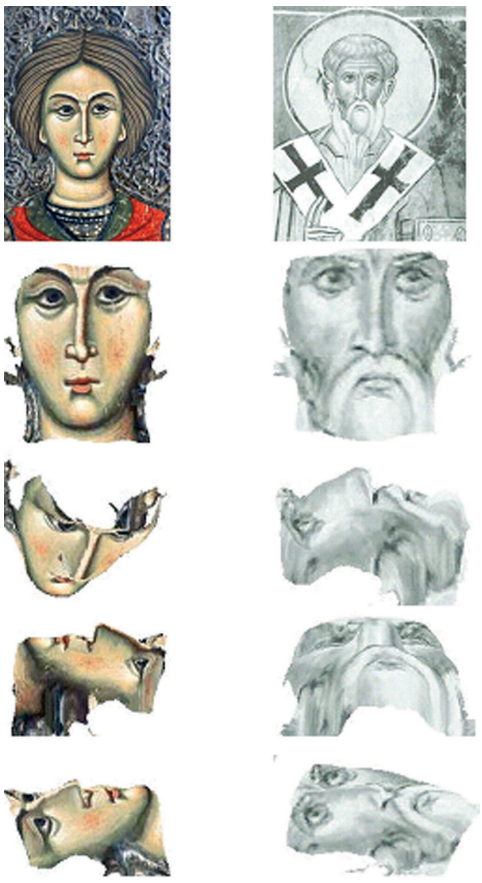
VSMM 2008

Digital Heritage

Proceedings of the 14th International Conference on Virtual Systems and Multimedia

Full Papers



20–25 October 2008
Limassol, Cyprus

M. Ioannides, A. Addison, A. Georgopoulos, L. Kalisperis (Editors)

Marinos Ioannides
Editor-in-Chief

Elizabeth Jerem
Managing Editor

Fruzsina Cseh, Elizabeth Jerem
Copy Editors

ARCHAEOLINGUA
Cover Design

Cover image: 3D reconstruction of faces appearing in Cultural Heritage Artifacts. The top row shows actual images showing the faces of Saint Mamas and Saint Tychikos from the wallpaintings in the Church of Panayia Phorbiotissa - Asinou in Cyprus (UNESCO World Heritage Site: http://whc.unesco.org/en/list/351). The remaining rows show the corresponding reconstructed 3D models as seen from different viewpoints. More information about the 3D reconstruction method can be found in the research paper "Reconstructing 3D Faces in Cultural Heritage Applications" by A. Lanitis and G. Stylianou.

ISBN 978-963-8046-99-4

Published by ARCHAEOLINGUA
Printed in Hungary by PRIMERATE

Budapest 2008

VSMM 2008

Digital Heritage
Proceedings of the 14th International Conference on Virtual Systems and Multimedia

20–25 October 2008

LIMASSOL, CYPRUS

Conference Chair
Marinos Ioannides CY

Co-Chairs of the International Scientific Committee (ISC)
Andreas Georgopoulos GR, Loukas Kalisperis CY/USA, Alonzo Addison USA

Paper Review Chair
Andreas Lanitis CY

Workshop Chair
Denis Pitzalis FR

Local Organizing Committee

Yiorgos Chrysanthou
Christis Z. Chrysostomou
Ioannis Eliades
Diofantos Hadjimitsis
Andreas Hadjiprokopis
Achilleas Kentonis
Andrew Laghos
Christos Lambrias
Andreas Lanitis
Anna Marangou
Antonis Maratheftis
Demetrios Michaelides
Stratos Stylianidis
Georgos Stylianou
Kyriakos Themistokleous
Marina Tryfonidou

