Assisting the user in image to geometry alignment

Tomas Franken, Paolo Cignoni, Fabio Ganovelli, Matteo Dellepiane,
Claudio Montani, Roberto Scopigno
ISTI-CNR, Via Moruzzi 1, 56124 Pisa Italy
www: `http://vcg.isti.cnr.it/`, e-mail: `first.last@isti.cnr.it`

KEY WORDS: 3D scanning, data integration, color sampling and mapping, interactive visualization

ABSTRACT: Producing good quality 3D scanning models which encode color sampling is still a complex task. The acquisition of surface properties, such as the apparent color, is often decoupled from the acquisition of the geometry, leading to data defined in different reference systems. A common situation is to have several pictures of the scanned object with no information about their relative position. The image–to–geometry alignment task can be very time consuming and, except in special cases, is never done automatically. The usual approach demands to the user the selection of correspondences between points in the pictures and points on the object. This paper proposes a novel system to assist the user in the alignment process by computing the minimum set of correspondences which have to be selected. The system is based on a graph representation where the nodes represent the pictures and the 3D object, and arcs encode correspondences. This graph is used to infer new correspondences from the ones specified by the user and from successful alignment of single images.

I. Introduction

The digitalization of real objects, usually referred as 3D scanning, has become one of the most discussed topics in computer graphics. This is due both to the many intrinsic difficulties and technical/theoretical problems involved (see [BR02] for a survey) and to the wide variety of applications of 3D scanning, which include virtual museums, restoration planning and documentation [CCG+04], industrial manufacturing, reverse engineering, etc.

Many applications require to sample not just the geometry but also the color information, for example when digitalizing piece of arts for building virtual museums. Acquiring the real color of an object, i.e. its surface reflection properties, is a complicated and time consuming task [LKG+03], [Goe04] 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 controllable to obtain good quality results). However, for most practical cases a series of pictures taken by a digital camera are simply stitched onto the surface of the object, trying to avoid shadows and highlights and taking pictures with favorable light conditions. However, even in this simplest case, the pictures need to be processed in order to build a plausible texture for the object[CCS02].

When managing color information, a fundamental problem is how to register the images with the geometric data. In some cases the problem is 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[PCD+97], [SWI97]. Unfortunately, in many cases the image are unregistered since a more simple setup is used (hand-held camera, color acquired in a second stage). There are two main reasons for that: the 3D scanner may be working in light conditions which are not optimal for cameras and viceversa or, most important, the pictures may have been taken by a professional photographer independently from the 3D scanning campaign.

A number of papers addressed the problem of registering 2D pictures to 3D geometry and a brief survey of state of the art approaches is presented in SectionII. We can still say that there is no fully automatic approach to register 2D images in the general case (large and complex object, with images depicting only a subset of its overall extent) . The user is usually required to provide correspondences, or hints on the correspondences, between the 2D images and 3D geometry and/or between two or more 2D images.

The main contribution of this work is a technique to minimize the user intervention in the process of registering a set of images with a 3D model. The main idea is to setup a *graph of correspondences*, where the 3D model and all the images are represented as nodes and two nodes are connected if a correspondence between them exists. The graph of correspondences is then used to automatically infer new correspondences and to find which is the shortest path, in term of the number of correspondences that must be provided by the user, to complete the registration of all the images. The technique is incorporated in an application which is also described in detail. The paper proceeds as follows: Section II briefly presents the method more closely related to our problem: Section III describes the method implemented to align a single image; Section IV describes the correspondence graph and its use to infer new correspondences; Section V show how the correspondence graph is used to minimize the user workload. Section VI shows a case study to evaluate the benefit from the proposed technique and finally conclusions and future work are reported in Section VII

II. Related work

Camera parameters estimation involves the computation of intrinsic parameters, i.e the ccd sizes, the focal length, the optical center and radial distortion introduces by the lens, and the extrinsic parameters, i.e. position and orientation of the camera in the global reference system. *Intrinsic parameters* can be estimated by providing 3D-2D correspondences or taking a picture of a known calibration pattern on a planar geometry [Tsa87], [Zha98]. In this second case the pattern can be detected automatically. A number of publicly available libraries for camera calibration are available (for example [Cor01]).

