Flow-Based Local Optimization for Image-to-Geometry Projection

Matteo Dellepiane, Ricardo Marroquim, Marco Callieri, Paolo Cignoni, and Roberto Scopigno

Abstract—The projection of a photographic data set on a 3D model is a robust and widely applicable way to acquire appearance information of an object. The first step of this procedure is the alignment of the images on the 3D model. While any reconstruction pipeline aims at avoiding misregistration by improving camera calibrations and geometry, in practice a perfect alignment cannot always be reached. Depending on the way multiple camera images are fused on the object surface, remaining misregistrations show up either as ghosting or as discontinuities at transitions from one camera view to another. In this paper we propose a method, based on the computation of Optical Flow between overlapping images, to correct the local misalignment by determining the necessary displacement. The goal is to correct the symptoms of misregistration, instead of searching for a globally consistent mapping, which might not exist. The method scales up well with the size of the data set (both photographic and geometric) and is quite independent of the characteristics of the 3D model (topology cleanliness, parametrization, density). The method is robust and can handle real world cases that have different characteristics: low level geometric details and images that lack enough features for global optimization or manual methods. It can be applied to different mapping strategies, such as texture or per-vertex attribute encoding.

Index Terms—Computer graphics, image color analysis.

1 Introduction

The acquisition of the appearance properties of real objects is a broad and complex field of research in the Computer Graphics and Computer Vision context. The main goal is to produce extremely accurate and realistic 3D models, encompassing both shape and color. While the acquisition of geometry has been greatly improved over the last few years, the acquisition and visualization of surface appearance properties has not yet reached such a mature level. Even if methods to treat small objects, specific materials and simple BRDF models have been proposed, no general approach can handle complex geometries with varying reflectance properties, especially when flexibility, portability on the field, and robustness is needed.

When confronting these situations, a popular, simple, and robust alternative is adding to the 3D shape a color information obtained by the mapping and back-projection of a set of photographs. Relying on an additional photographic data set is needed because, in most cases, the color information acquired by the acquisition devices is not accurate enough. These approaches generally start by computing an *image-to-geometry registration*, followed by an *integration strategy*. The first phase deals with the computation of the camera's intrinsic and extrinsic parameters, i.e., the information needed to compute the inverse projective transformation of the photos. On the other hand, the integration phase manages how to combine the many color samples available for each surface point. Most methods use some sort of weighting or interpolation of the samples to produce the final color.

Unfortunately, computing 100 percent correct *image-to-geometry registration* is never possible in the real world, due to several factors, such as: the specific geometrical features of the 3D model (the surface might lack representative geometric features to be matched with visual features in the image); the 3D model may not be sufficiently accurate due to poor scanning, or excessive simplification/smoothing introduced in the postprocessing phase; and/or the image may be distorted, cropped or there may be shadows/highlights which interfere with the image-to-geometry registration. Consequently, global optimization methods are unable accurately align all features simultaneously, leading to blurry details or the so-called "ghosting" effect (see Fig. 1) after color projection. This kind of artifact stands out significantly in the case of small and sharp color features, especially when the color detail is in areas with low geometric information (i.e., painted vases or nearly flat surfaces). Cultural Heritage is a context where many of these complex cases arise, since the artworks are usually decorated, and have densely textured details (grain of the material, deteriorations, scratches, or loss of material/paint, etc.). The problem is made more pressing by the requirements of the end users: if the 3D medium is going to be the main form of representation, we need to be able to produce 3D models with the same quality as digital photos.

This paper proposes a method for correcting small inaccuracies introduced in the image-to-geometry registration. Rather than solving the problem in a global manner by computing a new, congruent global registration over all images (which is sometimes impossible), our solution is based on a local warping of the source images aimed at improving the sharpness of the mapping, in a similar fashion to methods proposed for range maps registration [1]. This local intervention can correct small misalignment artifacts while, at the same time, preserves the initial

- *M. Dellepiane, M. Callieri, P. Cignoni, and R. Scopigno are with the Visual Computing Lab, ISTI-CNR, Via G. Moruzzi 1, Pisa 56124, Italy. E-mail: {Dellepiane, Callieri, Cignoni, Scopigno}@isti.cnr.it.*
- *R. Marroquim is with the Department of Systems Engineering and Computer Science, Universidade Federal do Rio de Janeiro, Cidade Universitária, Centro de Tecnologia, Bloco H, Sala 319, Rio de Janeiro, RJ, Brasil. E-mail: marroquim@cos.ufrj.br.*

Manuscript received 2 Aug. 2010; revised 9 Feb. 2011; accepted 14 Feb. 2011; published online 12 Apr. 2011.
Recommended for acceptance by K. Bala.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-08-0169.
Digital Object Identifier no. 10.1109/TVCG.2011.75.

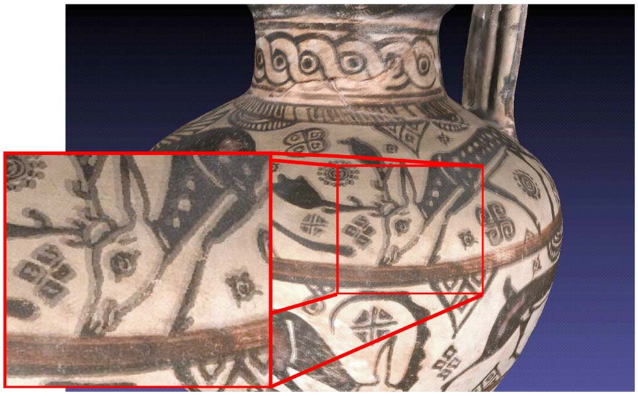


Fig. 1. An example of "ghosting" effect due to small inaccuracies in image registration.

registration at a higher level. An optical flow strategy is used to drive the warping of the images before projection, in order to produce an accurate and unambiguous mapping. Even though optical flow is a well-known technique, it has been mostly applied in the fields of motion tracking and reconstruction from stereo, where little scene variation between consecutive images is assumed, i.e., small displacements of the camera or the objects. In contrast, in this study it is used in a different scenario, where the input images are obtained from a set of cameras with significant position and orientation variation. Our correction algorithm also handles cases where several images map onto the same geometrical point (i.e., where continuous warping among several images is needed).

The main contributions of our method are:

- robustness in terms of correcting misalignments and preserving sharpness of small-scale details in colored 3D models, able to manage applications requiring a large number of images and very dense models;
- easy integration with state-of-the-art methods based on color projection and integration;
- the application of optical flow approaches to accurately drive the image-to-geometry projection over a set of images taken from a very sparse location, complied with specific input conditions that usually not accounted for in the optical flow literature;
- a method to combine multiple warping fields which exploits "importance" relationship between input to images, able to build a globally coherent color information.

2 Related Work

Our method deals with two important fields of research: color acquisition and projection, and optical flow. A brief overview of the most important studies and of the main issues will be presented. Additional references to other related fields of research can be found in other sections of the paper.

2.1 Optical Flow

The optical flow between two images is usually computed as a set of displacement vectors between corresponding pixels. One of its most prominent applications is motion tracking, where the images define a sequence in time and the flow field represents the *image velocity*. It has also been used in other areas such as image-to-image registration, template matching, video compression, and image morphing. In fact, whenever a pixel correspondence between images must be retrieved, optical flow methods may be applied.

Even though it has been a very active area of research over the last few decades, there is still no single optimal solution, since each problem is based on a different set of assumptions and requirements. Some important problems tackled by researchers have included the management of large deformations, rotations, illumination changes, and large displacements. In each case, some variables are restricted in order to bound the complexity of the algorithm to a practical scale. For example, many algorithms are based on the assumption of a global smooth field or constant pixel brightness between *images*: the *data conservation constraint* [2] assumes that the intensity is conserved for small time steps.

There are many approaches to compute the optical flow, and what type works best depends mostly on the application. In the context of image-to-geometry alignment, optical flow strategies were used in some specific settings, such as capturing real people under controlled illumination with a video camera [3], [4]. In another work Pulli et al. [5] employed optical flow to align range images with color information using a minimization approach to calibrate the camera parameters for each range map. Differently from many applications, a global matching solution is not required in our settings and, even more, smoothing should be avoided as to not cause blurring in the final model. Even though the contribution of this work is not a novel optical flow method by itself, in Section 3.1 we further discuss some strategies that handle well our test cases.