International Scientific Committee

Abdelaziz Abid, FR
Alonzo Addison, USA
Orhan Altan, TR
Angelos Amditis, GR
Alfredo Andia, USA
David Arnold, UK
Alessandro Artusi, IT
Manos Baltsavias, CH
Juan A. Barcelo, ES
Richard Beacham, UK
Anna Bentkowska-Kafel, UK
J-Angelo Beraldin, CA
Niels Ole Bernsen, DK
Massimo Bertoncini, IT
Nicoletta Di Blas, IT
Jan Boehm, DE
Paul Bourke, AU
Rosella Caffo, IT
Panagiotis Charalambous, CY
Stavros Christodoulakis, GR
Yiorgos Chrysanthou, CY
Christis Z. Chrysostomou, CY
Paolo Cignoni, IT
Sabine Coquillart, FR
Andrea D'Andrea, IT
Uzi Dahari, IL
Adel Danish, EG
Rob Davies, UK
Andy Day, UK
Martin Doerr, GR
Michael Doneus, AT
Pierre Drap, FR
Sabry El-Hakim ,CA
Ioannis Eliades, CY
Dieter W. Fellner, AT
Maurizio Forte, IT
Bernard Frischer, USA
Sakis Gaitatzis, CY
Andreas Georgopoulos, GR
Luc Van Gool, CH
Stephen M. Griffin, USA
Pierre Grussenmeyer, FR
Norbert Haala, DE
Diofantos Hadjimitsis, CY
Klaus Hanke, AT
Sven Havemann, AT
Sorin Hermon, IT
Jeremy Huggett, UK
Marinos Ioannides, CY
Babis Ioannidis, GR
Charalambos Ioannidis, GR
Yiannis Ioannidis, GR
Wassim Jabi, USA
Loukas Kalisperis, CY/USA
Sarah Kenderdine, AU
Timo Kunkel, UK
Marios Kyriakou, CY
Eleni Kyza, CY
Andrew Laghos, CY
Christos Lambrias, CY
Andreas Lanitis, CY
Celine Loscos, UK
Jose Luis Lerma, ES
Katerina Mania, GR
Keith May, UK
Despina Michael, CY
Demetrios Michaelides, CY
David Mullins, IE
Christiane Naffah, FR
Massimo Negri, IT
Steve Nickerson, CA
John Mackenzie Owen, NL
George Papagiannakis, CH
Petros Patias, GR
Sumanta Pattanaik, USA
Denis Pitzalis, FR
Daniel Pletinckx, BE
Chryssy Potsiou, GR
Mario Santana Quintero, BE
C. Renaud, FR
Julian D. Richards, UK
Seamus Ross, UK
Nick Ryan, UK
Robert Sablatnig, AT
Fathi Saleh, EG
Donald H. Sanders, USA
Pasquale Savino, IT
Michael Scherer, DE
Holly Schlaumeier, UK
Roberto Scopigno, IT
Stratos Stylianides, CY
Georgos Stylianou, CY
Nadia M. Thalmann, CH
Juan Carlos Torres, ES
Olga De Troyer, BE
Marina Tryfonidou, CY
Nicolas Tsapatsoulis, CY
Giorgio Verdiani, IT
Maria Luisa Vitobello, IT
Krzysztof Walczak, PL
Aloysius Wehr, DE
Martin White, UK

VSMM 2008
Digital Heritage – Proceedings of the 14th International Conference on Virtual Systems and Multimedia
M. Ioannides, A. Addison, A. Georgopoulos, L. Kalisperis

AN AUGMENTED 3D ALBUM BASED ON PHOTOS AND BUILDING MODELS

Pablo C. Elias [a], Asla M. Sá [a], Alberto Raposo [a], Paulo Cezar Carvalho[b], Marcelo Gattass [a]

[a] Catholic University of Rio de Janeiro, Computer Graphics Technology Group
R. M. de S. Vicente, 225, Rio de Janeiro, RJ, Brazil, 22453-900
(pelias, asla, abraposo, mgattass)@tecgraf.puc-rio.br
[b] IMPA, Instituto Nacional de Matemática Pura e Alicada, Rio de Janeiro - pcezar@impa.br

KEY WORDS: Calibration, Augmented Reality, 3D Photo Navigation, Image Processing, Camera Reconstruction

ABSTRACT:

Camera calibration is an important step in computer vision algorithms to interpret and reconstruct the three-dimensional structure of a real scene from a set of digital pictures or videos. Captured images are blended via computer vision techniques with synthetic images from scene models in order to render new applications, called augmented reality applications. Among the many uses for this technology this work has particular interest in those applications concerning augmented visits to buildings. These applications, produce images of buildings — typically old structures or ruins —reconstructed from virtual models inserted into captured pictures, allowing one to visualize the original appearance of those buildings. This work proposes an efficient, semi-automatic method to perform such reconstruction and to register virtual cameras from real pictures and models of buildings. This method makes it possible to compare photos and models through direct superimposition and to perform a three-dimensional navigation between the many registered pictures. Our method requires user interaction, but it is designed to be simple and productive.

1. INTRODUCTION

In photography collections on a same subject stored in data files, very often the file names carry little or no information about their specific contents. This makes storing and searching for pictures by content a difficult task.

With the possibility of 3D registration, many search functionalities for a given picture based on a region of the virtual scene can be developed in order to facilitate the management of image sets. The proposal of a three-dimensional picture filing system is not original, but what we propose herein is an application aimed at the cultural heritage context, focusing on the demands and the users involved in this field. For instance, in the case of incomplete buildings or ruins, it is possible to use a three-dimensional model of the finished building according to its historical registers, creating an augmented reality application in which real pictures can be compared with the complete virtual model.

