Int J Comput Vis (2017) 125:65–81
DOI 10.1007/s11263-017-1022-x

Automatic Registration of Images to Untextured Geometry Using Average Shading Gradients

Tobias Plötz[1] · **Stefan Roth**[1]

Received: 19 May 2016 / Accepted: 18 May 2017 / Published online: 6 June 2017

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. Moreover, we show that registration can be performed even for paintings despite lacking photo-realism.

Keywords Registration · 3D geometry · Gradients · Dense correspondence · Pose estimation

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 (Corsini et al. 2009; Neugebauer and Klein 1999). 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 (Matzen and Snavely 2014), 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, leading to a multitude of augmented reality applications.

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 (see Fig. 1). This is in contrast to many existing image-to-geometry registration approaches (Irschara et al. 2009; Li et al. 2008, 2010) 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 (Agarwal et al. 2009). 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;

Communicated by Rene Vidal, Katsushi Ikeuchi, Josef Sivic, Christoph Schnoerr.

This work was supported by the European Union FP7 Project "Harvest4D" (No. 323567).

✉ Tobias Plötz
tobias.ploetz@visinf.tu-darmstadt.de

Stefan Roth
stefan.roth@visinf.tu-darmstadt.de

[1] TU Darmstadt, Darmstadt, Germany

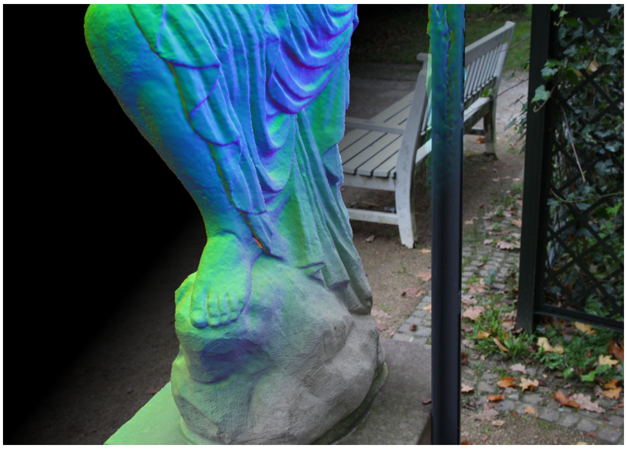


Fig. 1 Our algorithm finds the pose of an untextured 3D model of a potentially textured object from a single photograph

thus we need to work with the raw geometry information only (Corsini et al. 2012).

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, (e.g., Corsini et al. 2009). 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 stream of incoming 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., Dalal and Triggs 2005; Lowe 2004). Since we cannot hope to recover the texture gradients in renderings of the untextured 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 photographs. 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 match 2D keypoints on the image to 2D keypoints detected on renderings of randomly sampled viewpoints around Harris3D keypoints (Sipiran and Bustos 2011). Coarse poses are obtained by estimating an affine transformation between the matching patches of photograph and rendering. This initial estimate is refined in a second step that iteratively improves the pose using SIFT flow (Liu et al. 2011) 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 succeeded.

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 deal with a low ratio of correct patch correspondences by generating coarse image-to-geometry registration estimates from just a single correct correspondence. (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 been successful. 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.

This paper is an extended version of (Plötz and Roth 2015). In comparison to our previous work, we improved the refinement as described in Sect. 4.4 and give a proof for the bound on the average shading gradients in the appendix. We give further analysis of the approximation error incurred by our closed-form bound in Sect. 3. Furthermore, we provide an extensive study and discussion of the different parameters of the proposed feature descriptor and compare our method to an improved RANSAC baseline. Finally, we show that our registration pipeline is also capable of registering paintings and stylized photographs to untextured 3D meshes, despite their non-photorealistic depiction.

2 Related Work

The idea of using *rendered lines* for aligning 3D objects has a long history in computer vision (Lowe 1991) and is used in object-level pose estimation (Lim et al. 2013a; Stark et al. 2010; Zia et al. 2013), image-to-geometry registration (Russell et al. 2011), sketch-based shape retrieval (Eitz et al.

2012), and photo-to-terrain alignment (Baboud et al. 2011). 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 (DeCarlo et al. 2003), for example, are drawn where contour lines would occur if the view direction was altered slightly. Apparent ridges (Judd et al. 2007) 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. While we observe good results using a simple Lambertian shading model, incorporating global illumination effects like ambient occlusion (Shanmugam and Arikan 2007) 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 (Aubry et al. 2014; Irschara et al. 2009; Li et al. 2010). A pose is typically estimated from these 2D-to-3D correspondences using RANSAC. Irschara et al. (2009), Li et al. (2008, 2010) use previously registered images to derive image features and Wendel et al. (2011) extends (Irschara et al. 2009) by exploiting temporal coherency of camera poses when moving through a scene. In our work we drop the requirement of having pre-aligned images and instead use only a 3D model from which we render synthetic views. Aubry et al. (2014), Russell et al. (2011) also take this approach for aligning paintings to geometry. They leverage the ground plane to generate novel viewpoints, while we sample camera poses around key points on the 3D object. Also, while Aubry et al. (2014), Russell et al. (2011) use 3D models with texture information, we address the more general setting of having an untextured 3D model of a real-world object. While in our experiments the 3D model is represented as an untextured mesh, Sibbing et al. (2013) use colored point clouds from which they create renderings. They propose a novel splat rendering technique that generates complete synthetic views from sparse point clouds while maintaining sharp gradients. For ground-to-aerial image matching Shan et al. (2014) leverage a multi-view stereo reconstruction to warp ground images into novel views from aerial camera perspectives. This reduces viewpoint differences to real aerial images, allowing for robust matching of image features.

Our two-phase pose estimation strategy is related to (Li et al. 2008; Russell et al. 2011), which use GIST descriptors (Oliva and Torralba 2001) for retrieving similar views and thereby also first generate initial pose estimates, which are subsequently refined. In our work, the first phase relies on image patches instead of complete views, allowing for a wider sampling of viewpoints. Using affine transformations induced by single feature point correspondences for generating relative poses between two images has been used in image retrieval (Philbin et al. 2007) to re-rank initial search results based on spatial verification. Similarly, Lowe (2004) treats object detection as an image retrieval problem. Putative object poses are identified by matching image features on a test image to those obtained from training images of objects, where each feature correspondence induces an affine transformation of a training image. To remove spurious detections due to background and clutter, Lowe (2004) clusters these transformations with a Hough transform, which is related to our verification step. However, in this paper we use affine transformations for generating initial poses instead of validating feature matches. Since the difficulty of our registration problem comes from very few correspondences being correct, we postpone verification *after* the refinement step to obtain a robust verification criterion by looking at the mutual reprojection error of pairs of pose hypotheses.

Regarding the *descriptor* used for matching, we found standard HoG descriptors (Dalal and Triggs 2005) to perform well in our setting, but other specialized descriptors have been proposed as well. Baatz et al. (2012) use contourlets to match the horizon line of a query photograph to renderings of a digital elevation model, thus enabling accurate geo-localization in mountainous environments. Arandjelović and Zisserman (2011) incorporate domain knowledge into their descriptor by combining HoG features with an occupancy map derived from a figure-background segmentation. The segmentation is based on a superpixel classifier trained for distinguishing statues from background. The enhanced descriptor shows significant improvements for retrieving similar views of textureless statues from a database of real photographs. Liu and Stamos (2012), in contrast, rely on global features such as lines that are typically found in urban scenes.