To find the *extrinsic parameters*, i.e. the view specification associated to a given picture, if often done by providing 3D-2D correspondences and minimizing an error function which usually is the sum of the differences between each 3D point projected onto the screen and its 2D corresponding 2D feature. To avoid the tedious work to provide correspondences,landmarks can be placed onto the real object and detected automatically. When each image covers the entire object, the silhouette of the model and the 2D contour of the object can be used as matching features. This is done by minimizing the difference between the projection of the object and its image in the picture, which requires a 2D segmentation usually easy to achieve automatically (or anyway, with little user intervention) [NK99], [Low91], [MK99], [LHS00], [WWH97].

III. Aligning the 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 offer an optimal overlapping of the image and the model, with respect to the "world" reference system. As previously mentioned, camera parameters can be divided in two main groups:

- *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 internal geometry and the optical characteristics of the camera.

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

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

where R is the 3*3 rotation matrix and T is the translation vector. These are the parameters 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 intrinsic parameter *f*, which is the focal length:

$$X_u = f\frac{x}{z} \quad (2)$$

$$Y_u = f\frac{y}{z} \quad (3)$$

Another intrinsic parameter which can be considered is radial lens distortion: we can calculate undistorted image coordinates

$$X_d + D_x = X_u \quad , \quad Y_d + D_y = Y_u \quad (4)$$

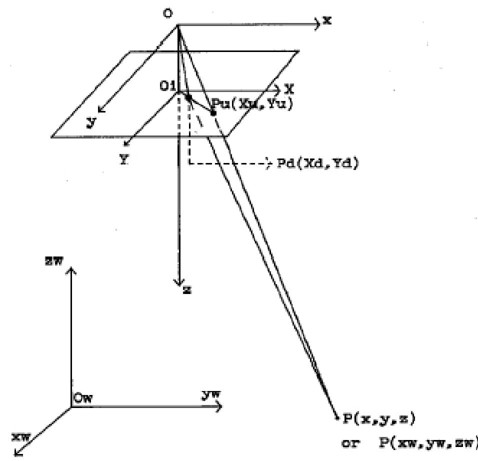

Fig. 1. Camera geometry with radial lens distortion

where

$$D_x = X_d(k_1 r^2 + k_2 r^4 + ...) \quad , \quad D_y = Y_d(k_1 r^2 + k_2 r^4 + ...) \quad (5)$$

and

$$r = \sqrt{X_d^2 + Y_d^2} \quad (6)$$

so the parameters to be calibrated are k_i.

The user-driven setup of a few correspondences between points in the image and on the 3D mesh is the standard approach to calculate all these parameters. As a matter of fact, the goal is to find the parameter values which 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). In our approach two kind of calibrations can be performed by the user: the first one, based on Tsai approach [Tsa87], needs at least 12 point correspondences, and it is able to optimize all extrinsic and intrinsic parameters. The second kind of calibration performs a non-linear method [DG97] derived from Faugeras and Toscani approach [FT86], which needs at least 4 correspondences and performs optimization on extrinsic parameters and focal length value. Another useful option is the possibility for to user to optimize only one (or more) of the variables, i.e. rotation, translation, focal length or lens distortion. This can be useful, for instance, if we are sure about the obtained optimization of 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.

IV. Inferring correspondences

The registration of a single 2D image to a 3D geometry can be performed using a number of linear and non-linear techniques, and the difficulty of operation is usually determined by the need to find a sufficient number of *useful correspondences*. Moreover, registering 15-20 images to a single 3D geometry can be very time consuming and hard to manage, especially when the geometry presents large flat areas with insufficient shape 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. An helpful solution could be the possibility to utilize correspondences between single images. The overlapping areas of 2D images, due to color changes and texture detail, could be much more useful to infer new correspondences than the corresponding section of the 3D model shape.

Our technique for graph-based setup of correspondences (addressing both *image-to-geometry* and *image-to-image* correspondences individuation) 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 use the information encoded in the graph (i.e. the links which connect single images) and a graph traversal approach to detect the "shortest path", i.e. the minimum number of new correspondences which have to be instantiated to complete the registration.

The structure of graph is quite simple: the 3D model and each image are represented by nodes; if a correspondence between any of the nodes exists, the nodes are connected by an arc. We show a very simple correspondence graph in the example in Figure 2: IMAGE1 is connected with the 3D mesh with three correspondences (i.e. three corresponding point pairs have been selected); IMAGE2 has four correspondences; moreover, IMAGE1 and IMAGE2 share a correspondence pair (arc *a*) in the mesh and are connected by a correspondence (arc *g*) which has no connection with the mesh, i.e. is a pair between points in the images (a *image-to-image* correspondence).

