Just look at the image: viewpoint-specific surface normal prediction for improved multi-view reconstruction

Silvano Galliani Konrad Schindler

Photogrammetry and Remote Sensing, ETH Zurich
{firstname.lastname@geod.baug.ethz.ch}

Abstract

We present a multi-view reconstruction method that combines conventional multi-view stereo (MVS) with appearance-based normal prediction, to obtain dense and accurate 3D surface models. Reliable surface normals reconstructed from multi-view correspondence serve as training data for a convolutional neural network (CNN), which predicts continuous normal vectors from raw image patches. By training from known points in the same image, the prediction is specifically tailored to the materials and lighting conditions of the particular scene, as well as to the precise camera viewpoint. It is therefore a lot easier to learn than generic single-view normal estimation. The estimated normal maps, together with the known depth values from MVS, are integrated to dense depth maps, which in turn are fused into a 3D model. Experiments on the DTU dataset show that our method delivers 3D reconstructions with the same accuracy as MVS, but with significantly higher completeness.

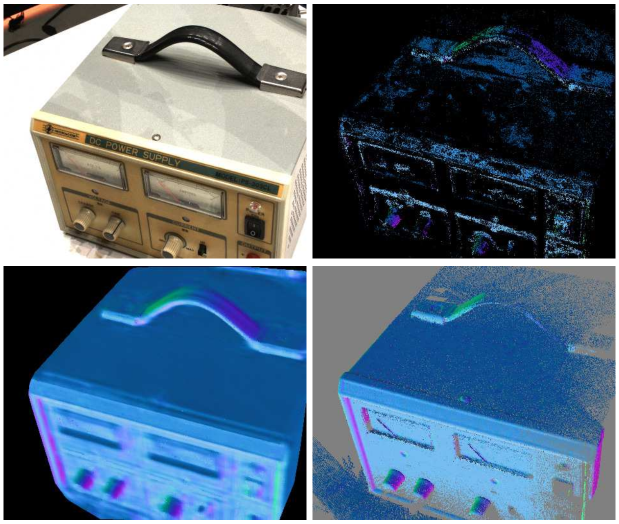

Figure 1. Illustration of our reconstruction method. Given an input image (**top left**) and an incomplete normal map from multi-view stereo (**top right**), we reconstruct the missing normals by CNN regression on the image (**bottom left**). The normal maps are then integrated to dense 3D models (not shown). **Bottom right**: Ground truth.

1. Introduction

The reconstruction of 3D surfaces from images is a central problem of computer vision. The dominant approach is multi-view stereo (MVS): densely match image points in multiple views with known camera poses, then triangulate the corresponding rays to 3D points. MVS algorithms have greatly improved over the past decades and nowadays deliver high-quality point clouds, respectively surfaces derived from those point clouds [23, 34]. Yet MVS, being based on point correspondences between different images, only works in areas with sufficient texture. If no correspondence can be established, the methods fails. Most commonly this happens in surface regions with uniform albedo and on specular highlights, where matching is ambiguous due to a lack of high-frequency brightness/color variations. A further recurrent problem are occlusions, where many viewing rays are blocked and do not reach the surface point. In such regions the only options are to either not reconstruct 3D points, leaving holes in the surface; or to interpolate, which can lead to inaccurate or even totally wrong results.

We propose to fill in the missing regions with the help of shading information. It is well-known that, complementary to MVS, shape reconstruction from shading requires only a single view, and works best for uniform albedo. Yet, recovering 3D surface normals from shading has proved remarkably difficult in practice, mainly because a number of important influence factors are hard to model. In real data the illumination can be quite complex and the illumination direction(s) are not exactly known. Most importantly, the reflectance properties (the bi-directional reflectance distribution function, or BRDF) of the surfaces in the scene are usually unknown.

The starting point for the present paper is that if one wants to reconstruct surface shape from shading, it might not be necessary to model the global illumination and the complete reflectance distribution. Rather, one only needs to cover the specific illumination, viewpoint and surface properties that are present in a given image. We exploit this by implicitly learning the view-specific shading patterns in a discriminative manner. Given that in most images there are pixels for which the surface normals are known (from 3D points reconstructed via multi-view stereo), we propose to learn a regression directly from raw RGB patches to surface normal directions, using a convolutional neural network (CNN).