Recently, *learning-based* approaches for object-level pose estimation have shown great success. Brachmann et al. (2014) estimate the 6 degrees-of-freedom (DoF) extrinsic camera pose from a single RGB-D image for a predefined class of objects. First, pixel-wise estimates of 3D object coordinates are regressed and afterwards the local noisy information is aggregated to a robust pose estimate using RANSAC. Instead of scoring pose hypotheses using a handcrafted energy, Krull et al. (2015) employ a convolutional neural network (CNN) to model the energy; optimization is still done by RANSAC. As shown by Kendall et al. (2015), Kendall and Cipolla (2016) it is also possible to directly regress the 6D camera pose from a single image using a CNN. These networks are trained on images with known camera poses that were previously obtained using structure-from-motion. For retrieval of single objects and joint pose estimation, Bansal et al. (2016) propose to first predict an intermediate 2.5D representation from a 2D input image by regressing surface normals. Subsequently, a CNN trained on rendered views of CAD models predicts object pose and style. It remains open, how well synthesized images can be used as a proxy of real training data without sacrificing performance. To bridge the gap between synthetic renderings and realistic images, Su et al. (2015) propose a pipeline for rendering 3D objects in common poses onto realistic backgrounds. Based on this, Massa et al. (2016) learn a mapping from CNN features computed on a realistic photo to features from a rendering, both showing the same object in the same pose, thus improving matching. In contrast to learning-based methods, our registration algorithm is non-parametric since it leverages a database of 2D image descriptors. Hence, it is possible to update this database online, e.g. for incorporating prior successful registrations, without the need to retrain a model.

Techniques for *pose refinement* often involve optimizing some measure of alignment between the photograph and a rendering of the model. Viola and Wells (1997) pioneered mutual information based alignment, which assumes that pixel values are spatially independent, but come from a joint 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. Corsini et al. (2009) propose a blending of normal and ambient occlusion maps and Dellepiane and Scopigno (2013) additionally render colors induced from other images whenever possible. Other refinement approaches try to align the silhouette lines of the renderings and photograph (Neugebauer and Klein 1999). This approach, however, requires the full object to be depicted, whereas our approach does not assume silhouette lines to be visible. Also 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

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

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. Under this simple shading model, we can express the image I of the untextured 3D model given a certain camera pose in terms of a normal map $\mathbf{n}$ and lighting direction $\mathbf{l}$ as

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

Inserting Eq. (2) into (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 (DeCarlo et al. 2003). 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 order to be able to nevertheless recognize these gradients, 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*

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

$$= \int_{\mathbf{S}} \Big[\Big(h_x * \max(0, -\mathbf{n}^\top \mathbf{l})\Big)^2 + \Big(h_y * \max(0, -\mathbf{n}^\top \mathbf{l})\Big)^2 \Big]^{\frac{1}{2}} \; d\mathbf{l}. \tag{3b}$$

Even for the simple Lambertian shading model computing the average gradient magnitude in Eq. (3a) 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 $\max(0, -\mathbf{n}^\top \mathbf{l})$ by $\frac{1}{2}(\mathbf{n}^\top \mathbf{l})$, since the square of the dot product is symmetric in the light direction and every pixel is visible for only half of the lighting directions that we integrate over. In other words, 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 Sect. 5). As a second approximation, we apply Jensen's inequality, which allows deriving a closed form upper bound on the approximation as follows:

$$\begin{align}
\overline{\|\nabla I\|} &\approx \frac{1}{2}\int_{\mathbf{S}} \sqrt{\left(h_x * (\mathbf{n}^\top \mathbf{l})\right)^2 + \left(h_y * (\mathbf{n}^\top \mathbf{l})\right)^2}\ \mathrm{d}\mathbf{l} \tag{4}\\
&\le \frac{1}{2}\sqrt{\int_{\mathbf{S}} \left(h_x * (\mathbf{n}^\top \mathbf{l})\right)^2 + \left(h_y * (\mathbf{n}^\top \mathbf{l})\right)^2\ \mathrm{d}\mathbf{l}} \notag\\
&= \frac{1}{2}\sqrt{\int_{\mathbf{S}} \left((h_x * \mathbf{n})^\top \mathbf{l}\right)^2\ \mathrm{d}\mathbf{l} + \int_{\mathbf{S}} \left((h_y * \mathbf{n})^\top \mathbf{l}\right)^2\ \mathrm{d}\mathbf{l}} \notag\\
&= \sqrt{\frac{\pi}{3}}\sqrt{\sum_{i=1}^{3}(h_x * \mathbf{n}_i)^2 + (h_y * \mathbf{n}_i)^2}. \tag{5}
\end{align}$$

To obtain the last equality, we linearize the squared filter response by applying the following transformation to all vectors:

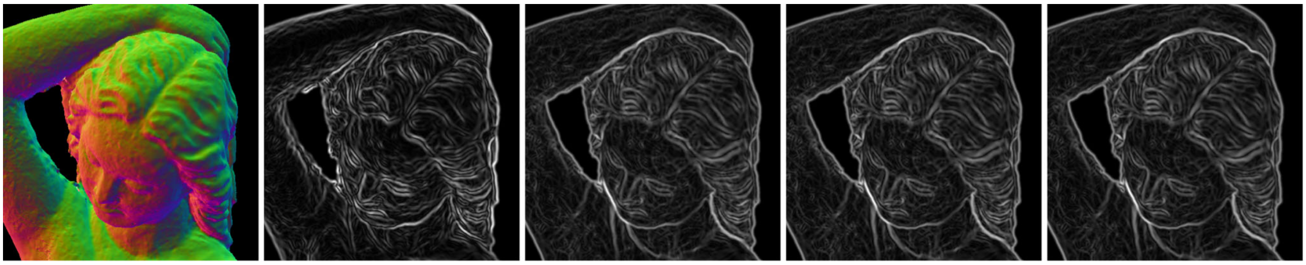
$$\hat{\mathbf{x}} = [\mathbf{x}_1^2\ \ \mathbf{x}_2^2\ \ \mathbf{x}_3^2\ \ 2\mathbf{x}_1\mathbf{x}_2\ \ 2\mathbf{x}_1\mathbf{x}_3\ \ 2\mathbf{x}_2\mathbf{x}_3]^\top, \tag{6}$$

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

$$\begin{align}
\int_{\mathbf{S}} \left((h * \mathbf{n})^\top \mathbf{l}\right)^2\ \mathrm{d}\mathbf{l} &= \int_{\mathbf{S}} \widehat{(h * \mathbf{n})}^\top \hat{\mathbf{l}}\ \mathrm{d}\mathbf{l} \notag\\
&= \widehat{(h * \mathbf{n})}^\top \int_{\mathbf{S}} \hat{\mathbf{l}}\ \mathrm{d}\mathbf{l} \notag\\
&= \frac{4}{3}\pi \sum_{i=1}^{3} (h * \mathbf{n}_i)^2, \tag{7}
\end{align}$$

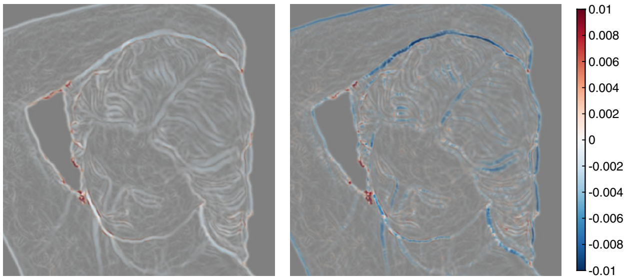
where the $\mathbf{n}_i$ denote the x, y, z components of the normal field. A proof for the last equality can be found in the appendix. The bound from Eq. (5) is very efficient to compute as it only involves convolutions of the normal map and pixel-wise operations. While it may seem intuitive to compute gradient information from the normal map, it is not immediately clear how to do this due to its multivariate nature. Our result in Eq. (5) shows how this intuitive idea can be realized and, moreover, formally justified as an approximation of the average shading gradient. Furthermore, having this gradient information allows deriving common image features, e.g. HoG (Dalal and Triggs 2005), from the normal map. Undirected gradient orientations in the range $[0, \pi]$ are obtained by computing image derivatives on the average shading gradient image with a central difference gradient operator.

Benefits Figure 2 shows an example of the gradient magnitudes of a Lambertian shading model for the normal map of a statue. For this illustration, we compute the average shading gradient using Monte Carlo estimation, i.e. we sample 2000 light directions and approximate the integral by averaging the gradients of the shaded images. We also show a Monte Carlo estimate of our approximation to the true average shading gradients (Eq. 4). Note that this exhaustive computation is not practical, however. Assuming only a single light direction avoids this issue, but when making a "headlight" assumption (b, DeCarlo et al. (2003)), 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 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. Our two approximations (d, e) to the exact average shading gradient have little visible impact, hence retain all its benefits. Crucially, our closed-form bound is much more efficient to compute. It requires roughly the same computational effort as calculating the gradients of the shaded image for a single light direction. To further investigate the approximation error, Fig. 3 shows the differences between the Monte Carlo estimate of the true average shading gradients and the Monte Carlo estimate with the approximated Lambertian shading, as well as the differences to the gradients obtained by evaluating our closed-form bound. Overall, the error is very small in both cases with less then one percent of the maximum gradient magnitude. Moreover, we can see that in areas of low curvature the bound is tight, whereas in areas of high curvature and along the silhouette the closed-form bound overestimates the gradient magnitude slightly.