2.2 The Camera Calibration Problem

The first step in the color projection pipeline is image registration, since in most cases the camera parameters associated with each image are not known in advance. Several automatic [6], [7], [8], [9] and semiautomatic [10] methods for image-to-geometry registration have been proposed. They are mainly based on an analysis of the geometric features of the model (e.g., silhouette and orthogonality), or on some input given by the user (2D-to-3D correspondences). Although these solutions can be extremely accurate, the quality of the results is influenced by several factors:

- **The geometric properties of the model.** If the model lacks peculiar geometric features, both the automatic and semiautomatic methods might not have enough data to provide an accurate solution.
- **The quality of the 3D model.** If the 3D model has been generated with an insufficiently accurate method (e.g., low cost scanners, manual modeling, or photogrammetry), the geometry may be different from that depicted in the images, so that a precise estimation of the camera parameters becomes almost impossible.
- **The use of a camera model.** All the above cited methods attempt to fit a perspective camera model. However, this classic camera model does not always fit the real case perfectly: this can happen for some particular cameras, or when images have undergone some kind of processing, such as cropping.

In these cases, it is often not possible to produce perfect results, which leads to ghosting effects and discontinuities.

2.3 Color Acquisition and Visualization

The most correct way to represent the material properties of an object is to describe them through a reflection function (i.e., BRDF), which attempts to model the observed scattering behavior of a class of real surfaces. A detailed presentation of its theory and applications can be found in Dorsey et al. [11]. Unfortunately, state-of-the-art BRDF calculation approaches rely on controlled and complex illumination setups [12], [13], [14]: this limits their application in the context of complex scanning projects (big artifacts and on-the-field acquisitions, such as those performed in museums). Other approaches, based on simplified assumptions [15], [16] are also difficult to apply in more general cases.

A less accurate, but more robust, solution is the direct use of images to transfer the color to 3D surfaces. In these cases, the apparent color value, as sampled in digital photos, is mapped on the digital object's surface by registering these photos w.r.t. the 3D model (by estimating the camera parameters), and then applying an inverse projection. In addition to other important issues (briefly cited in other sections of the paper), such as the image registration and how to store color information, there are numerous difficulties in selecting the correct color when multiple candidates are in different images. Some of these are: how to deal with discontinuities caused by color differences between photos that cover adjacent areas, and how to reduce illumination-related artifacts, i.e., shadows, highlights, and peculiar BRDF effects. This is also true when it's possible to use the color acquired directly by 3D scanners [17].

To solve these problems, one group of methods selects, for each part of the surface, a portion of a representative image following a specific criterion (in most cases, the orthogonality between the surface and the view direction [18], [19], [8]). However, artifacts caused by the lack of consistency between overlapping images are visible on the borders between surface areas that receive color from different images. These can be partially removed by working on the border between two images [18], [19], [8]. Also aiming to solve this issue, Chuang et al. [20] employed color gradients to seamless reconstruct colored surfaces from scans.

Another group "blends" the contribution of all the images by assigning a weight to each one or to each input pixel (this value expresses the "quality" of its contribution), and selecting the final surface color as the weighted average of the input data, as in Pulli et al. [21]. The weight is usually a combination of various quality metrics [17], [22], [23]. In particular, Callieri et al. [24] presented a flexible weighting system that can be extended in order to accommodate additional metrics. These methods provide better visual results, and their implementation permits very complex data sets to be used, i.e., hundreds of images and very dense 3D models. Nevertheless, the blending approach produces undesirable ghosting effects when the starting set of calibrated images is not perfectly aligned. In the case of the first group of methods, this kind of artifact is visible only along the frontier of regions mapped to different images.

Another issue, which is common to all the cited methods, is the projection of lighting artifacts on the model, since the lighting environment is usually not known in advance. In order to correct (or not project) the lighting artifacts, two possible approaches include the estimation of the lighting environment [25], [26], and the use of easily controllable lighting setups [27].

2.4 Image Warping in Color Mapping

Warping input images, in order to obtain a better color projection from a set of photos on a 3D model, is an idea discussed in some recent papers. However, in the literature, most of the works are focused toward 3D models built from multiview stereo matching and structure-from-motion approaches. Hence, the data to be managed are composed of a quite simple geometry and the same photographic data set used to generate it. Our aim is to deal with the more general case with an arbitrary 3D model and photographic data set.

In Harmonised Texture Mapping [28], Takai et al. first modify the geometry according to the texture inconsistency, obtaining a simpler triangulation where it is easier to define warp fields to reduce image misalignments. Then, using view-dependent texture mapping, they combine the obtained warp fields for the current viewpoint according to the view angle. The results are nice, but a globally valid color mapping is not produced.

Aganj et al. [29] use, as a starting point for the image warping, the same set of points used for the stereo matching. The matched points are reprojected onto the surface, each defining a warp direction on the images. These directions are then combined in each image using a thin-plate spline to minimize warp discordance. Input images are warped statically using the resulting field.

Gal et al. [30] also consider geometries obtained by multiview stereo matching and work at triangle level, employing a multilevel optimization strategy to integrate the different warp directions. Each triangle is then textured using a single photo, smoothing a very small area of border between different photos. Since a smooth warp field is not available, it is not possible to exploit the redundancy in the overlapping images, which could have been used to better smooth the transition. Moreover, working at triangle level requires a small, topologically clean geometry, which is not the case in the more general scenario.

The Floating Textures system [31] uses optical flow to obtain warp data between images, and then combining the warp fields linearly in the space of the current viewpoint. In this way it is possible to work in realtime. Again, the linear, view-dependent combination of warp fields prevent the creation of a globally valid texture. Besides the different nature of data set, our approach proposes a more elaborate, but still easily implementable, method to combine the discordant warp fields produced by the optical flow, based on the local importance of the input images. Using this method, it is possible to combine the different warp fields such that the resulting color mapping is globally coherent, overcoming the limitation of a single viewpoint.

3 The Correction Algorithm

The goal of our system is to fulfill three main requirements:

- **Generality.** The local misalignment problem can be approached in different ways, but very few guarantee wide applicability. For example, global camera optimization methods may not work in the case of low quality data sets (as detailed in previous section), and local per-triangle warping [18] can only be used on small meshes with a good surface parameterization. We need a method which is independent of the properties of the 3D mesh, i.e., mesh resolution or geometry quality/accuracy, and which can work with camera calibrations provided by different tools.

- **Scalability.** Current practical 3D geometry and color acquisition projects produce very dense models and hundreds of images. We believe it is possible to implement the color data processing in an out-of-core fashion, in order to deal with very large amounts of raw data.
- **Automation.** Human intervention, such as fine tuning of camera calibration or local blending correction (or even local texture manipulation via image editing tools), is only possible in very simple cases. The correction system must not require user intervention.

The basic idea of the algorithm is to locally warp input images in order to minimize small-scale misalignment of high frequency color features, thus obtaining a sharper color mapping. The proposed solution has two components: 1) a method to determine the image warps necessary to obtain a coincident 2D-to-3D color projection and 2) a strategy to combine the resulting image warps to obtain a coherent warping to be used in color mapping.

Instead of calculating the warp directions on a limited set of points, we opted to use a more dense search method, based on Optical Flow. Since the relative size of photographic misalignment may be very small, working at a higher level may not be enough to cope with fine details. Moreover, we believe that, by selecting the correct flow calculation it is possible to obtain a warp field which is globally smooth enough to produce continuous result but also locally sharp enough to precisely correct higher frequency color features.

Each overlapping pair of photos, however, result in a warping field which is generally not coherent with the other fields; in order to generate a color mapping, it is necessary to combine multiple warp fields. Our aim is to produce a globally coherent color mapping. Hence, we need a way to combine the warps all over the geometry, without relying on view-dependent texture. Instead of trivially combining the warp fields using linear interpolation or using numerical methods to minimize the combination error, we choose to locally weight the different fields using quality metrics. By evaluating the local quality of the photos, it is possible to determine which image is more representative for a certain area of the mesh, and to locally warp the other contributions in order to be coherent with the dominant one. The smoothness of the transition between image dominance derived by the weight scheme ensures a continuous warp interpolation. Additionally, since the warp combination is continuous, we can still fully exploit data redundancy between the photos, effectively blending overlapping areas to minimize inconsistency.

