Visual Comput (2005)
DOI 10.1007/s00371-005-0309-z

ORIGINAL ARTICLE

Thomas Franken
Matteo Dellepiane
Fabio Ganovelli
Paolo Cignoni
Claudio Montani
Roberto Scopigno

Minimizing user intervention in registering 2D images to 3D models

Published online: 31 August 2005

T. Franken · M. Dellepiane ·
F. Ganovelli · P. Cignoni (✉) ·
C. Montani · R. Scopigno
ISTI – CNR via Moruzzi 1, Pisa, Italy
firstname.lastname@isti.cnr.it

Abstract This paper proposes a novel technique to speed up the registration of 2D images to 3D models. This problem often arises in the process of digitalization of real objects, because pictures are often taken independently from the 3D geometry. Although there are a number of methods for solving the problem of registration automatically, they all need some further assumptions, so in the most general case the process still requires the user to provide some information about how the image corresponds to geometry, for example providing point-to-point correspondences. We propose a method based on a graph representation where the nodes represent the 2D photos and the 3D object, and arcs encode correspondences, which are either *image–to–geometry* or *image–to–image* point pairs. This graph is used to infer new correspondences from the ones specified by the user and from successful alignment of single images and to factually encode the state of the registration process. After each action performed by the user, our system explores the states space to find the shortest path from the current state to a state where all the images are aligned, i.e. a final state and, therefore, guides the user in the selection of further alignment actions for a faster completion of the job. Experiments on empirical data are reported to show the effectiveness of the system in reducing the user workload considerably.

Keywords Image registration · 3D scanning · Automatic acquisition

1 Introduction

The digitization of real objects, usually referred to as 3D scanning, has become a hot topic in computer graphics. This is due both to the many intrinsic difficulties and technical/theoretical problems involved (see [1] for a survey) and to the wide variety of applications of 3D scanning, which include industrial manufacturing, reverse engineering, character modelling in the movie industry, cultural heritage modelling (virtual museums, restoration planning and documentation [2]), etc.

Many applications require to sample not just the geometry, but also the color information, for example 3D models for the design of virtual museums require reflection attributes. Acquiring the real color of an object, i.e. its surface reflection properties, is a complicated and time consuming task [7, 9] and the methods proposed usually make some assumptions on the photometric property of the material and/or on the light conditions (which usually have to be controlled to obtain good quality results). For most practical cases a simpler approach is adopted: a series of pictures taken by a digital camera are stitched onto the surface of the object, trying to avoid shadows and highlights and taking pictures under favorable light conditions. However, even in this simpler case, the pictures need to be processed in order to build a plausible texture for the object [3].

A basic problem in managing color information is how to register the images with the geometric data. In some cases, the problem has been solved by fixing the camera onto the 3D scanner, so that the relative position of the two devices is known and 2D and 3D data are already aligned [13, 15]. Unfortunately, images are often unregistered since a simpler setup is used (hand-held camera, color acquired in a second stage with respect to 3D scanning). Two main reasons justify the latter choice: the 3D scanner could require light conditions and a scanning setup that is not optimal for cameras and vice versa (sampling resolution is usually very different, and thus the selected set of scanning poses may not be optimal or redundant for the photographic campaign); or the pictures may have been taken by a professional photographer independently from the 3D scanning campaign.

Many papers have addressed the problem of registering 2D pictures to 3D geometry (see Sect. 2). We can still say that there is no fully automatic approach to register 2D images in the general case (i.e. a large and complex object, where each image covers only a subset of its overall extent). The user is usually required to provide correspondences, or hints on the correspondences, which link the 2D images and 3D geometry.

The main goals of this work are: to reduce the user intervention in the process of registering a set of images with a 3D model; to improve the robustness of the process by giving the user the possibility of selecting correspondences which link either 2D points to 3D geometry (*image–to–geometry* correspondences) or 2D points to 2D points (*image–to–image* correspondences). The main idea is to setup a *graph of correspondences*, where the 3D model and all the images are represented as nodes and a link is created for any correspondence defined between two nodes. This graph of correspondences is then used to automatically infer new correspondences and to find the shortest path, in terms of the number of correspondences that must be provided by the user, to complete the registration of all the images. The technique has been implemented in a new image alignment system, which allows us to manage large sets of images on complex models produced with accurate 3D scanning. The paper proceeds as follows: Sect. 2 briefly presents the previous work; Sect. 3 describes the canonical methods used to align a single image, given a set of *image–to–geometry* correspondences; Sect. 4 describes our correspondence graph and its use to infer new correspondences; Sect. 5 shows how the correspondence graph is used to minimize the user workload. Sect. 6 shows a case study to evaluate the benefit of the proposed technique, and finally conclusions and future work are reported in Sect. 7.

2 Related work

Camera parameters estimation involves the computation of *intrinsic camera parameters* (the focal length, the optical center and radial distortion introduced by the lens), and the *extrinsic parameters* (position and orientation of the camera in the global reference system).

Intrinsic parameters can be estimated by providing *image–to–geometry* correspondences or by taking a picture of a known calibration pattern on a planar geometry [16, 18]. In this second case, we can design patterns that can be detected automatically. A number of publicly available libraries for camera calibration are available [4].