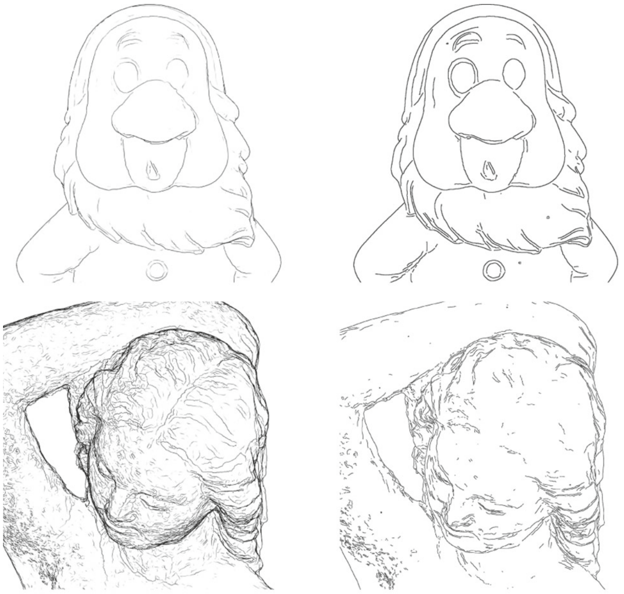
(a) Normal map **(b)** "Headlight" assumption **(c)** Average shading gradient **(d)** Approximation, Eq. (4) **(e)** Bound, Eq. (5)

Fig. 2 Image gradients for the normal map from (**a**). *From left to right* **b** Gradient magnitude computed with Lambertian shading and "headlight" assumption (DeCarlo et al. 2003). Monte Carlo estimate of the average gradient magnitude using the (**c**) correct (Eq. 3a) and **d** approximated (Eq. 4) Lambertian shading. **e** Our closed-form bound (Eq. 5)



(a) Approximation, Eq. (4) **(b)** Bound, Eq. (5)

Fig. 3 *Left:* differences between Fig. 2c and d. *Right:* differences between Fig. 2c and e. *Red and blue colors* mean that the true average shading gradients are stronger or weaker, respectively, than the compared gradient magnitudes. *Shades of grey* indicate that the gradient magnitudes coincide. Differences are normalized by the largest gradient magnitude in Fig. 2c



(a) Apparent ridges **(b)** Average shading gradient

Fig. 4 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*)

Connection to apparent ridges Judd et al. (2007) 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. 4, 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 Sect. 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 (Shrivastava et al. 2011), painting-to-geometry registration (Aubry et al. 2014), and location estimation (Irschara et al. 2009; Li et al. 2010). As matching to untextured models leads to more false correspondences, we divide the registration process into two steps. First, we estimate a set of coarse poses, each from just a single correspondence between an image patch and a patch in the database of rendered views of the model. Compared to matching entire rendered images (Russell et al. 2011), matching individual patches increases flexibility and reduces the size of the database of renderings, since translations and scalings of patches do not need to be considered at this stage. In the second step we refine the coarse poses into full 11 degrees-of-freedom poses by alternating between dense flow estimation between rendering and photograph, and re-estimating the pose from the correspondences induced by the flow field. Finally, a verification step assesses whether the registration process was successful at all, since we do not expect our algorithm to register all images perfectly. Figure 5 illustrates this pipeline.

4.1 Patch Database

To populate the database with rendered patches, we randomly sample camera poses from which the 3D model can be rendered. To reduce the space of possible camera poses, we first identify characteristic points on the 3D model that likely give rise to discriminative features in renderings that show this point. We find 100 characteristic points using Harris3D (Sipiran and Bustos 2011), a 3D keypoint 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 randomly chosen subset of all vertices, and use non-maximum suppression in 3D space to yield thinned out keypoints. For each keypoint we randomly sample 10 camera poses viewing this

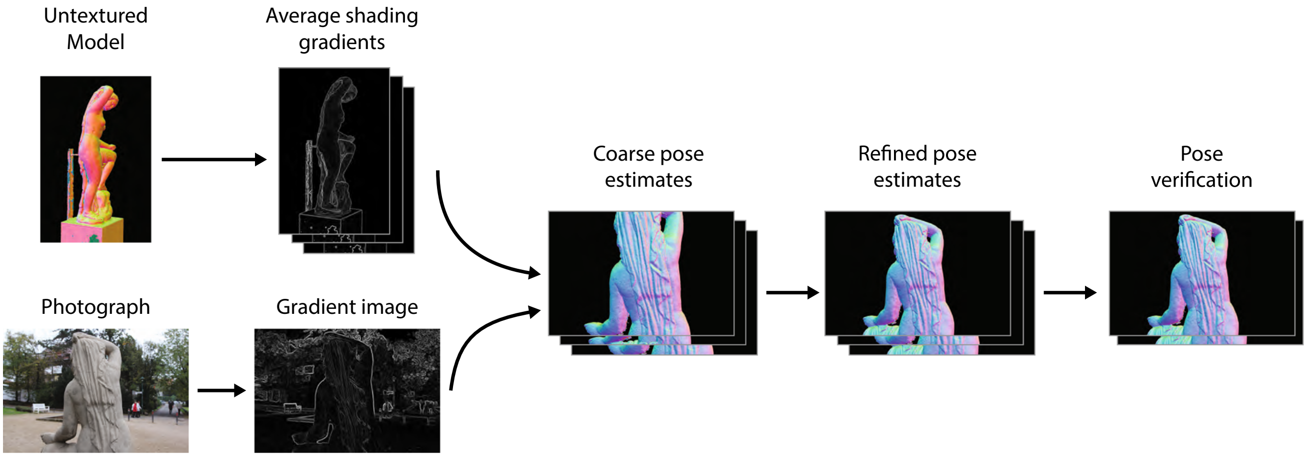


Fig. 5 Registration pipeline using average shading gradients

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 such that the relative distance to a mean distance is Gaussian. This allows to sample camera poses nearby and far away. We choose the mean to be a fixed value relative to the size of the 3D model. However, one could also make it dependent on a measure of scale of the 3D keypoint. 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. 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. While this random sampling of poses gives our method the flexibility to register photographs from many different viewpoints, it is also possible to constrain the sampling process if prior knowledge about the query photographs is available.

We then render each view using the average shading gradient from Sect. 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 (Lowe 2004), 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 image and the gradient image of a query photograph, even in the presence of additional edges in one modality. 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.

4.2 Feature Descriptor

We compute a HoG descriptor (Dalal and Triggs 2005) from the gradient patches. To this end, we consider two different strategies: (i) *HoG from first-order gradients*, where gradient magnitudes from the photograph respectively the normal map are binned directly into gradient histograms; (ii) *HoG from second-order gradients*, where we apply a central difference gradient operator on the gradient magnitude images of the photograph and normal map, respectively. The oriented gradients obtained from this second gradient operator are then sorted into the gradient histograms of HoG. The reason for evaluating the second alternative is that Eitz et al. (2012) found a similar representation to be beneficial for sketch-based shape retrieval. They compute a descriptor from the responses of oriented Gabor filters on line drawings and line renderings. Their GALIF descriptor can also be seen as calculating histograms over oriented gradient information of a gradient image.

When inserting the gradients into the histogram bins we linearly interpolate spatially and in terms of orientation. We use 8×8 blocks on a regular grid with 9 undirected orientation bins, resulting in a 576-dimensional descriptor, which is stored in the database. We also try a circular HoG descriptor with 4 undirected orientation bins in a circular layout with 8 angular bins and 4 bins along the radius resulting in a 132 dimensional descriptor. Note that we do not use non-maximum suppression on the gradients, as we found this to deteriorate performance.