The color mapping is then carried out considering, in each point, the local importance of the contributing input images, and interpolating the warp fields according to their relative quality. The result is a global, view-independent color mapping which preserves high frequency features and color continuity.

3.1 Computation of Optical Flow between Image Pairs

Our application makes no assumption about camera positions, and usually deals with very large displacements. The only requirement to compute the correspondence is that the two images contain mutually overlapping regions, which are the regions we are interested in.

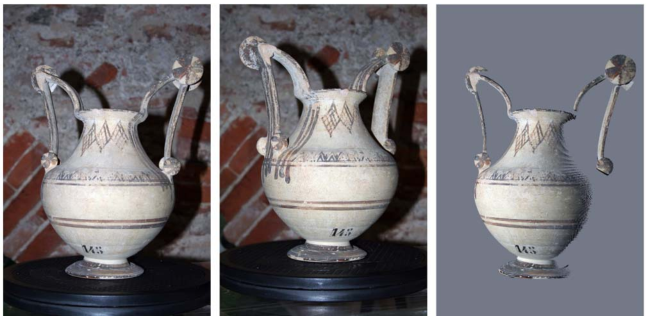


Fig. 2. Left: Input Image1. Center: Input Image2. Right: The projection of Image2 on the plane of Image1.

In contrast, a common assumption of most optical flow applications is that the camera position has a small variation between two images, or that the elements of the scene undergo small movements between subsequent frames. To establish an initial condition that can be handled in a straightforward way, we start by estimating the camera parameters for each image and the 3D model. Then, for every image pair (Image1 and Image2 in Fig. 2) and using the associated camera parameters, Image2 is projected onto the 3D model, and reprojected back on the camera plane of Image1 (see Fig. 2 right) [32]. Given an input set of n images, a maximum number of $n(n\text{-}1)$ projections need to be generated. In practice this number is much lower, since each image usually has a consistent overlap with no more than 4-5 other images.

As aforementioned, since a global perfect alignment is not always possible, the images are not perfectly superimposed when projected onto the model. One important contributing factor is the different samplings of color and geometry, which, as noted by Pulli et al. [5], is an inherent issue. Even though these discrepancies are generally only of a few pixels, which is an ideal initial condition to compute an accurate optical flow, there are still many issues, such as aliasing and blurring or differences in lighting, which might be present in the original set of images, or introduced during the projection phase.

Since it is very difficult to handle all cases at the same time, we have chosen solutions that provide a satisfactory result for a wide range of input sets with regards to the ambient conditions and camera settings. We tested three different approaches: a brute force template matching and its hierarchical variation (both implemented with CUDA), and the GPU implementation of Brox et al. [33] provided by the Floating Textures system [31].

The first two strategies do not use a minimization strategy to solve the objective function, but apply template matching in the following way:

$$q(Ii(p), \Delta x) = \frac{\sum_{c=0,1,2} \sum_t \left[f(p, \Delta t, c)^2 + \gamma g(p, \Delta t, c)^2 \right]}{\sum |t|},$$

where

$$\begin{aligned} f(p, \Delta t, c) &= I_i(p + \Delta t)_c - I_j(p + \Delta x + \Delta t)_c, \\ g(p, \Delta t, c) &= (I_i(p + \Delta t)_c - \bar{I}_{ic}) \\ &\quad - (I_j(p + \Delta x + \Delta_t)_c \bar{I}_{jc}), \end{aligned}$$

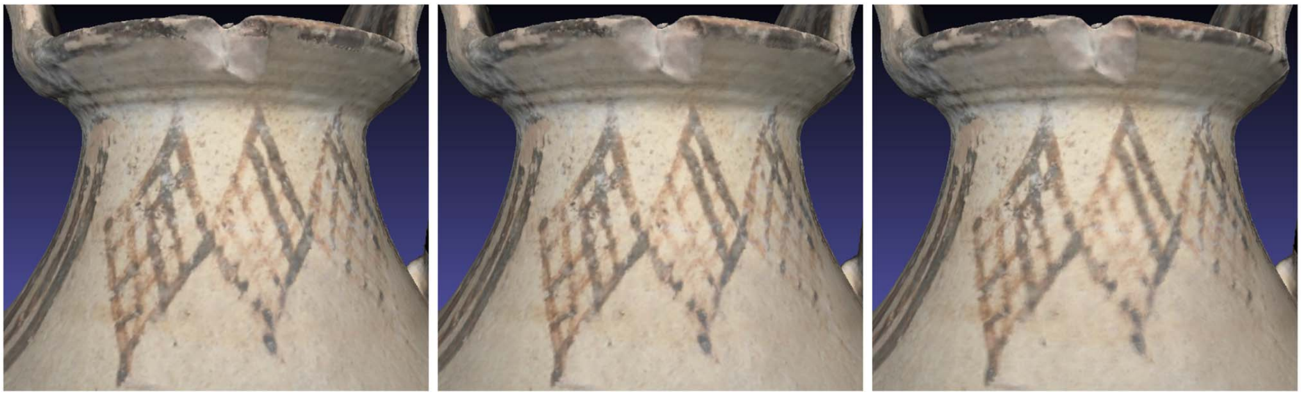


Fig. 3. Left: Brute force method. Center: Hierarchical template matching. Right: Floating Texture's optical flow implementation [31]. Note how even though the third method better preserves the borders (top of the vase), the patterns in the center of the figure are sharper with the first two methods.

where I_j is the search space in the target image, $I_i(p)c$ is one of the color channels of a pixel from the source image, $\bar{I}_{ic}$ is the average per color channel inside the template centered at p, and Δx and Δt are, respectively, the displacements of the search space and inside each template. The final flow for a pixel is given by the minimum value of q inside the search region. For our tests, we set $\gamma = 2$ and the search space and matching template, respectively, to 41^2 and 15^2 pixels, i.e., the maximum pixel displacement is 20 units in a each direction. For the hierarchical method, we reduce the search and matching templates to 15^2 and 5^2 pixels, respectively.

Although the source and target images may initially seem very similar, they present many high frequency variations, and create a deviation in image sharpness (see Fig. 4). This problem tends to be aggravated by warping methods—as already noted by Steinbruecker et al. [34]—since fine details are lost in coarser levels and, consequently, cannot be matched at the corresponding scale. Fortunately, the projection limits the flow to only a few pixels and allows us to use only a few levels of the hierarchy; in our tests we limited the maximum level to three.

Analyzing the visual results using the three methods, all data sets were handled in similar ways, although each method treats some particular regions better than others. Even though the Floating Texture method may achieve more precise results, it depends in some degree on fine-tuning the parameters for each specific input. The template matching strategies, on the other hand, were able to handle all tested data sets with the same configuration. Fig. 3 illustrates the difference between the three methods with a detail of the vase model.

It is important to point out that even though we are not giving shadows and highlights any special treatment, when the diffuse component predominates, which is indeed true for our tested sets, these issues can be properly treated. An example is the shadow in the sample images of the Painted Cave data set (see Fig. 10 top). In fact, the same phenomenon was already observed by Theobalt et al. [3], however, we believe that no single optical flow method will handle appropriately all possible inputs, thus strong illumination effects, such as specular highlights, might mislead the flow. However, Fig. 5 shows an example where noise is automatically removed during optical flow calculation.

Disregarding which approach is used, the optical flow operation produces a map for each projection of a source image onto another image's camera space. In order to continue with the image correction algorithm, it is necessary to "retro-project" the flow, i.e., bring it back to the camera space of the source image. This is carried out by first applying the flow to the projected image, and then projecting back the displaced pixel to its original camera space. The difference between the new and the original pixel positions defines the correction displacement to be applied to the image.

3.2 Using the Optical Flow

Once the optical flows for all the mutually overlapping image pairs have been calculated, they are used to coherently project the color information.

