Registering Images to Untextured Geometry using Average Shading Gradients

Tobias Plötz Stefan Roth
Department of Computer Science, TU Darmstadt

Abstract

Many existing approaches for image-to-geometry registration assume that either a textured 3D model or a good initial guess of the 3D pose is available to bootstrap the registration process. In this paper we consider the registration of photographs to 3D models even when no texture information is available. This is very challenging as we cannot rely on texture gradients, and even shading gradients are hard to estimate since the lighting conditions are unknown. To that end, we propose average shading gradients*, a rendering technique that estimates the average gradient magnitude over all lighting directions under Lambertian shading. We use this gradient representation as the building block of a registration pipeline based on matching sparse features. To cope with inevitable false matches due to the missing texture information and to increase robustness, the pose of the 3D model is estimated in two stages. Coarse pose hypotheses are first obtained from a single correct match each, subsequently refined using SIFT flow, and finally verified. We apply our algorithm to registering images of real-world objects to untextured 3D meshes of limited accuracy.*

1. Introduction

Registering images to 3D models of real-world objects or places is an important prerequisite for transferring information between images and a 3D model of the scene [6, 26]. For example, color information from images can be used to texture a 3D model that was previously acquired using range scans. More broadly speaking, the 2D image may provide diverse information that can be used to annotate, or possibly even update [24], the 3D model. Going in the opposite direction, it is possible to annotate images with information from the corresponding part of the 3D scene, once we know the camera pose from which the image was taken.

In this paper, we introduce a method for registering individual photographs to 3D models even in the absence of any information on the texture of the object. This is in contrast to many existing image-to-geometry registration approaches [14, 16, 17] that rely on pre-registered images to which a newly arriving photograph is aligned through matching of features. Such pre-registered images are available, for example, when the 3D geometry is acquired through multi-view stereo [1]. However, this scenario is not always applicable, *e.g.* when acquiring a 3D model by non-photometric methods, such as range scans. Although some range scanners are able to measure the reflectance of a surface point, this color information is not very reliable and only available if the scanning is performed during daytime. However, it is not unusual that scanning campaigns are required to take place at night; thus we need to work with the raw geometry information only [5].

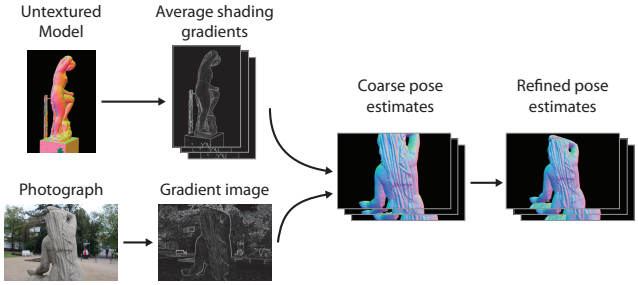


Figure 1. Registration pipeline using average shading gradients.

Our method estimates the pose of the depicted 3D model by searching for sparse correspondences between features found on the photograph and image features found on renderings of the 3D model. Existing methods, in contrast, typically aim to maximize the statistical dependency between the photograph and a rendering [6]. The resulting registration criterion is dense, but leads to a highly non-convex optimization problem with many local optima, necessitating good initialization. Therefore, dense registration methods are by and large bootstrapped with user interaction or some other prior information on the camera pose. While this may be suitable for smaller scanning campaigns, this does not scale to registering a continuous incoming stream of images to a geometric model of the scene. Our work is complementary to these dense methods in that it automatically provides registration hypotheses, which can be further refined, if needed, without requiring user interaction.

Gradients are the most common building block for many image features, *e.g.* [7, 23]. Since we cannot hope to recover the texture gradients in renderings of the 3D model, we need

to rely on gradients due to the shading of the object, if we aim to use well-proven image features for describing image patches. In absence of prior information on the lighting and reflectance properties of the object, we assume a simple, yet effective, Lambertian shading model with a single point light source, and estimate the observable gradient magnitude averaged over all directions of the point light. This *average shading gradient* directly relates to the magnitude of standard image gradients that are computed with the same linear operator, yet neither requires a known lighting direction nor any ad-hoc assumptions about it. Bringing both rendering and photograph into a gradient representation allows us to establish sparse 2D-to-3D correspondences.

However, in the absence of texture, the ratio of correct correspondences tends to be lower than when matching images. To cope with this, we estimate the camera pose in two stages. First, coarse poses are generated from just a single correspondence each. To that end we render patches from randomly sampled viewpoints around Harris3D keypoints [31] and match them to the image. The coarse pose is obtained by estimating an affine transformation between image and matching rendering. This initial estimate is refined in a second step that iteratively improves the camera pose using SIFT flow [20] on the gradient representation. While registration does not always succeed due to the difficulty of the problem, a final automatic verification step can predict reliably whether the registration was successful.

The contributions of this paper are as follows: *(1)* We present average shading gradients, a novel way of computing a gradient representation from renderings of an untextured 3D model in the absence of any lighting information. The representation directly relates to gradients found on real images. *(2)* We introduce a method for generating coarse image-to-geometry registration estimates from just a single correct patch correspondence. Compared to other work in image-to-geometry registration [2, 14, 28], we are not restricted to specific (*e.g.*, ground-level) viewpoints. *(3)* We propose an iterative pose refinement technique based on SIFT flow that substantially increases the registration accuracy. *(4)* To make our pipeline fully automatic, we suggest a verification step that accurately predicts whether the registration has succeeded. Our experiments show that average shading gradients coincide well with gradient information of corresponding images and robustly cope with "noisy" geometry. Moreover, we demonstrate the efficacy of our entire pipeline on 3D meshes of varying complexity and accuracy.