The graph encodes all the connections between images and geometry (either of type *image-to-geometry* or *image-to-image*). Let us introduce what do we mean with inferring new correspondences.

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* ($I2I$) correspondence pairs, since we may infer an implicit *image-to-geometry* ($I2G$) 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 Figure 2, which shows what automatically happens when IMAGE2 is aligned to the mesh: the point on IMAGE2 associated to correspondence g is projected on the model, creating at the same time an "indirect" correspondence g (represented by a bold broken 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 mesh, 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 *image-to-image* and *image-to-geometry*, approach the user can set some correspondences with other images that present overlaps with the one covering a region with insufficient shape features. As soon as the connected images are aligned in an accurate manner to the geometry, new "indirect" correspondences are created for the more "challenging" images, helping the user to complete the alignment.

V. Minimization of user workload

The graph is *complete* w.r.t the alignment task when an arc (either explicit or inferred) has been defined for each model-image pair. The possibility of automatically inferring new correspondences between images and the 3D model implies that, given an intermediate completion status of the graph, there are different "paths" which could be added to complete the graph and to find a solution to the registration problem. Each path has a different cost in terms of number of necessary point-to-point correspondences which are needed to align the corresponding nodes. Managing the registration of many images can be quite challenging for the user, since a complex graph has to be initialized. An handy alignment system should give advices to the user on which images and how many point pair correspondences should be added in order to minimize the cost of registration and to produce a valid complete alignment. In order to do this it's necessary to explore all the possible paths who lead to the end of alignments. Since for a high number of images the exploration of state graph could be time consuming, a pruning operation is necessary to give an answer to user in a short time.

We show a simple example to demonstrate that different costs can be required for different registrations of the same set of images (Figure 3); in this case, we start from the hypothesis that at least five correspondences are needed to align an image to the 3D model (single *image-to-geometry* alignment). In figure 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 Figure 3, with a cost of 1 since we added just 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 2 more correspondence (lines q and r), we may also create 4 new connections (dashed arc i,l,m,n) between IMAGE1 and the 3D model (right-most graph on second line, Figure 3). At this point, IMAGE1 can be aligned without any new intervention by user and the total cost to complete registration is 3. But if we decide, going back to the starting graph layout, 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 Figure 3) IMAGE1 earns 4 new inferred connections to 3D model. At this point, IMAGE1 can be aligned without any added cost (Figure 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 (two new arcs).

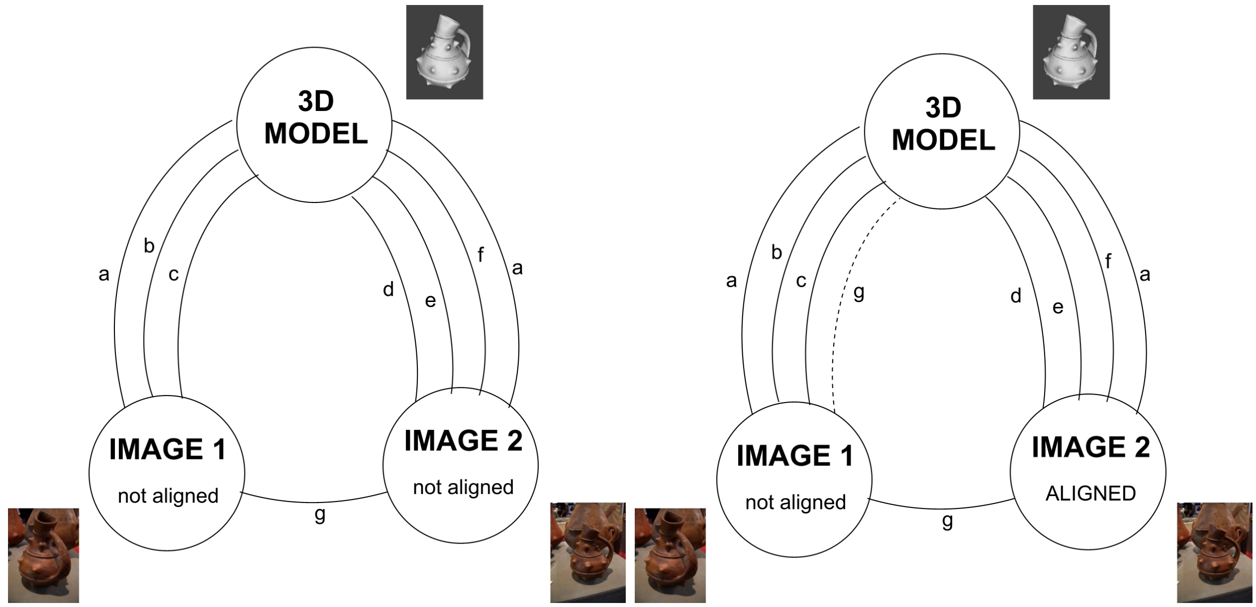

Fig. 2. Example of correspondences graph (on the left). A new *image-to-geometry* correspondence (g) for IMAGE1 is inferred automatically given an *image-to-image* correspondence which links IMAGE1 with IMAGE2 (graph on the right).