Let us start with some notation. We have a set of n images $I_1, \ldots, I_n$ registered over a 3D model and let's assume that we know how to compute the "real" color of each point p_o of the mesh as a weighted blend [24] of the colors of all the images involved

$$c(p_o) = \frac{\sum_j I(\phi_j(p_o))B_j(\phi_j(p_o))}{\sum B_j(\phi_j(p_o))},$$

where $\phi_i, (p_o)$ is the inverse camera projection that, given a 3D point p_o in object space, provides its 2D projection onto the image I_i. Analogously, $\phi_i^{-1}(p_i)$ is the direct camera projection, which projects a 2D point p_1 on a 3D model. In addition, $B_j(\phi_j(p_o))$ is a function that associates each point with a blending weight. To find out the relation between corresponding points on different images we use the $\psi_{i,j}(p_i)$ function

$$\psi_{i,j}(p_i) = \phi_j(\phi^{-1}(p_i)),$$

that computes where the point p_i from image I_i falls on image I_j. With $I_i(p)$ we denote the pixel at location p over the ith image. If there were no color variations, errors, and ignoring occlusions we would have

$$I_i(p_i) = I_j(\psi_{i,j}(p_i)).$$

Let us now introduce the flow function, that we computed for each significant pair of images (e.g., for each image pair that has a significant overlap in object space). The result of the flow computation between each pair of images i, j is a warping function $W_{i,j}$ such that

$$I_i(p_i) \approx I_j(W_{i,j}(p_i)).$$

In other words, the warping $W_{i,j}$ finds the corresponding position of a pixel of image i on image j. Note that the

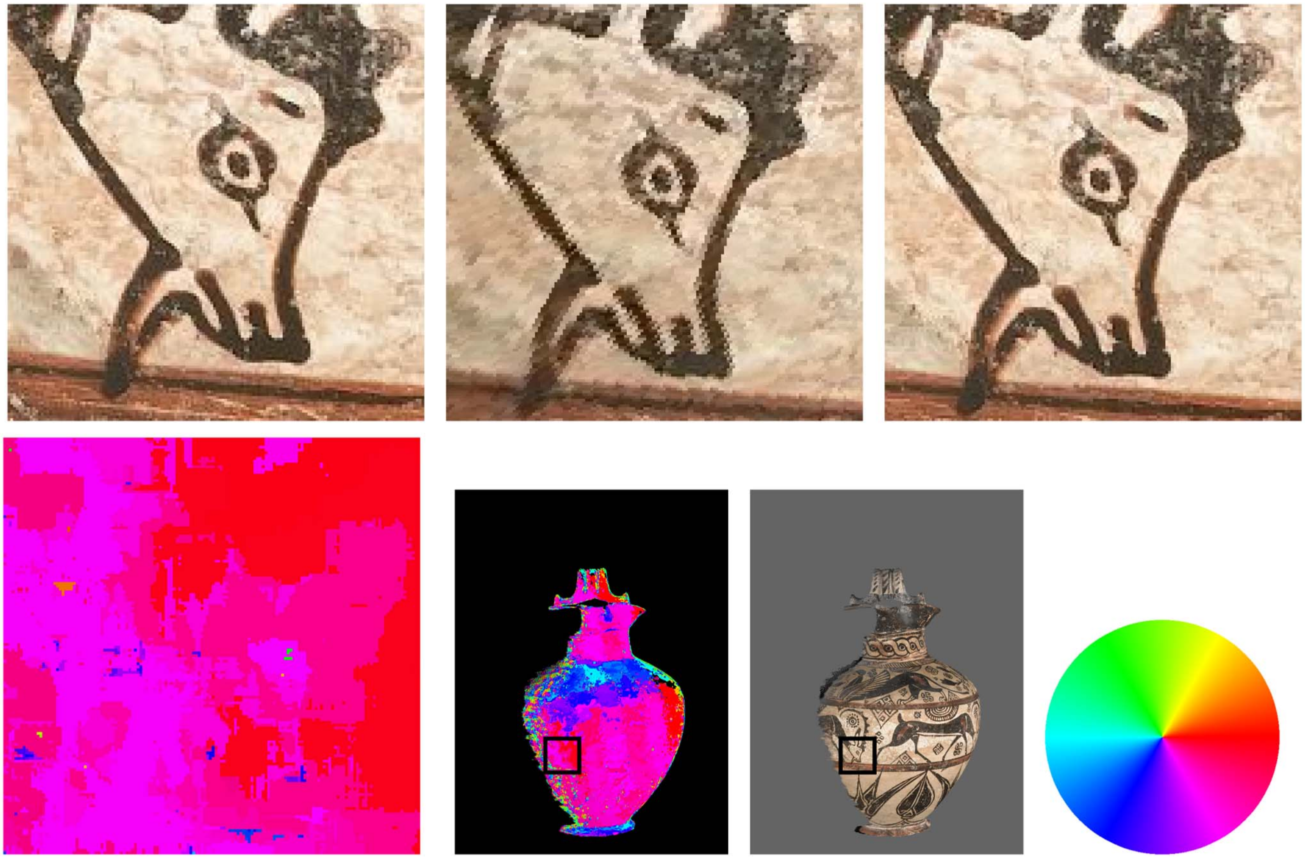


Fig. 4. Sequence of images from the flow calculation of the Louvre Vase. Top row: details of original image (target), projected image (source), and target image warped to source's plane using the flow. Bottom row: detail of the correspondent optical flow, entire optical flow using the first method, projected image, and color code for flow directions. Note that along the border of the vase the flow becomes highly irregular specially due to the low sampling quality of projected images in that region; however, this part is usually not used for blending since it obtains low scores.

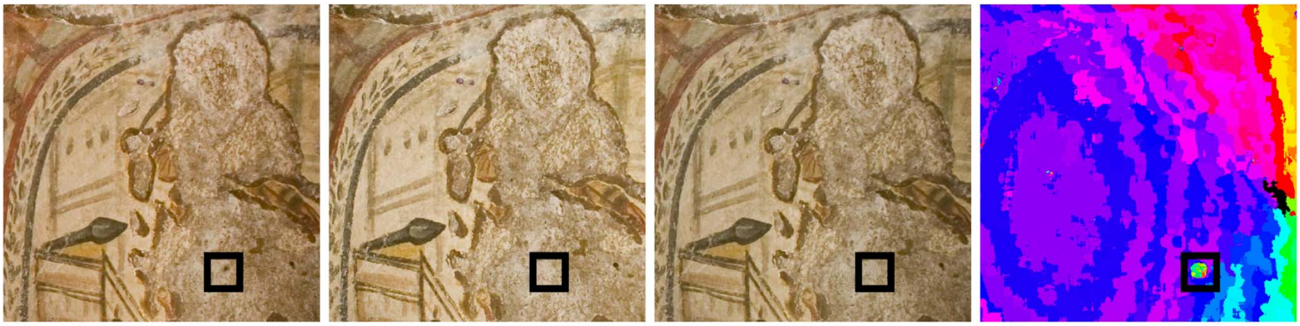


Fig. 5. Sequence of detail images from the flow calculation of the Painted Catacomb. From left to right: original image (target), projected image (source), target image warped with flow, and the flow field. Note that the flow is also able to remove noise (inside the black square in evidence) from the original image. The flow pattern indicates that there is not only a translation between the images, but also a radial distortion possibly caused by the camera lenses or image projections.

warping functions are generally not invertible, i.e., usually $W_{i,j} \neq W_{j,i}^{-1}$ due to occlusions and discontinuities. We define the displacement generated by the warping function as $\Delta_{i,j}(p_i)$, so that

$$\Delta_{ij}(p_i) = W_{ij}(P_i) - \psi_{i,j}(p_i),$$
$$W_{ij}(P_i) = \psi_{i,j}(p_i) + \Delta_{i,j}(P_i).$$

Fig. 6 shows an example of our notation. In practice, the warping finds the necessary Δ in order to correct the errors due to the nonperfect correspondence between the digitally scanned model and its real shape (captured by photos).