4.3 Coarse Pose Estimation

Given an input photograph that should be registered, we apply the same feature detection pipeline as for the renderings, using the same linear gradient operator. To obtain 2D-to-3D patch correspondences, for each feature in the 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. (2014):

$$s(\mathbf{d}_q, \mathbf{d}_{db}) = (\mathbf{d}_{db} - \boldsymbol{\mu})^\top \Sigma^{-1} \mathbf{d}_q. \quad (8)$$

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. Equation (8) can be interpreted as the calibrated classification score of $\mathbf{d}_q$ for an one-*vs*-all classifier that discriminates $\mathbf{d}_{db}$ from all other descriptors using linear discriminant analysis (LDA) (Aubry et al. 2014). Like Aubry et al. we found that transforming the database descriptors increases the matching quality over the raw descriptors as non-discriminative features have less influence.

As we do not rely on textured 3D models, we need to deal with an increased number 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. Moreover, we found that the commonly used ratio test (Lowe 2004) is not applicable in our scenario as the distribution of the ratios is almost identical for both inlier and outlier correspondences. Hence, regular RANSAC (Fischler and Bolles 1981) is likely to fail as we need to sample 3 (Kneip et al. 2011) or more correct correspondences (Lepetit et al. 2009) to estimate the extrinsic camera pose, or at least 6 correspondences to estimate the full pose.

To cope with this, 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 isotropic scale of the Harris keypoints. Philbin et al. (2007) observed this similarity transform to provide very good performance in the context of re-ranking image retrieval results and that additionally estimating an anisotropic scale or shear factor yields only marginal improvements. After applying the 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. 6). 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 Sect. 5 that this provides a good and efficient initialization for pose refinement.

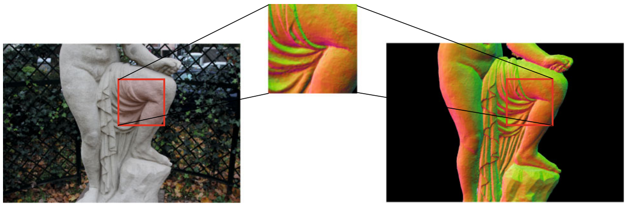


Fig. 6 Estimating a camera pose from a single correspondence: the query patch (*red box on the left*) was matched to a database patch (*middle*). We generate a coarse estimate of the true camera pose by concatenating the known pose of the database patch with the relative scale and translation of the matching 2D Harris keypoint. This figure shows the photograph and the aligned normal map for better visualization; the matching uses gradient representations

4.4 Pose Refinement

The coarse pose estimates are ranked based on two criteria. First, the number of inlier correspondences, i.e. those whose 3D point projects within a 50 pixel distance to the 2D point. Second, the matching score of the descriptors as a high matching score indicates that a highly discriminative descriptor was matched well. We select up to 20 coarse poses for iterative refinement by taking the 10 top ranked poses for both criteria, respectively.

For refining the coarse poses we alternate between two steps. First, we estimate a dense flow field between the average shading gradients of the rendering given the current camera pose on the one hand and the gradient representation of the photograph on the other hand. Second, we compute a refined pose by leveraging the 2D-to-3D correspondences that are induced by the flow field. For computing the dense correspondences, we propose to use the SIFT flow algorithm (Liu et al. 2011), which is similar in spirit 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 (Liu et al. 2011), as well as the default parameters as provided by the authors' implementation. Deviating from our previous work (Plötz and Roth 2015), we adapt the SIFT flow energy such that the pairwise term is only active between pixels that actually show a part of the 3D scene. Thus, the flow at those pixels becomes independent of the flow at pixels that do not show the scene. This avoids a bias in flow estimation for the visible pixels, which may be caused by empty parts of the rendering being matched against big constant areas in the photo such as the sky. We also experimented with Deformable Spatial Pyramid Matching (Kim et al. 2013), another dense scene matching algorithm, but found the performance to be subpar compared to SIFT flow. Standard optical flow algorithms tend to fail since brightness constancy is often violated when matching across the two modalities.

The resulting flow field is now used to compute dense 2D-to-3D correspondences. In contrast to the coarse step, we can use RANSAC to estimate a refined pose from these, 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 (Hartley and Zisserman 2004). Empirically, we found that only few iterations of RANSAC suffice to find a good refinement. We use three iterations of this alternating refinement in a coarse-to-fine fashion: 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 refinement serves two purposes. First, for coarse poses that originate from a wrong matching, the refinement will usually diverge. Second, the refinement of coarse poses from a "true" correspondence will usually converge to poses near the ground truth. These two effects combined allow for a robust pose verification.

4.5 Pose Verification

Our pose verification step detects whether the registration process was successful. For this we use the pairwise mutual reprojection error between refined poses. 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:

$$\delta(\mathcal{P}, \mathcal{P}') = \frac{1}{2}\left(\frac{1}{|\mathcal{V}|}\sum_{\mathbf{x}\in\mathcal{V}} \|\mathcal{P}(\mathbf{x}) - \mathcal{P}'(\mathbf{x})\|_2 + \frac{1}{|\mathcal{V}'|}\sum_{\mathbf{x}\in\mathcal{V}'} \|\mathcal{P}(\mathbf{x}) - \mathcal{P}'(\mathbf{x})\|_2\right). \tag{9}$$

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, we regard 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., Corsini et al. 2012).

While typical registration or localization approaches (e.g., Irschara et al. 2009; Li et al. 2010) employ a geometric verification based on the reprojection error of individual points, e.g. to determine the number of inliers in the inner RANSAC loop, our verification considers all visible points of the pose hypotheses, thus reducing the probability that hypotheses are compatible by chance. The robustness of the verification is reflected in our experiments by the low false positive rate of verified poses.

5 Experiments

To quantitatively 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 (Corsini et al. 2009) with manual initialization. The mesh is of 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.

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 (MVE, Fuhrmann et al. 2015). 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 backgrounds and in 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 proposed average shading gradients (ASG) match gradients and edges found on real images. As rendering baseline we use apparent ridges (Judd et al. 2007), 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 gradient rendering method also after non-maximum suppression (NMS).

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
ASG + NMS/sketch tokens	110.9	64.6	63.7
ASG + NMS/gradients + NMS	130.6	70.6	65.2
ASG/gradients	**159.3**	**82.5**	**72.4**

Results with highest similarity score marked in bold

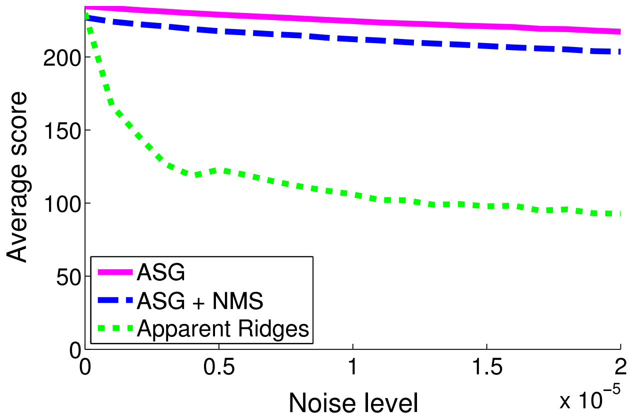


Fig. 7 Similarity score (Eq. 8) between descriptors from renderings of a noiseless mesh and of meshes with artificial noise on the vertex positions. Higher scores mean more robustness to noise

On the photograph, we compute gradients or detect edges using the gradient operator of the well-known Canny detector (Gradients, Canny 1986), as well as using sketch tokens (Lim et al. 2013b), 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. (8) from patches in correct correspondence. Higher scores mean higher similarity. Since the coarse registration algorithm (Sect. 4.3) 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 7 shows the similarity score (Eq. 8) 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.

5.2 Pose Estimation

Next we evaluate our registration pipeline in terms of success rate and accuracy of the registration as well as the robustness of the verification.

Experimental Setting In order to disentangle the effects of the different parts of our descriptor, we create baselines by changing descriptor parameters along the following design dimensions:

- We compare the proposed average shading gradients to *headlight-shading gradients* (HSG), i.e. gradients from a single Lambertian shading under a headlight assumption where the light direction of the single point light coincides with the camera viewing direction.
- We compare the case of 1st-order and 2nd-order gradients as described in Sect. 4.2.
- We compare the HoG descriptor obtained on a regular grid (HoG) to the circular HoG descriptor (CHoG) to assess the importance of the layout.