As this simple example shows, only with a smart exploration of all the possible future states we can foresee the path which leads to the completion of the registration with the lowest cost. In our approach we try to predict the evolution of the graph, so we use a simplified version of a correspondences graph, called Stategraph, which contains only some useful information: total cost of registration, and, for each non aligned node, the number of correspondences to 3D model and to the other non-aligned images. When we simulate the alignment of an image, Stategraph is "updated". The final Stategraph is the one with all nodes aligned, and a final registration cost. The algorithm should follow every possible path until it reaches a graph status corresponding to complete registration. All possible paths should be tested in order to discover the shortest one (that is the one with lowest registration cost) . As for the example in Figure 3, the application would suggest the user to align IMAGE2 as a first step, and then to consider IMAGE3 and IMAGE1.

Unfortunately, if the user is managing a quite big number of images (e.g. 20 or more) the graph of possible future states can be quite complicate, so that a complete exploration of all possible future states results in an excessive time overhead. To avoid this exhaustive search, we need a strategy of Stategraph pruning. Since, at the moment, no particular indication can be extracted by the 3D model and images features, pruning strategy must take account of user previous behavior in order to infer what images can be considered more strictly related between each other. Analyzing a state, the pruning strategy must select a certain number (three in our algorithm) of non-aligned images. These images are selected considering two main 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 image would produce. When a non-aligned image is selected, a simulation of its alignment produces a new Stategraph, where all connections of the image with other non-aligned images are transformed in inferred correspondences. The total cost of registration is increased by the minimum number of new correspondences needed to align the selected image. Moreover, the algorithm suggests to the user one or more nodes that could be useful to complete the alignment of this image: one of them is always the 3D model, but if the algorithm finds an aligned image which already has some connections with the selected image, it argues that the image can be well overlapped with the selected one, and it indicates it as another good node to set new correspondences to. For each Stategraph which is calculated, three new possible Stategraphs are produced, until all the paths are completed. The path with the lowest cost is the one the algorithm will indicate to user. Exploration of paths to registration is useful to user only when the algorithm can work on a set of data where some correspondences between similar images are already set, and where some "simpler-to-align" images are already aligned with the 3D model. This is because the algorithm cannot infer anything about images quality and percentage of overlapping. Path exploration is very useful when the user has set a good number of connections, and he needs some advice to complete registration quickly. For example, sometimes the user could not notice that an image, due to inferred connections, does not need other correspondences to be aligned. Advice mechanism will let him know about it.

VI. Results and Discussion

We present in this section a simple concrete example where graph correspondence helps the user in aligning an image. We suppose to have two images to be aligned over the digital model of Michelangelo's David. Images are shown in