In contrast to other recent work that predicts surface normals in a purely data-driven fashion [10, 27, 32] we do *not* aim for generality across different lighting and viewing conditions, and thus do not need a diverse training set that covers all possible conditions. Rather, we learn an individual, view-specific shading model per image, trained on reprojected 3D normals that we reconstruct from high-confidence MVS points. Such a model only needs to cover a subset of the BRDFs of (usually few) visible materials, under constant lighting, thus it can be expected to predict more accurately. *I.e.*, we argue that the image itself, together with an incomplete range/disparity image, contains sufficient information to predict surface orientation, without a globally valid shading model.

Our method is able to estimate surface normals with an accuracy similar to (sometimes even slightly better than) that of the training data. To complete the pipeline we integrate the dense normal field per image, together with the known 3D points from MVS, into a dense and hole-free depth map, and fuse the depthmaps from multiple views to obtain a more complete 3D model.

2. Related works

Normals in MVS. Many multi-view stereo methods only estimate depth, *e.g*. [7, 30, 36]. If normal vectors are required, they are found in post-processing by fitting local tangent planes to the point cloud [21, 29]. There are however a number of MVS methods that explicitly reconstruct the local tangent plane as part of their internal parametrization, and thus directly deliver surface normals on top of depth maps (respectively, 3D points). Notable examples include the well-known PMVS method [16], as well as the multi-view variant [18] of the PatchMatch stereo algorithm [5]. Methods that directly deliver normals at the reconstructed surface points naturally lend themselves to our problem. We use [18], on the one hand for its computational efficiency, and on the other hand because it provides an explicit parameter to trade off completeness *vs*. accuracy and ensure sufficiently clean training normals.

There are also methods which from the beginning constrain MVS reconstruction with strong a-priori assumptions about the surface normal. *E.g*., Zeisl *et al*. [42] focus on indoor scenarios consisting only of horizontal floor and ceiling planes connected by vertical walls. Furukawa *et al*. [15] go even further and assume a Manhattan world [8]. At the extreme end of the spectrum (though somewhat outside the scope of our work) come model-based methods, which align the images with an existing 3D template of the object and reconstruct by deforming the template to better fit the geometric or photometric evidence, *e.g*. [26, 37].

Use of shading cues in MVS. The first attempts to combine multi-view geometry and shading for 3D reconstruction date back at least 30 years [4]. Since then, the topic has been somewhat overshadowed by the development of pure stereo, respectively multi-view matching, but has received constant attention [9, 14, 33]. The complex interplay between surface orientation, light sources, and surface BRDFs proved difficult to handle outside the lab, and most works focus on one of these components. Wu *et al*. [38] assume a Lambertian surface but consider general illumination, approximating the incoming illumination with spherical harmonics. Jin *et al*. [25] propose a joint variational framework for the estimation of shape, normal and a single light source, assuming a Lambertian surface with piecewise constant albedo. Haines and Wilson [19] integrate information from shading and stereo via belief propagation to estimate fine surface details. Beeler *et al*. [3] detect and eliminate ambient occlusion to improve surface estimation.

Surface normal estimation. A number of recent works have posed surface normal prediction as a machine learning problem. Fouhey *et al*. [11] mine for distinctive, repeatedly occurring shape and appearance primitives in indoor RGB-D data, and match those primitives to new images to obtain a normal map. Later that method was augmented with shape priors for rooms and an explicit model of crease edges [12]. Ladicky *et al*. [27] directly predict normals from image features extracted in a pixel's neighborhood. They turn normal estimation into a classification problem, by clustering the normals to a discrete set of directions on the unit sphere and interpolating between neighboring directions. Instead, Eigen and Fergus [10] learn a direct regression from image to normal (alternatively also to depth or semantic label) with a multi-scale convolutional architecture.