Extrinsic parameters, i.e. the view specification associated to a given picture, are often retrieved by providing *image–to–geometry* correspondences and minimizing an error function, which usually is the sum of the differences between each 3D point projected onto the screen and its corresponding 2D feature. The selection of these correspondences can be tedious when many images have to be processed, or complex when we have images that depict regions of the 3D shape with insufficient shape features (e.g. nearly-planar or smoothly curved surface sections).

To avoid the tedious work of providing correspondences, landmarks can be placed onto the real object and can be detected automatically. The use of landmarks has the disadvantage that some image parcels contain the marker rather than the surface color; moreover, stitching markers on valuable or delicate works of art is often prohibited.

When each image covers the entire object, the silhouette of the model in the image and the silhouette contour of the rendered 3D object can be used as matching features. This is done by minimizing the difference between the projection of the synthetically-rendered object and its silhouette in the photo [8, 10–12, 17]. Silhouette extraction requires a 2D segmentation usually easy to achieve automatically (or anyway, with little user intervention). The disadvantages of the silhouette-based methods are that the entire object has to be visible in each image (preventing the use of this method on complex objects where we need a dense photo sampling), and the lack of robustness in the case of symmetrical objects.

3 Aligning a single image

The alignment of a single image to a 3D model is performed by defining all the parameters of the virtual camera whose position and calibration gives an optimal inverse projection of the image on the 3D model. As previously mentioned, camera parameters can be divided in two main groups:

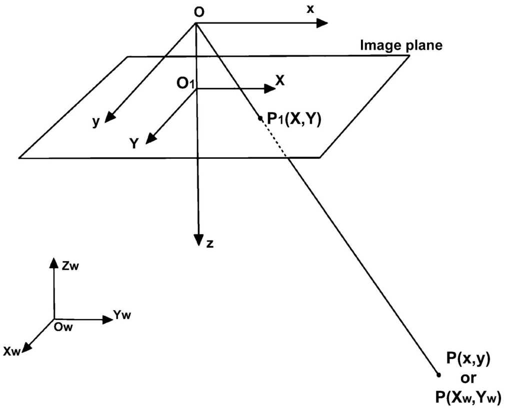

Fig. 1. Camera geometry without radial lens distortion

– *extrinsic parameters*, which model the location and orientation of the camera with respect to a world coordinate system, and
– *intrinsic parameters*, which model the behavior of the optical characteristics of the camera.

Figure 1 shows an example of the camera geometry. Extrinsic parameters can be inferred by the rigid body transformation from a world coordinate system (x_w, y_w, z_w) to the camera 3D coordinate system (*x,y,z*):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{1}$$

where R is the 3×3 rotation matrix and T is the translation vector. These are the parameters that we have to optimize to derive the position and orientation of the camera. The transformation from 3D camera coordinates to distorted image coordinates (u_d, v_d) is regulated by the intrinsic parameter f, which is the focal length:

$$X_u = f\frac{x}{z} \ , \ \ Y_u = f\frac{y}{z} \tag{2}$$

Another intrinsic parameter that can be considered is radial lens distortion. We can calculate the undistorted image coordinates

$$X_d + D_x = X_u \ , \ \ Y_d + D_y = Y_u \tag{3}$$

where

$$D_x = X_d(k_1r^2 + k_2r^4 + ...) \ , \ \ D_y = Y_d(k_1r^2 + k_2r^4 + ...) \tag{4}$$

and

$$r = \sqrt{X_d^2 + Y_d^2} \tag{5}$$

so the parameters to be calibrated are k_i.

The user-driven setup of a few correspondences between 2D points in the image and 3D points on the model is the standard approach to calculate all these parameters. As a matter of fact, the goal is to find the parameter values that minimize the *error function* value, defined as the distance between the selected image points and the projection of the points selected on the 3D model (projected back on the image by using the computed camera intrinsic and extrinsic parameters).

Two different calibrations are implemented in our system and can be selectively used by the user. The first one, based on the Tsai approach [16], needs at least 11 point correspondences (for a fully optimized calibration), and it is able to optimize all extrinsic and intrinsic parameters. The second calibration uses a non-linear method [5] derived from the approach of Faugeras and Toscani [6], which needs at least four correspondences and performs optimization on extrinsic parameters and the focal length value.

Another useful feature of our system gives the user the possibility of optimizing only one (or more) of the variables, e.g. rotation, translation, focal length or lens distortion. This can be useful, for instance, if we are sure about the evaluation obtained for the intrinsic parameters (e.g. because we have performed a pre-calibration of the camera) and thus we need only to optimize the rotation and the translation to align the image.

4 Inferring correspondences

The registration of a single 2D image to a 3D geometry can be performed by linear and non-linear techniques, as presented in the previous section. To solve the task we need to find a sufficient number of *useful correspondences*. Registering 15–20 images to a single 3D geometry can be very timeconsuming and hard to manage, especially when the geometry presents large flat areas with insufficient 3D features or when single images cover a too small section of the 3D model. The former could be the case, for example, of a large mosaic or a very simply-shaped building. The task can be simplified if the user can add correspondences between pairs of images (*image–to–image*). The overlapping areas of the images, due to color changes and texture detail, can often be more useful for inferring new correspondences than the corresponding section of the 3D model.