2. Related Work

The idea of using *rendered lines* for aligning 3D objects has a long history in computer vision [22] and is used in object-level pose estimation [18, 32, 35], image-to-geometry registration [28], sketch-based shape retrieval [10] and photo-to-terrain alignment [3]. In addition to simple line rendering techniques, such as silhouettes, contours, ridges and valleys, more sophisticated and view-dependent methods have been proposed. Suggestive contours [8], for example, are drawn where contour lines would occur if the view direction was altered slightly. Apparent ridges [15] use a notion of view-dependent curvature to compute ridges and valleys. The obtained lines do not necessarily coincide with high principal curvature, but rather with large perceived curvature. Both line rendering techniques are geared to convey shape to human users. In contrast, the average shading gradient proposed here aims at matching the gradients observable from a real image of the 3D object. Our technique is also more robust to noise and fine surface detail, as it is computed in screen space. Incorporating global illumination effects like ambient occlusion [29] into the shading model could further improve the shading gradient.

Feature-based pose estimation matches image features on the photograph to features stored in a database and anchored to 3D points [2, 14, 17]. A pose is typically estimated from these 2D-to-3D correspondences using RANSAC. [14, 16, 17] use previously registered images to derive image features. [34] extends [14] by exploiting temporal coherency in a sequence. In contrast, our work does not require pre-aligned images, but only a 3D model from which we render synthetic views instead. [2, 28] take this approach for aligning paintings to geometry, however assuming that camera poses only occur at ground level, with a fixed set of horizontal and vertical orientations. This limits the applicability when registering photographs from elevated viewpoints. We instead sample camera poses for rendering around key points on the 3D object. Also, while [2, 28] use 3D models with texture information, we address the more general setting of having an untextured 3D model of a real-world object. Our two phase pose estimation strategy is related to [16, 28], which use GIST descriptors [27] for retrieving similar views and thereby also first generate initial pose estimates, which get subsequently refined. In our work, the first phase relies on image patches instead of complete views, allowing for a wider sampling of viewpoints. [21] in contrast relies on global features such as lines that are typically found in urban scenes.

Techniques for *pose refinement* often involve optimizing some measure of alignment between the photograph and a rendering of the model. Most prominent is the seminal work on mutual information alignment [33], which assumes that pixel values are spatially independent, but come from a joint probability distribution over pixel values of photograph and rendering. The objective is to maximize their statistical dependency. This results in a highly non-convex optimization problem, hence good initialization is crucial. The rendering technique itself turns out to be crucial as well. [6], for example, proposed a blending of normal and ambient occlusion maps. This is extended by [9] to render colors in-

duced from other images whenever possible. Other refinement approaches try to align the silhouette lines of the renderings and photograph [26]. However, these approaches typically require the full object to be depicted, whereas our approach is not limited to photos that depict any silhouette line. Note that our approach for generating coarse pose hypotheses complements these refinement algorithms.

3. Average Shading Gradients

To match feature points between renderings of untextured models and photographs, we need to define a suitable representation that allows assessing their similarity. This representation should depend on local image variation that is present in both source modalities. Here, we propose to use gradients from shading, since they are detectable in both photographs and on renderings of the 3D model. In general, the gradient magnitude of an image is defined as

$$\|\nabla I\| = \sqrt{(h_x * I)^2 + (h_y * I)^2}, \tag{1}$$

where I denotes the image, h_x and h_y are derivative filters in x and y direction, and $*$ denotes the convolution operation. All other operations are pixel-wise.

Aside from the 3D geometry and camera pose, the image formation process also depends on the context of the scene (*e.g.*, the background), as well as the lighting conditions and the reflectance model of the 3D surface. Without prior knowledge, we assume the background to be constant and the reflectance model to be Lambertian with constant albedo. For the lighting, we assume a single point light source with unknown lighting direction. Hence, we can express the image I of the 3D model given a certain camera pose in terms of a normal map $\mathbf{n}$ and lighting direction $\mathbf{l}$ as

$$I = \max(0, -\mathbf{n} \cdot \mathbf{l}). \tag{2}$$

Inserting Eq. (2) into Eq. (1) allows to compute gradients on the rendered image. However, the light direction $\mathbf{l}$ is still unknown. Assuming a fixed lighting direction is possible; setting it to coincide with the camera viewing direction ("headlight" assumption), for example, results in a gradient magnitude that is related to suggestive contours [8]. However, for a fixed lighting direction some discontinuities in the normal map will not give rise to gradients. Yet, these discontinuities may be strongly present for other lighting directions. In this paper we thus average the gradient magnitude over all possible light directions of the unit sphere $\mathbf{S}$. Specifically, we propose the *average shading gradient*

$$\overline{\|\nabla I\|} = \int_{\mathbf{S}} \|\nabla I(\mathbf{l})\| \, \mathrm{d}\mathbf{l} \tag{3}$$

$$= \int_{\mathbf{S}} \Big[(h_x * \max(0, -\mathbf{n} \cdot \mathbf{l}))^2 + (h_y * \max(0, -\mathbf{n} \cdot \mathbf{l}))^2 \Big]^{\frac{1}{2}} \mathrm{d}\mathbf{l}. \tag{4}$$