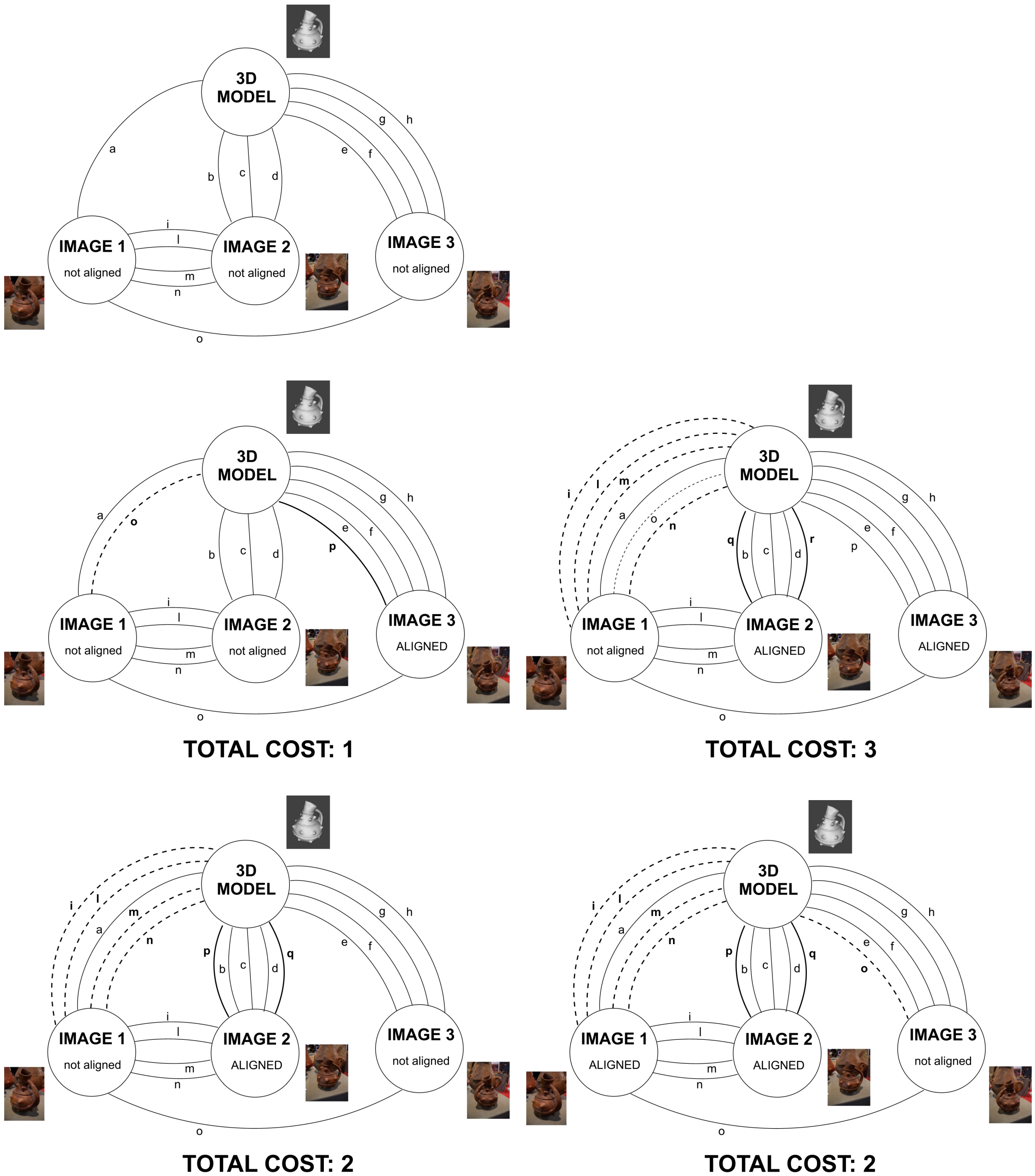

Fig. 3. Different workloads in registration completing. Top: starting state. Middle: registration aligning Image 3 for first, then Image 2. Bottom: registration aligning Image 2 for first, then Image 1.

Figure 4: the first one encodes visually some shape features that can be used to set correspondences with the 3D model (hand, chest, hair...). The second image depicts a partially overlapped view of the statue where there are almost no relevant surface features. Nevertheless, in Figure 5 we see how lots of correspondences can be set in the overlapping area of the images, due to the presence of a lot of small holes and brown spots which are well evident in the images. The screenshot presented in Figure 5 shows the structure of our application: in the *Workspace* mode the thumbnail of all loaded images are listed in the lower part of the window, and any pair can be loaded to set new *image-to-image* connections (marked as green crosses in the images, or as solid points on the 3D model). Figure 5 shows how matching texture features can be used to set some connections between images. The *Calibration* mode, shown in Figure 6, shows the result of the alignment of the first image with the 3D model, set by instantiating 7 pixel-to-3Dpoint correspondences. Once the first image is aligned, we can use the connections between images 1 and 2 to align the second one on the 3D model (see Figure 7). If we ask, after the registration of the first image, the workload minimizer for an advice (see Figure 8), it will suggest to align image 2, pointing out that no more correspondences are needed: 3D Model and Image 1 seem to be the most suitable to set new correspondences, if the user wants to refine calibration (see red rectangle). After

Fig. 4. An example of two partially overlapping images (Michelangelo's David).