Our technique addresses the selection of both *image-to–geometry* (I2G) and *image-to–image* (I2I) correspondences and has been developed to help the user to complete the registration of all images in a shorter time, setting a lower number of *image–to–geometry* correspondences.

We define a *correspondence graph*, where the 3D model and the images are represented by nodes. Two nodes are connected by an arc *if* there is a correspondence

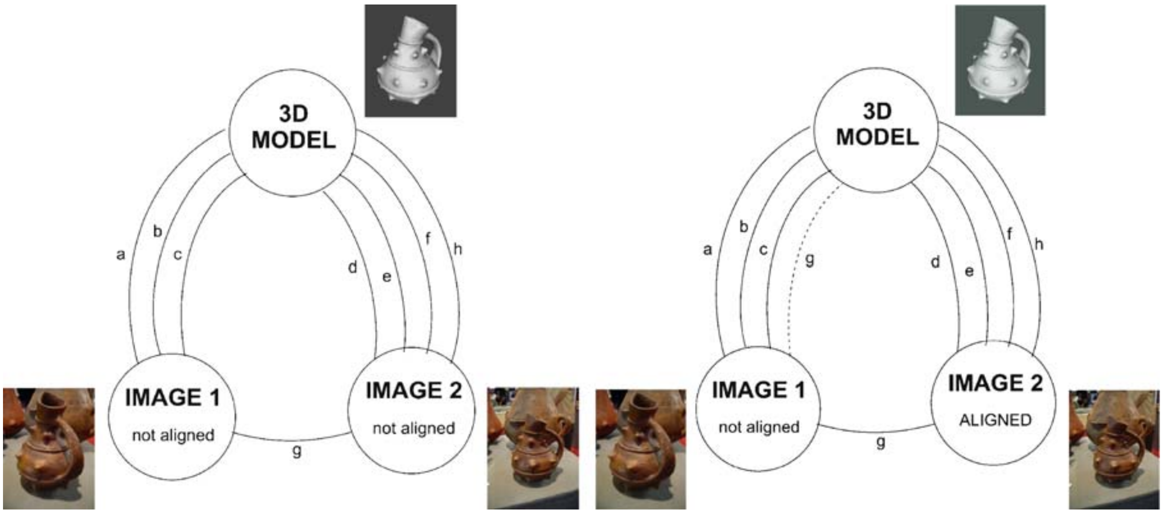

Fig. 2. Example of correspondences graph (on the left). A new *image-to–geometry* correspondence (g) for IMAGE1 is inferred automatically given an *image-to–image* correspondence that links IMAGE1 with IMAGE2 (graph on the right)

between the respective entities, of either type I2G or I2I. We show a very simple correspondence graph in the example in Fig. 2: IMAGE1 is connected with the 3D mesh with three correspondences (i.e. three corresponding point pairs have been selected); IMAGE2 has four correspondences and MAGE1 and IMAGE2 are connected by an arc g, which is a correspondence between points in the images (a I2I correspondence).

Let us exlain what we mean by inferring new correspondences from the I2G and I2I arcs defined in the graph. Automatic inferring of new *image-to–geometry* correspondences can be performed whenever an image I_2 is aligned to the 3D geometry M and to another image I_1 via some *image-to–image* correspondence pairs, since we may infer an implicit *image-to–geometry* between I_1 and M by taking into account the composition of $I2G(I_2, M)$ with $I2I(I_2, I_1)$. This composition is performed by mutually projecting corresponding points: given a point pair (p, q), which defines the correspondence $I2I(I_2, I_1)$ with $p \in I_1$ and $q \in I_2$, by projecting q on the geometry according to $I2G(I_2, M)$, we indirectly connect the point p of image I_2 to the model M. This mechanism is shown in the right-most graph in Fig. 2, which shows what happens when the graph is automatically augmented: the point on IMAGE2 associated to correspondence g is projected on the model, at the same time creating an "indirect" correspondence g (represented in the figure by a dashed line) between IMAGE1 and the mesh. In this case, the act of mutually aligning an image pair caused the creation of a new correspondence between an image and the 3D model, without intervention by the user. This approach can be very useful when an image covers a region of the 3D shape where it is hard to find shape-based correspondences. Using a mixed I2G and I2I approach, the user can now also set correspondences between overlapping images; these correspondences can be used by the system for augmenting the I2G correspondences in model regions that present insufficient shape features. These "indirect" correspondences can be easily created by the system for the more "challenging" images, substantially helping the user to complete the alignment.

5 Minimization of user workload

Automatically inferring new correspondences is an advantage with respect to standard solutions, but exploiting this feature manually can be challenging even with a few images. We show an example of how taking different choices leads to different results in terms of user workload (Fig. 3). Let us start from the hypothesis that at least five correspondences are needed[1] to align an image to the 3D model (single I2G alignments). In Fig. 3, the starting state (uppermost graph) shows that IMAGE1 has only one connection to the 3D model, IMAGE2 has three connections and IMAGE3 has four connections. Observing only the current state, the wisest choice seems to be the creation of a single correspondence from IMAGE3 to the mesh. This leads to the state in the second line in Fig. 3, with a cost equal to one since we added only a new correspondence (arc p). A new correspondence (dashed arc o) can be inferred from IMAGE1 to the 3D model. If we align IMAGE2 to the 3D model, with the cost of two more correspondence (lines q and r), we may also create four new connections (dashed arc i, l, m, n) between IMAGE1 and the 3D model (right-most graph on

[1] The number of correspondences can vary if the image is totally uncalibrated or some parameters (for example, the focal length) are known.