Computing the average gradient magnitude in Eq. (3) in closed form is challenging due to the complex form of the integrand. Hence, we make two approximations to arrive at a more tractable expression. First, we replace $\max(0, -\mathbf{n}\cdot\mathbf{l})$ by $\frac{1}{2}(\mathbf{n} \cdot \mathbf{l})$, since the square of the dot product is symmetric in the light direction and we integrate over all lighting directions. *I.e.*, pixels on the normal map, for which the inner product is positive, will be clipped for the opposite light direction, and vice versa. Only when the stencil of the derivative filter covers an area across which the visibility (*i.e.* the sign of the dot product) changes, this approximation is inexact. However, we found this effect to be negligible in practice (see Fig. 2 and Sec. 5). As a second approximation, we apply Jensen's inequality, which allows deriving a closed form bound as follows:

$$\overline{\|\nabla I\|} \approx \frac{1}{2} \int_{\mathbf{S}} \sqrt{(h_x * (\mathbf{n} \cdot \mathbf{l}))^2 + (h_y * (\mathbf{n} \cdot \mathbf{l}))^2} \, \mathrm{d}\mathbf{l} \tag{5}$$

$$\leq \frac{1}{2} \sqrt{\int_{\mathbf{S}} (h_x * (\mathbf{n} \cdot \mathbf{l}))^2 + (h_y * (\mathbf{n} \cdot \mathbf{l}))^2 \, \mathrm{d}\mathbf{l}}$$

$$= \frac{1}{2} \sqrt{\int_{\mathbf{S}} ((h_x * \mathbf{n}) \cdot \mathbf{l})^2 \, \mathrm{d}\mathbf{l} + \int_{\mathbf{S}} ((h_y * \mathbf{n}) \cdot \mathbf{l})^2 \, \mathrm{d}\mathbf{l}}$$

$$= \sqrt{\frac{\pi}{3}} \sqrt{\sum_{i=1}^{3} (h_x * \mathbf{n}_i)^2 + (h_y * \mathbf{n}_i)^2}. \tag{6}$$

To obtain the last equality, we transform the squared filter response as

$$\hat{\mathbf{x}} = [\mathbf{x}_1^2 \;\; \mathbf{x}_2^2 \;\; \mathbf{x}_3^2 \;\; 2\mathbf{x}_1\mathbf{x}_2 \;\; 2\mathbf{x}_1\mathbf{x}_3 \;\; 2\mathbf{x}_2\mathbf{x}_3], \tag{7}$$

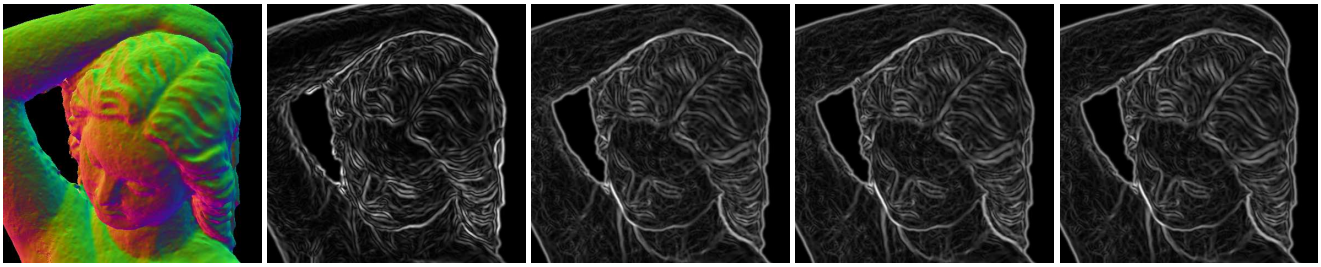
which maps a three-dimensional vector into a six-dimensional space such that $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} = (\mathbf{x} \cdot \mathbf{y})^2$. We obtain

$$\int_{\mathbf{S}} ((h * \mathbf{n}) \cdot \mathbf{l})^2 \, \mathrm{d}\mathbf{l} = \int_{\mathbf{S}} \widehat{(h * \mathbf{n})} \cdot \hat{\mathbf{l}} \, \mathrm{d}\mathbf{l}$$

$$= \widehat{(h * \mathbf{n})} \cdot \int_{\mathbf{S}} \hat{\mathbf{l}} \, \mathrm{d}\mathbf{l} = \frac{4}{3}\pi \sum_{i=1}^{3} (h * \mathbf{n}_i)^2, \tag{8}$$

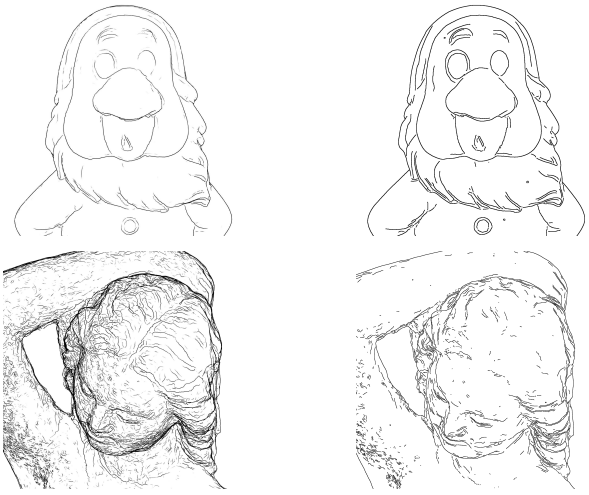
where the $\mathbf{n}_i$ denote the x, y, z components of the normal field. The bound from Eq. (6) is very efficient to compute as it only involves convolutions and pixel-wise operations.

