TO APPEAR AT THE WORKSHOP ON 3D REPRESENTATION FOR RECOGNITION (3DRR-11) AT ICCV 2011

Automatic alignment of paintings and photographs depicting a 3D scene

Bryan C. Russell
INRIA*

Josef Sivic
INRIA*

Jean Ponce
École Normale Supérieure*

Hélène Dessales
École Normale Supérieure†

Abstract

This paper addresses the problem of automatically aligning historical architectural paintings with 3D models obtained using multi-view stereo technology from modern photographs. This is a challenging task because of the variations in appearance, geometry, color and texture due to environmental changes over time, the non-photorealistic nature of architectural paintings, and differences in the viewpoints used by the painters and photographers. Our contribution is two-fold: (i) we combine the gist descriptor [23] with the view-synthesis/retrieval of Irschara et al. [14] to obtain a coarse alignment of the painting to the 3D model by view-sensitive retrieval; (ii) we develop an ICP-like viewpoint refinement procedure, where 3D surface orientation discontinuities (folds and creases) and view-dependent occlusion boundaries are rendered from the automatically obtained and noisy 3D model in a view-dependent manner and matched to contours extracted from the paintings. We demonstrate the alignment of XIXth Century architectural watercolors of the Casa di Championnet in Pompeii with a 3D model constructed from modern photographs using the PMVS public-domain multi-view stereo software.

1. Introduction

Given a set of photographs depicting a static 3D scene, it is often possible to automatically align the photographs and recover the corresponding camera parameters and pose, along with a model of the underlying 3D scene. The ability to reliably find and match features at interest points, along with robust procedures for bundle adjustment, has allowed for the recovery of large-scale 3D models from many images taken from a variety of viewpoints and under many viewing conditions [8, 9, 10]. This has allowed several applications, such as image retrieval [24], virtual exploration of a site [29], preserving cultural heritage [12], and computational rephotography [2].

*WILLOW project-team, Laboratoire d'Informatique de l'École Normale Supérieure ENS/INRIA/CNRS UMR 8548.

†Laboratoire Archéologies d'Orient et d'Occident et textes anciens, ENS/CNRS UMR 8546.

While much success has been shown for aligning photographs, we are not aware of work on automatically aligning non-photographic depictions of a scene, such as paintings and drawings (herein, for convenience, we will refer to both paintings and drawings as *paintings*). Such depictions are plentiful and comprise a large portion of our visual record. In this paper, we seek to automatically align paintings and photographs depicting a static 3D scene. Our main focus in this work is the ability to handle historical architectural paintings where the artist has made an effort to accurately portray a 3D scene. Such paintings are often made with the aid of a *camera lucida*, so we will assume, at least to first order, that the paintings are perspective scene renderings. In spite of this assumption, we will see that this is still a challenging scenario, and progress in this direction would be of benefit to fields where paintings were used to record the state of a given site, such as archaeology. We believe that the overall system is relevant to scenarios where local feature matching fails, e.g. texture-less or wiry objects with significant 3D structure (e.g. chairs, mugs, tables) or changes due to time/season, which present significant challenges for current feature-based instance-level matching/recognition techniques. Image to 3D model matching is also important for applications in urban planning and civil engineering, where a reference 3D model may be available but may contain errors or unexpected changes (e.g. something built/destroyed).

A number of difficulties arise when trying to align paintings and photographs. These difficulties are primarily due to what makes paintings fundamentally different from photographs. At first glance, a realistic painting of a scene does not appear to be very different from a photograph taken from a similar viewpoint. However, upon closer inspection, they are usually quite different, since the color, geometry, illumination, shading, shadows and texture may be rendered by the artist in a realistic, but "non physical" manner. These differences can be quite difficult to spot, which artists often exploit to "fake" visual realism with much success [5].

Because of these differences, the main issue is establishing visual correspondences between paintings and photographs. Current successful systems for 3D reconstruction of large-scale data (e.g. PhotoSynth [29]) completely

fail when considering paintings. For example, using SIFT matching [19], the painting in Figure 5(a), which at first-glance looks like a clean image rendering of the scene, only finds 121 putative matches total across all images in our dataset. Upon visual inspection, none of the putative painting-image matches are correct. We believe that local feature matching, as currently implemented in Bundler, would not work for this problem.

In this work, we investigate features for reliably aligning paintings and photographs. We primarily build upon recent success in recovering dense 3D points and a triangular mesh from a set of photographs [9, 15]. We believe that the use of such a 3D model offers several advantages to aid in alignment. First, the 3D model allows us to extract viewpoint independent features, such as folds and creases, and viewpoint dependent ones, such as occlusion boundaries. We argue that such features are better-suited for matching with paintings, and are not necessarily easily extracted by reasoning about the photographs independently. Furthermore, while impressive results have been shown to densely align challenging image pairs exhibiting drastic appearance changes [13, 27, 30, 31], these approaches do not use a 3D model and rely on the image pair to have nearby viewpoints. Such direct matching is difficult when a photograph from a similar viewpoint is not available. In our case, using a 3D model allows us to cope with new, previously unseen and potentially different viewpoints.