These methods are related to ours in that they pose normal estimation as a learning problem, and in some cases also use CNNs as regression engine. Beyond this technical similarity, there are however two fundamental differences. On the one hand, our model is more specific w.r.t. illumination and reflectance: we do not learn a generic model that is supposed to cover the shading behavior of "the world", or at least of an entire dataset; rather we rely on MVS to generate sparse training data tailored to the specific image, such that for that image the prediction is more accurate, while no external training data is needed. On the other hand, our model is more generic w.r.t. geometry. We rely only on the local shading and the position in the image, but do not depend on the presence of a small number of vanishing directions or recurrent geometric primitives (such as for example those present in the NYU2 Dataset [35]).

Richter and Roth [32] also relax the requirement for external training data and instead use synthetic training data. They assume knowledge of the object's silhouette in the image. The distance from the silhouette is used to guess a rough initial normal map, which in turn serves to derive a quadratic approximation of the reflectance map and relight the synthetic training data appropriately.

Normal extrapolation from MVS. Few authors have explored the idea to use an incomplete cloud of MVS points as reference for normal prediction. Xu *et al*. [40] seemingly also use the appearance around known points/normals, together with smoothness of the normal field, to fill holes in an image-based surface reconstruction. Unfortunately, no details are given in their paper. Ackermann *et al*. [1] use MVS to bootstrap photometric stereo. Instead of directly modeling lighting and reflectance, they extract per-pixel material coefficients at the MVS points and predict unknown normals by minimizing the photometric differences to the known points.

Integrating normals to surfaces. Shading-based methods in most cases estimate normal vectors, which still need to be integrated to surfaces. Reconstructing a function from known gradients is a classic problem in computational geometry as well as in computer vision. Perhaps the most popular method, already employed by Horn and Brooks [22], is to solve the Poisson equation that arises as a necessary condition in variational least-squares reconstruction. Here we also follow this standard approach. It has also been attempted to replace the least-squares error function by more robust norms to improve the robustness to outliers [2]. Some authors prefer to use the computationally more efficient eikonal equation [17, 20]. Further approaches include integration in the frequency domain [13], which is limited to dense vector fields; and direct line-by-line integration, which only works for noise-free data [39].

3. Method

We start with an overview of our complete surface reconstruction pipeline. As input data, we require multiple images of the same scene, with known camera poses. The first step is a conventional MVS reconstruction. We use a multi-view version of PatchMatch Stereo. That method has been shown to deliver state-of-the-art performance [18], and it returns point-wise normals in 3D scene space as a byproduct; but other algorithms could be plugged in as well. The next step is to predict normals for pixels where multi-view stereo failed to compute a reliable depth. This is done separately for every viewpoint. From the points reconstructed successfully by MVS, we train a convolution neural network (CNN) to perform regression from raw image patches to surface normals. With the network, we densely predict all missing normals (Sec. 3.2). The dense vector fields are turned into a 3D surface model by first integrating them to depth maps with masked Poisson reconstruction (Sec. 3.3) and then fusing the depth and normal maps from multiple views.[1]

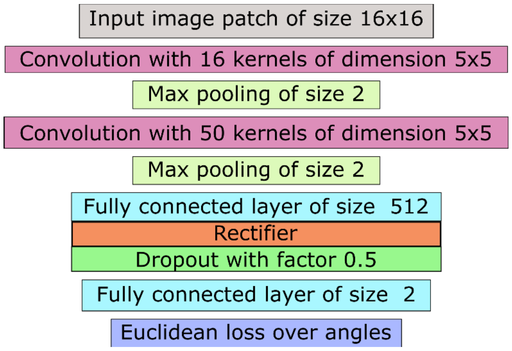
Input image patch of size 16x16
Convolution with 16 kernels of dimension 5x5
Max pooling of size 2
Convolution with 50 kernels of dimension 5x5
Max pooling of size 2
Fully connected layer of size 512
Rectifier
Dropout with factor 0.5
Fully connected layer of size 2
Euclidean loss over angles

Figure 2. CNN architecture for regression from image patches to surface normals.