Benefits. Figure 2 shows an example of the gradient magnitudes of a Lambertian shading model for the normal map of a statue. First, we note that averaging over light directions (c, Eq. 3) as proposed here appears superior to making an arbitrary assumption on the lighting direction. When making a "headlight" assumption (b, [8]), *i.e.* the light comes from the viewing direction, certain characteristic structures like the contour of the chin get lost. On the arm of the statue



(a) Normal map (b) "Headlight" assumption (c) Average shading gradient (d) Approximation, Eq. (5) (e) Bound, Eq. (6)

Figure 2. Image gradients for the normal map from (a). From left to right: (b) Gradient magnitude computed with Lambertian shading and "headlight" assumption [8]. Monte Carlo estimate of the average gradient magnitude using the (c) correct (Eq. 3) and (d) approximated (Eq. 5) Lambertian shading. (e) Our closed-form bound (Eq. 6).



(a) Apparent ridges (b) Average shading gradient

Figure 3. Comparison of apparent ridges (a) and our average shading gradients (b), after non-maximum suppression and hysteresis, on a high quality mesh (top) and a noisy mesh (bottom).

it can be seen, moreover, that gradients tend to vanish for surfaces pointing towards the camera in the headlight case, while they are present for our average shading gradient. We also see that the two approximations (d, e) to the exact average shading gradient have little visible impact.

Connection to apparent ridges. Judd *et al*. [15] observed that apparent ridges coincide well with the output of a Canny edge detector on renderings assuming Lambertian shading, averaged over many light configurations. This suggests interpreting our gradient rendering algorithm as a screen space approximation to apparent ridges. We compare both in Fig. 3, after non-maximum suppression and hysteresis, as in a Canny edge detector. On a high quality mesh (top) the obtained lines for both renderings coincide very well, whereas on a mesh with a noisier surface (bottom), especially on slanted parts, apparent ridges produce more spurious lines that are not related to meaningful edges. In Sec. 5 we show the improved noise behavior of our average shading gradients quantitatively. Additionally, our approach can be used with any linear gradient operator and is more efficient as it avoids the costly computation of the view-dependent curvature in object space for each frame.

4. Pose Estimation

To estimate the camera pose of an input image relative to the untextured 3D model, we now match patches of the input image to patches generated from renderings of the 3D model, using gradients as basic building block of the representation. This yields 2D-to-3D point correspondences from which a pose is then estimated. Similar approaches have recently been used for image-to-painting alignment [30], painting-to-geometry registration [2], and location estimation [14, 17]. As matching to untextured models leads to more false correspondences, we divide the registration process into two steps. First, we estimate a coarse pose from just a single correspondence between an image patch and a patch in the database of rendered views of the model. In a second step we refine this pose into a final, full 11 degrees-of-freedom (DOF) pose. Figure 1 illustrates the pipeline.

4.1. Patch database

To populate the database with rendered patches, we randomly sample camera poses from which the model can be rendered. To reduce the space of possible camera poses, we first identify characteristic points on the model that likely give rise to discriminative features in renderings that show this point. Compared to matching entire rendered images [28], this significantly reduces the pose space, since translations do not need to be considered at this stage. We find 100 characteristic points using Harris3D [31], a 3D key point detector for point clouds and meshes. It approximates the local surface around a vertex as a two-dimensional quadratic function, and applies a continuous version of the well-known Harris operator. This yields a score that correlates well with the local curvature around the vertex, favoring corners or spike-like structures.

Specifically, we evaluate the Harris3D score at a ran-

domly chosen subset of all vertices, and use non-maximum suppression in 3D space to yield thinned out key points. For each key point we randomly sample 10 camera poses that show this particular point. To cover a reasonable range of different viewpoints, we sample uniformly across all camera directions from which the surface point is visible; the camera distance is sampled from a log-normal distribution (*i.e*. the distance relative to the mean is Gaussian). Note, that we do not need to estimate a ground plane and we do not introduce a bias toward camera poses that are at a certain height above ground, or have a fixed set of possible viewing angles relative to the 3D object as in previous work [2, 14, 28]. We only assume a photographer's bias to upright pictures; *i.e*. we choose the in-plane rotation such that the up-axis of the model coincides with y-axis of the view.

We then render each view using the average shading gradient from Sec. 3, after which we identify 2D keypoints that we can match to those of the image to be registered. In our experience blob detectors, such as the difference of Gaussians [23], do not lead to stable keypoints. The reason is that photographic images also contain texture gradients not present in the average shading gradient-representation of the 3D model, which can have significant influence on blob localization. In contrast, corners are stable features that can be localized reliably in both the average shading gradient and the gradient image of a query photograph. Note that in both cases we compute gradients using the same linear operator. We detect corner points on multiple scales using a (2D) Harris detector, and extract patches of size 120σ, where σ is the scale of the key point. All extracted patches are resized to 256×256 pixels to gain scale invariance. Finally, we compute a HoG descriptor [7] from the gradient patches. Note that we do not use non-maximum suppression on the gradients, as we found this to deteriorate performance. We use 8×8 blocks with 9 orientation bins, resulting in a 576-dimensional descriptor, which is stored in the database.

4.2. Coarse pose estimation

Given the descriptors from a 2D query image, we search the nearest neighbor within the database. To compare a query descriptor $\mathbf{d}_q$ to a database descriptor $\mathbf{d}_{db}$, we use the similarity score proposed by Aubry *et al*. [2]:

$$s(\mathbf{d}_q, \mathbf{d}_{db}) = (\mathbf{d}_{db} - \boldsymbol{\mu})^T \Sigma^{-1} \mathbf{d}_q. \quad (9)$$

Here, Σ and $\boldsymbol{\mu}$ are the covariance matrix and mean, respectively, over all descriptors in the database. At query time, evaluating $s(\mathbf{d}_q, \mathbf{d}_{db})$ can be done by taking the inner product between $\mathbf{d}_q$ and a transformed set of database descriptors, which can be pre-computed. Eq. (9) can be interpreted as the calibrated classification score of $\mathbf{d}_q$ for a one-*vs*-all classifier that discriminates $\mathbf{d}_{db}$ from all other descriptors using linear discriminant analysis [2]. Like Aubry *et al*. we found that transforming the database descriptors increases the matching quality over the raw descriptors.