For our study, we use modern photographs and historical drawings and paintings depicting the Casa di Championnet, which is located amongst the ancient ruins of the Roman town of Pompeii. As the site sits outside and exposed to the elements, various components of the site have changed over time. For example, many of the wall murals that existed in the earlier paintings have almost completely disappeared in its present state. Furthermore, the central columns have changed shape, with one suffering major collapse since its initial excavation. This site is of particular interest to archaeologists, who wish to study these changes, in addition to digitally restoring the site. This poses an additional challenge for alignment, which must account for these structural changes across time.

Some of the above issues are partially addressed in prior work. Procedural modeling is used to build a 3D model of the Pompeii site [21], with the models manually created with the aid of archaeologists. The 4D cities project seeks to align photographs across time [26]. The Piranesi project [16] re-photographs historical paintings by manually finding a similar viewpoint as the painting. While there have been significant efforts to align photographs to 3D models, e.g. [14, 17, 18, 25], paintings have been overlooked so far. Dense alignment of challenging image pairs (e.g. infrared and visible spectrum imagery) has only been demonstrated on photographs [13, 27, 30, 31], and the focus has been on 2D alignment models. Recently, there has been work to align silhouettes extracted from topological maps to photographs, with the initial view obtained by GPS [1]. In contemporary work [28], painting-to-image matching and retrieval over a large database has been explored. However, this work only considers image-based similarity and does not reason about a 3D model.

Inputs: painting $\mathcal{I}$, set of photographs $\mathcal{J}$
Output: triangular mesh $\mathcal{M}$, camera parameters Θ for $\mathcal{I}$

1. Recover triangular mesh $\mathcal{M}$ from the set of photographs $\mathcal{J}$ using Bundler [29], PMVS [9], and Poisson surface reconstruction [15] (Section 2).
2. Coarse alignment by view-sensitive retrieval (Section 3)
   (a) Generate virtual cameras that uniformly sample the recovered 3D scene and render the views [14].
   (b) Find nearby virtual viewpoint Θ to $\mathcal{I}$ by gist feature matching [23].
3. Fine alignment by matching view-dependent contours (Section 4)
   (a) Extract contours (ridges and valleys [22] and occluding contours) for Θ from $\mathcal{M}$.
   (b) Use shape context features [3] to iteratively match contours for Θ to gPB contours [20] extracted from $\mathcal{I}$ and estimate Θ from the correspondences.

Figure 1. Algorithm overview.

Our contribution is the use of a 3D model to extract features suitable for matching paintings and photographs. To the best of our knowledge, this is the first system to automatically align paintings and drawings with a 3D model computed from photographs. On the technical level, our contribution over the current state-of-the-art is two-fold. First, we combine the gist descriptor [23] with the view-synthesis/retrieval of Irschara et al. [14] to obtain a coarse alignment of the painting to the 3D model. The view-synthesis/retrieval allows reasoning about unseen viewpoints [14] while the gist descriptor is sensitive to viewpoint [17] and has been shown to generalize across appearance variation for scene category recognition [23]. Second, we develop a fine alignment procedure based on the Iterative Closest Point (ICP) algorithm [4], where 3D surface orientation discontinuities (folds and creases) and view-dependent occlusion boundaries are rendered from the automatically obtained and noisy 3D model in a view-dependent manner and matched to contours extracted from the paintings. By aligning view-dependent contours, we hope to overcome the limitations of local feature matching for this problem by exploiting the ability to recover reliable contours from the painting and 3D model and by building on the success of contour matching [3]. We demonstrate the alignment of XIXth Century architectural watercolors of the Casa di Championnet in Pompeii with a 3D model constructed from modern photographs.

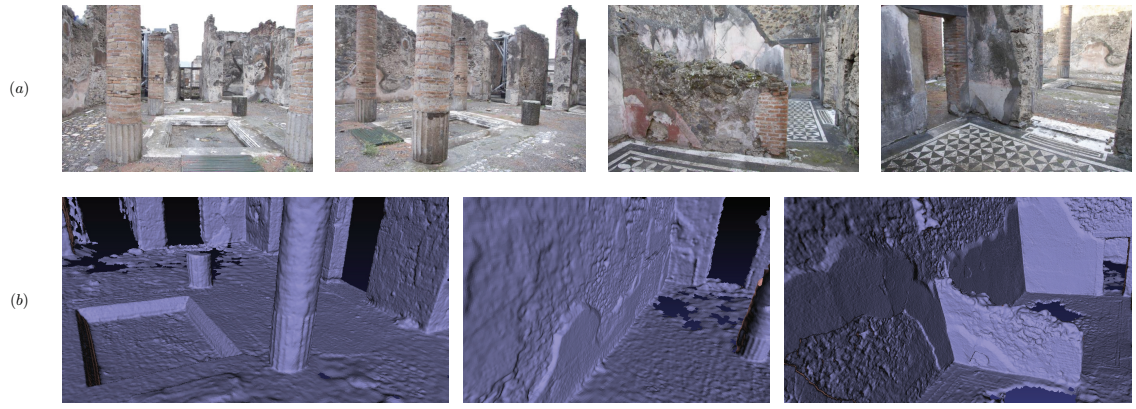


Figure 2. (a) Example photographs captured of the Pompeii site (563 photographs are used in total). (b) Rendered viewpoints of the recovered 3D model. Notice the fine-level details that are captured by the model.