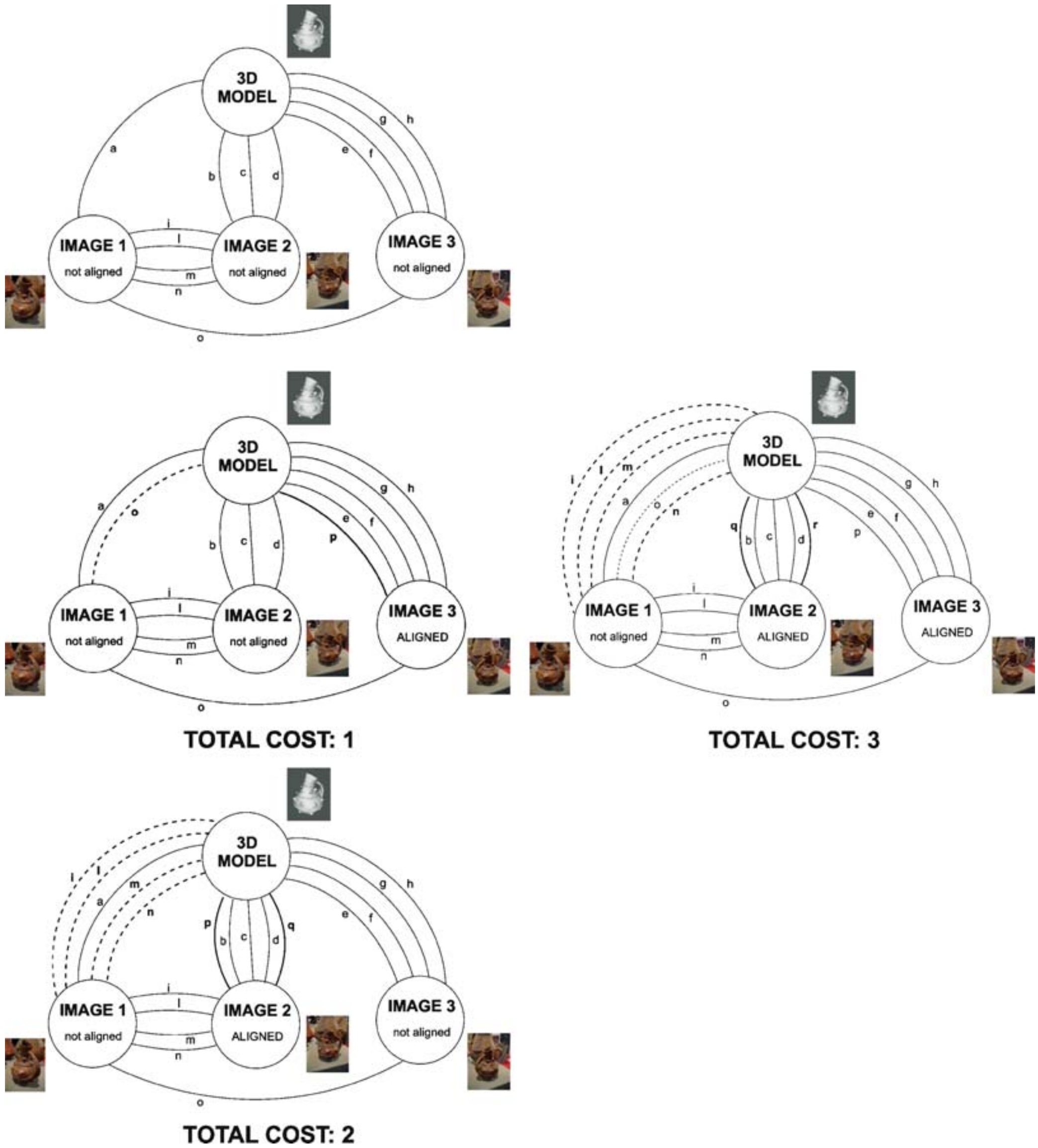

Fig. 3. Different workloads in registration completing. Top: starting state. Middle: registration aligning IMAGE3 for first, then IMAGE2. Bottom: registration aligning IMAGE2 for first, then IMAGE1

the second line, Fig. 3). At this point, IMAGE1 can be aligned without any new intervention of the user and the total cost of the complete registration is equal to three. Conversely, going back to the starting graph layout, if we decide to align first IMAGE2 to the the 3D model, we discover that with a cost of two new correspondences (arcs p and q, bottom-left graph of Fig. 3) IMAGE1 earns four new inferred connections to the 3D model. At this point, IMAGE1 can be aligned without any added cost (Fig. 3, bottom-right) and IMAGE3 has a new inferred connection (dashed arc o). We have reached the minimum number of connections to complete the initial alignment: just two new arcs. Therefore, the second strategy is cheaper than the first one.