3.1. Generation of normals for training

The first stage of our method is a standard MVS reconstruction to obtain an initial (incomplete) cloud of reliable 3D object points. Among the many available algorithms we choose the fast multi-view PatchMatch implementation of [18]. In a nutshell, that method first generates a depthmap in each camera, by propagating depth values along slanted tangent planes of the surface so as to maximize photo-consistency across multiple views. In a second step it employs a consensus mechanism to robustly fuse the individual depth maps into a 3D point cloud. We pick this method for two reasons. On the one hand, it computes and outputs, by construction, not only 3D points but also explicit surface normals at those points. Since our further processing needs those normals, PatchMatch is a natural fit. On the other hand, the depthmap fusion relies on a consensus mechanism that checks both the consistency of the depth values *and* of the normal directions across several views. As a result, points with unreliable normals are discarded during fusion, which is important for our purposes, since

[1] The integration and fusion steps could potentially be solved jointly. We prefer to keep them separate, which is more efficient and adds a further checkpoint to explicitly identify inconsistencies between different views.

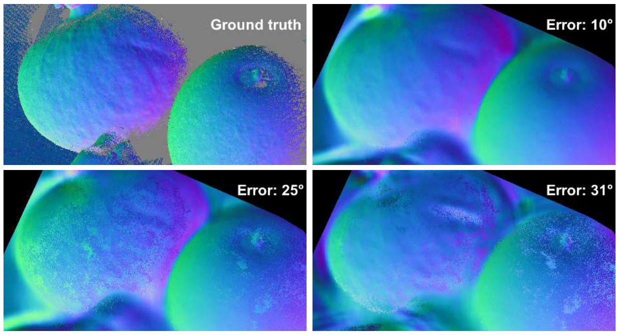

Figure 3. Comparison of different strategies for normal prediction. A model trained for a specific image (**top right**) works better than one trained on multiple views of the same scene (**bottom left**) or a generic model trained for a whole database (**bottom right**).

those normals will later serve as training data. It is interesting to note that the method achieves a high completeness of the MVS reconstructions [18], in spite of the rather strict consistency check.

3.2. Normal prediction

The philosophy of our second, shading-based stage is to learn the relation between surface normals and the appearance of the corresponding surface patches. That relation can then be used to predict surface normals at locations where no MVS points could be reconstructed. As explained, we prefer to initially do this on a per-image basis and again fuse the results afterwards. Estimating the normals individually in each image simplifies the learning problem, because in a single exposure the lighting conditions are constant; and it also simplifies the implementation, because one can work on the pixel grid rather than discretise the 3D scene surfaces.

We also experimented with a single model for all views, effectively trying to learn the shading variation for a given object, under any viewpoint. This did not work well, see Fig. 3. We see two possible reasons. On the one hand, the learning problem obviously gets a lot more complicated and ambiguous if one has to cover two additional degrees of freedom (for the viewing direction) in the BRDF. On the other hand, it may well be that for certain materials the CNN also learns context and texture cues that are not independent of the viewpoint.

Training data. As part of the MVS reconstruction, we have a surface normal map for each individual view, which holds, at every pixel, either a normal vector in camera-centric coordinates or a flag that no normal could be reconstructed. In order to ensure clean training data for CNN training, we filter those surface normal maps. Our goal at this point is high precision even at the cost of a bit lower recall, *i.e*. we try to ensure that only correct and accurate normals are retained. As a first filter, we remove all normals that did not survive the multi-view fusion (meaning that they did not fit the consensus). For those pixels which did contribute to the reconstruction of a 3D normal vector, we reproject the 3D vector and replace the original entry. This can be expected to improve the accuracy of the valid normals, because the inliers to the consensus voting are averaged during fusion to suppress noise. On very slanted surfaces it can, in rare cases, happen that the averaged normal points away from the camera; such normals are discarded. The final normal maps have entries only where the original matcher found a depth, and thus also a normal, and that depth and normal were confirmed as correct and visible by a consensus over multiple viewpoints.