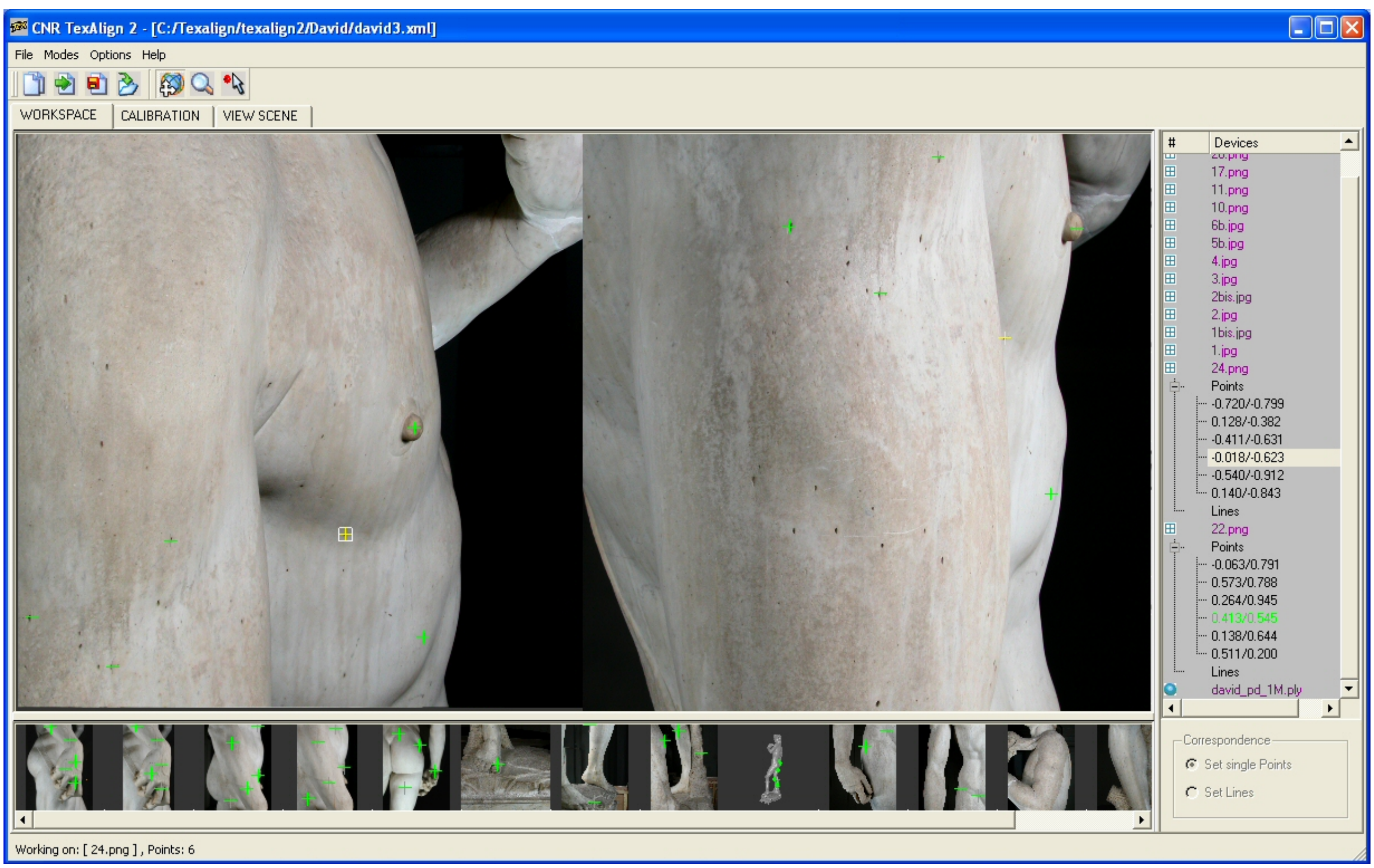

Fig. 5. Setting correspondences between two images, based on *image-to-image* correspondences based on image features (small dark spots corresponding to small holes on the David's marble).

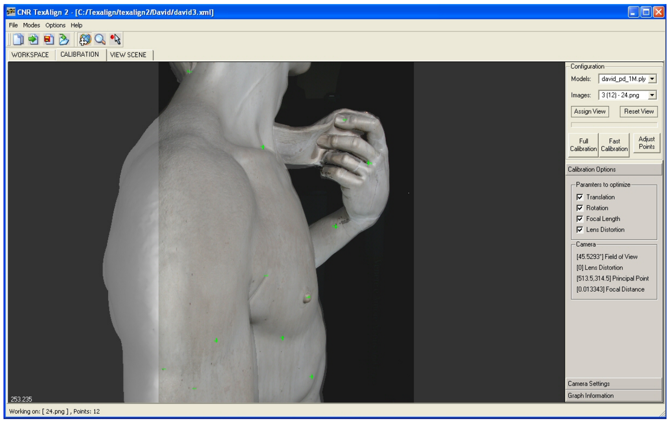

Fig. 6. Alignment of Image 1 with seven *image-to-geometry* correspondences.