2. Overall approach

Given a painting $\mathcal{I}$ and a set of photographs $\mathcal{J}$ depicting a 3D scene, our goal is to automatically recover a model of the 3D scene, along with parameters describing the viewpoint of the painting with respect to the recovered model. We represent the model of the 3D scene by a triangular mesh $\mathcal{M}$ and assume that points in the 3D scene are rendered onto the painting by perspective projection. In other words, we treat the painting as if it were an image, with the desired goal of recovering the standard "camera" parameters Θ that describe the painting's internal calibration and viewpoint with respect to $\mathcal{M}$. The parameters that we seek to recover are the camera center, rotation, focal length, and principal point (we assume zero-skew and square pixels). Our algorithm is summarized in Figure 1.

To start, our aim is to build a high-quality triangular mesh of the 3D scene from the input set of photographs. For this, we use existing algorithms that have shown much success on a variety of real-world scenes. We start by running Bundler [29] on the input set of photographs. Bundler matches features extracted at interest points to recover a sparse 3D point set, along with camera parameters for each photograph. The recovered camera parameters are then used as input for the PMVS algorithm [9], which uses photometric matching and region growing to recover a dense 3D point set. Finally, the dense 3D point set is used as input for the Poisson surface reconstruction algorithm [15], which solves a spatial Poisson problem to recover a triangular mesh. To perform these steps, we use existing public-domain software. An example triangular mesh recovered from a set of photographs is shown in Figure 2 (more details about the photographs and mesh are given in Section 5).

Given the triangular mesh $\mathcal{M}$ extracted from the photographs, we wish to recover the parameters Θ of the painting with respect to $\mathcal{M}$. We begin by coarsely aligning the painting to the 3D model by matching to virtual viewpoints that are uniformly sampled across the 3D model [14]. In this way, we get a set of initial guesses of the painting viewpoint. Next, we use the initial viewpoints to validate and more accurately align the painting to the 3D model. We discuss in more detail these steps in the following sections.

3. Coarse alignment by view-sensitive retrieval

Our aim is to find a viewpoint that is sufficiently close to the painting viewpoint, where the depicted scene objects in the painting are close to their 3D model projection. We wish to build upon recent successes in view-sensitive matching in large photo collections [17] and matching to synthetic views of 3D models [14]. This will allow us to be sensitive to different viewpoints, while reasoning about unseen views and generalizing across large appearance variations.

We generate virtual camera matrices that uniformly sample viewpoints of the 3D model. To densely sample viewpoints from the model, we assume that the paintings are drawn upright at eye-level. Hence, we limit our virtual viewpoints to live in a plane at approximately eye-level looking in a direction parallel to the ground plane. To find the ground plane, we search for the dominant scene plane using the Hough transform [7] on the PMVS point set. We determine the eye-level height by finding the average distance of the camera centers for the input photographs to the recovered ground plane. We sample camera centers in a grid and use 12 horizontal orientations at each sample point. The grid spacing is determined empirically (4% of the scene width) to ensure reasonable scene coverage. For the camera intrinsic parameters, we assume zero-skew square pixels, with the principal point in the center of the image and use the average focal length of the recovered cameras.

Next, we render each virtual viewpoint using the set of PMVS points. Figure 3 shows example rendered virtual viewpoints of the 3D model. We discard viewpoints in which the 3D model projects to less than 25% of the pixels in the rendered viewpoint.

Given the large pool of virtual viewpoints, we wish to retrieve a small set of candidate nearby viewpoints to the