The main functional requirements for the proposed application are: interactive positioning of virtual cameras based on a sparse set of pictures and computer models; ability to register the several reconstructed cameras in the synthetic scene; and finally, three-dimensional navigation among the registered cameras. The proposed technique assumes a known geometric model of the building and presents original techniques to position the pictures in relation to the model.

Camera reconstruction is one of the critical issues of computer vision. It consists in retrieving parameters that define the mathematical model of a camera (known as virtual camera) from the image of a real scene. Our method approaches the problem of computing the camera positions from a single picture, without using calibration markers applied to the building scene. To allow for a general position system the user must interactively navigate in the virtual model to provide an approximate camera position.

The proposed method uses integrated techniques that help the user create a set of corresponding features throughout digital images and the virtual model of a building. Based on these associations, the position of the camera that originated the real picture related to the model can be retrieved, thus providing navigation possibilities over the image set provided.

Although we propose a semi-automatic method, our basic rule is to demand little user interaction, minimizing the user's efforts. The interface concept proposed allows the user to handle *only the image* in order to perform the detection of associations with points of the world – direct associations between segments of the image and the model are not necessary. Nonetheless, camera reconstruction can also be made through a set of image-model associations explicitly provided by the user.

The test application implemented provides a retrieval solution and camera registration based on the semi-automatic matching between CAD models and pictures. The application provides tools to compare them, and a new three-dimensional navigation experience over the several pictures registered.

The case study presented involves a set of pictures and a model of a monastery from the 17th century, currently in ruins. However, as will be seen, there are no restrictions regarding the type of model used. Therefore, it is also possible to use models of objects whose structure is more precisely known, such as engineering or CAD models of modern buildings.

The paper is organized as follows: the next section summarizes the two main references related to this work and points out the differences in our proposal. In Section 3 the pre-processing steps required to the main method are presented. In Section 4

the camera position recovery technique is presented. Results obtained with our case study are shown in Section 5. Finally, conclusions and future work are presented in the last section.

2. RELATED WORK

Two photogrammetric systems have inspired the present work: Microsoft's Photosynth (Microsoft Live Labs, 2008) originated by Noah Snavely's work (Snavely et al., 2006), and Façade, proposed by Paul E. Debevec (Debevec, 1996).

Photosynth's input is a dense set of pictures of an object with overlapping regions. It retrieves clouds of three-dimensional points and camera models of these pictures based only on the matching features among them, which are calculated using the RANSAC (Fischer and Bolles, 1981) and SIFT (Lowe, 2004) algorithms. After matching features are located automatically, epipolar geometry is used to compute the fundamental matrix and retrieve world points, being refined according to the number of matches found. Even though this technique is successful, presenting efficient results especially regarding camera retrieval, it imposes an operational condition that is not desired in the present work: it assumes the availability of a dense set of pictures with overlapping regions, and it requires a long processing time.

Façade, on the other hand, adopts a semi-automatic approach in which a sparse set of pictures is used and the model's geometry is partially known. The camera model is reconstructed with the purpose of retrieving the proportions from a parametric model that was previously created by the user, associating several marks in several images. Debevec was based especially on (Taylor and Kriegman, 1995) to reconstruct the three-dimensional structure of a scene based on multiple pictures. He has also proposed a new method to estimate an initial camera rotation using orientations that are known in the scene. The camera adjustment technique and the method for matching models and images that we propose herein were motivated by such work.

In the initial stage of our research, the technique proposed by Debevec was considered. We concluded that it demands too much user interaction, conflicting with our requirements. Moreover, the final results of this technique are as good as the quality of the parametric model created and the marks made to the images. However, Debevec uses this technique with the principal goal of reconstructing the proportions of a rough model, which will subsequently be used as input for other integrated techniques, such as View-Dependent Texture Mapping (Debevec, 1996), depth map generation using the model as restriction, and other techniques that increase the quality of the final rendering – these are also outside our goals. The Façade system created by Debevec became the base for other commercial systems, and is successfully used to effectively retrieve model proportions and camera positions.

The fact that we assume the model's geometry to be known implies a substantial difference in relation to Photosynth and Façade's purposes, as well as allows a simplification of the camera reconstruction process. However, despite the different approaches, some concepts of Snavely's final application were used here to develop the test application, such as the calibration system using EXIF tags and the general concept of three-dimensional photo album.

3. PRE-PROCESSING

Two pre-processing steps are performed independently and concluded before the main method starts, namely: camera intrinsic parameter retrieval, and geometry loading and processing.

3.1 Camera Intrinsic Parameter Retrieval

A calibrated camera is one in which any given point **x** in its projection plane can be related to the respective ray **d** connecting its optical center to **x**. Formally, this means finding a calibration matrix **K** representing a transformation between this point **x** and the ray's direction, i.e., such that $\mathbf{d} = \mathbf{K}^{-1}\mathbf{x}$ (Hartley and Zisserman, 2003).