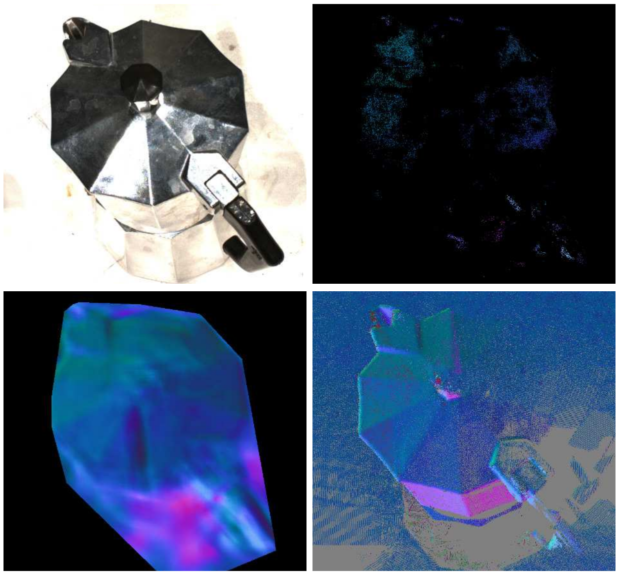

Figure 4. Normal prediction for a particularly difficult scene (DTU object n.° 77). Even with few training points of a highly specular object the regressor is able to recover reasonable normals in many regions. **Top left**: input image. **Top right**: training normals from MVS. **Bottom left**: predicted normals. **Bottom right**: ground truth.

Regression with CNN. Having found a set of reliable normal vectors to serve as training data, we learn, separately for each view, a convolutional neural network (CNN) to predict unknown surface normals. Note that no manually labelled training data is required, the regressor is trained only from automatically reconstructed MVS points. As input, the network takes 16 × 16 pixel RGB patches, downsampled from 64 × 64 pixel patches of the original image. As output, it returns the estimated normal vector at the center pixel of the patch, parameterized by two polar angles θ and ϕ (a.k.a. *azimuth* and *elevation*, or *yaw* and *tilt*). The patch size has been determined empirically: much smaller patches do not work as well, it seems that they do not capture sufficient shading information; larger patches slow down the computation without improving performance.

As loss function, we directly minimize the minimal planar angle $\alpha = \langle \mathbf{n}_{\text{true}}, \mathbf{n}_{\text{pred}} \rangle$ between the true normal and the predicted one. Our architecture follows the *LeNet* framework [28]: a convolution layer with 16 kernels of window size 5×5, followed by max-pooling over 2×2 blocks; a second convolutional layer with 50 kernels of size 5×5, again followed by 2×2 max-pooling; a fully connected layer of 512 neurons, with $ReLU$ rectification and 50% drop-out; and a final fully connected layer with 2 output neurons for the angles θ and ϕ; See Fig. 2. The network is implemented in the Caffe framework [24], and trained with stochastic gradient descent, with a fixed momentum of 0.9 and a learning rate of 0.001. Training and prediction take ≈30 min per view, on a single PC.

It is clear that several other regression methods, like for example regression forests, would be computationally more efficient. We plan to test alternative regressors in future work. The following reason motivated us to use a CNN: the perhaps biggest strength of CNNs and related deep learning methods, and the main reason for their phenomenal success in computer vision, has been the capability to learn good image representations from raw RGB data. We feel that this end-to-end learning, which relieves us from finding a suitable feature set, is particularly useful for our problem. Compared to well-researched vision tasks like pedestrian detection or semantic segmentation, little is known about the right choice of features for discriminative normal estimation, hence finding good features might end up being a lengthy trial-and-error process. We also point out that in the recent work of [43] CNNs were shown to perform well (and superior to regression forests) for a related regression task from visual appearance to a spatial direction, namely image-based gaze estimation.

After training the regressor, we apply it to the same image, and estimate normal vectors densely for all pixels except for the training data, which already possess normals from MVS. To avoid excessive extrapolation, we only predict inside the convex hull of the training pixels.