To coherently apply the warping function on the whole surface in a consistent way we have to choose some base reference system (all the warping functions denote relative movements). We assume that for each point p_0 of the digital scanned surface of the object, we know which images are projecting something on that point and that we have a scalar *scoring* value $Q_i(p_o)$ that tells us the best images projecting on it; we used the scoring value calculated by the system proposed by Callieri et al. [24]. This score is basically the product of several, normalized per-pixel quality values. Some of the basic quality are quite straightforward, such as the view angle, distance from the sensor, distance in image-space from photo borders, and depth discontinuities (inner silhouettes). Other metrics can be combined, such as image

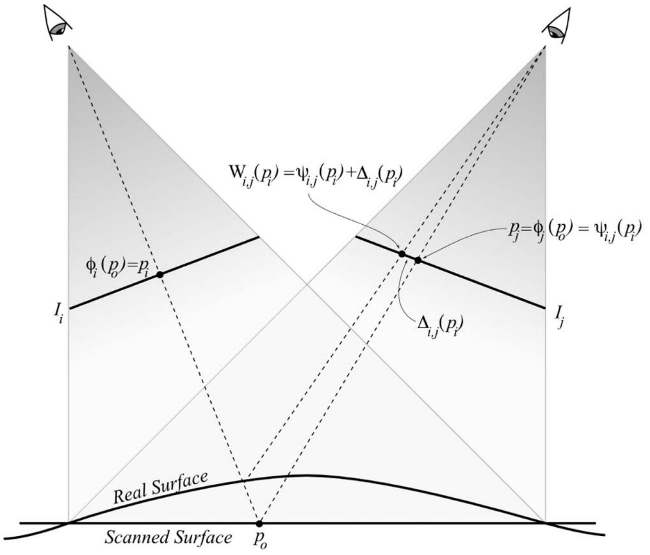

$W_{i,j}(p_i) = \psi_{i,j}(p_i) + \Delta_{i,j}(p_i)$
$p_j = \phi_j(p_o) = \psi_{i,j}(p_i)$
$\phi_i(p_o) = p_i$
I_i
$\Delta_{i,j}(p_i)$
I_j
Real Surface
Scanned Surface
p_o

Fig. 6. The notation used to denote correspondence between (warped) points and images.

sharpness and custom-made masks. The use of a product to combine the weights ensures that zeros are conserved. Please refer to the paper for further details. This scoring mechanism induces a partition $V^1 = \{V_1^1, \ldots, V_n^1\}$ of the mesh into regions, such that V_i^1 is the portion of the mesh where the image I_i has the highest score. For each region, we use the best image I_i as a reference and, when blending, warp the other images to the space of I_i

$$C^i(P_o) = \frac{\sum_j I_j(W_{i,j}(\phi_i(P_o)) \cdot B_j(W_{i,j}(\phi_i(p_o))}{\sum B_j(W_{i,j}(\phi_i(p_o))}.$$

In other words, for each image I_j, instead of directly using the pixel $I_j(\phi_j(p_o))$, we use its warped image $I_j((W_{i,j}(\phi_i(p_o)))$. For the sake of readability, we define the shorthand notation

$$\phi_j^i(p_o) = W_{i,j}(\phi_i(p_o)),$$

which denotes the fact that, when we need a pixel of I_j for blending in the space of I_i, we need to go through the warping function $W_{i,j}$. The blending function can now be written as

$$C^i(P_o) = \frac{\sum_j I_j(\phi_j^i(p_o)) \cdot B_j(\phi_j^i(p_o))}{\sum B_j(\phi_j^i)(p_o))}.$$

Unfortunately, as shown in Fig. 8 (middle), this approach creates seam artifacts on the boundary between different regions. In fact, for each region, all warpings are based on a different base image, i.e., on the boundary between regions V_i^1 and V_j^1, on one side pixels of image I_i are left unmoved and pixels of image I_j are warped by $\Delta_{i,j}$, while on the other side pixels of I_j are fixed and those of I_i are warped by $\Delta_{j,i}$. It is possible to correct this issue under the assumption that the warping function is continuous and bijective

$$\Delta_{i,j}(\phi_i(p_o)) = -\Delta_{j,i}(\phi_j(p_o)),$$

and computing for each region, a continuous prewarping $\theta_k^{ij}(p)$ of all the involved images, so that it counter balances boundary differences (see Fig. 8 (bottom)).

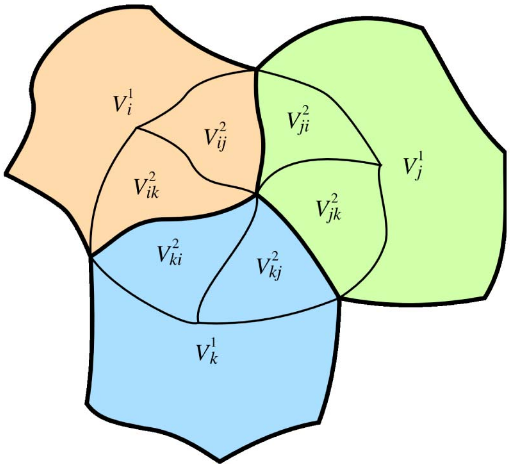


Fig. 7. We partition the surface of the mesh according to an image scoring value defining the regions V_i^1, where each image I_i is the best one. The region is further subdivided in subpartitions $V_{i,j}^2$, where the images I_i, I_j are, respectively, the first two best options.

The image scoring mechanism is used to make a further sub-partitioning of the surface. Let $V^2 = \{V_{1,1,\ldots,}^2 V_{i,j,\ldots,}^2 V_{n,n}^2\}$ be the partition of the mesh into regions such that $V_{i,j}^2$ is the portion of the mesh where I_i and I_j are the exactly two best images that project onto that region. For each region $V_{i,j}^2$, we consider $\delta(p_o)$ as a weighted distance from the boundary shared with the region $V_{j,i}^2$. Fig. 7 illustrates this kind of partitioning. Now we can define the warping of a pixel from I_k, to the region defined by the two best images I_i and I_j, by

$$\theta_k^{i,j}(p_o, \alpha) = \phi_k(p_o) + \Delta_{i,k}(p_i) - (1 - \alpha) \cdot \frac{\Delta_{i,j}(p_i)}{2},$$

where α is a value that is equal to 0 on the boundary between the two regions and less than 1 elsewhere. This alpha is calculated as the pixel distance, in the dominant image space, divided by the maximum size of the overlapping area (this value may be calculated at the beginning, while computing weights and optical flow, or imposed as a fixed threshold to limit the extent of warp blending).

For the points far from the border (where α is one) this equation simplifies to the previous case, while on the boundary between two regions $V_{i,j}^2$ and $V_{j,i}^2$ (where α is zero), it computes a warping equivalent to that of the adjacent region.

The θ function does not affect the accuracy of the matching done by the flow based warping, because in the blending equation it modifies all the image accesses in a consistent way. It can be considered as a *prewarping* that deforms the image space in the same way for each point of each region.

Nonetheless, warping is generally not bijective and the δ are not antisymmetric. In most cases, this is due to occlusions that lead to discontinuities in the mapping process. It thus becomes impossible to achieve a continuous warping, and so we compromise with the average of the Δ involved

$$\theta_k^{i,j}(p_o, \alpha) = \phi_k(p_o) + \Delta_{i,k}(p_i) - (1 - \alpha) \cdot \frac{\Delta_{i,j}(p_i) - \Delta_{j,i}(p_j)}{4}.$$

The above approach could cause some other minor discontinuities in the boundary between regions $V_{i,j}^2$ and

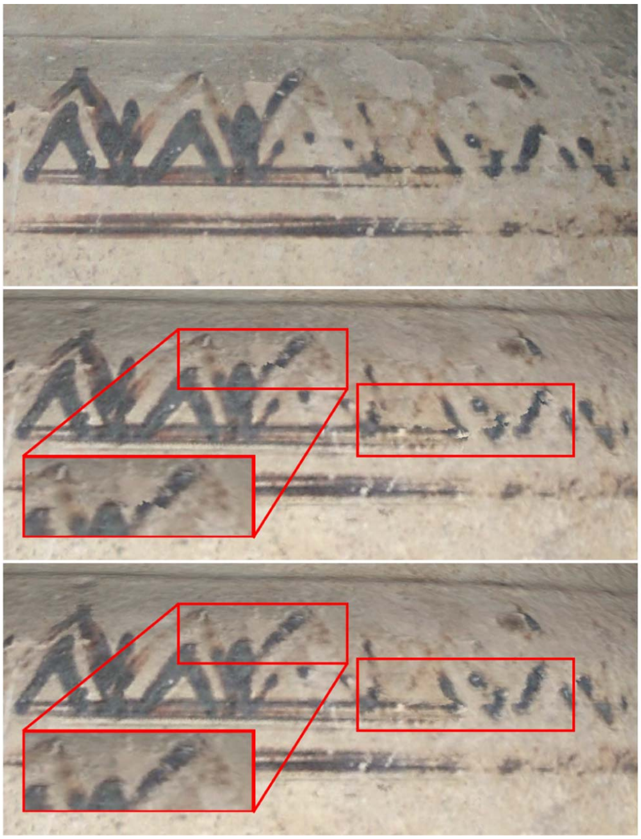


Fig. 8. Top: a portion of one of the images used for projection. Middle: the colored model obtained using optical flow correction, without taking into account the boundary between images (visible artifacts in the red boxes). Bottom: the colored model obtained using a continuous prewarping function.

