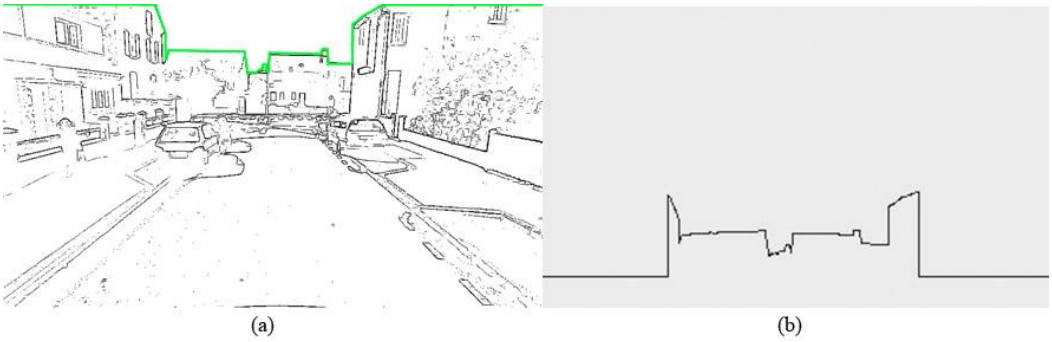

Fig.3. This image show the extracted sky line (green) by using adaptive threshold method

Besides, a morphologic filter closing operation on original image is also used to reduce the artefacts caused by thin structures such as: telegraph poles, wires and road lights. Then, each extracted skyline point is also transformed into the same 2D coordinate system (see Figure 4.b) as the panoramic skyline using the real camera internal parameters. One remark is that all non-skyline intersections are set to zero and labelled as “no-sky” in order to ignore their influence during the skyline matching process.

4 Skyline matching method

We now have both the panoramic skyline (B) from GIS and the partial skyline (A) from the real image (See Figure 5.A and 5.B for an example). Both of them are assumed to be extracted from the same camera position. In order to find their correspondence, we rely on the similarity of their shapes. As they are already transformed into the same scale irrelevant 2D coordinate metric, they are supposed to have the same energy level. This fact simplifies the comparison process.

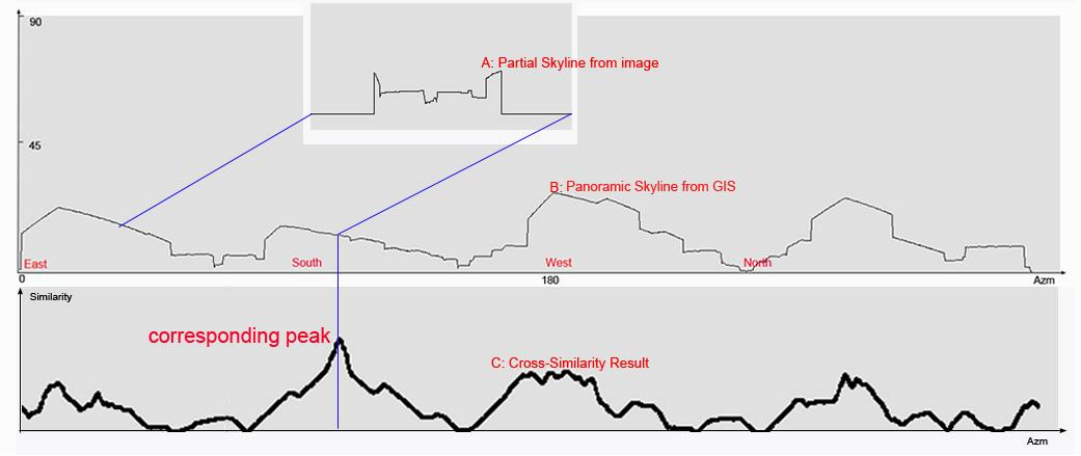

Fig.4. The result of Cross-Similarity where curve A is the partial skyline extracted from image; B is the Panoramic Skyline generate from GIS and C is the Cross-Similarity result