From this very simple example the reader can visualize how complex the correspondence graph can become when we have to manage some tens of images. With many tens or hundreds of connections to be selected, the process becomes very difficult for a human operator. The graph of possible future states can be quite complicated, so that a complete exploration of all possible future states results in an excessive time overhead.

So, what we need is a system that suggests the best strategy in order to minimize the number of correspondences to be placed manually. We pose the problem as a *state space search* problem [14].

Note that the correspondence graph encodes the *state* of the alignment, i.e. the set of correspondences that have been placed. If the system is in a state s, and the user places a correspondence, the system moves to the state s'. A goal state, a graph in which all the images are aligned, is a correspondence graph where every image is connected to the model by at least 11 (direct or indirect) corresponding I2G point pairs (the number of correspondences is seven if the intrinsic parameters are already known). More formally, we can define the state space as the quintuple:

$$S = \{N, I, G, A, \sigma\}$$

where N is the set of states, I is the current state when the search is performed, $G \subseteq N$ is the set of goal states, A is the set of actions (in this case, the singleton {*place_a_correspondence*}) and $\sigma : N \times A \rightarrow N$ is the set of transactions.

An exhaustive search on this space is prohibitive, since the branching factor would be $n!/2$, with n number of nodes. In fact, given a correspondence graph, the user can place a correspondence between any pair of nodes. Therefore, we redefine A as the action of aligning an image, i.e. of placing all the correspondences necessary to align an image. In this way, the branching factor becomes n, even if the optimum in terms of the number of correspondences placed is no longer guaranteed, since we will visit only a subset of the state space.

We use a Best First approach: starting from the current state, all the actions that can be performed are evaluated with a *heuristic function* and the corresponding states are put in a priority queue. The algorithm ends when a goal state is found and the corresponding path is reported.

Heuristic evaluation of an action. . The action of aligning an image is evaluated considering two factors: the number of (direct and inferred) correspondences to the mesh, and the number of correspondences to other non-aligned images. The first factor describes the proximity of the image to a possible alignment, the second factor describes the number of new correspondences that the alignment of the image would produce. The result of the search is used to suggest to the user what action to take next, and the search is done after each action. Back to the example mentioned at the beginning of this section, our graph-based system would suggest to the user to align IMAGE2 as a first step, and then to consider IMAGE3 and IMAGE1.

Fig. 4. The dataset used for the test (3D model and all eight images), representing a ceramic dish

6 Results and discussion

We present a simple concrete example where we analyze the improvement brought on by the use of graph correspondences and a workload minimizer. The sample dataset, shown in Fig. 4, consists of a 3D model (nearly 500K faces) of a painted ceramic dish and a set of 8 pictures, taken directly by the rgb unit of the scanner (Konica Minolta VI910). The pictures present quite big overlapping areas and each one covers a small section of the dish. Moreover, the very simple geometry of the plate makes the registration of pictures quite challenging, due to the difficulty in finding relevant surface features.

The screenshot presented in Fig. 5 shows the structure of our application: in the *Workspace* mode the thumbnails of all loaded images are listed in the lower part of the window, and any of them can be dragged to the main part to set new connections (marked as green crosses in the images, or as solid points on the 3D model). The *Calibration* mode, shown in Fig. 6, shows the result of the alignment of an image with the 3D model. The white arrow indicates advice given by the workload minimizer.

In the first test, we asked two subjects to perform a complete registration of the images using our system. The first subject had already used the application previously, hence was "experienced". The other subject tried the application for the first time directly in the test. The subjects performed two registrations of the same dataset (see the screenshot of an intermediate step in Fig. 7), the first time using only I2G correspondences, the second time

Table 1. Results of first test

	Experienced user		Unexperienced user	
	I2G only	I2G and I2I	I2G only	I2G and I2I
Completion time	~35 min	~28 min	~50 min	~35 min
No. of correspondences selected	42	33	51	38
Avg. no. of correspondences for each image (min-max)	9 (8-11)	11 (8-13)	11 (7-14)	12 (9-13)