$V^2_{i,k}$. However, these discontinuities are hardly noticeable and can be removed by applying the previous blending approach to a partitioning $V^3_{i,j,k}$ according to the first three most important images i, j, k. On such a partition, it is possible to blend the warping vectors Δ based on barycentric interpolating coordinates centered on the "region corners," where three regions share a pair of indexes. While such an approach is more *theoretically correct*, we found the previous technique simpler, sufficient, and more practical from an implementation point of view.

4 Results

We applied our method on a number of test sets of different qualities and sizes. In this section, five examples are shown and analyzed, all were computed using the second optical flow calculation method (hierarchical template matching), which proved to be the best trade-off between speed and accuracy.

The camera parameters for the images to be aligned were obtained in several different ways, but the projection of color information was performed by extending the blending approach proposed by Callieri et al. [24]. This system proved to be extremely flexible and scalable, and since our proposed technique is not dependent on any global optimization, it can be easily integrated and used even on complex cases, where dense 3D models and hundreds of images are used.

The first test was performed on the 3D model of an ancient vase, made up of a 5 million triangle model and 12 images. In this case, the geometry was acquired using a triangulation scanner, hence, it was extremely accurate. The images were acquired under a not ideal lighting setup and were of medium quality.

The images were aligned on the model using a semiautomatic approach [10]. Fig. 9 (first row) shows that the use of the correction optical flow preserves the detail in the decorations of the vase: visible ghosting artifacts are completely removed, and the resolution of the color information is almost the same as the original images (see also Fig. 8).

The second test case was another vase from the Louvre collection (inv. A 316). While the quality of the nine images was extremely good, the 3D model had several inaccuracies, especially around the handle and the lip, which are crucial in order to obtain an accurate image alignment. The alignment quality obtained using the silhouette based approach [8] was thus not good enough to preserve the quality of the starting images. Our method managed to remove the annoying ghosting effect on the colored model (Fig. 9 (second row)).

The third test (Fig. 9 (third row)) involved a 3D model of a prehistoric skull, obtained using 3D scanning, on which 12 images were mapped. In this case both the model and the images exhibit high accuracy, but the alignment is not accurate enough to avoid blurring. The use of our method brings it to a visible improvement. The fourth test regarded a portion of a painted cave, acquired at high resolution. The images were not acquired under optimal lighting conditions, but the main issue was in the quasiplanar shape of the geometry, which prevented an accurate estimation of parameters of the camera. The misalignments shown in Fig. 10 (first row) were solved by our method.

In the fifth test set, the geometry was generated using a commercial system which starts from images [35]. In addition, the camera parameters associated with the six images were extracted from the internal data of the system. This meant that it was impossible to correct any inaccuracy. In fact, the color projection produced blurry details, which were corrected by our system (see Fig. 10 (second row)).

In conclusion, the test sets presented above showed that the correction system can improve the quality of the colored model in a variety of cases. The method is compatible with both per-vertex and texture color encoding, so that it can be applied on both high and low-level detail 3D models. Table 1 shows a summary of the features of the data sets used. The optical flow was calculated only on the projected images on which the overlapping pixels were more than 10 percent of whole image: hence, the fourth column is an indirect measure of the overlap between the projected images.

Finally, the values in the last three columns show the computational time needed to calculate the optical flow for the presented approaches: Brute Force Templates (BF), Hierarchical Templates (HT), and Floating Textures (FT). Time values are influenced not only by the image resolution, but also by the peculiar nature of the data set (the amount of overlap between images).

While solving the problem of small misalignments and subsequent blurring, the proposed method inherits some of

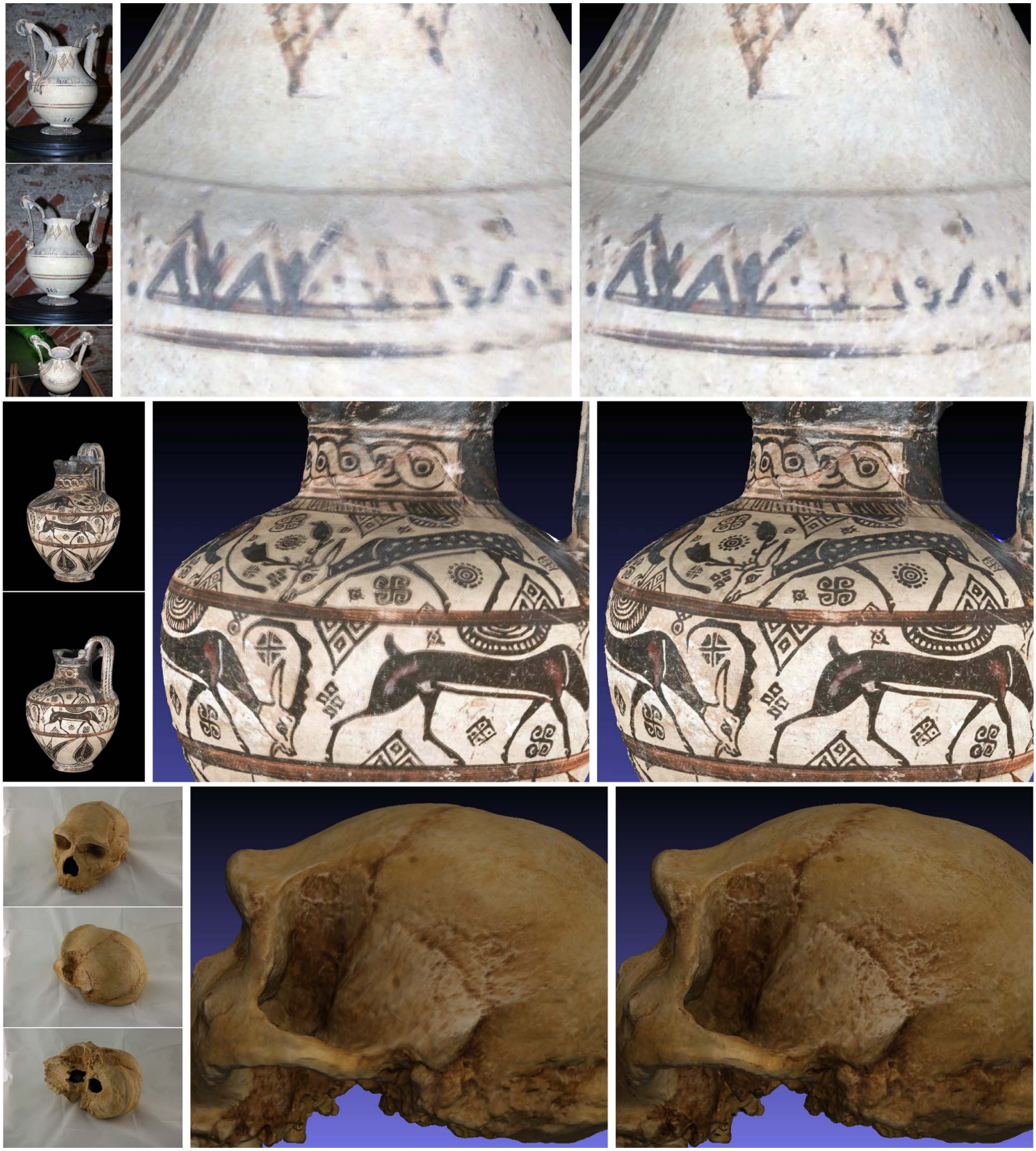


Fig. 9. Three examples of color projection. For each row, a sample of the images, a rendering of the colored model without and with the flow correction are presented.

the limitations of the image projection approaches. First of all, if the misalignment between images is very big (tens of pixels) it could be impossible to calculate the flow. It could be thought of trying different flow bands to adapt to this, but in the case of repeating high frequency features this could bring to further inaccuracies. The presence of lighting artifacts (i.e., specular highlights, shadows) can mislead the flow calculation. Finally, if the quality of images is extremely low, or there are severe differences in lighting conditions, flow calculation can be extremely difficult. Nevertheless, these drawbacks are shared with almost all of the state-of-the-art approaches in the field.