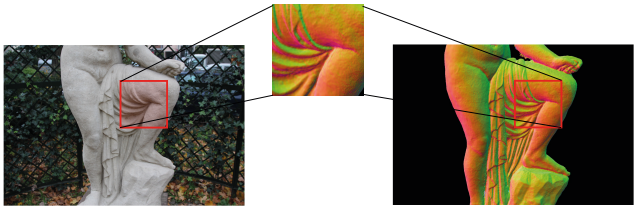


Figure 4. Estimating a camera pose from a single correspondence: The query patch (red box on the left) was matched to a database patch (middle). We generate a coarse estimate of the true camera pose by concatenating the known pose of the database patch with the relative scale and translation of the matching Harris keypoints. This figure shows the photograph and the aligned normal map for better visualization; the matching uses gradient representations.

As we do not rely on textured 3D models, we need to deal with an increased amount of false correspondences in the matching process. For example, on the Statue dataset shown in Fig. 2, on average only 4% of all putative correspondences from nearest neighbors are correct in the sense that the 3D point projects within a distance of 50 pixels to the matched 2D point. A regular RANSAC [11] approach would fail as we need to sample 3 or more correct correspondences to estimate the extrinsic camera pose, *e.g*. using [25], or at least 6 correspondences to estimate the full pose.

To deal with this issue, we first estimate a coarse pose from just a single correspondence, making this viable even for low rates of correct putative correspondences. For every correspondence between an image and a database patch, we compute an affine transformation from the relative position and scale of the Harris keypoints. After applying this transformation to the known pose of the rendered view, the support of the rendered patch is transformed to the support of the patch within the image (see Fig. 4). Note that the admissible poses *relative* to the pose of the rendered view in the database are limited to scaled and translated variants. However, we argue and show in Sec. 5 that this provides a good and efficient initialization for pose refinement.

4.3. Pose refinement and verification

The coarse pose estimates are ranked based on the number of inlier correspondences, *i.e*. those whose 3D point projects within a 50 pixel distance to the 2D point. The 20 top ranked poses are then iteratively refined. We propose to use SIFT flow [20] for computing a dense flow field from the average shading gradient-rendering, given the current camera pose, to the gradient of the query image. The SIFT flow algorithm is similar to optical flow algorithms, but matches dense feature vectors instead of raw intensities. The flow field is estimated by minimizing the L1-norm between warped image features, while simultaneously regularizing the flow spatially and in magnitude (favoring slow

and smooth flows). Since we did not find the refinement to be very sensitive to the choice of image features, we used SIFT as originally proposed [20], as well as the default parameters as provided by the authors' implementation.

The resulting flow field is then used to compute dense 2D-to-3D correspondences. In contrast to the coarse step, we can use RANSAC to estimate a refined pose, as there are now many inliers if the coarse pose was sufficiently close to the true one. In each iteration of the inner RANSAC loop we sample 6 correspondences to estimate both the extrinsic and intrinsic parameters using the direct linear transformation algorithm [13]. Empirically, we found that only few iterations of RANSAC suffice to find a good refinement. We use three iterations of coarse-to-fine estimation: First a downscaled version of both rendering and photograph is used to refine the pose from which a new rendering is created; this is repeated on progressively finer resolutions.

The refined poses on the finest resolution allow for a robust pose verification step to detect whether the registration process was successful. For this we use their mutual reprojection error. Specifically, let $\mathcal{P}$ be a pose that projects a 3D point onto the 2D image plane and $\mathcal{V}$ the set of vertices that are projected inside the image area, *i.e.* visible within the image. Then the mutual reprojection error δ between two poses $\mathcal{P}$ and $\mathcal{P}'$ measures the average 2D Euclidean distance of projected vertices visible in either view:

$$\delta(\mathcal{P},\mathcal{P}') = \frac{1}{2}\left(\frac{1}{|\mathcal{V}|}\sum_{\mathbf{x}\in\mathcal{V}}\|\mathcal{P}(\mathbf{x})-\mathcal{P}'(\mathbf{x})\|_2 + \frac{1}{|\mathcal{V}'|}\sum_{\mathbf{x}\in\mathcal{V}'}\|\mathcal{P}(\mathbf{x})-\mathcal{P}'(\mathbf{x})\|_2\right) \tag{10}$$

We compute the mutual reprojection error for every pair of refined poses and treat them as compatible if the error is below 5% of the longest image dimension. The compatibility relation defines a graph on the refined poses, in which we find the largest connected component $\mathcal{C}$. Finally, our algorithms regards a photograph as correctly registered if $\mathcal{C}$ consists of at least 3 poses. Otherwise, our algorithm rejects the photograph as not registered. The verified poses in the largest connected component constitute the final output of our algorithm and can be further refined by bootstrapping existing dense registration approaches, *e.g.* [5].

5. Experiments

To evaluate our gradient rendering method as well as our approach for image-to-geometry registration, we use three different datasets. The first is a 3D mesh of a *Gnome* along with 9 real images, which were registered using mutual information-based alignment [6] with manual initialization. The mesh is high quality with little noise on the vertex positions and normals. The photographs are taken under controlled conditions and show the gnome figurine on a smooth background and under diffuse illumination. These are favorable conditions for a good registration.