In this work, we have adopted the CCD pinhole camera model, which is similar to the classical pinhole model but taking into account that the pixels in the projection plane might not be square. In this case, the matrix **K** is given by:

$$K = \begin{bmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Entries in **K** are usually called the camera's intrinsic parameters. We refer to the reconstruction of those intrinsic parameters as the camera calibration stage. Such parameters are computed for each input image provided. If the image has proper EXIF information, the camera's geometry can be extracted directly from the image file metadata. If such information is not available, the calibration is made based on the vanishing points of the three principal directions.

Several authors (Rother, 2002; Schaffalitzky and Zisserman, 2000; McLean and Kotturi, 1995; Gamba et al., 1996) developed techniques for the automatic detection of vanishing points, but they only work in cases with little noise and in the presence of good straight line segments that can be detected in the image. Besides, the probability of false positives is high, i.e., often segments that do not belong to any of the three directions are detected, because straight lines that are perpendicular to the world can be projected as being almost parallel to the image.

Therefore, we have opted for a semi-automatic approach, with the principal directions of the scene being provided by the user, as illustrated in Figure 1. This picture was taken in the 1980's, so it hasn't EXIF tags and is very noisy, due to the digitalization process, representing a very difficult case for a full automatic approach.

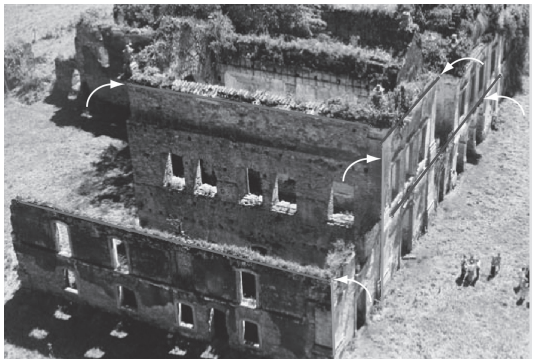


Figure 1: Example of simple vanishing point calibration. Marked lines are indicated by the white arrows.

The method implemented for camera calibration is the one suggested by (Hartley and Zisserman, 2003). Although the vanishing point calibration provides a good estimation of the camera's orientation, only its intrinsic parameters are calculated at this stage. Camera orientation and translation will be obtained subsequently.

3.2 Structural Edge Extraction

Geometric meshes generated in modeling applications often present many edges that cannot correspond to any segment of the input image, either because they are not visible or because they result from mesh representation processes with rendering purposes rather than favoring an economical structural description of the model.

The virtual model pre-processing stage is comprised of a sequence of simple operations, resulting in a new list of edges called *structural edges* of the model. These are used to facilitate the location of matches between the model and the image. The pre-processing operations carried out on the original model are the following:

- Discarding duplicated edges;
- Discarding coplanar edges;
- Aggregating sequences of collinear edges;
- Normalizing the mesh and subsequently discarding short edges;
- Possibility of manually discarding edges.

Figure 2: Part of the original mesh of a model displayed in wireframe.

Tests of the simplification process using a model with over 1,000,000 edges resulted in a list with a little over 3,000 edges (0.3%), most of which will be discarded during the occlusion test that will be described below. In fact, in the model illustrated in Figures 2 and 3, a little over 90 visible edges were selected on average, to mark the relations between model and image.

Figure 3: Some of the selected visible structural edges marked in yellow on the model.

When a model's mesh is first loaded, it is processed and the list of structural edges is stored in an auxiliary file (.simp). The model's original geometry will be used again to enhance users visual experience and for the occlusion test stage described below.

4. CAMERA POSITION RECOVERY TECHNIQUE

To retrieve the point of view of a given camera, the identification of corresponding features between the picture and the virtual model is essential. The approximate virtual model of the scene captured in the images plays the role of calibration pattern for the cameras. This requires knowing its real proportions and processing the images so that the corresponding features between the images and the virtual model can be identified.

Since our case study is applied to buildings, the identification of edges rather than points was preferred, because they are easier to track in pictures of buildings. The approach adopted gives the user an important role manipulating the picture-model associations, especially where, due to reasons regarding the quality of the input image, it is not possible to detect a sufficient amount of good-quality edges.

The proposed method has the purpose of helping the user as much as possible, so that in simple cases where the image has little noise the user might not need to provide extra segments for the application to retrieve the camera. In these cases, the user will only need to provide one adequate initial camera position.