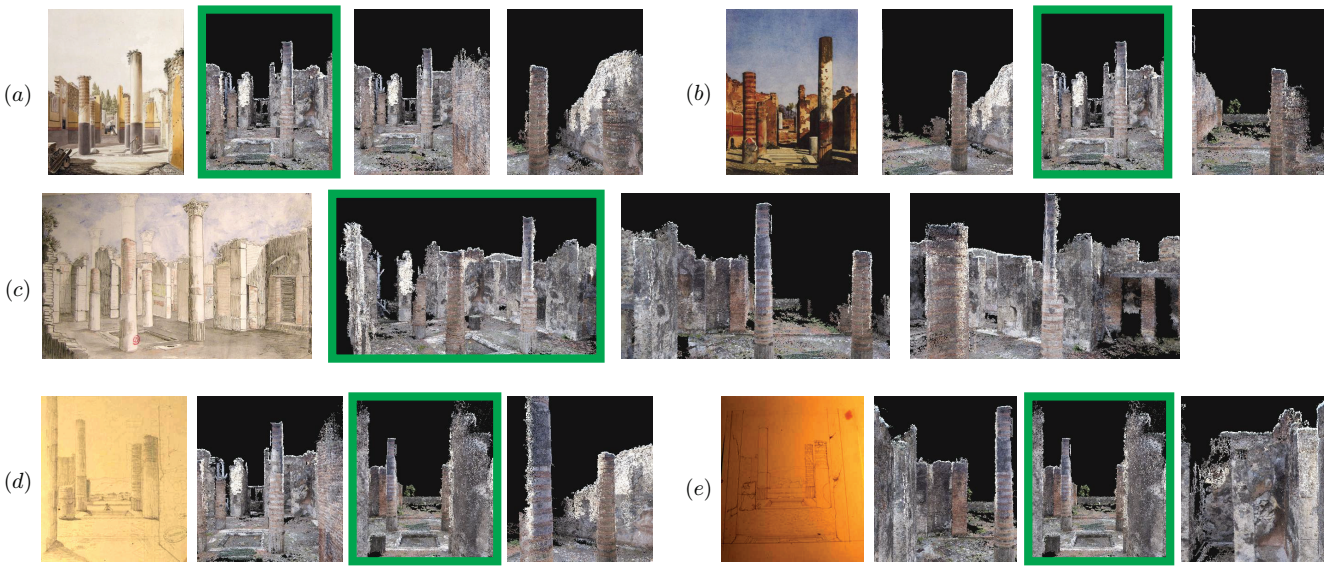


Figure 3. **Coarse alignment by view-sensitive retrieval.** For each painting (left), the top three nearest neighbor virtual viewpoints generated from the 3D model using gist feature matching [23] are shown. The correctly retrieved viewpoint is highlighted in green. Notice that the correct viewpoints are retrieved in the top nearest neighbors in all cases.

painting. For this, we match the appearances of the painting and virtual viewpoints using the gist descriptor [23]. The gist descriptor has been used for recognizing scenes despite significant changes in appearance and has been shown to work well in retrieving similar viewpoints of the same 3D scene [17]. We find the gist descriptor to provide a good degree of robustness to the large changes of appearance between the painting and the rendered viewpoints of the 3D model, while being sensitive to viewpoint. Also, the descriptor is robust to the noisy rendering of the 3D scene, which contains holes and incomplete structures. We normalize the descriptors to be unit length and retrieve the set of nearest neighbor viewpoints that minimize L2 distance. Figure 3 shows example retrievals. In all cases the correct viewpoint is retrieved within the top 3 nearest neighbors.

4. Fine alignment by matching view-dependent contours

Given the retrieved coarsely matching viewpoint, we wish to refine the viewpoint and align the painting to the 3D model. For this, we need to find correspondences between the 3D model and painting. We find that standard local features, such as SIFT, are not suitable for establishing reliable correspondences between paintings and photographs due to extreme appearance changes. Instead, we establish correspondences by matching contours that are extracted from the painting and the 3D model. As we have a 3D model of the scene, we can extract reliable view-dependent contours that are informed by the shape of the 3D surface. With the established correspondences, we can recover the camera matrix for the painting via camera re-sectioning [11].

Contours for 3D model and painting. Given the difficulty in matching the appearance of paintings, we wish to extract features from the 3D model that can be reliably matched to the painting. For this, we extract contours corresponding to folds, creases, and occlusions from the 3D model, as these are often depicted in the paintings.

To recover folds and creases, we use the ridges and valleys algorithm of Ohtake et al. [22]. This algorithm draws lines at places with high surface curvature. For this, we use publicly available software [6], which also extracts occlusion contours. We perform smoothing on the surface normals to remove spurious contours, which result from fine undulations in the 3D model. Example extracted line drawings are shown in Figure 4. Notice that the lines follow closely the scene folds, creases, and occlusions.

To match the contours extracted from the 3D model, we find edges in the painting. For this, we find edges in the painting using the global probability of boundary (gPB) detector [20]. Each gPB edge point $\mathbf{x}$ has a response strength and edge orientation ϕ. The edge orientation denotes the edge direction at the given point, with the orientations quantized into 8 directions (between 0 and π). As a preprocessing step, we perform adaptive histogram equalization in Lab color space on the original painting. We threshold the gPB response strength (we use a threshold of 0.05) to obtain binary images $B_{\mathcal{I}}(\mathbf{x}, \phi)$ for each edge orientation. We perform similar thresholding for the contours extracted from the 3D model for the nearby virtual viewpoint Θ and measure the response of a second derivative operator tuned to the 8 edge orientations to obtain binary images $B_{\mathcal{M}}(\mathbf{x}, \phi, \Theta)$.

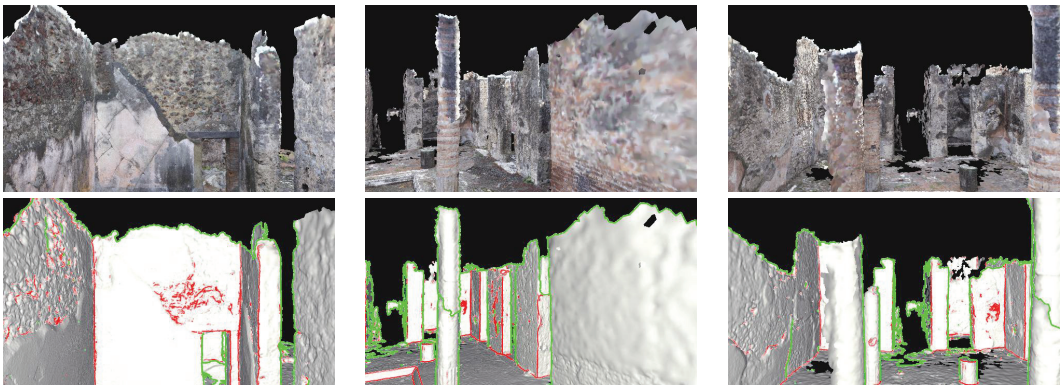


Figure 4. Lines for creases, folds, and occlusions extracted from the 3D model. Top row: rendered viewpoints. Bottom row: lines corresponding to ridges and valleys [22] (red) and occlusion boundaries (green). Notice that the drawn lines follow closely the creases, folds, and occlusions in the 3D scene.