5 Conclusions and Future Work

We have presented a system to correct misalignment artifacts in an image-to-geometry projection procedure, which otherwise would lead to ghosting artifacts in the final 3D colored model. Unlike previous methods, we made no assumption about the existence of a perfect global

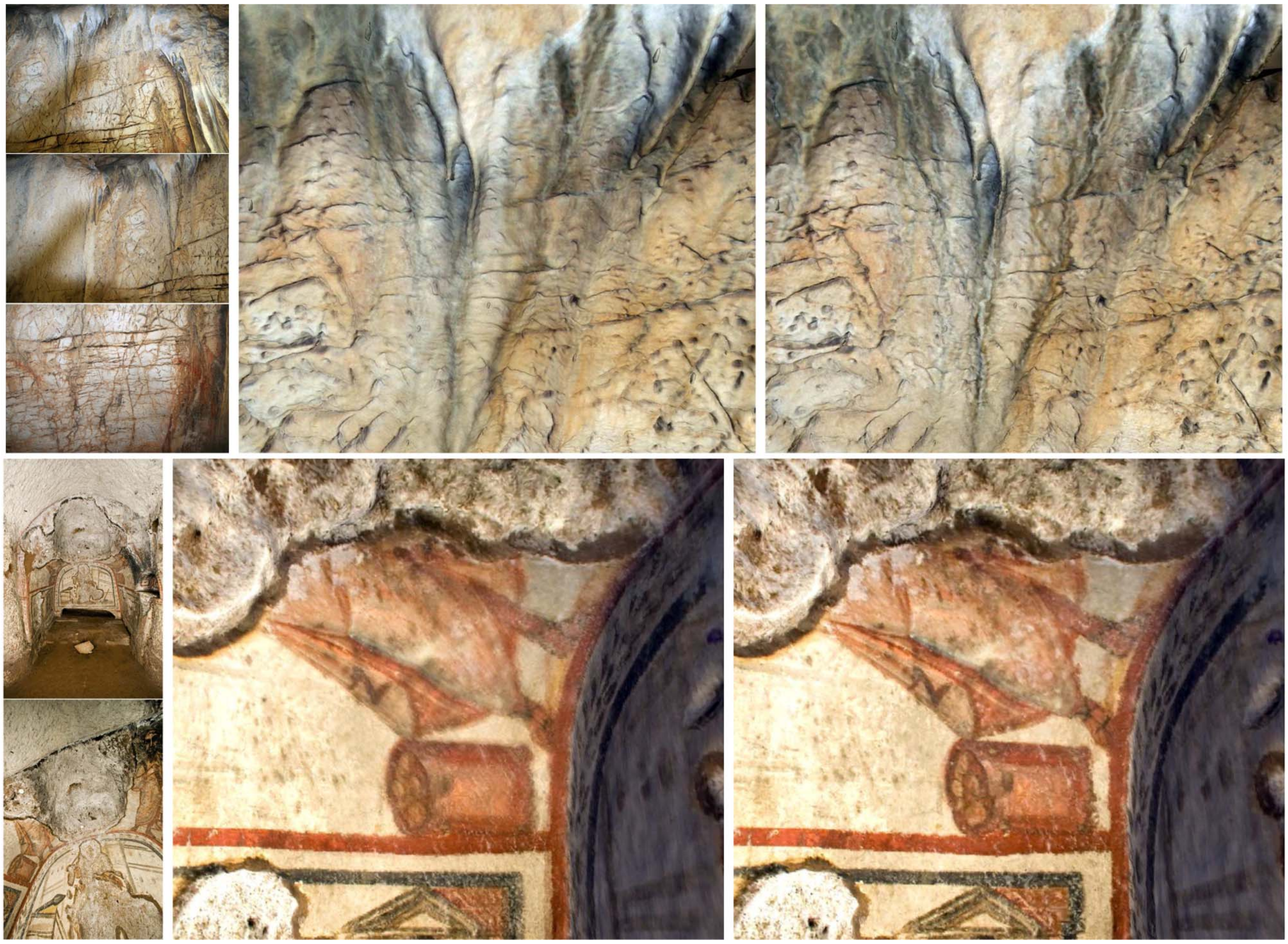


Fig. 10. Two examples of color projection. For each row, a sample of the images, a rendering of the colored model without and with the flow correction are presented.

TABLE 1
Overview of the Test Sets

Test Case	3D Model size, Color coding	N. of images(Resolution)	Overlapping (Max)	BF	HT	FT
1- Vase	5M, per-vertex	12 (1664x2496)	103 (132)	~4 h.	82 m.	45 m.
2- Louvre Vase	150K, texture(4096x4096)	9 (2004x2672)	54 (72)	~4 h.	95 m.	70 m.
3- Prehistoric Skull	5M, per-vertex	12 (2500x1673)	102 (132)	~3 h.	60 m.	27 m.
4- Painted Cave	6M, per-vertex	8 (2700x1800)	40 (56)	~5 h.	81 m.	60 m.
5- Painted Catacomb	100k, texture(4096x4096)	6 (3000x1996)	36 (36)	~5 h.	84 m.	42 m.

alignment between the geometry and the images. Hence, we performed a local optimization by warping the overlap between images to create a consistent mapping.

We made use of optical flow techniques that can handle many issues encountered in real cases. It is performed over pairs of prealigned images by using the underlying geometry to project each image onto the plane of the others. The resulting flow is then retroprojected and used together with a partition of the space into best fitting images, to perform the local warps and the final blending.

We have shown, using different examples, that the system can handle input data sets that do not possess ideal characteristics for alignment, either because of issues in the acquisition process, or due to inherited issues in the nature of the models. Another advantage of our method is that it scales well with the size of the input data, and is able to deal with large models in term of the geometry, the number of images in the set, and image resolution. Although we showed the method in the context of color projection, it can also be exploited in a number of other applications (medical, restoration, 3D reconstruction from images) which use registered images as an input.

We are currently integrating a measure of flow quality with the blending function to better drive the final color estimation. Furthermore, we are investigating how to treat more drastic cases in terms of deviation in illumination between images, and projection of shadows.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (/FP7/2007-2013), through the 3D-COFORM project, under grant agreement n 231809. We would like to thank all the people who helped us with feedback, data, and support.

REFERENCES

[1] B. Brown and S. Rusinkiewicz, "Global Non-Rigid Alignment of 3-D Scans," *ACM Trans. Graphics* , vol. 26, no. 3, p. 21, Aug. 2007.

[2] M.J. Black and P. Anandan, "A Framework for the Robust Estimation of Optical Flow," *Proc. Fourth Int'l Conf. Computer Vision,* pp. 231-236, 1993.

[3] C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, and H.-P. Seidel, "Seeing People in Different Light-Joint Shape, Motion, and Reflectance Capture," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 4, pp. 663-674, July/Aug. 2007.

[4] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High Resolution Passive Facial Performance Capture," *ACM Trans. Graphics,* vol. 29, no. 4, pp. 1-10, 2010.

[5] K. Pulli, S. Piiroinen, T. Duchamp, and W. Stuetzle, "Projective Surface Matching of Colored 3D Scans," *Proc. Fifth Int'l Conf. 3-D Digital Imaging and Modeling (3DIM),* pp. 531-538, 2005.

[6] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, and Y. Okamoto, "The Great Buddha Project: Digitally Archiving, Restoring, and Analyzing Cultural Heritage Objects," *Int'l J. Computer Vision,* vol. 75, no. 1, pp. 189-208, 2007.

[7] L. Brunie, S. Lavallée, and R. Szeliski, "Using Force Fields Derived from 3D Distance Maps for Inferring the Attitude of a 3D Rigid Object," *ECCV '92: Proc. Second European Conf. Computer Vision,* pp. 670-675, 1992.