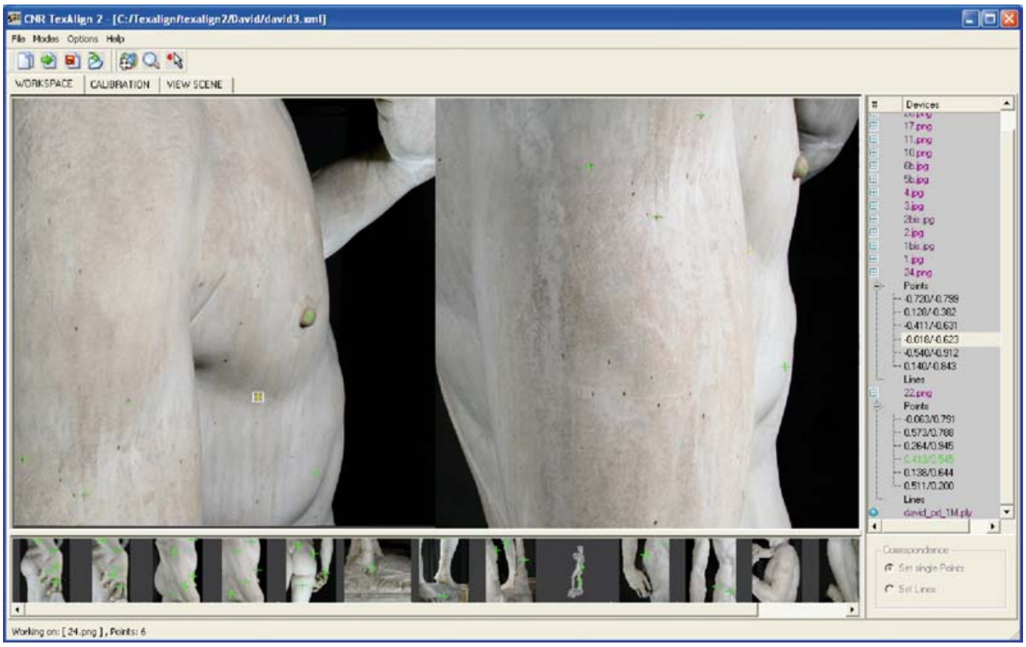

Fig. 5. Workspace of our application, some correspondences between two images of Michelangelo's David

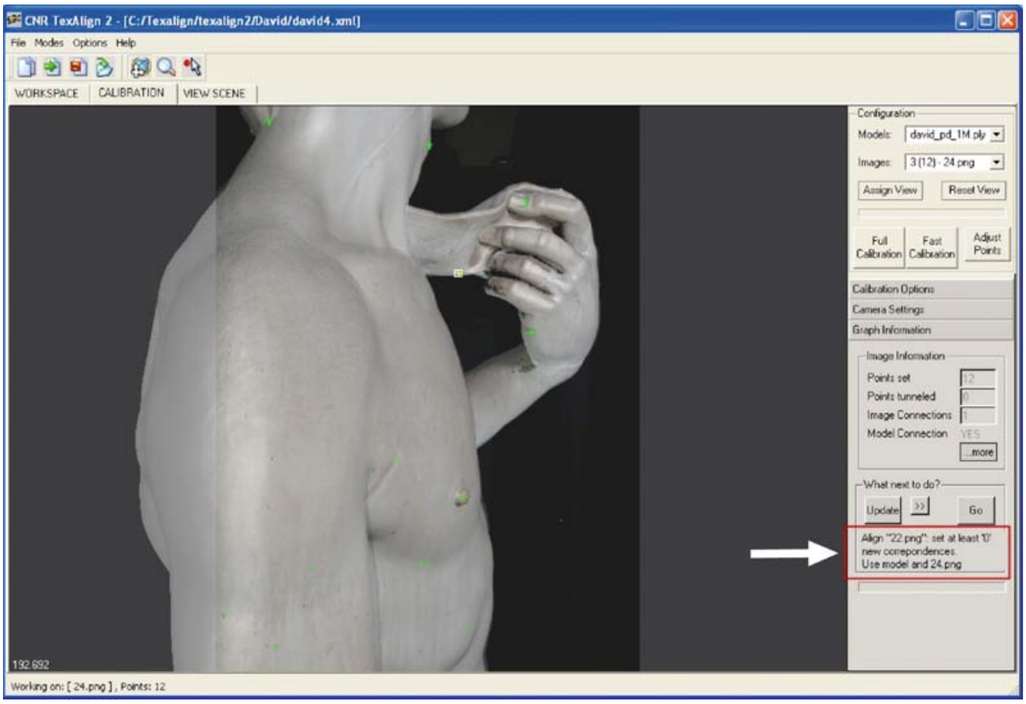

Fig. 6. Calibration space, alignment of an image and use of workload minimizer

using I2G and I2I correspondences. A comparison of results is shown in Table 1. The experienced user had an improvement of nearly 7 minutes in registration time, and, by the end of the registration, the number of explicit correspondences set from images to the 3D model shows a 25% reduction, even if the medium number of correspondences defined for each image is improved, due to the new I2G correspondences inferred from the I2I ones. The unexperienced user had an improvement of nearly 15 minutes (partially due to the improved skill gained while using the

Table 2. Results of second test

	Experienced user		Unexperienced user	
	Without minimizer	With minimizer	Without minimizer	With minimizer
Completion time	∼23 min	∼12 min	∼32 min	∼17 min
Total cost (no. of new correspondences)	22	17	25	18

Fig. 7. Alignment of one of the images over the dish mesh

system), reduced by 13 the number of selected correspondences and also obtained an improvement in the number of total correspondences (explicit and implicit) for each image. This very simple test shows that the use of graph correspondences can be very helpful to the user, reducing time and improving the registration quality.