4.1 Initial Solution

The initial position of the virtual model, corresponding approximately to the point of view of the picture whose camera is to be retrieved, is assumed to be provided by the user. To position the model, the user enters manually, through a trackball-type manipulator, a translation and a rotation to be applied to the virtual camera. The goal is to align, as much as possible, the model's structural edges with the corresponding edges of the picture that is mapped onto the camera projection

plane. From the observer's point of view, the image is always fixed in the screen space, while the model moves (Figure 4).

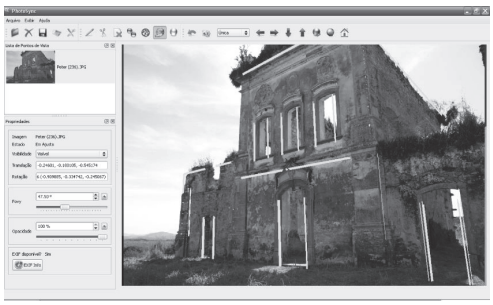


Figure 4: Model superimposed by an image with opacity of 100%. Even with maximum opacity, the guidelines are still visible and the model can be manipulated.

Using the user-provided initial position, the structural edges of the virtual model undergo an occlusion test that takes the original model into account, with the purpose of discarding structural edges which are not visible from the point of view of the initial position.

To implement the occlusion test, OpenGL's Occlusion Query functionality is used. The procedure is simple: the original model is rendered; then each of the structural edges is rendered; then the Occlusion Query verification is carried out. If the rendered edge generates visible pixels, then it is visible and must be maintained. Otherwise, the edge is discarded. The final result consists of a set V of visible structural edges.

4.2 Matching Structural Edges and Images

The semi-automatic method proposed to solve the problem of associating features of the picture to the virtual model consists of 3 stages: 1) edge detection on the image; 2) interpretation of the correspondence between the detected edges and the visible structural edges of the virtual model; 3) intervention and validation of matching features by the user.

4.2.1 Edge Detection on the Image

To locate straight line segments, Canny's filter is applied to the pictures to highlight the edges (Canny, 1986). Then, specific edge-detection algorithms are applied. We have evaluated the performance of two algorithms: the Standard Hough Transform (SHT) (Trucco and Verri, 1998) and the Progressive Probabilistic Hough Transform (PPHT) (Matas et al. 1998). The latter provides straight line segments on the image rather than straight line equations, as in SHT, as well as being significantly more efficient. The OpenCV library (Intel, 2008) was used for the implementation of both methods.

It was observed that each method presents a different performance depending on the input image quality, therefore it is not possible to single one out as the best in all cases. PPHT is more efficient when dealing with images with a significant amount of noise (such as the one in Figure 5), as was already mentioned; however, it is less precise. The choice for one of these two methods depends directly on the nature of the input image. The number of quality segments detected by PPHT was bigger than the test made with SHT, and the execution time was considerably smaller in high-resolution images.

Figure 5: Difficult case. Over 400 edges detected, most of them on the vegetation.

As a result of this stage, we obtained a set $U = \{u_1,...,u_n\}$ of edges from the image.

4.2.2 Matching Image and Structural Edges

The structural edges of set $V = \{v_1,...v_n\}$ those that passed the occlusion test – are then projected onto the image space using the intrinsic camera parameters obtained in the pre-processing stage. Thus, a set $V' = \{v'_1,...,v'_n\}$ of straight line segments, given by the projection of the edges in set V onto the image space, is created.

Our idea is to use the model to restrict the area where the corresponding segments in the image will be searched, creating a window around each segment v'_i and thus defining a Region Of Interest $ROI(v'_i)$ relative to each edge. Each ROI works as a sub-image, restricting the search area for edges of set U that are candidate to match a structural edge v_i.

To select an edge from U that corresponds to an element within v_i, it is reasonable to analyze only those elements u_i contained in $ROI(v'_i)$, giving preference to elements with inclination, size and extremities comparable to those of V', because we assume the initial position of the model provided by the user to be approximately adjusted to the picture.

To objectively classify the edges of U that are matching candidates, i.e. that are inside the effective region of a ROI, we have adapted the error function proposed in (Taylor and Kriegman, 1995). Thus, each edge u_i from U inside $ROI(v'_i)$, is classified according to the following function:

$$rank(u_i) = \frac{l}{h_1^2 + h_1 h_2 + h_2^2} \qquad (2)$$

where l is the size of segment u_i and h_1 and h_2 are the distances from the extremities of segment u_i to the straight line L_i, built from the endpoints of a projected model segment v_i (Figure 6).

The edge u_i with the best classification according to function *rank* is chosen as the one in the image that matches the structural edge v_i from the virtual model.