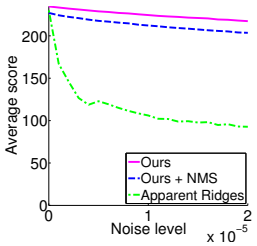


Figure 5. Similarity score (Eq. 9) between descriptors from renderings of a noiseless mesh and of meshes with artificial noise on the vertex positions. Higher scores mean more robustness to noise.

Additionally, we use two real world datasets – *Statue* and *Notre Dame* – acquired from photographs via multi-view stereo reconstruction using the publicly available multi-view environment software package [12]. While this is a convenient way of acquiring 3D models with registered images for evaluation, the models are significantly "noisier" than the Gnome model, posing a greater challenge to our registration algorithm. The *Statue* surface is quite porous but this fine detail is not reflected in the 3D geometry, thus acting like a texture. Many of the images show the 3D mesh on cluttered background and changing light conditions, further contributing to the difficulty of registration. While the photographs from the *Statue* dataset were taken with the intent of reconstructing the geometry, the *Notre Dame* dataset consists of community photos. We emphasize that the images used for evaluation were only used to create the 3D model and not in any part of our pipeline. For testing, we sampled 69 diverse images from Statue, and 70 images from Notre Dame. The query images are resized such that the longest dimension has 1024 pixels.

5.1. Average shading gradients

We first evaluate how well our gradient rendering method matches gradients and edges found on real images. As rendering baseline we use apparent ridges [15], a standard technique for conveying 3D shape via line drawings. To have a fair comparison to apparent ridges which yield thin lines, we show results for our average shading gradient method also after non-maximum suppression (NMS). On the photograph, we compute gradients or detect edges using the gradient operator of the well-known Canny detec-

Table 1. Similarity score between photograph and rendered patches for various combinations of gradient/edge representations.

	Gnome	Statue	Notre Dame
Apparent ridges / Sketch tokens	131.5	53.4	52.5
Apparent ridges / Gradients + NMS	145.8	52.6	46.6
Ours + NMS / Sketch Tokens	110.9	64.6	63.7
Ours + NMS / Gradients + NMS	130.6	70.6	65.2
Ours / Gradients	**159.3**	**82.5**	**72.4**

Table 2. Registration success rate. For each query image only the pose with the most inliers is considered.

	Gnome	Statue	Notre Dame
RANSAC	0.89	0.10	0.46
Shaded (coarse)	0.67	0.13	0.40
Ours (coarse)	1.00	0.43	0.66

tor [4] (Gradients), as well as using sketch tokens [19], a state-of-the-art, learned edge detector.

To measure how well the representations for rendering and photograph match, we compute the descriptor similarity score from Eq. (9) from patches in correct correspondence. Higher scores mean higher similarity. Since the coarse registration algorithm (Sec. 4.2) is based on nearest neighbors in descriptor space, this directly relates to its ability to find a correct image-to-model correspondence. Table 1 shows the results on the three datasets. As can be seen, the highest descriptor similarity is achieved between our average shading gradient-representation of the 3D geometry and gradients extracted on corresponding images. This confirms our intuition that average shading gradients computed from the normal map of an untextured surface are highly correlated to the gradients of photographs. Moreover, our gradient representation clearly outperforms apparent ridges, except after NMS on the easy *Gnome* dataset. Note however, as mentioned before, that NMS generally does not help here.

In a second experiment we analyze the robustness to geometric noise. We take the high-quality *Gnome* model and add increasing amounts of Gaussian noise to each vertex along its normal. As before, we render the meshes from different poses and extract descriptors on the rendering. Figure 5 shows the similarity score (Eq. 9) between descriptors from renderings of the original mesh and from the noisy mesh. The noise level denotes the standard deviation of the Gaussian noise, as a fraction of the object diameter. It can be seen that apparent ridges are sensitive to even small amounts of noise, while average shading gradients degrade gracefully with the noise level.

5.2. Pose estimation

We evaluate our full registration pipeline, with and without refinement, and compare to two baselines. The first baseline replaces the proposed average shading gradients with a simple Lambertian shading under a "headlight" illumination. We, moreover, compare to a standard RANSAC approach that generates poses as follows: The correspondences between 2D feature points on the input photograph and 3D key points on the model form the putative inliers. In each of 5000 iterations of the inner RANSAC loop we sample 4 correspondences and estimate the extrinsic pose (*i.e.* camera rotation and translation) with the efficient PnP algorithm of Moreno-Noguer *et al.* [25]. We then compute the number of consistent inlier correspondences, and finally refit the extrinsic pose on the inliers. The optimistic RANSAC baseline assumes the true intrinsics to be known.

Table 3. True positive and true negative rates of verification step.

	Gnome	Statue	Notre Dame
true positives (TP)	1	1	0.98
true negatives (TN)	1	0.81	0.7

We measure the registration quality by means of the mutual reprojection error (Eq. 10). Table 2 shows the success rate for the RANSAC baseline, for the shading baseline, as well as for the coarse step of our registration pipeline, both considering only the top-ranked hypothesis. We count a coarse registration with $\delta < 150$ as successful, since empirically this is accurate enough for the refinement to improve the pose significantly. Fig. 6 plots the fraction of correctly registered photographs among the top k hypotheses. Recall, that hypotheses are ranked based on the number of inlier 2D-to-3D correspondences. We find that our approach achieves consistently better registration rates than using RANSAC, despite RANSAC assuming known intrinsics. Moreover, average shading gradients significantly outperform registering on a shaded image itself. Nonetheless, since the setting of registering images of an arbitrary viewpoint to untextured geometry is challenging, it is to be expected that coarse registration does not always succeed.