ICP-like fine alignment. We seek to align the binary images over edge orientations for the painting and 3D model. Let $S_{\mathcal{I}}(\phi) = \{\mathbf{x} \mid B_{\mathcal{I}}(\mathbf{x}, \phi) = 1\}$ be the set of edge points for orientation ϕ in the binary image for the painting. Particular camera parameters Θ define a similar set of edge points $S_{\mathcal{M}}(\phi, \Theta)$ for the 3D model. We wish to find the viewpoint with camera parameters $\hat{\Theta}$ minimizing the following cost function

$$\min_{\Theta} \sum_{\phi} \sum_{\mathbf{x}_i \in S_{\mathcal{I}}(\phi)} \min_{\mathbf{x}_j \in S_{\mathcal{M}}(\phi,\Theta)} \min\left(|\mathbf{x}_i - \mathbf{x}_j|^2, \gamma\right) \quad (1)$$

where γ is an inlier threshold. This is similar to the truncated reprojection error of 3D points along folds, creases, and occlusions from the 3D model to the edge points in the painting. The truncation allows robustness to outlier correspondences. We set the inlier threshold γ to be 0.5% of the painting diagonal.

To solve the optimization problem of Equation (1), we begin by searching for putative correspondences between the oriented edge points for the painting and 3D model. For this, we use the shape context descriptor [3], which is a representation used to describe the shape about a given location and has shown success in aligning binary image patterns and for digit recognition.

The shape context descriptor is formed by spatially accumulating sampled edge points about a given location. We sample 1000 edge points from the painting and 3D model, which empirically provides sufficient coverage of the scene. Spatial log-polar bins are formed around each sampled point, and sample points landing in each spatial bin are accumulated. In addition, we use the edge orientation information and accumulate points having the same edge orientation. In total, we use 3 radial bins (the minimal and maximal bin sizes are 7% and 15% of the painting diagonal), 12 angular bins, and 8 edge orientations, which results in a 288 dimensional feature vector. The feature vector is normalized to sum to one.

Given shape context features f_i, f_j and edge orientations ϕ_i, ϕ_j for points belonging to the painting and the 3D model, respectively, we use the following match score:

$$R_{i,j} = (1 - \eta)\, \mathcal{X}(f_i, f_j) + \eta\, \mathcal{E}(\phi_i, \phi_j) \quad (2)$$

where $\mathcal{X}(f_i, f_j)$ is the Chi-squared distance and $\mathcal{E}(\phi_i, \phi_j)$ is the unsigned difference in edge orientation. We set $\eta = 0.1$ to the default setting used in [3]. A putative correspondence is formed for each sampled point in the 3D model by matching it with the best scoring sample point in the painting. For each putative correspondence $(\mathbf{x}_i, \mathbf{x}_j)$, sampled points belonging to the 3D model are back-projected to retrieve the corresponding 3D point $\mathbf{X}_j$. The putative correspondences $(\mathbf{x}_i, \mathbf{X}_j)$ are used for camera resectioning.

Given the putative correspondences, we use RANSAC to recover the camera parameters Θ and to find inlier correspondences. To speed-up RANSAC, we use a simpler model (known intrinsics) within the RANSAC loop, followed by iterative re-estimation using the full camera model [24]. In particular, we assume reasonable values for the intrinsic camera parameters (square pixels, focal length equal to image width and principal point at the image center) and recover only the extrinsic parameters corresponding to the camera rotation and center inside the RANSAC loop. This results in six parameters, which are computed using three putative correspondences at each RANSAC iteration (note that there are up to four solutions, which must all be validated). We run RANSAC for 1000 iterations.

The camera parameters $\hat{\Theta}$ that minimize the cost function given by Equation (1) when projecting the 3D points corresponding to the points in $S_{\mathcal{M}}(\phi, \Theta)$ are returned. In other words, the sparse set of shape context points propose camera parameters, which are then validated (scored) with the dense set of 3D points along the scene creases, folds, and occlusions. The dense set of painting edge points that are inliers (i.e. lie within γ distance to a projected 3D model point), along with the closest 3D model points, are used to