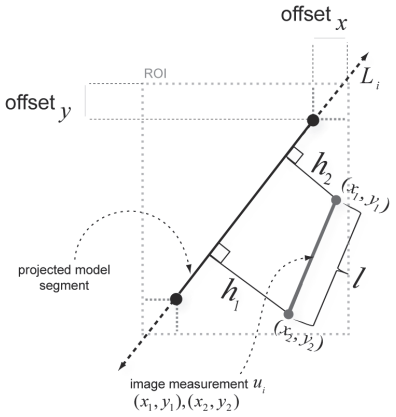


Figure 6: Edge classification within a ROI.

As can be observed, the *rank* function classification prioritizes large edges over smaller ones, but small edges that are closer and more aligned have priority over larger edges that are farther away or less aligned. The quality of the classification obtained depends on the position of the user-provided virtual model.

As this is a greedy local-search method, if there are intersections between *ROIs*, it might be the case that the same edge u_i in the image be classified as matching two or more different structural edges of the model, which is an inconsistent solution. To overcome this problem, the tie-breaker criterion adopted to select the structural edge that best corresponds to edge u_i in the image consists in comparing the value of $rank(u_i)$ in relation to the different edges v_i classified as matching the same u_i, then again adopting a greedy strategy.

4.2.3 Validation and Manual Edition of Matching Features

The matching features between the model and the picture obtained automatically are finally displayed for the user using visual information to allow the user to discard them, approve them or complement them. The complementation entered by the user constitutes of new segments made only on the image. Based on these segments and on the initial camera position provided by the user, the best candidates to corresponding edges of the model are classified. Only in cases where it is impossible to detect matching features based on proximity (due to the low quality of the image), direct manual association between model and image is necessary.

Complementation is required when it is not possible to detect edges in the three principal directions of the scene, which is an essential requirement for the subsequent stage of camera adjustment (minimization). This can occur especially due to noise or occlusion. In this case, the user is required to complement the set of detected edges by marking new edges in regions where no or few segments were found.

In resume, the output of the matching step is a set of pairs $\{(v_1,u_1), (v_2,u_2), \ldots, (v_n,u_n)\}$ to be used as input to the minimization step.

4.3 Optimizing Camera Position

Using the set of associations between model and image edges, already validated by the users, and the camera initial position, the next step is the camera position adjustment, in order to maximize the alignment between corresponding segments in the image and in the model.

To achieve this maximum alignment, it is necessary to measure the error between each pair (v_i, u_i). With these measures, it is possible to recover the position and orientation of a calibrated camera, by means of the minimization of an objective function that depends on the external parameters and the obtained correspondences.

We use again the error function proposed by (Taylor and Kriegman, 1995), already presented in the previous section. That is related to the image formation process, modeled as a projection function $\mathbf{P}(\mathbf{R},\mathbf{T},v_i)$, where $\mathbf{R}$ is the rotation matrix and $\mathbf{T}$ is the translation matrix. The projection function produces an ideal bi-dimensional segment that is used to measure an error between the projected model and the user detected image lines u_i, given by:

$$O = \sum_{i=1}^{t} Error(\mathrm{P}(\mathbf{R},\mathbf{T},v_i),u_i) = \\ = \sum_{i=1}^{t} \mathbf{m}^T(\mathbf{A}^T\mathbf{B}\mathbf{A})\mathbf{m} \tag{3}$$

where,

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix}, \mathbf{B} = \frac{l}{3(m_x^2 + m_y^2)} \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \tag{4}$$

and $\mathbf{m} = \{m_x, m_y, m_z\}^T$ is the normal vector shown in Figure 7.

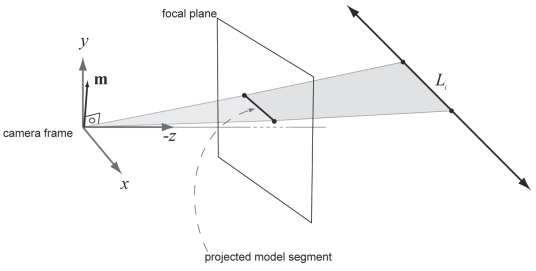


Figure 7: The representation of the line L_i by the normal vector **m**.

This function measures the total error between model and image edges in function of the camera external parameters, which are recovered using a minimization process to find their optimized values.

This objective function to be minimized is not linear. Therefore, in order to achieve the minimization, we approximate the non-linear function by a quadratic function and optimize it using a gradient-descent method.

5. RESULTS

5.1 Case Study

The set of pictures used as case study represents the ruins of the São Boaventura convent, which was built around 1660. The convent's model was created based on historic documents and photographic registers. The model is not totally precise in relation to the ruins' structure, but it suffices to illustrate and test the application and the method proposed.

The pictures employed include a lot of vegetation and noise, adding to the high complexity of this case study. Nonetheless, a set of 10 cameras was successfully retrieved from the respective pictures, in less than 10 minutes.'