3.3. Surface normal integration

The previous step yields a dense map of normals for every viewpoint. Since our goal is 3D surface reconstruction, we need to convert that normal map into a dense depth map, which however is constrained to pass through the known depth values from MVS. We do this with a masked version of the 2D Poisson equation. Formally, we face an interpolation problem: interpolate depth values at all points not reconstructed by MVS, such that they best agree with the predicted surface normals. To distinguish points with known MVS depth from those without one, we define two separate depth functions: f_{mvs} for MVS points is known, whereas f is the unknown to be recovered. The domain of f_{mvs} is only the discrete set $\mathcal{A}$ of MVS points, and f is defined everywhere in the image plane Ω excepts at the points $\mathcal{A}$. The vector field g consists of the gradients of both functions,

$$\forall x \in \Omega : g(x) = \begin{cases} \nabla f_{mvs}, & if\ x \in \mathcal{A} \\ \nabla f, & else \end{cases} \tag{1}$$

Our task is to find an interpolant f over $\Omega \backslash \mathcal{A}$ that minimizes the squared error

$$\min_f \iint_{\Omega \backslash \mathcal{A}} \|\nabla f - g\|^2 \quad . \tag{2}$$

This leads to the Poisson equation

$$\Delta f = \operatorname{div} g \ , \tag{3}$$

with $\operatorname{div}(\cdot)$ the divergence operator and $\Delta(\cdot)$ the Laplacian. The MVS points in $\mathcal{A}$ each contribute a Dirichlet boundary condition, ensuring that the depth map will pass through f_{mvs}. Together with standard von Neumann boundary conditions at the image border the equation has a unique solution. Since the domain is irregular, one must fall back to an iterative solver for (3), we use the Gauss-Seidel scheme with successive overrelaxation [31, 41].

3.4. Depth map fusion

Our setting is that we have multiple overlapping views of a scene — otherwise we could not perform MVS reconstruction. Having recovered depth maps in all these views, the last step is to fuse them into a consistent 3D model. We apply a robust consensus mechanism across different views, similar to the one in the MVS step, to filter out incorrect depth values and at the same time denoise correct ones.

To minimize the number of outliers we prefer to do the filtering conservatively, *i.e*. examine every depth map individually and remove all points whose 3D scene space coordinates are not consistent with other depth maps. Let Π_i^{-1}, Π_i be the forward, respectively backward projection operators between a camera C_i and the 3D scene space. For every image C_* in turn, we forward-project the points p_* of the disparity map d_* into 3D points, and back-project those points to other cameras $\{C_i\}$ whose viewfields overlap with the one of C_*. Which viewfields overlap can easily be determined from the known camera poses and is already known from the initial MVS step. In each C_i we test two conditions: the disparity $\Pi_i(\Pi_*^{-1}(d_*))$ should coincide with the observed value d_i, up to a threshold ε. Our default value is $\varepsilon = 0.3$ pixel. And the angle between the projected normal vector $\Pi_i(\Pi_*^{-1}(n_*))$ and the observed n_i should also lie below a threshold β. We set $\beta = 10°$.

If both conditions are fulfilled in $K \geq 3$ other cameras, then we warp the corresponding points from all consistent views into scene space and average the 3D points $\Pi_i^{-1}(p_i)$