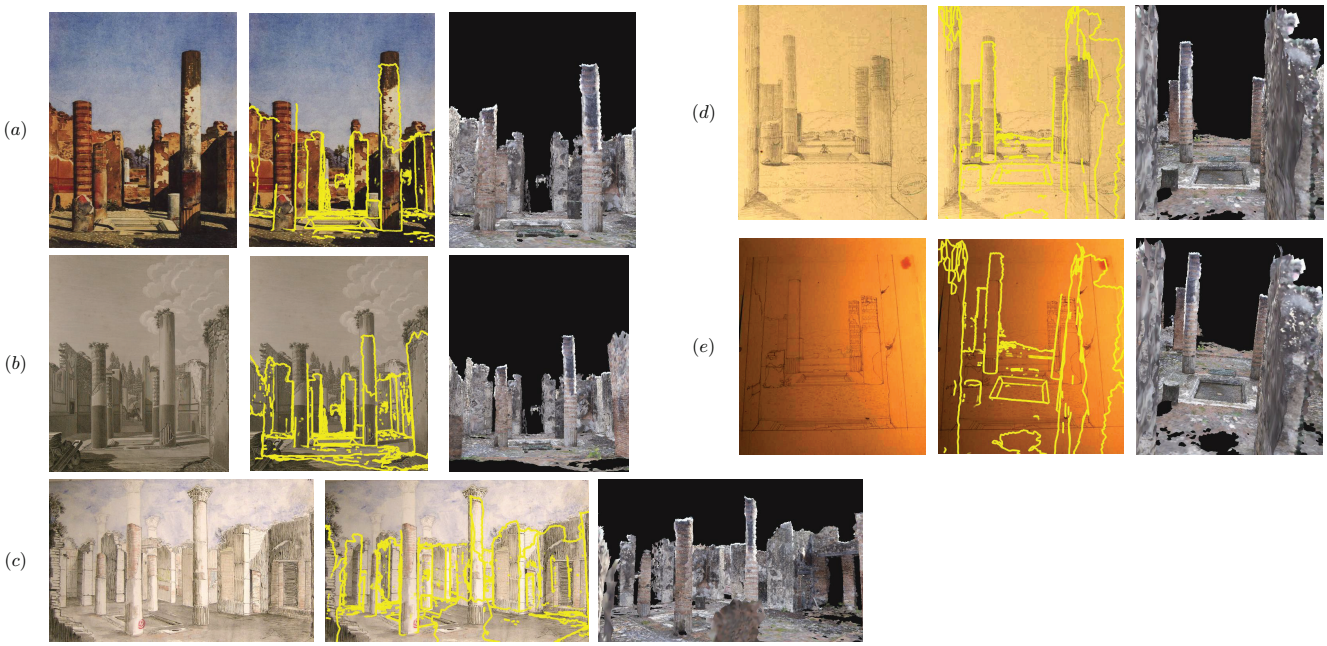


Figure 5. Final alignment between the paintings and 3D model. For each example, left: painting; middle: 3D model contours projected onto painting; right: synthesized viewpoint from 3D model using recovered camera parameters Θ. For the examples in (a-c), note how the final alignment is close to the painting. Our system handles paintings that depict the 3D structure of the scene over time and span different artistic styles and mediums (e.g. water colors, cross-hatching, copies of originals on engravings). Notice how the site changes over time, with significant structural changes (e.g. the wall murals decay over time, the columns change). Example failure cases are shown in (d,e).

minimize geometric reprojection error over all of the camera parameters (intrinsic and extrinsic).

Given the camera parameters $\hat{\Theta}$, a new viewpoint is generated and corresponding new set of contours are extracted from the 3D model. The new set of contours are used to iteratively optimize Equation (1). At iteration i, we limit the search for shape context correspondences to be within $\epsilon = \frac{\epsilon_0}{2^i}$ pixels, with ϵ_0 set to be 20% of the painting diagonal. The entire process is stopped when $\epsilon \leq 1$, and the best scoring set of camera parameters are returned.

5. Experimental results

We have tested our system on paintings of the Casa di Championnet, which is located amongst the ancient ruins of the Roman town of Pompeii (VIII 2, 1). The city was buried during a volcanic eruption in 79 AD, rediscovered in 1599, and was first excavated in 1748. The Casa di Championnet was excavated in 1798. For our study, we have gathered 9 paintings and drawings depicting the 3D structure of the site[1]. The paintings were rendered over different periods and provide a glimpse of the site as it changed over time. Example paintings are shown in Figures 3 and 5. We focus on paintings where the artist had intended to accurately capture the 3D scene structure. In some cases, it is believed that a camera lucida was used to assist the artist. The paintings were manually gathered from archaeological archives, which required an expert to correctly identify that the depictions are of the site of interest. Notice that different styles are represented, ranging from watercolors to cross-hatching. Furthermore, drawings and watercolors were used to produce engravings in the XIXth century publications, with strong rendering differences.

To recover the dense 3D model, we use 563 photographs (4752×3164 resolution) that were captured of the Pompeii site over two days during sunrise and sunset (to avoid strong cast-shadows). Figure 2(a) shows example photographs of the site. The final mesh contains 10M vertices and 20M triangles. A snapshot of the mesh is seen in Figure 2(b).

Coarse alignment retrieval results. To coarsely align the painting to the 3D model, we synthesize 16,548 virtual viewpoints and discard viewpoints that back-project to less than 25% of the pixels in the rendered viewpoint. This results in 8,379 virtual viewpoints, which are used to retrieve a similar viewpoint to the input painting. We show results for coarse alignment retrieval in Figure 3 for different paintings in our dataset.

For validation, we obtain correspondences between each painting and the 3D model by hand-clicking on average 19 correspondences at key points in the scene. We use the correspondences to obtain ground-truth camera matrices for the

[1]The paintings and drawings were collected from publications found in European archives and libraries: Bibliothèque Nationale de France, Ecole nationale supérieure des Beaux Arts, Bibliothèque de l'Institut National d'Histoire de l'Art, Museo archeologico nazionale di Napoli, National Museum of Stockolm.