The descriptor proposed in our previous work (Plötz and Roth 2015) corresponds to using average shading gradients, binning second-order gradients, and using the regular HoG layout. The gradient operator h for average shading gradients is a central difference filter together with Gaussian smoothing with standard deviation of 2 pixels. This descriptor is used in the experiments of this paper if not stated otherwise. The “shaded” baseline in our previous paper (Plötz and Roth 2015) corresponds to gradients from headlight shading that are directly binned into a regular HoG descriptor (denoted “1st (no smoothing)” here). To better compare the two gradient renderings, we also evaluate headlight shading with additional smoothing (denoted “1st”). For evaluating descriptor similarity for CHoG we use the inner product on the raw descriptors, since the LDA transformation led to significantly worse performance here.

Moreover, we compare to a RANSAC baseline for estimating just the extrinsic pose while assuming the intrinsic pose to be known. Note, that our algorithm estimates a full 11 DoF pose without knowledge of the intrinsic parameters. The RANSAC baseline operates on the original descriptor (i.e. HoG from 2nd-order gradients, using ASG on the normal

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

Pose estimation	Descriptor	Gradient type	Gradient order	Gnome	Statue	Notre Dame
Our	CHoG	HSG	1st (no smoothing)	1	0.04	0.44
Our	CHoG	HSG	1st	1	0.16	0.47
Our	CHoG	HSG	2nd	0.89	0.09	0.56
Our	CHoG	ASG	1st	1	0.26	0.64
Our	CHoG	ASG	2nd	1	0.20	0.64
Our	HoG	HSG	1st (no smoothing)	1	0.13	0.54
Our	HoG	HSG	1st	0.89	0.28	0.53
Our	HoG	HSG	2nd	1	0.38	0.59
Our	HoG	ASG	1st	1	0.38	0.71
Our	HoG	ASG	2nd	1	0.42	0.66
RANSAC	HoG	ASG	2nd	0.78	0.16	0.56

map) in the following way: The correspondences between 2D feature points on the input photograph and 3D keypoints on the model form putative inliers. In each of up to 50,000 iterations of the RANSAC loop we sample 3 correspondences using the PROSAC method (Chum and Matas 2005) and ranking correspondences according to the scores from descriptor matching. Next, we estimate the extrinsic pose (i.e., camera rotation and translation) with the P3P algorithm of Kneip et al. (2011), yielding four estimates of the extrinsic pose. Poses that pass the $T_{1,1}$ test (Matas and Chum 2002) on a fourth sampled correspondence are improved further with a local optimization (Chum et al. 2003), where we employ 10 inner RANSAC iterations on the inlier set. In each inner iteration, we sample 6 correspondences from the inlier set and use the non-minimal EPnP pose solver of Lepetit et al. (2009) to calculate an improved extrinsic pose.

We measure the registration quality by means of the mutual reprojection error (Eq. 9) to the ground truth pose.

Registration Success Rate In a first experiment,[1] Table 2 shows the success rate of the coarse registration stage with the different variants of the descriptor and the RANSAC baseline. More precisely, we count a registration as successful if the coarse pose with most inliers achieves a mutual reprojection error below 150 pixels, since empirically this is accurate enough for the refinement to improve the pose significantly. Likewise, we apply the same threshold to the poses found by RANSAC to determine if it has found a correct registration. We make the following observations from the results: First, the HoG descriptor on the regular grid is superior to CHoG, which we attribute to the low dimensionality of the CHoG descriptor. Second, our proposed average shading gradients lead to a consistently higher registration success than when using the headlight shading gradients, even when the same amount of smoothing is applied in both cases. This fact holds across all combinations of the other descriptor parameters, emphasizing the superiority of average shading gradients. Third, there is no clear ranking between using first- or second-order gradients. Fourth, our coarse registration stage is superior to RANSAC in finding suitable pose estimates, despite RANSAC assuming known intrinsics. This effect is especially pronounced on the challenging *Statue* dataset that exhibits a low ratio of correct matches.

For the *Statue* and *Notre Dame* datasets and the HoG descriptor only, Fig. 8 plots the fraction of correctly registered photographs among the top k coarse hypotheses, ranked according to the number of inlier 2D-to-3D correspondences. We see that findings for the top hypothesis carry over when looking at more hypotheses. We also note, that performance saturates, especially on the *Notre Dame* dataset, meaning that for some photographs the coarse matching does not produce a single good hypothesis. 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.

Registration Accuracy Next, we evaluate the accuracy of the estimated poses. Here, we concentrate our analysis on registration with HoG since it outperforms CHoG in terms of registration success rate and since the refinement step uses SIFT descriptors in both cases. To assess the registration accuracy, we evaluate the statistics of the reprojection error (Eq. 9) of coarse and refined poses from our pipeline as well as the RANSAC baseline. Specifically, we evaluate the reprojection error on those images that can be successfully registered by *all* considered methods, thus enabling a

[1] Numbers differ compared to (Plötz and Roth 2015) due to stochasticity in the algorithm and improvements to the refinement.

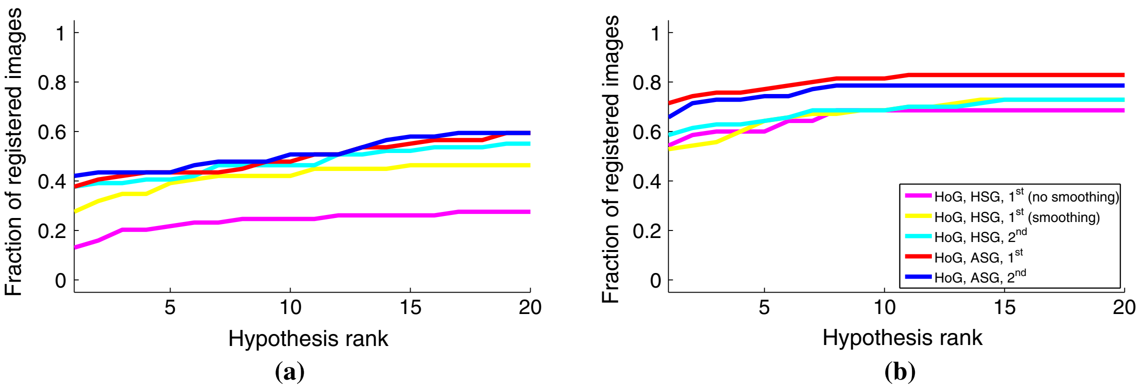


Fig. 8 Fraction of correctly registered photographs when considering the first k ranked coarse hypotheses on two datasets. **a** *Statue* dataset. **b** *Notre Dame* dataset

Table 3 Median reprojection error, as well as lower and upper quartiles for the pose with most inliers on images that can be registered correctly

Coarse poses	Refined	Gradient type	Gradient order	Gnome	Statue	Notre Dame
Our	No	HSG	1st (no smoothing)	17.6 (12.7/22.1)	46.2 (33.3/50.8)	20.7 (15.5/31.8)
Our	No	HSG	1st	16.0 (13.1/24.0)	28.2 (25.5/36.7)	24.5 (17.1/29.7)
Our	No	HSG	2nd	22.1 (11.1/28.8)	36.0 (29.0/40.0)	23.9 (18.7/29.3)
Our	No	ASG	1st	20.9 (18.4/24.4)	27.8 (23.0/33.3)	25.3 (14.6/30.7)
Our	No	ASG	2nd	20.5 (18.1/27.4)	25.4 (22.7/42.5)	19.5 (14.8/25.0)
Our	Yes	HSG	1st (no smoothing)	66.5 (33.8 / 98.5)	20.5 (11.1 / 46.0)	7.9 (2.6/15.6)
Our	Yes	HSG	1st	9.3 (8.4/9.9)	4.9 (4.1/14.4)	5.0 (2.6/9.7)
Our	Yes	HSG	2nd	8.8 (7.3/14.0)	7.9 (4.4/10.5)	7.5 (3.6/12.7)
Our	Yes	ASG	1st	8.6 (6.1/9.4)	4.8 (3.7/6.0)	4.0 (2.6/5.1)
Our	Yes	ASG	2nd	8.5 (4.7/9.5)	3.8 (3.6/6.6)	4.4 (2.5/8.5)
RANSAC	No	ASG	2nd	13.2 (7.9/17.0)	30.7 (18.4/43.7)	16.4 (10.4/19.5)
RANSAC	Yes	ASG	2nd	8.4 (4.4/9.4)	3.8 (3.5/4.6)	2.7 (2.1/4.2)

fair comparison across methods. In total we thus evaluate 6 images on *Gnome*, 5 on *Statue* and 20 on *Notre Dame*. For each of these images we calculate the reprojection error for the pose that has the highest number of inlier correspondences. To robustly summarize the distribution of the reprojection error, we evaluate the median as well as well as the upper and lower quartiles across all considered images. We also evaluate applying our refinement procedure to the poses resulting from the RANSAC baseline.