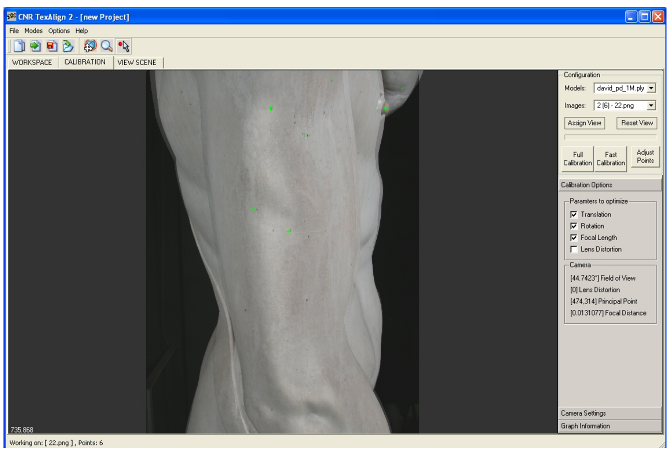

Fig. 7. Alignment of Image 2 using inferred correspondences.

being registered, texture features of image 2 can be useful for other images alignment. This simple example shows how the proposed approach goes beyond the concept of single *image-to-geometry* alignment: it introduces the possibility to use the connections between images, which in some cases are much more stronger or easy to detect visually than the ones between the image and the 3D model. Moreover, since the possibility to set connections between all nodes can become confusing when the user tries to align many images, the workload minimizer helps the user to get a complete registration with lower effort.

VII. Conclusion and Future Work

A technique to help user in minimizing the workload in the registration of images to scanned 3D models has been presented in this paper. The main idea is the representation of all the connections between the model and the images using a graph representation. Using a graph-based approach it is possible to create new links between the images and the 3D model, using *images-to-images* correspondences between images to infer implicit *images-to-geometry* correspondences. It's also possible, exploring the graph and simulating possible following states, to identify the series of actions which can lead to the end of the registration with the lowest work load. The proposed technique proves to be very helpful when many images (¿16) have to be registered; moreover, we 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 good number of connections is set, and some images are already aligned to the geometry. The graph-based approach is incorporated in a user-friendly application, which helps user in managing all images. It also gives the possibility to calibrate only some (extrinsic or intrinsic) parameters of the camera used.

Even if the registration turns out to be easy and fast, the proposed technique extracts information out of the graph by analyzing only user's choices: unfortunately, no automatic feature extraction mechanism is provided. As a future extension, we would like to search for automatic image feature matching solutions which could, for instance, determine 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 to speed up calibration and improve ”shortest path” definition.

Aknowledgements. 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 are courtesy of *Rabatti & Domingie Photographers* and Gallerie dell'Accademia, Florence, Italy.

References

[BR02] F. Bernardini and H. E. Rushmeier. The 3D Model Acquisition Pipeline. *Computer Graphics Forum*, 21(2):149–172, March 2002.

[CCG+04] M. Callieri, P. Cignoni, F. Ganovelli, G. Impoco, C. Montani, P. Pingi, F. Ponchio, and R. Scopigno. Visualization and 3d data processing in David restoration. *IEEE Computer Graphics & Applications*, 24(2):16–21, Mar.-Apr. 2004.

[CCS02] M. Callieri, P. Cignoni, and R. Scopigno. Reconstructing textured meshes from multiple range rgb maps. In *7th Int.l Fall Workshop on Vision, Modeling, and Visualization 2002*, pages 419–426, Erlangen (D), Nov. 20 - 22 2002. IOS Press.

[Cor01] Intel Corporation. Open source computer vision library. http://www.intel.com/research/mrl/research/opencv/, 2001.

[DG97] Fadi Dornaika and Christophe Garcia. 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*, pages 123–133, 1997.

[FT86] Olivier D. Faugeras and Giorgio Toscani. The calibration problem for stereo. In *Proceedings CVPR '86, Miami Beach, Florida*, pages 15–20, 1986.

[Goe04] Michael Goesele. *New acquisition techniques for real objects and light sources in computer graphics*. PhD thesis, Universität Saarbrücken, 2004.