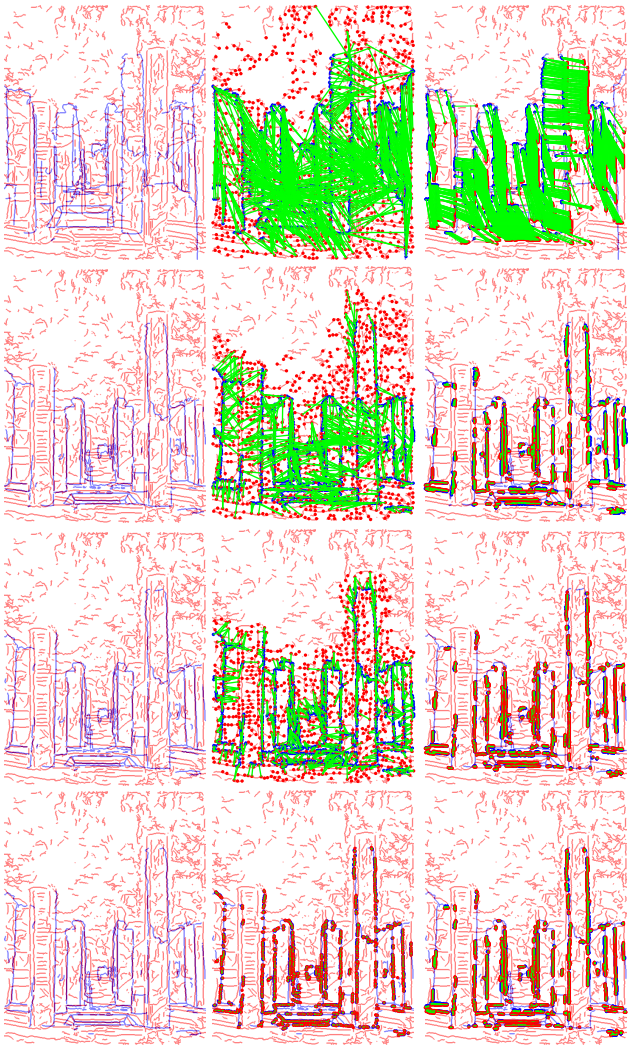
paintings via camera resectioning by minimizing geometric error [11]. To declare a virtual viewpoint as correct, we look at the reprojection error when projecting the 3D points lying in the painting onto the virtual viewpoint. Correct viewpoints are highlighted in green in Figure 3. For all paintings, the correct viewpoint is retrieved in the top 3 nearest neighbors. This is noteworthy given the size of the set of virtual viewpoints. In all cases, the correctly retrieved viewpoint is close to the input painting and forms a good initialization for fine alignment.

Fine alignment results. Given the retrieved view-sensitive coarse alignment, we run the ICP-like fine alignment procedure five times and return the output with lowest cost in Equation (1). In Figure 5, we show the final alignment of paintings in our dataset to the 3D model. We show the input painting, extracted contours from the 3D model for the final viewpoint overlaid on the painting, and a rendering of the 3D model for the final viewpoint.

For the paintings in Figure 5(a-c), the output viewpoint is close to the depicted viewpoint and the rendered 3D model contours mostly follow the contours depicted in the painting. Moreover, our system is able to successfully align a variety of different painting styles. Figure 6 shows snapshots of the fine alignment procedure at different iterations. Example failure cases of the fine alignment procedure are shown in Figures 5(d,e). Many of these failures are due to unreliable features extracted from the painting. In particular, pencil strokes indicating shading is difficult for our system. However, note that for the failure cases, a close-by viewpoint is successfully retrieved, as shown in Figure 3.

In Table 1 we quantify the fine alignment procedure for each painting by measuring the average reprojection error (both in pixels and as a percentage of the length of the painting diagonal) using the set of hand-clicked correspondences. The fine alignment procedure successfully finds a viewpoint with average reprojection error within 3% of the painting diagonal for 5 out of 9 paintings in our dataset, which corresponds visually to a nearby viewpoint.

In assessing the quality of the final alignment, one difficulty is in determining the exact source of alignment error. Some 3D model contours do not tightly snap to the painting contours (e.g. the top of the left-column in the painting in Figure 5(a)). These errors are often due to the alignment procedure and features. However, for some depicted features in the painting, the error could be due to drawing error. For example, the columns in Figure 5(b) are depicted closer to the edge of the central pool than in the reconstructed 3D model. This is also suggested by the reprojection error for the ground-truth camera matrix, as shown in Table 1. The set of paintings that were rendered from the viewpoint depicted in Figure 5(b) consistently yield higher ground-truth reprojection error (normalized errors well above 1%). For the scene in Figure 5(c), the paintings do not all appear to be painted with the same accuracy, as determined by the reprojection errors of the ground-truth matrices.



(a) (b) (c)

Figure 6. **Fine alignment by matching view-dependent contours.** (a) Extracted edges (painting - red; 3D model - blue). (b) Shape context sample points and putative correspondences (green lines). (c) Dense edge inlier correspondences found by RANSAC. After each iteration, the viewpoint is updated. Notice that there are many inliers in the putative set and that the 3D model gets closer to the input painting with each iteration. The matching converges after seven iterations, with iterations 1, 2, 3, and 7 shown.