We use a similarity factor based on the SAD (sum of absolute differences), with an outlier removal procedure, to measure the correspondence between A and B of each azimuth angle within A as follows:

$$CS(fg)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} \begin{cases} 1 - \frac{|f[m]-g[n+m]|}{f[m]} & if\ 1 - \frac{|f[m]-g[n+m]|}{f[m]} < 0.75 \\ 0 & else \end{cases} \quad (4)$$

Where f represents partial skyline A and g represents panoramic skyline B. We call equation (4) the Cross-Similarity function expressed as $CS(fg)[n]$. For outlier removal, points with similarity below 75% are considered as outliers and discarded from the similarity evaluation. Its purpose is to make the correspondence peaks sharper and more significant. This method is under the assumption that if the recent searching point is the corresponding point of two skylines, then their energy difference (difference of elevation angle) should be less than 25%. Figure 4 shows one result of this Cross-Similarity method process with one partial skyline from a real image and one panoramic skyline generated in the same position. From this result we can find a quite significant corresponding peak which can help us to determine the camera's orientation in the horizontal plane.

5 Experiments and Results

In order to test the accuracy and robustness of our method, we conducted several experiments. First we tested the accuracy of the algorithm by using a set of synthetic images which is generated by

using the GIS model and known projection matrices. The result is quite satisfying; our method can correctly recognize the camera orientation as long as we have more than 10% skyline appearance along the image width. Then we applied our method in several images which are taken in the city center of Rennes, France. During these experiments we not only compared the partial skyline extracted from the image with the panoramic skyline generated in the position provided by GPS. Instead we compare it with a set of panoramic skylines located around this position in order to pick out the position with the highest similarity value, through which we not only estimated the orientation of the camera but also refined the translation vector. Figure 5 shows two results we got from our experiment, from which we can see the GIS model (blue wireframe) is well aligned with the buildings. As we can see from the result that the upper boundary of the building is better registered than the vertical boundary, which is because the algorithm is only focused on getting the best matching between skylines which is only related to the upper boundary of the building. Such characteristic also makes this method fail when the upper boundary of the building is quite different from the GIS model (as our GIS model does not contain any roof slope information it always fails when buildings have significant roof height) or such boundary is sheltered by trees or something else.