[LHS00] H.P. Lensch, W. Heidrich, and H.P. Seidel. Automated texture registration and stitching for real world models. In *Proc. 8th Pacific Graphics 2000 Conf. on Computer Graphics and Application*, pages 317–327, Los Alamitos, CA, 2000. IEEE.

[LKG+03] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transaction on Graphics*, 22(2):234–257, April 2003.

[Low91] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(5):441–450, May 1991.

[MK99] Kenji Matsushita and Toyohisa Kaneko. Efficient and handy texture mapping on 3D surfaces. In P. Brunet and R. Scopigno, editors, *Computer Graphics Forum (Eurographics '99)*, volume

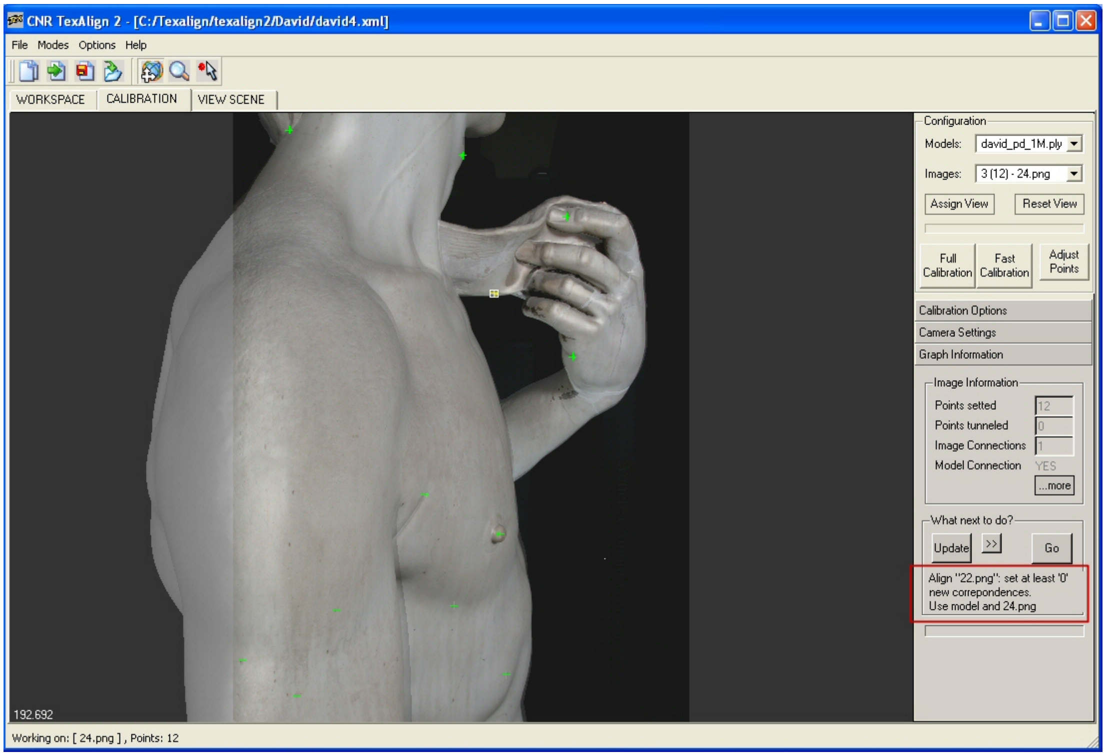

Fig. 8. Use of the workload minimizer: see a suggestion produced in the framed area (red rectangle).

18(3), pages 349–358. The Eurographics Association and Blackwell Publishers, 1999.

[NK99] P.J. Neugebauer and K. Klein. Texturing 3d models of real worls objects from multiple unregistered photographic views. *Computer Graphics Forum (Eurographics'99 Proc.)*, 18(3):245–255, 1999.

[PCD+97] K. Pulli, M. Cohen, T. Duchamp, H.Hoppe, L. Shapiro, and W. Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 23–34. Springer Wien, June 1997. ISBN 3-211-83001-4.

[SWI97] Y. Sato, M.D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In Turner Whitted, editor, *Comp. Graph. Proc., Annual Conf. Series (SIGGRAPH 97)*, pages 379–388. ACM SIGGRAPH, Aug. 1997.

[Tsa87] R. Tsai. 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), August 1987.

[WWH97] S. Winkler, P. Wunsch, and G. Hirzinger. 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)*, pages 129–136, 1997.

[Zha98] Zhengyou Zhang. A flexible new technique for camera calibration, April 16 1998.