The second test was performed to analyze the usefulness of the graph-based workload minimizer. The same users as in the first test were given an "intermediate" state of registration on the same *dish* dataset, where some I2G and I2I correspondences were already set, and two out of the eight images were already aligned to the geometry. Users had to complete registration with and without the use of the advice proposed by the workload minimizer. The minimizer estimated a minimum number of 15 new correspondences needed. In Table 2 we present a comparison of results. Without the help of the minimizer, the experienced user took nearly 23 minutes to complete the registration, setting 22 new correspondences. Using the minimizer, the completion time was almost halved, with only 17 new correspondences (for two images the alignment became satisfactory with one connection more than the ones indicated by the minimizer). The second user produced similar results, using approximately one half of the original times. The workload minimizer proves to be very helpful, especially when we can use I2I connections: the user does not have to align images one by one, he/she can set some I2I connections between images, taking advantage of texture features, and then start aligning the less challenging ones selecting I2G correspondences. Then he/she can ask the system to augment the graph, adding implicit I2G arcs, and to suggest the actions to be performed.

7 Conclusions and future work

We have presented a new technique to help the user in minimizing the workload in the registration of images to scanned 3D models and a system implemented according to this new approach. The main idea is to represent all the correspondences between the model and the images using a graph; graph arcs represent both *images-to–geometry* and *images-to–images* correspondences. Using a graph-based approach it is possible to automatically augment the graph, using *images-to–images* correspondences to infer implicit *images-to-geometry* correspondences. It is also possible, by exploring the graph and simulating possible following states, to identify the series of actions that can lead to the end of the registration with a smaller workload. The proposed technique proves to be very helpful when many images (more than 15) have to be registered. Moreover, we have proved empirically that it is much easier to set correspondences between the overlapping parts of images than between images and the 3D model. The technique works best when a subset of possible *images-to–images* connections is set, and some images are already aligned to the geometry. The graph-based approach is incorporated in a user-friendly and interactive registration system. It also provides the possibility of calibrating selected subsets of the extrinsic/intrinsic parameters of the camera used. Although the choices for the search strategies do not guarantee the optimal solution in terms of the number of correspondences, because of pruning of the state space and the use of an heuristic function, in the practical cases observed the choices made by the system where always optimal, so a future direction of work is certainly to adopt more powerful strategies for a more complete exploration of the state space.

Even if the registration turns out to be easy and sufficiently fast, the proposed technique extracts information out of the graph by analyzing only the user's choices: unfortunately, no automatic feature extraction mechanism is provided. As a second future extension, we would like to search for automatic image feature matching solutions that could, for instance, determine in an unattended manner the mutual rough alignment of images or find overlapping images. This automatic feature extraction (e.g. based on edge and shape detection, or on color analysis) could speed up calibration and improve the "shortest path" definition. Testing the application on a bigger number of users and comparing camera parameters obtained with different techniques could be very helpful in analyzing the usefulness of the approach and in inserting new user-friendly features in the alignment process. We are now processing the David dataset (76 high resolution images to be mapped on a very complex geometry which presents many smooth, feature-poor surface regions), which is a very compelling testbed.

Acknowledgement This work was supported by the EU NoE "EPOCH" project. The David model is courtesy of the Digital Michelangelo Project, Stanford University. The color images of the David model are courtesy of Rabatti & Domingie Photographers and Galleria dell'Accademia, Florence, Italy.

References

1. Bernardini, F., Rushmeier, H.E.: The 3D Model Acquisition Pipeline. Computer Graphics Forum 21(2), 149–172 (2002)
2. Callieri, M., Cignoni, P., Ganovelli, F., Impoco, G., Montani, C., Pingi, P., Ponchio, F., Scopigno, R.: Visualization and 3d data processing in David restoration. IEEE Computer Graphics & Applications 24(2), 16–21 (2004)
3. Callieri, M., Cignoni, P., Scopigno, R.: Reconstructing textured meshes from multiple range rgb maps. In: 7th Int.l Fall Workshop on Vision, Modeling, and Visualization 2002, pp. 419–426. IOS Press, Erlangen (D) (2002)
4. Corporation, I.: Open source computer vision library. URL http://www.intel.com/research/mrl/research/opencv/ (2001)
5. Dornaika, F., Garcia, C.: Robust camera calibration using 2d to 3d feature correspondences. In: Proceedings of the International Symposium SPIE – Optical Science Engineering and Instrumentation, Videometrics V, Volume 3174, pp. 123–133 (1997)
6. Faugeras, O., Toscani, G.: The calibration problem for stereo. In: Proceedings CVPR '86, Miami Beach, Florida, pp. 15–20 (1986)
7. Goesele, M.: New acquisition techniques for real objects and light sources in computer graphics. Ph.D. Thesis, Universität Saarbrücken (2004). URL http://scidok.sulb.uni-saarland.de/volltexte/2004/338/
8. Lensch, H., Heidrich, W., Seidel, H.: Automated texture registration and stitching for real world models. In: Proc. 8th Pacific Graphics 2000 Conf. on Computer Graphics and Application, pp. 317–327. IEEE, Los Alamitos, CA (2000)
9. Lensch, H., Kautz, J., Goesele, M., Heidrich, W., Seidel, H.P.: Image-based reconstruction of spatial appearance and geometric detail. ACM Transaction on Graphics 22(2), 234–257 (2003). URL http://visinfo.zib.de/EVlib/Show?EVL-2003-10
10. Lowe, D.: Fitting parameterized three-dimensional models to images. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-13(5), 441–450 (1991)
11. Matsushita, K., Kaneko, T.: Efficient and handy texture mapping on 3D surfaces. In: P. Brunet, R. Scopigno (eds.) Computer Graphics Forum (Eurographics '99), Vol. 18(3), pp. 349–358. Blackwell Publishers (1999)
12. Neugebauer, P., Klein, K.: Texturing 3d models of real worls objects from multiple unregistered photographic views. Computer Graphics Forum (Eurographics'99 Proc.) 18(3), 245–255 (1999)
13. Pulli, K., Cohen, M., Duchamp, T., Hoppe, H., Shapiro, L., Stuetzle, W.: View-based rendering: Visualizing real objects from scanned range and color data. In: J. Dorsey, P. Slusallek (eds.) Eurographics Rendering Workshop 1997, pp. 23–34. Springer Wien (1997). ISBN 3-211-83001-4
14. Russell, S., Norvig, P.: Artificial Intelligence a Modern Approach. AI. Prentice_Hall (1995)
15. Sato, Y., Wheeler, M., Ikeuchi, K.: Object shape and reflectance modeling from observation. In: T. Whitted (ed.) Comp. Graph. Proc., Annual Conf. Series (SIGGRAPH 97), pp. 379–388. ACM SIGGRAPH (1997)
16. Tsai, R.: A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation RA-3(4) (1987)
17. Winkler, S., Wunsch, P., Hirzinger, G.: A feature map approach to real-time 3-D object pose estimation from single 2-D perspective views. In: Proc. 19th DAGM Symposium (Braunschweig, Germany, September 15–17), pp. 129–136 (1997)
18. Zhang, Z.: A flexible new technique for camera calibration (1998). URL http://citeseer.ist.psu.edu/316762.html; http://www.research.microsoft.com/~zhang/Papers/TR98-71.pdf