Table 3 shows the error statistics. We make the following observations: First, we generally observe that the proposed refinement step greatly increases the registration accuracy over the coarse poses. On *Statue* refined poses are approximately six times more accurate when using average shading gradients and second-order gradients. On *Notre Dame* the error is decreased to less than a quarter compared to coarse poses. This observation holds even true for applying our refinement to the RANSAC poses. Furthermore, average shading gradients almost always lead to more accurate coarse poses than when using the corresponding descriptor with gradients from headlight shading. On *Statue* coarse poses from average shading gradients with second-order gradients are 30% more accurate than when using headlight shading. On *Notre Dame* the error is decreased by 18%. For refined poses we observe a similar trend. When comparing to the RANSAC baseline, we see that coarse poses from our pipeline are outperforming RANSAC poses on *Statue*, while RANSAC is better on the other two datasets. However, the coarse poses from RANSAC are already subject to a refinement by the local optimization step. Moreover, recall that the RANSAC baseline leads to a significantly smaller registration success rate.

Verification Accuracy In a final experiment, we evaluate the accuracy of the verification stage. As can be seen from Table 4, the verification proposed in Sect. 4.5 is able to identify very reliably when the registration succeeds. We do not observe any false positives, meaning that each photo-

Table 4 True positive and true negative rates of verification step

	Gnome	Statue	Notre Dame
True positives (TP)	1	1	1
True negatives (TN)	1	0.79	0.68

graph that our system believes to be registered properly is indeed correctly registered. This is an important property if the estimated camera poses are to be used for subsequent applications, e.g. color transfer, since it is very undesirable to transfer false information. Note that we observe some false negatives, showing that our system errs on the cautious side. These results, moreover, suggest that our approach can be used as a fully automatic registration system.

Qualitative Examples Figure 9 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.

5.3 Painting-to-Geometry Registration

We furthermore test our registration method on the challenging task of registering paintings to 3D models. For this we downloaded a set of 26 paintings and stylized photographs of the Notre Dame front and applied our registration pipeline on it. This task is more challenging than the registration of natural photographs since paintings usually do not follow a precise perspective projection and do not depict the world in a realistic way on purpose.

Figure 10 shows examples of paintings for which our algorithm finds a set of poses that pass the verification step. Each image pair shows the query image as well as the verified pose with most inliers. As can be seen, the accuracy of the registration varies. This is mainly attributable to the refinement step. Since the refinement estimates a full 11 DoF pose, it can choose an unusual focal length to accommodate for the deviations from a perspective projection. This results in a distorted view of the 3D model. Restricting the set of admissible poses during refinement to sensible values of the intrinsic parameters could help to avoid these artifacts. In total 13 out of 26 images resulted in a verified set of poses and visual inspection revealed that all verified poses were sensible. This again highlights the robustness of our verification step.

Figure 11 shows paintings for which our algorithm could not find a valid registration, i.e. no set of hypotheses passed the verification step. The main reason for this failure is that already the coarse matching does not succeed in finding good proposals for the refinement. A more elaborate matching would certainly improve results on these hard examples.

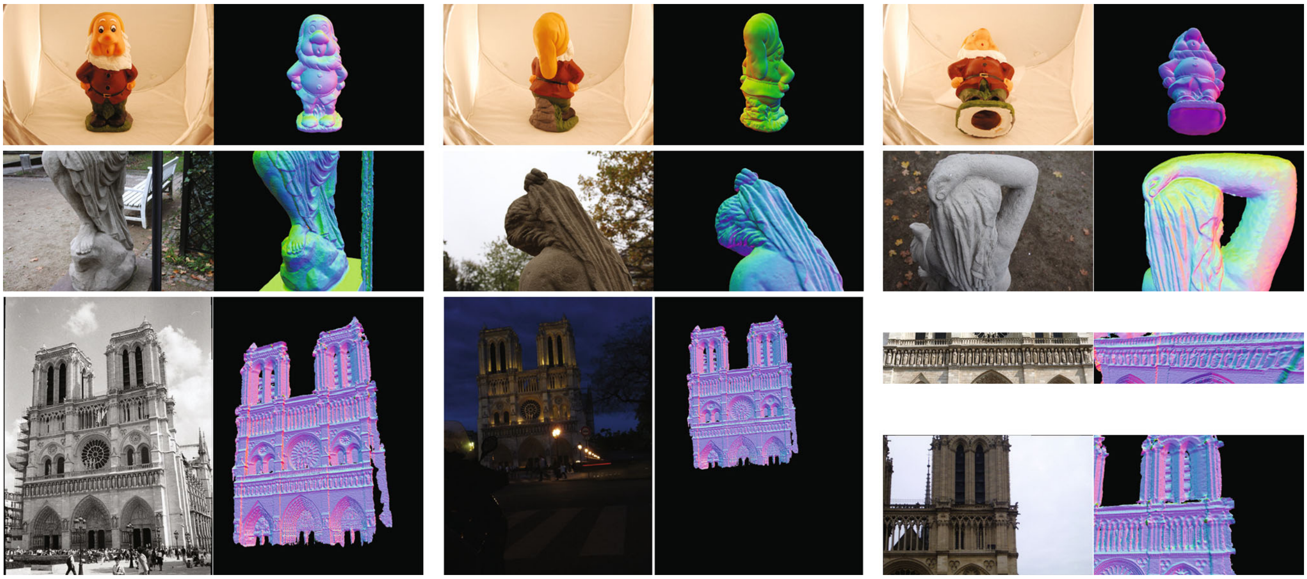


Fig. 9 Examples of successful registrations: the query photograph is shown on the *left*, the top-ranked verified pose on the *right*