Fortunately, the verification step proposed in Sec. 4.3 is able to identify very reliably when the registration succeeds, as can be seen in Table 3. Note that we observe some false negatives, suggesting that our system errs on the cautious side. These results, moreover, suggest that our approach can be used as a fully automatic registration system. To demonstrate that, we evaluate the statistics of the mean reprojection error among those registrations that are in the set $\mathcal{C}$ of verified poses, obtained by the verification step. For a fair comparison, the error for the coarse poses is evaluated on the set of poses that pass the verification after refinement. After computing the mean reprojection error per image over all verified registrations, we take its median as well as well as the upper and lower quartiles across all accepted images. For RANSAC we report the error of the pose with the most inliers among the correctly registered images.

Table 4. Median mean reprojection error, as well as lower and upper quartiles for images that passed the verification step. For RANSAC only images that can be registered correctly are used.

	Gnome	Statue	Notre Dame
RANSAC	25.3 (14.0 / 46.2)	36.7 (16.7 / 66.2)	39.9 (14.4 / 70.1)
Ours (Coarse)	24.8 (20.8 / 33.2)	33.9 (27.0 / 39.9)	41.2 (32.4 / 61.3)
Shaded (Ref.)	22.8 (22.8 / 22.8)	43.4 (26.8 / 401.7)	10.6 (7.5 / 19.8)
Ours (Ref.)	12.6 (12.1 / 19.9)	6.4 (3.8 / 12.1)	9.1 (6.4 / 14.4)

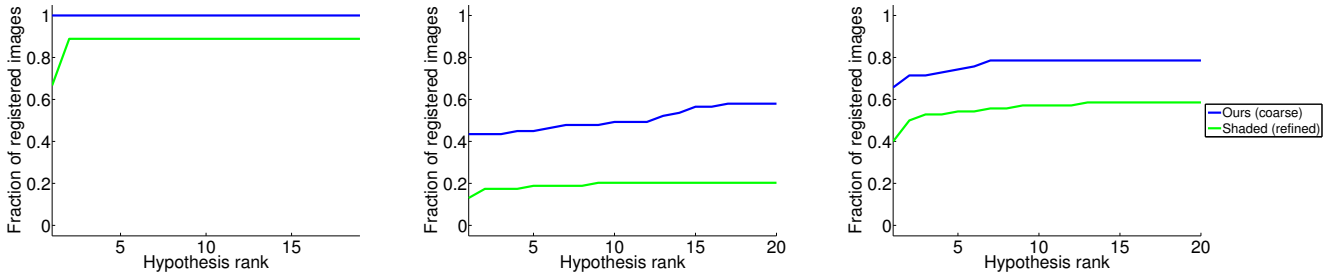


Figure 6. Fraction of correctly registered photographs when considering the first k ranked hypotheses. We compare coarse poses computed with average shading gradients to refined poses computed with Lambertian shaded renderings.

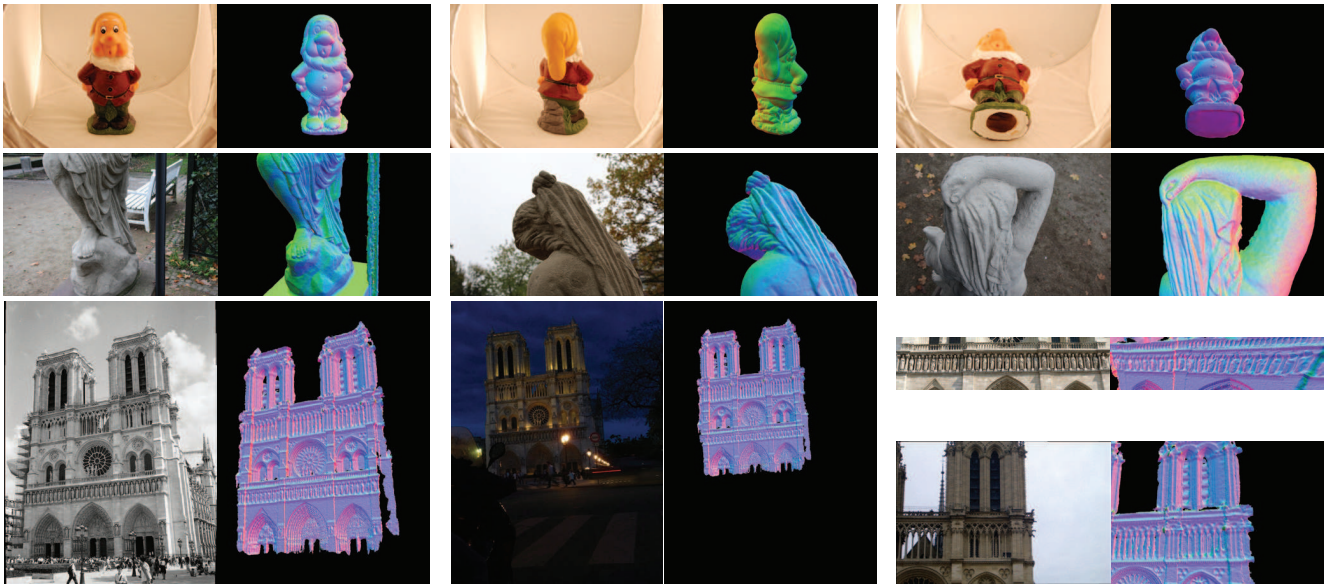


Figure 7. Examples of successful registrations: The query photograph is shown on the left, the top-ranked verified pose on the right.