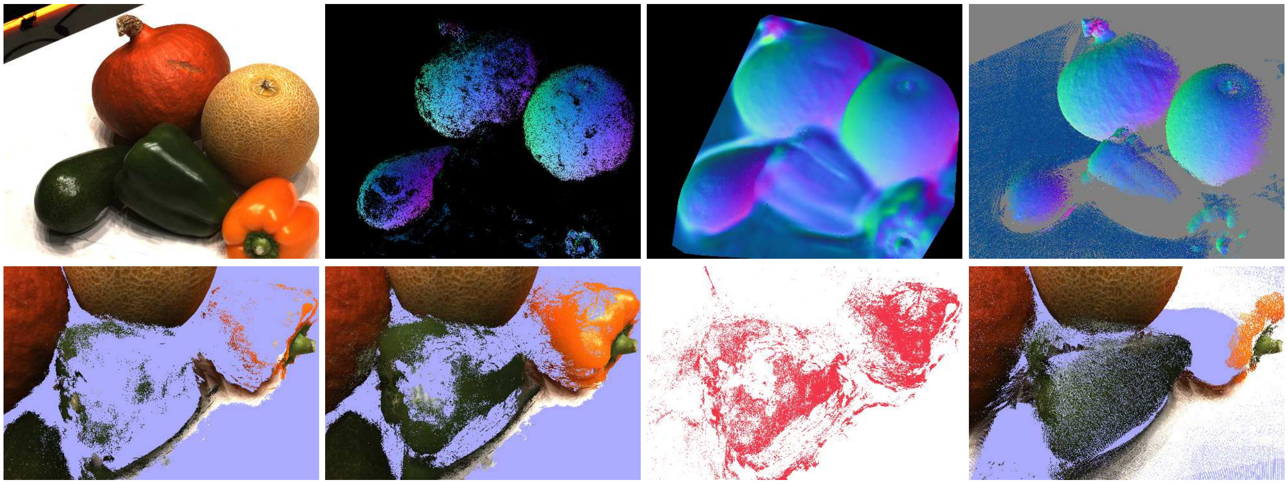

Figure 5. **Top**: Normal prediction: input image, training points, reconstruction, ground truth. **Bottom**: Reconstruction closeup of the peppers after normal integration and depth fusion. **From left to right**: Input. Our result. Difference. Ground truth. Our reconstruction with normal prediction is able to complete parts missed both by MVS and by the structured light scanner used for the ground truth.

and the normals $\Pi_i^{-1}(n_i)$ to suppress noise. Otherwise, if fewer than K other views confirm the estimate (d_*, n_*), the point is discarded.

Obviously the strict consistency check means that quite many of the points reconstructed by the normal prediction will be rejected. Still, a significant portion survives. This shows that in many locations the appearance-based normal prediction (and subsequent integration) yields comparable accuracy to multi-view stereo, which uses similar fusion criteria. Obviously, the fusion parameters ε, β, K provide a simple interface to tune accuracy *vs*. completeness of the reconstruction. With strict values, fewer but more reliable points survive (*e.g*., for applications in industrial metrology). With more generous settings the completeness of the reconstruction increases, at the cost of lower accuracy (*e.g*., for graphics and visualization purposes).

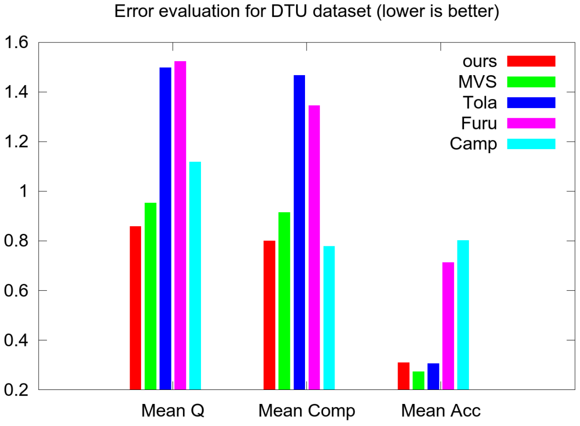

Figure 6. Quantitative comparison with our initialization and other pure MVS methods [6, 16, 36]. Lower values are better.

4. Results

To validate our method, we use a subset of 14 objects from the extensive DTU multi-view stereo dataset [23]. The dataset is, to our knowledge, the only large MVS testbed that is publicly available. It features a variety of objects and materials, and provides complete coverage with 49 images per scene. Ground truth of adequate density has been recorded with a structured light scanner. The large selection of shapes and materials, ranging from simple diffuse surfaces to specular plastic and metal objects, is well-suited to test our normal prediction under realistic conditions. Importantly, the dataset is difficult enough to challenge multi-view stereo: even state-of-the-art methods, including the one that we use for MVS [18], do not manage to reconstruct large parts of some scenes. And it is also complex enough to defy shading methods based on simple Lambertian reflectance, with materials of different color, texture and specularity. We use the variant of the data recorded under standard (relatively diffuse) lighting conditions, because this is the only one for which multiple recent works have reported results. In principle it would be possible (and potentially beneficial) for our method to include images with various lighting conditions, in the hope that a certain illumination is better suited for certain parts of the scene than others.