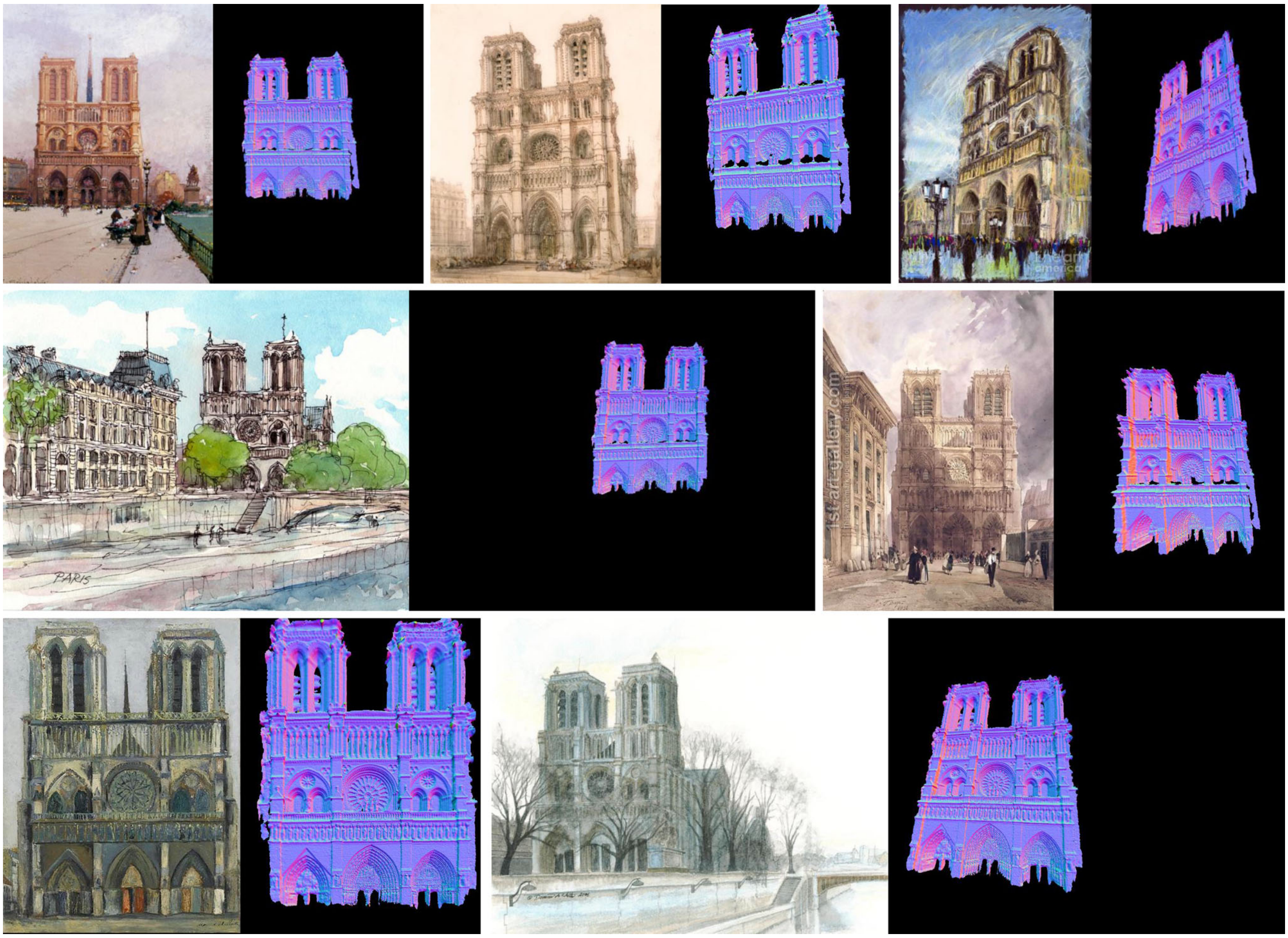


Fig. 10 Examples of verified pose estimates for painting-to-geometry registration. Images courtesy of David Roberts (*1st row, center*), Yuriy Shevchuk (*1st row, right*), Andre Voyy (*2nd row, left*) and Dominic White (*3rd row, right*)

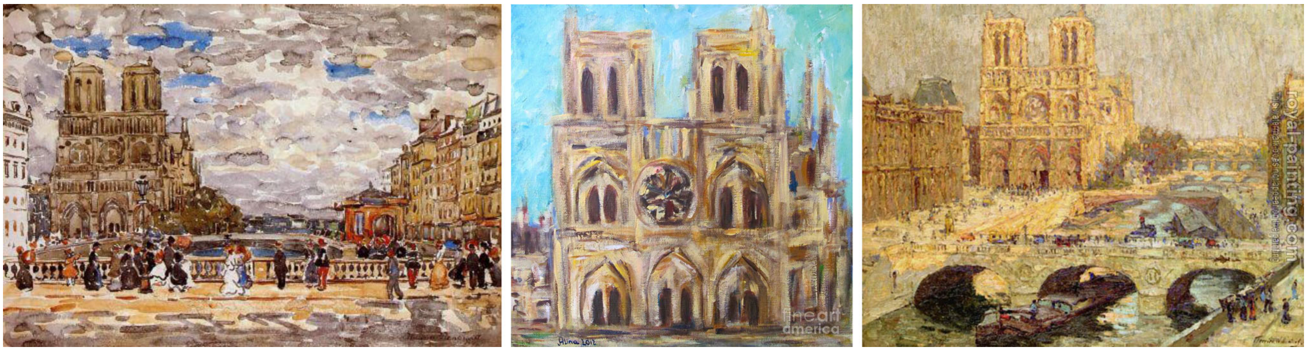


Fig. 11 Input paintings for which our algorithm fails to produce valid registrations. *Center* image courtesy of Alina Vidulescu

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 viewpoints and illumination conditions, without requiring initialization or any other form of manual intervention.

Moreover, it is also capable of aligning paintings to untextured 3D models despite the lack of photo-realism.

Acknowledgements We want to thank Gianpaolo Palma as well as Michael Goesele's research group for giving access to their datasets. Moreover, we want to thank the reviewers for giving us detailed and very helpful feedback. Portions © 2015 IEEE, reprinted, with permission, from (Plötz and Roth 2015).

Appendix A: Proof of Eq. (7)

Here, we show that

$$\widehat{(h * \mathbf{n})}^\top \int_{\mathbf{S}} \hat{\mathbf{l}}\, \mathrm{d}\mathbf{l} = \frac{4}{3}\pi \sum_{i=1}^{3} (h * \mathbf{n}_i)^2. \tag{10}$$

We begin by first rewriting the integral over the surface of the unit sphere by parameterizing the unit sphere with spherical coordinates:

$$\int_{\mathbf{S}} \hat{\mathbf{l}}\, d\omega = \int_0^{\pi} \int_0^{2\pi} \hat{\mathbf{l}} \cdot \sin\theta \; d\phi\, d\theta, \tag{11}$$

where θ denotes the polar angle and ϕ the azimuth. The factor $\sin\theta$ within the integral is the surface element. We now express the light direction $\mathbf{l}$ and also its augmented vector $\hat{\mathbf{l}}$ in terms of spherical coordinates as

$$\mathbf{l} = \begin{bmatrix} \sin\theta \cdot \cos\phi \\ \sin\theta \cdot \sin\phi \\ \cos\theta \end{bmatrix}, \quad \hat{\mathbf{l}} = \begin{bmatrix} \sin^2\theta \cdot \cos^2\phi \\ \sin^2\theta \cdot \sin^2\phi \\ \cos^2\theta \\ 2\sin^2\theta \cdot \cos\phi \cdot \sin\phi \\ 2\sin\theta \cdot \cos\phi \cdot \cos\theta \\ 2\sin\theta \cdot \sin\phi \cdot \cos\theta \end{bmatrix}. \tag{12}$$

We can integrate the vector $\hat{\mathbf{l}}$ component-wise and we will see that the first three and the last three components of $\hat{\mathbf{l}}$ integrate to the same values, respectively. Here we show how to integrate the first component of $\hat{\mathbf{l}}$:

$$\int_0^{\pi} \int_0^{2\pi} \sin^2\theta \cdot \cos^2\phi \sin\theta \; d\phi d\theta \tag{13a}$$

$$= \int_0^{\pi} \sin^3\theta \; d\theta \int_0^{2\pi} \cos^2\phi \; d\phi \tag{13b}$$

$$= \frac{4}{3} \int_0^{2\pi} \cos^2\phi \; d\phi \tag{13c}$$

$$= \frac{4}{3}\pi. \tag{13d}$$

The first equality is straightforward. The second and third equality can be arrived at by applying basic trigonometric identities. Integrating the second and third component of $\hat{\mathbf{l}}$ works similarly and again yields the constant $\frac{4}{3}\pi$. Now we show how to integrate the fourth component of $\hat{\mathbf{l}}$:

$$\int_0^{\pi} \int_0^{2\pi} 2\sin\theta \cdot \cos\phi \cdot \sin\phi \sin\theta \; d\phi d\theta \tag{14a}$$

$$= \int_0^{\pi} \sin^2\theta d\theta \int_0^{2\pi} 2\cos\phi \cdot \sin\phi \; d\phi \tag{14b}$$

$$= \int_0^{\pi} \sin^2\theta \; d\theta \cdot 0 \tag{14c}$$

$$= 0. \tag{14d}$$

The second equality can be seen by noting that $\cos\phi \cdot \sin\phi$ is an uneven function around π as $\cos\phi$ is even around π and $\sin\phi$ is uneven around π. Integrating components five and six also yields 0; the derivations can be obtained in a similar fashion.