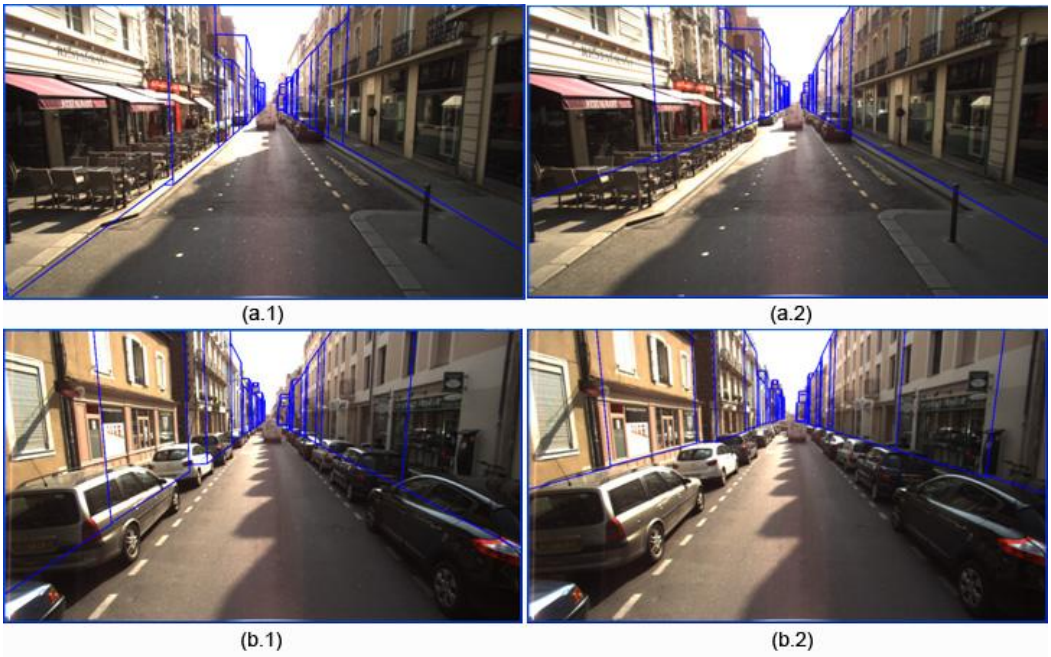

Fig.5. Two experiment results of skyline matching method with the photos taken in the city center of Rennes. (a.1) and (b.1) are initial poses estimated by GPS differences; (a.2) and (b.2) are refined pose after applied skyline matching method; blue frame represents the GIS models.

6 Conclusion

Based on the concern of developing an efficient and pure geometry solution for Video/GIS registration, we investigated the possibility to use skyline feature as a bridge to establish the

correspondence between real images and GIS models. Through our skyline matching method we find out a way to transform an image/model matching problem into a curve matching problem which significantly reduces the computation complexity (time for one Cross-Similarity matching is around 30ms and for a complete skyline matching without optimization is around 160ms in a PC platform with Inter Core2 CPU E8400. 3.00GHz) and makes it a possible solution for real-time applications. Its accuracy and robustness was tested on a set of both synthetic and real images. Despite its simplicity, the skyline matching method gives promising results in the case of an artificial environment like a city center. Our future work will be focused on improving its robustness in case of occlusions between building and sky such as tall trees and also improving its efficiency by using GPU calculation. On the other hand we will try to ally this method with vanishing point estimation which can be used to rectify the image into horizontal position which is the basic demand of this method.

References:

1. T. Langlotz *et al.,* Sketching up the world: in situ authoring for mobile Augmented Reality. Pers. Ubiquit. Comput., 1-8 (2011).
2. G. Sourimant, T. Colleu, V. Jantet, L. Morin, K. Bouatouch, Toward automatic GIS–video initial registration, Ann. Telecommun **67**, 1-13 (2011).
3. T. Colleu, G. Sourimant, L. Morin, Automatic Initialization for the Registration of GIS and Video Data. *3DTV'08 Conference,* 49-52 (2008)
4. R.Y. Tsai, A Versatile Camera Calibration Technique for High-Accuracy 3D Maching. IEEE T. Robotic Autom. **RA-3**, 323-344 (1987).
5. W. Zhang, J. Kosecka, Image Based Localization in Urban Environments. *3DPVT'06*, 33-40 (2006).
6. C. Arth, D. Wagner, M. Klopschitz, A. Irschara, D. Schmalstieg,Wide area localization on mobile phones. *2009 IEEE/ACM -ISMAR*, 73-82 (2009).
7. G. Reitmayr, T. Drummond, Going out: robust model-based tracking for outdoor augmented reality. *2006 IEEE/ACM-ISMAR*, 109-118 (2006).
8. N. Bioret, M. Servières, G. Moreau, Urban Localization based on Correspondences between Street Photographs and 2D Building GIS Layer. *CORESA-2009* (2009).
9. M. Pollefeys *et al.,* Detailed Real-Time Urban 3D Reconstruction from Video. Int. J. Comput. Vis. **78**, 143-167 (2007).
10. S. Ramalingam, S. Bouaziz, P. Sturm et al., SKYLINE2GPS: Localization in Urban Canyons using Omni-Skylines, *IROS' 10*, 3816-3823 (2010).
11. A. Nüchter, S. Gutev, D. Borrmann et al., Skyline-based registration of 3D laser scans, Geo-spatial Info. Sci. **14(2)**, 85-90 (2011)
12. W. Lie, T. T. Lin, K. Hung, A robust dynamic programming algorithm to extract skyline in images for navigation. Pattern. Recogn. Lett. **26**, 221-230 (2005).