6. Conclusion

We have shown successful alignment of historical paintings depicting the Casa di Championnet in Pompeii to a 3D model constructed from modern photographs of the site. To achieve this we have developed a two-stage alignment procedure. We find an approximate viewpoint by retrieval and then refine the viewpoint by matching to view-dependent contours extracted from the 3D model. Renderings of the

	Fig. 5(a)	Fig. 5(b)			Fig. 5(c)			Fig. 5(d)	Fig. 5(e)	Mean
Res.	476×600	456×550	474×578	459×550	640×393	640×388	640×390	480×547	475×550	
GT	3.60 (0.47)	7.88 (1.10)	8.16 (1.09)	9.79 (1.37)	3.60 (0.48)	6.82 (0.91)	6.31 (0.84)	2.52 (0.35)	4.05 (0.56)	5.86 (0.80)
Alg.	5.65 (0.74)	12.06 (1.69)	17.80 (2.38)	18.25 (2.55)	10.51 (1.40)	128.92 (17.23)	112.52 (15.01)	36.95 (5.08)	46.03 (6.33)	43.19 (5.82)

Table 1. Reprojection error for 19 hand-clicked correspondences between the paintings and 3D model. Res: Painting resolution in pixels. GT: Error for ground-truth camera matrix computed from hand-clicked correspondences by minimizing geometric error. Alg: Error for algorithm output. The reprojection error is measured in pixels. In parenthesis, the error is given as a percentage of the length of the painting diagonal. The scenes depicted in Figures 5(b,c) have additional paintings (not shown in Figure 5) that are rendered in different styles (the first sub-columns correspond to the displayed paintings in Figure 5). Notice that some paintings have higher error for the ground-truth camera matrix. This provides an indication of the difficulty of producing an accurate alignment, which may be due to drawing errors.

recovered viewpoint are close to the viewpoint depicted in the paintings, with contours extracted from the 3D model following closely the depicted structures in the paintings. As demonstrated, our method copes with challenges in appearance that paintings provide, such as color and texture. Such challenges are difficult for current systems relying on local feature matching.

While collecting the 3D model was relatively easy, collecting the paintings that were used was challenging, as it required an expert to correctly identify from archaeological archives paintings rendered with the intention of accurately depicting the site of interest. We believe that scaling up to additional sites is an interesting future direction that may require successful large-scale painting retrieval [28].

Acknowledgments. We thank Soprintendenza archeologica di Napoli e Pompei, for access to the site and the possibility of study; Margareta Staub Gierow (Archäologisches Institut, Albert-Ludwigs-Universität, Freiburg), for her collaboration in collecting archives in the National Museum of Stockolm. Supported in part by the MSR-INRIA joint laboratory, the ANR project DETECT (ANR-09-JCJC-0027-01) and the EIT ICT labs (activity 10863).

References

[1] L. Baboud, M. Cadik, E. Eisemann, and H.-P. Seidel. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *CVPR*, 2011. 2

[2] S. Bae, A. Agarwala, and F. Durand. Computational rephotography. *ACM Trans. Graph.*, 29(3), 2010. 1

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(4):509–522, 2002. 2, 5

[4] P. J. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE PAMI*, 14(2):239–256, 1992. 2

[5] P. Cavanagh. The artist as neuroscientist. *Nature*, 434:301–307, 2005. 1

[6] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *SIGGRAPH*, 22(3):848–855, 2003. 4

[7] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15:11–15, 1972. 3

[8] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a cloudless day. In *ECCV*, 2010. 1

[9] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE PAMI*, 32(8), 2010. 1, 2, 3

[10] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007. 1

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4, 7

[12] K. Ikeuchi, A. Nakazawa, K. Hasegawa, and T. Oishi. Digital presentation and restoration of cultural heritage through computer vision techniques. In *International Conference on Artificial Reality and Telexistence*, 2003. 1

[13] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *ICCV*, 1998. 2

[14] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 1, 2, 3

[15] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 2006. 2, 3

[16] R. Langenbach. Outside of the frame: Piranesi's perspective and composition, re-explored in the digital age. In *ICOMOS*, 2008. 2

[17] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, 2008. 2, 3, 4

[18] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010. 2

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[20] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008. 2, 4

[21] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. V. Gool. Procedural modeling of buildings. *ACM Trans. Graph.*, 25(3):614–623, 2006. 2

[22] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23(3):609–612, 2004. 2, 4, 5

[23] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 1, 2, 4

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 1, 5

[25] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 66(3):231–259, 2006. 2

[26] G. Schindler, F. Dellaert, and S. Kang. Inferring temporal order of images from 3d structure. In *CVPR*, 2007. 2

[27] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 2

[28] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. In *SIGGRAPH ASIA*, 2011. 2, 8

[29] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *IJCV*, 80(2):189–210, 2008. 1, 2, 3

[30] P. A. Viola and W. M. W. III. Alignment by maximization of mutual information. *IJCV*, 24(2):137–154, 1997. 2

[31] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE PAMI*, 29(11):1973–1989, 2007. 2