Some results are shown in Figures 8 to 11:

Figure 8: A match between the model and a picture of the convent.

Figure 9: The same picture as in Figure 9, shown with zoom using a different orientation.

Figure 10: The model and its structural edges shown in yellow.

Figure 11: The match for the same camera position and orientation shown in Figure 11. The structural edges of the model are shown over the picture.

5.2 3D Photo Navigation

A test application was created to demonstrate the proposed method, performing semi-automatic retrieval of camera positions of pictures from a building in relation to its virtual model. Each retrieved camera can be registered, composing a picture filing and facilitating the management of the photographs in 3D space.

Moreover, the system provides a three-dimensional navigation experience over the pictures, offering a number of tools to compare details on the pictures and the model, such as zoom and transparency control.

The tridimensional navigation was implemented in a simple way to provide an efficient and easy mechanism for searching for pictures. Although such mechanism do not intend to be the best or the final solution for tridimensional navigation between camera positions, it can be used to illustrate the utility of the virtual camera recovering process. With such virtual cameras in hand, a new set of possibilities is opened allowing the creation of new features for picture filing, search and comparison between pictures and building models.

The tridimensional navigation algorithm receives as input a specific direction and the set of recovered cameras and produces and output that can be empty (in the case that no camera is sufficiently near) or represent another camera in the given set (chosen as being nearest in the given direction). Given a specific direction, one can navigate to another near camera in a specific direction using the criteria shown below:

$$dist(\hat{\mathbf{d}}, \hat{\mathbf{v}}) = \frac{1}{\langle \hat{\mathbf{d}}, \hat{\mathbf{v}} \rangle} \|\mathbf{d}\| \quad (5)$$

where *dist* is the distance classification function between a camera (*i*) in the given set and the current camera, $\hat{\mathbf{d}}$ is the unit vector that marks the direction to the optical center of the next camera *i* from the current camera and $\hat{\mathbf{v}}$ is the chosen navigation direction, represented in the local space of the current camera (Figure 12).

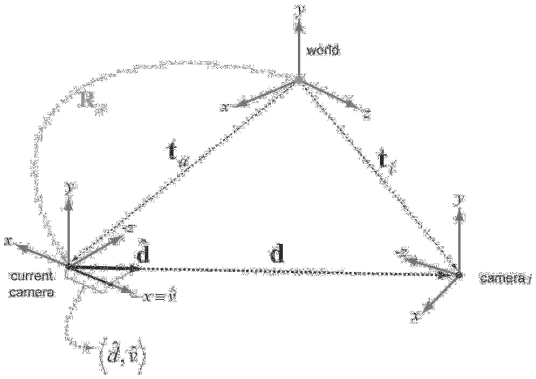


Figure 12: A simple strategy for tridimensional navigation between recovered cameras.

By using virtual cameras it is also possible to recover the position from where a picture was taken in the real world if one of the dimensions of the real model is known in one of the pictures of the used set. This allows one to build observations of a specific part of the model during the time (using the same world position). For example, an old picture can reveal that some parts of the real model are currently damaged or inexistent, as illustrated in Figures 13 and 14. If the position this old picture was taken is recovered, it is possible to take new pictures of the model from the same position, which makes possible to observe the same part of the model across the time.

Figure 13: A match between the model and a picture of the convent. The picture is shown with 100% of the opacity. This picture was taken in the early 80's so no EXIF info is available. Vanishing point calibration was used.

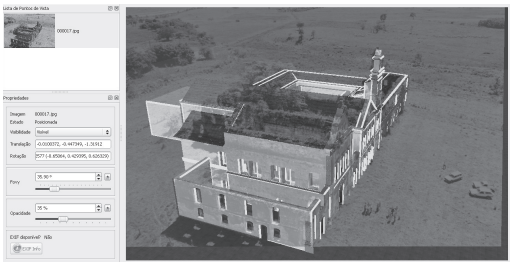


Figure 14: The same match shown in the Figure 13, using around 30% of opacity.

Another very useful functionality related to the production of archives for cultural heritage items is the possibility of verifying whether the set of pictures provided offers a complete view of a given building. To implement such functionality, one can simply project the pictures registered on the model by applying an inverse projective transformation using matrix $\mathbf{K}^{-1}$, and observe the parts of the model that are not covered by the pictures.

6. CONCLUSIONS AND FUTURE WORK

The present work has proposed a method for matching models and images of buildings using a set of integrated techniques, with camera reconstruction as the main strategy. To perform such reconstruction and successfully match and catalogue the pictures, first we needed to solve the problem of identifying correspondences between elements of the image and the model, which is one of the fundamental problems in computer vision. The approach proposed to solve it was to use the building's model, positioning it in order to restrict the search for matching features on the image. This strategy assumes the virtual model to be manipulated by the user in such a way that the edges can serve as guidelines to locate corresponding features in the image, using a local search strategy in the neighborhood of the projection of the model's edges.