Plugging these results into the left hand side of Eq. (10) proves the equality by noting that the first three components of $\widehat{(h * \mathbf{n})}$ are as follows:

$$\widehat{(h * \mathbf{n})}_i = (h * \mathbf{n}_i)^2, \qquad \forall i \in \{1, 2, 3\}. \tag{15}$$

References

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., & Szeliski, R. (2009). Building Rome in a day. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 72–79).

Arandjelović, R., & Zisserman, A. (2011). Smooth object retrieval using a bag of boundaries. In *IEEE international conference on computer vision* (ICCV) (pp. 375–382).

Aubry, M., Russell, B. C., & Sivic, J. (2014). Painting-to-3D model alignment via discriminative visual elements. *ACM Transactions on Graphics*, *33*(2), 14.

Baatz, G., Saurer, O., Kser, K., & Pollefeys, M. (2012). Large scale visual geo-localization of images in mountainous terrain. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.) *European conference on computer vision* (ECCV), Springer, Lecture Notes in Computer Science (Vol. 7573, pp. 517–530).

Baboud, L., Čadík, M., Eisemann, E., & Seidel, H. P. (2011). Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 41–48).

Bansal, A., Russell, B., & Gupta, A. (2016). Marr revisited: 2D–3D alignment via surface normal prediction. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 5965–5974).

Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., & Rother, C. (2014). Learning 6D object pose estimation using 3D object coordinates. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.) *European conference on computer vision* (ECCV), Springer, Lecture Notes in Computer Science (Vol. 8690, pp. 536–551).

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Chum, O., & Matas, J. (2005). Matching with PROSAC—Progressive sample consensus. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 220–226).

Chum, O., Matas, J., & Kittler, J. (2003). Locally optimized RANSAC. In B. Michaelis, & G. Krell (Eds.) *Pattern recognition, Proceedings of DAGM-symposium*, Springer, Lecture Notes in Computer Science (Vol. 2781, pp. 236–243).

Corsini, M., Dellepiane, M., Ganovelli, F., Gherardi, R., Fusiello, A., & Scopigno, R. (2012). Fully automatic registration of image sets on approximate geometry. *International Journal of Computer Vision*, *102*(1–3), 91–111.

Corsini, M., Dellepiane, M., Ponchio, F., & Scopigno, R. (2009). Image-to-geometry registration: A mutual information method exploiting illumination-related geometric properties. *Computer Graphics Forum*, *28*(7), 1755–1764.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 886–893).

DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., & Santella, A. (2003). Suggestive contours for conveying shape. *ACM Transactions on Graphics*, *22*(3), 848–855.

Dellepiane, M., & Scopigno, R. (2013). Global refinement of image-to-geometry registration for color projection. In *Digital heritage international congress* (DH) (pp. 39–46).

Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., & Alexa, M. (2012). Sketch-based shape retrieval. *ACM Transactions on Graphics*, *31*(4), 31.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Fuhrmann, S., Langguth, F., Moehrle, N., Waechter, M., & Goesele, M. (2015). MVE—An image-based reconstruction environment. *Computers and Graphics*, *53*(A), 44–53.

Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.

Irschara, A., Zach, C., Frahm, J. M., & Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 2599–2606).

Judd, T., Durand, F., & Adelson, E. (2007). Apparent ridges for line drawing. *ACM Transactions on Graphics*, *26*(3), 19.

Kendall, A., & Cipolla, R. (2016). Modelling uncertainty in deep learning for camera relocalization. In *IEEE international conference on robotics and automation* (ICRA) (pp. 4762–4769).

Kendall, A., Grimes, M., & Cipolla, R. (2015). PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *European conference on computer vision* (ICCV) (pp. 2938–2946).

Kim, J., Liu, C., Sha, F., & Grauman, K. (2013). Deformable spatial pyramid matching for fast dense correspondences. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 2307–2314).

Kneip, L., Scaramuzza, D., & Siegwart, R. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 2969–2976).

Krull, A., Brachmann, E., Michel, F., Yang, M. Y., Gumhold, S., & Rother, C. (2015). Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. In *European conference on computer vision* (ICCV) (pp. 954–962).

Lepetit, V., Moreno-Noguer, F., & Fua, P. (2009). EPnP: Efficient perspective-n-point camera pose estimation. *International Journal of Computer Vision*, *81*(2), 155–166.

Li, X., Wu, C., Zach, C., Lazebnik, S., & Frahm, J. M. (2008). Modeling and recognition of landmark image collections using iconic scene graphs. In D. Forsyth, P. Torr, & A. Zisserman (Eds.) *European conference on computer vision* (ECCV), Springer, Lecture Notes in Computer Science (Vol. 5302, pp. 427–440).

Li, Y., Snavely, N., & Huttenlocher, D. P. (2010). Location recognition using prioritized feature matching. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.) *European conference on computer vision* (ECCV), Springer, Lecture Notes in Computer Science (Vol. 6312, pp. 791–804).

Lim, J. J., Pirsiavash, H., & Torralba, A. (2013a). Parsing IKEA objects: Fine pose estimation. In *IEEE international conference on computer vision* (ICCV) (pp. 2992–2999).

Lim, J. J., Zitnick, C. L., & Dollar, P. (2013b). Sketch tokens: A learned mid-level representation for contour and object detection. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 3158–3165).

Liu, C., Yuen, J., & Torralba, A. (2011). SIFT flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 978–994.

Liu, L., & Stamos, I. (2012). A systematic approach for 2D-image to 3D-range registration in urban environments. *Computer Vision and Image Understanding*, *116*(1), 25–37.

Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 441–450.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Massa, F., Russell, B. C., & Aubry, M. (2016). Deep exemplar 2D–3D detection by adapting from real to rendered views. In *IEEE conference on computer vision and pattern recognition* (CVPR) (pp. 6024–6033).

Matas, J., & Chum, O. (2002). Randomized RANSAC with Td,d test. In *British machine vision conference* (BMVC) (pp. 448–457).

Matzen, K., & Snavely, N. (2014). Scene chronology. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.) *European conference on computer vision* (ECCV), Springer, Lecture Notes in Computer Science (Vol. 8695, pp. 615–630).

Neugebauer, P. J., & Klein, K. (1999). Texturing 3D models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, *18*(3), 245–256.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3), 145–175.

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE conference on computer vision and pattern recognition* (CVPR).

Plötz, T., & Roth, S. (2015). Registering images to untextured geometry using average shading gradients. In *IEEE international conference on computer vision* (ICCV) (pp. 2030–2038).

Russell, B. C., Sivic, J., Ponce, J., & Dessales, H. (2011). Automatic alignment of paintings and photographs depicting a 3D scene. In *International IEEE workshop on 3D representation and recognition* (3dRR) (pp. 545–552).

Shan, Q., Wu, C., Curless, B., Furukawa, Y., Hernandez, C., & Seitz, S. M. (2014). Accurate geo-registration by ground-to-aerial image matching. In *IEEE international conference on 3D vision* (3DV) (pp. 525–532).

Shanmugam, P., & Arikan, O. (2007). Hardware accelerated ambient occlusion techniques on GPUs. In *Symposium on interactive 3D graphics and games* (pp. 73–80).

Shrivastava, A., Malisiewicz, T., Gupta, A., & Efros, A. A. (2011). Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics*, *30*(6), 154.

Sibbing, D., Sattler, T., Leibe, B., & Kobbelt, L. (2013). SIFT-realistic rendering. In *IEEE international conference on 3D vision* (3DV) (pp. 56–63).

Sipiran, I., & Bustos, B. (2011). Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. *The Visual Computer*, *27*(11), 963–976.

Stark, M., Goesele, M., & Schiele, B. (2010). Back to the future: Learning shape models from 3D CAD data. In *British machine vision conference* (BMVC).

Su, H., Qi, C. R., Li, Y., & Guibas, L. J. (2015). Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In *European conference on computer vision* (ICCV) (pp. 2686–2694).

Viola, P., & Wells, W. M. I. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, *24*(2), 137–154.

Wendel, A., Irschara, A., & Bischof, H. (2011). Natural landmark-based monocular localization for MAVs. In *IEEE international conference on robotics and automation* (ICRA) (pp. 5792–5799).

Zia, M. Z., Stark, M., Schiele, B., & Schindler, K. (2013). Detailed 3D representations for object recognition and modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2608–2623.