We make three observations: Already the coarse poses have a clearly lower reprojection error than the RANSAC pose. Moreover, the average shading gradients significantly increase the registration accuracy compared to using Lambertian shading. They show a particularly big benefit on the Gnome and Statue datasets, which do not have a lot of intricate geometric details. Finally, we observe that the proposed refinement step greatly increases the registration accuracy.

Figure 7 shows some examples of successful registrations for the top-ranked verified pose. It can be seen that our system is able to register photographs with a great variety of viewing angles and scales due to putting only few constraints on the sampled camera poses for creating the database. Our system is also able to register photographs on which only parts of the full 3D model are depicted, and successfully copes with different lighting conditions.

6. Conclusion

We presented a novel approach for the challenging problem of registering images to untextured geometry, based on sparse feature matching between the query image and rendered images obtained from the 3D model. Since we cannot rely on textural information for matching, we propose average shading gradients, a rendering technique for the untextured geometry that averages over all lighting directions to cope with the unknown lighting of the query image. As our experiments have shown, average shading gradients coincide well with shading-related gradients in real photographs. Our fully automatic registration pipeline consists of two stages, and is able to accurately register images across a wide range of view points and illumination conditions, without requiring initialization or any other form of manual intervention.

Acknowledgments: This work was supported by the EU FP7 project "Harvest4D" (no. 323567). We want to thank Gianpaolo Palma as well as Michael Goesele's research group for giving access to their datasets.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *CVPR 2009*, pages 72–79.

[2] M. Aubry, B. C. Russell, and J. Sivic. Painting-to-3D model alignment via discriminative visual elements. *ACM T. Graphics*, 33(2):14, Mar. 2014.

[3] L. Baboud, M. Čadík, E. Eisemann, and H.-P. Seidel. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *CVPR 2011*, pages 41–48.

[4] J. Canny. A computational approach to edge detection. *IEEE T. Pattern Anal. Mach. Intell.*, 8(6):679–698, Nov. 1986.

[5] M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, A. Fusiello, and R. Scopigno. Fully automatic registration of image sets on approximate geometry. *Int. J. Comput. Vision*, 102(1–3):91–111, Aug. 2012.

[6] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno. Image-to-geometry registration: A mutual information method exploiting illumination-related geometric properties. *Comput. Graph. Forum*, 28(7):1755–1764, Oct. 2009.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*, pages 886–893.

[8] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM T. Graphics*, 22(3):848–855, 2003.

[9] M. Dellepiane and R. Scopigno. Global refinement of image-to-geometry registration for color projection. In *Digital Heritage 2013*, pages 39–46.

[10] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *ACM T. Graphics*, 31(4):31, July 2012.

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[12] S. Fuhrmann, F. Langguth, and M. Goesele. MVE - A multi-view reconstruction environment. In *Graphics and Cultural Heritage*, 2014.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[14] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR 2009*, pages 2599–2606.

[15] T. Judd, F. Durand, and E. Adelson. Apparent ridges for line drawing. *ACM T. Graphics*, 26(3):19, 2007.

[16] 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*, volume 1, pages 427–440.

[17] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV 2010*, volume 2, pages 791–804.

[18] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing IKEA objects: Fine pose estimation. In *ICCV 2013*, pages 2992–2999.

[19] J. J. Lim, C. L. Zitnick, and P. Dollar. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR 2013*, pages 3158–3165.

[20] C. Liu, J. Yuen, J. Sivic, and A. Torralba. SIFT flow: Dense correspondence across different scenes. In *ECCV 2008*, volume 3, pages 1–17.

[21] L. Liu and I. Stamos. A systematic approach for 2D-image to 3D-range registration in urban environments. *Comput. Vis. Image Und.*, 116(1):25–37, 2012.

[22] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE T. Pattern Anal. Mach. Intell.*, 13:441–450, 1991.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[24] K. Matzen and N. Snavely. Scene chronology. In *ECCV 2014*, volume 7, pages 615–630.

[25] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative O(n) solution to the PnP problem. In *ICCV 2007*, pages 1–8.

[26] P. J. Neugebauer and K. Klein. Texturing 3D models of real world objects from multiple unregistered photographic views. *Comput. Graph. Forum*, 18(3):245–256, Sept. 1999.

[27] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, Feb. 2001.

[28] B. C. Russell, J. Sivic, J. Ponce, and H. Dessales. Automatic alignment of paintings and photographs depicting a 3D scene. In *3dRR 2011*, pages 545–552.

[29] P. Shanmugam and O. Arikan. Hardware accelerated ambient occlusion techniques on GPUs. In *I3D 2007*, pages 73–80.

[30] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM T. Graphics*, 30(6):154, Dec. 2011.

[31] I. Sipiran and B. Bustos. Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. *The Vis. Comput.*, 27(11):963–976, Nov. 2011.

[32] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3D CAD data. In *BMVC 2010*.

[33] P. Viola and W. M. I. Wells. Alignment by maximization of mutual information. *Int. J. Comput. Vision*, 24(2):137–154, Sept. 1997.

[34] A. Wendel, A. Irschara, and H. Bischof. Natural landmark-based monocular localization for MAVs. In *ICRA 2011*, pages 5792–5799.

[35] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D representations for object recognition and modeling. *IEEE T. Pattern Anal. Mach. Intell.*, 35(11):2608–2623, Nov. 2013.