The method is semi-automatic, beginning with an initial solution provided by the user which allows a local search for image-model associations rather than exploring the model's global information. If the input image contains significant noise and the photographed model has complex geometry, solving the matching problem becomes naturally difficult, and the method proposed herein becomes more dependent on user actions and prone to some degree of imprecision. In simple cases, on the other hand, the process is largely automatic and robust in relation to the model's initial position, as it is simpler to compute image-model correlations.

As final result, we have developed an application that implements the proposed method and provides a complete solution for the camera registration problem over pictures related to their virtual model. The system also provides various mechanisms to help the user compare pictures with the model, and navigate spatially over the several registered images.

Several improvements can be considered, especially in relation to the correspondence between points in the image and the world. In this sense, the use of NPR (Non-Photorealistic Rendering) techniques can be explored to detect structural edges in the virtual model and in the images.

A natural extension would be the application of this method to videos. The technique could be further developed to explore space and time coherence in frame sequences obtained from real videos. This extension to frame sequences can be facilitated by the very nature of the technique, which searches locally for corresponding features between model and image. When a sequence of frames from a film is assumed to present space and time coherence, the proposed method can be applied directly to adjust the sequence to the model, moving automatically from the initial frame position to the next frame. The process could begin with a manual positioning of the first video frame by the user, resulting in the retrieval of a complete camera path along a given time span.

As future work, we intend to use a set of pictures well distributed in space as input to a method that correlates them to a virtual model, seeking to improve the quality of an approximate virtual model of the represented building. The model's structural edges could be parameterized in order to be adjusted to the marks made to the pictures. This idea is very similar to the one developed by Paul Debevec (Debevec, 1996), but it would use models directly created by commercial modelers, which would benefit from the convenience of using a good modeler without demanding modeling precision. In other words, a model lacking the exact measures could be quickly created by a designer, and then its proportions would be adjusted based on the input pictures.

Another possibility to extend this work would be to use a dense set of pictures to reconstruct completely the correct proportions of a model, using only associations among the images. This strategy is inspired in the work developed by (Snavely et al., 2006) but it would use as base an imprecise model that could be adjusted and used as restriction for points reconstructed based on picture correlations using epipolar geometry. In (Snavely et al., 2006) only pictures are used, without any information on the model.

REFERENCES

Canny, J., 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), pp. 679-698.

Debevec, P., 1996. Modeling and Rendering Architecture from Photographs. PhD thesis, University of California at Berkeley.

Fischer, M. A., and Bolles, R. C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM,* 24(6), pp. 381-395.

Gamba, P., Mecocci, A., and Salvatore, U., 1996. Vanishing Point Detection By A Voting Scheme. In: *IEEE International Conference on Image Processing*, Lausanne, Switzerland, pp. 301-304.

Hartley, R., and Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridgeshire, UK, 2nd edition.

Intel, 2008. "Open Source Computer Vision Library". http://www.intel.com/technology/computing/opencv/ (accessed 16 Jun. 2008).

Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision,* 60(2), pp. 91-110.

Matas, J., Galambos, C., and Kittler, J., 1998. Progressive Probabilistic Hough Transform. In: *British Machine Vision Conference*, pp. 256-265.

McLean, G., and Kotturi, D., 1995. Vanishing Point Detection by Line Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11), pp. 1090-1095.

Microsoft Live Labs, 2008. "Photosynth". http://labs.live.com/photosynth/ (accessed 16 Jun. 2008).

Rother. C., 2002. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10), pp. 647-655.

Schaffalitzky, F., and Zisserman, A., 2000. Planar Grouping for Automatic Detection of Vanishing Lines and Points. Technical Report, Department of Engineering Science, University of Oxford.

Snavely, N., Seitz, S. M., and Szeliski, R., 2006. Photo Tourism Exploring Photo Collections in 3D. Technical Report, Microsoft at Microsoft Research, Redmond WA, USA.

Taylor, C. J. and Kriegman, D. J., 1995. Structure and Motion from Line Segments in Multiple Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 17(11), pp. 1021-1032.

Trucco, E., and Verri, A., 1998. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Inc, Upper Saddle River, NJ.

ACKNOWLEDGEMENTS

The TecGraf (Computer Graphics Technology Group) is one of the laboratories of the Computer Science Department at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio) and is mainly supported by Petrobras. Alberto Raposo is also funded by CNPq, Process Number 472967/2007-0.