     

THOMAS FRANKEN was born on 1980-05-06 in Cologne (Germany) as the youngest of six children of an Italian mother and a German father. At the age of 11 he went to a boarding school near the Lake of Konstanz and completedaccomplished high school in 1999. He started to study Information Engineering at the University of Konstanz and absolved his Bachelor in 2004. For six months he worked with the Visual Computing Group at the ISTI-CNR Pisa, focused on texture registration. Currently he is a Master's student at the University of Konstanz and is interested in texture registration and 3D object modelling.

MATTEO DELLEPIANE was born on 1979-01-03 in Genova. He received an Engineering Advanced Degree (Laurea) in 2002 from the University of Genova, with a thesis on image processing. He has worked on the use of 3D scanning technologies and 3D modelling for the reconstruction of ancient monuments as Research Fellow with the university of Genova. Currently he is a Research Fellow at the Istituto Scienza e Tecnologie dell'Informazione (formerly I.E.I.-CNR) of the National Research Council in Pisa, Italy. Among other activities, he is one of the developers of the Visualization and Computer Graphics Library (http://vcg.sf.net), an Open Source library for handling simplicial complexes.

FABIO GANOVELLI graduated in 1995 at the University of Pisa, where he also got his PhD in 2001 with a thesis on deformable objects modelling, where he proposed a multiresolution model for deformable objects supporting cuts and lacerations. He is currently engaged in research projects concerned with Scientific Visualization, Multiresolution, Deformable Object Modelling and 3D Scanning. Among other activities, he is active in organizing scientific events, as co-chairman of VriPhys (http://www.vriphys.org), local co-chairman of Vast05, program co-chairman of Eurographics2005 (short papers).

PAOLO CIGNONI received the Advanced Degree (Laurea) in 1992 and a PhD degree in computer science in 1998 from the University of Pisa. He is a research scientist at the Istituto Scienza e Tecnologie dell'Informazione (formerly I.E.I.-CNR) of the National Research Council in Pisa, Italy. His research interests include computational geometry and its interaction with computer graphics, scientific visualization, the application of 3D scanning technologies for preserving and restoring cultural heritage artifacts, and the exploitation of simplification and multiresolution techniques for efficient management of huge 3D datasets. In 2004 he received the first Eurographics Young Researcher Award.

ROBERTO SCOPIGNO received the Advanced Degree (Laurea) in computer science from the University of Pisa in 1984. He is a senior research scientist at the Istituto Scienza e Tecnologie dell'Informazione (formerly CNUCE – CNR) of the National Research Council in Pisa, Italy. He is currently engaged in research projects concerned with scientific visualization, volume rendering, web-based graphics, multiresolution data modeling and rendering, 3D scanning, and applications of 3D computer graphics to cultural heritage. He is a member of the IEEE Computer Society, Eurographics, and Siggraph. Since 2001 he has been the co-editor-in-chief of Computer Graphics Forum.

CLAUDIO MONTANI received an Advanced Degree (laurea) in computer science from the University of Pisa (Italy) in 1977. He is a research director with the Istituto Scienza e Tecnologie dell'Informazione (formerly I.E.I.-CNR) of the National Research Council in Pisa, Italy. His research interests include data structures and algorithms for volume visualization, surface-based graphics (simplification, multiresolution representation, editing), and automatic acquisition of 3D objects with particular emphasis on the digital representation of 3D cultural heritage. He is a member of the IEEE and of AICA, the Italian Association for Computer Science.