[8] H. Lensch, W. Heidrich, and H. Seidel, "Automated Texture Registration and Stitching for Real World Models," *Proc. Eighth Pacific Conf. Computer Graphics and Applications,* pp. 317-327, 2000.

[9] L. Liu, I. Stamos, G. Yu, G. Wolberg, and S. Zokai, "Multiview Geometry for Texture Mapping 2D Images Onto 3D Range Data," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 2293-2300, 2006.

[10] T. Franken, M. Dellepiane, F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno, "Minimizing User Intervention in Registering 2D Images to 3D Models," *The Visual Computer,* vol. 21, nos. 8-10, pp. 619-628, http://vcg.isti.cnr.it/Publications/2005/CDFGMS05, Sept. 2005.

[11] J. Dorsey, H. Rushmeier, and F. Sillion, *Digital Modeling of Material Appearance,* Morgan Kauf./Elsevier, http://artis.inrialpes.fr/Publications/2007/DRS07, 2007.

[12] P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-Based Reconstruction of Spatially Varying Materials," *Proc. 12th Eurographics Workshop Rendering Techniques,* pp. 104-115, June 2001.

[13] S.R. Marschner, S.H. Westin, E.P.F. Lafortune, K.E. Torrance, and D.P. Greenberg, "Image-Based Brdf Measurement Including Human Skin," *Proc. 10th Eurographics Workshop Rendering,* pp. 139-152, June 1999.

[14] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the Reflectance Field of a Human Face," *SIGGRAPH '00: Proc. 27th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 145-156, 2000.

[15] Y. Sato, M.D. Wheeler, and K. Ikeuchi, "Object Shape and Reflectance Modeling from Observation," *Proc. ACM SIGGRAPH '97,* pp. 379-387, 1997.

[16] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, "Inverse Global Illumination: Recovering Reflectance Models of Real Scenes from Photographs," *Proc. ACM SIGGRAPH '99,* pp. 215-224, 1999.

[17] F. Bernardini, I. Martin, and H. Rushmeier, "High-Quality Texture Reconstruction from Multiple Scans," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, no. 4, pp. 318-332, Oct.-Dec. 2001.

[18] M. Callieri, P. Cignoni, and R. Scopigno, "Reconstructing Textured Meshes from Multiple Range Rgb Maps," *Proc. Seventh Int'l Fall Workshop Vision, Modeling, and Visualization,* pp. 419-426, Nov. 2002.

[19] N. Bannai, A. Agathos, and R. Fisher, "Fusing Multiple Color Images for Texturing Models," *Proc. Second Int'l Symp. 3D Data Processing, Visualization and Transmission (3DPVT '04),* pp. 558-565, 2004.

[20] M. Chuang, L. Luo, B.J. Brown, S. Rusinkiewicz, and M. Kazh-dan, "Estimating the Laplace-Beltrami Operator by Restricting 3D Functions," *Proc. Symp. Geometry Processing,* pp. 1475-1484, July 2009.

[21] K. Pulli, H. Abi-Rached, T. Duchamp, L. Shapiro, and W. Stuetzle, "Acquisition and Visualization of Colored 3D Objects," *Proc. 14th Int'l Conf. Pattern Recognition (ICPR '98),* pp. 11-15, 1998.

[22] A. Baumberg, "Blending Images for Texturing 3D Models," *Proc. British Machine Vision Conf. (BMVC '02),* 2002.

[23] V. Rankov, R. Locke, R. Edens, P. Barber, and B. Vojnovic, "An Algorithm for Image Stitching and Blending," *Proc. SPIE,* vol. 5701, pp. 190-199, 2005.

[24] M. Callieri, P. Cignoni, M. Corsini, and R. Scopigno, "Masked Photo Blending: Mapping Dense Photographic Data Set on High-Resolution 3D Models," *Computer and Graphics,* vol. 32, no. 4, pp. 464-473, http://dx.doi.org/10.1016/j.cag.2008.05.004, http://vcg. isti.cnr.it/Publications/2008/CCCS08, Aug. 2008.

[25] P. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," *SIGGRAPH '98: Proc. 25th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 189-198, 1998.

[26] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec, "Capturing and Rendering with Incident Light Fields," *Proc. 14th Eurographics Workshop Rendering (EGRW '03),* pp. 141-149, 2003.

[27] M. Dellepiane, M. Callieri, M. Corsini, P. Cignoni, and R. Scopigno, "Improved Color Acquisition and Mapping on 3D Models via Flash-Based Photography," *ACM J. Computers and Cultural Heritage,* vol. 2, no. 4, pp. 1-20, http://vcg.isti.cnr.it/Publications/2010/DCCCS10, Feb. 2010.

[28] T. Takai, A. Hilton, and T. Matsuyama, "Harmonised Texture Mapping," *Proc. 3D Data Processing Visualization and Transmission (3DPVT '10),* May 2010.

[29] E. Aganj, P. Monasse, and R. Keriven, "Multi-View Texturing of Imprecise Mesh," *Proc. Asian Conf. Computer Vision,* Sept. 2009.

[30] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or, "Seamless Montage for Texturing Models," *Computer Graphics Forum,* vol. 29, no. 2, pp. 479-486, 2010.

[31] M. Eisemann, B.D. Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent, "Floating Textures," *Computer Graphics Forum,* vol. 27, no. 2, pp. 409-418, 2008.

[32] P.E. Debevec, C.J. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry-and Image-Based Approach," *SIGGRAPH '96: Proc. 23rd Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 11-20, 1996.

[33] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," *Proc. Eighth European Conf. Computer Vision,* vol. 4, pp. 25-36, May 2004.

[34] F. Steinbruecker, T. Pock, and D. Cremers, "Large Displacement Optical Flow Computation without Warping," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* 2009.

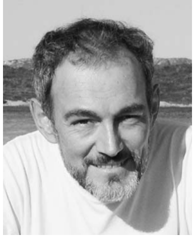
[35] "Adam Technology Website," More Info on: http://www.adamtech.com.au/, 2011.

Matteo Dellepiane received the advanced degree in telecommunication engineering (Laurea) from the University of Genova in 2002, and the PhD degree in information engineering from the University of Pisa in 2009. He is a researcher at Istituto di the Scienza e Tecnologie dell'Informazione (ISTI) of the National Research Council (CNR) in Pisa, Italy. His research interests include 3D scanning, digital archeology, color acquisition and visualization on 3D models, and perceptual rendering.

Ricardo Marroquim received the MSc and DSc degrees in computer and system engineering from the Federal University of Rio de Janeiro (UFRJ) in 2005 and 2008, respectively. In 2009 he was an ERCIM postdoc fellow at the Visual Computing Lab (CNR-Pisa), and since 2010 is a full professor at the Computer Graphics Lab (LCG) at UFRJ. His recent research topics include real-time rendering, point-based graphics, and 3D model reconstruction.

Marco Callieri received the master's degree in computer science in 2001 and the PhD degree in computer science from the University of Pisa in 2006. He is a researcher at the Istituto di Scienza e Tecnologie dell'Informazione (ISTI) of the National Research Council (CNR) in Pisa, Italy. His research interests include 3D scanning, color and appearance acquisition, large 3D data set manipulation and rendering.

Paolo Cignoni received the PhD degree in computer science at the University of Pisa in 1998. He is a senior research scientist with CNR-ISTI. He has been awarded "Best Young Researcher" by the Eurographics association in 2004. His research interests cover Computer Graphics fields ranging from visualization and processing of huge 3D data sets, to 3D scanning in the cultural heritage field and to Scientific Visualization. He has published more than 100 papers in international refereed journals and conferences.



Roberto Scopigno graduated in computer science at the University of Pisa in 1984. He is a research director with ISTI-CNR and leads the Visual Computing Lab. He is engaged in research projects concerned with 3D scanning, surface reconstruction, multiresolution, scientific visualization, and cultural heritage. He published more than 150 papers in international refereed journals/conferences. He has been responsible person for CNR-ISTI in several EU projects and co-chaired several international conferences. He served as chair of the Eurographics Association (2009-2010) and co-editor in chief of the Computer Graphics Forum journal (2001-2010).