We first evaluate the surface normal prediction separately, and then present an end-to-end comparison with the final 3D reconstructions.

4.1. Normal prediction

To quantify the accuracy of the normals predicted by the CNN, we measure the angular error w.r.t. to ground truth normal derived from the reference point cloud. As a first step, we compare the error on the "test" normals predicted by the regression to the one for the "training" normals estimated by MVS. Ideally, these errors should be similar, meaning that the appearance-based predictions would be as good as the multi-view estimates. We observe, not surprisingly, that the relation depends a lot on the difficulty of the scene. For simple, piecewise planar objects with little reflection, the predicted normals are even slightly more accurate than the training normals. Presumably, this is so because the learning problem is easy, and the "averaging" over training samples from the same surface reduces noise. *E.g.*, although the MVS result is rather sparse in Fig. 1, it is sufficient to obtain sensible predictions for most of the object. The corresponding mean and median errors are 13° and 9°, respectively, for the MVS points; and 12°, respectively 6° for the CNN prediction.

On the contrary, specular materials and complicated surface geometry, *e.g.* sharp creases, make the prediction more difficult. The most difficult object in the DTU database is the coffee-maker in Fig. 4. Even in that case, the appearance-based regression surprisingly gives reasonable predictions in many parts. However, the mean and median errors rise from 13° and 10° at the MVS points to 17° and 12° for the predicted ones.

In Fig. 5 the shiny surface of the peppers poses a serious problem for both MVS and for the structured light scanner that acquired the ground truth. Our method is able to predict normals in these areas. While the mean and median errors are significantly higher than at the MVS points (17° and 10°, compared to 8° and 6°), they are still good enough to reconstruct an important part of the missing surfaces to a depth accuracy of 0.3 pixels in disparity. Note that especially on the yellow pepper our reconstruction is also a lot more complete than the ground truth from the structured light scanner, which fails on very specular surfaces, too.

Over all 14 objects, the mean angular error is 11° for the training normals from MVS, and 18° for the predicted normals. The mean-of-median over all objects is 9° for the MVS normals and 16° for the predicted ones. The mean is consistently only a bit above the median, which indicates a relatively even error distribution not contaminated by many large outliers.

4.2. Improved multi-view reconstruction

Our overall goal is a better reconstruction of 3D point clouds, respectively surfaces. We thus go on to quantify the accuracy and completeness of the resulting 3D models. As baselines, we use the initial MVS reconstruction without normal prediction, as well as three further MVS methods for which results on DTU are available.

To ensure a fair comparison to pure MVS, we set the same fusion parameters (Sec. 3.4) both for fusing MVS depthmaps and for fusing depthmaps after normal integration. *I.e.*, points found with shading are added to the MVS reconstruction only if they fulfill the same strict reliability criteria.

Fig. 6 shows quantitative results averaged over all reconstructed objects. The proposed prediction and integration of the normals improves the mean completeness of the MVS initialization by 14 %, at the cost of a negligible increase in accuracy (accuracy is measured only at the reconstructed points, hence an improvement is virtually impossible when adding additional points to an existing, sparse reconstruction). Moreover, our results compare favorably w.r.t. other methods. In terms of accuracy, we are on par with the best result by [36], but with much higher completeness (≈ 83% better). In terms of completeness, we are second best, narrowly behind [6], which however has a lot lower accuracy (61 % higher error).

Any multi-view reconstruction method can trade off accuracy against completeness. Tuning for high accuracy means strict consistency checks that reject many points and drive down completeness. Conversely, tuning for completeness means accepting more points, even if they have higher error. We thus also compute the overall *quality* of a reconstruction, defined as the geometric mean of accuracy and completeness $Q = \sqrt{acc^2 + prec^2}$, similar in spirit to the $F1$-score. On that measure our method clearly performs best, leading by 11% over the MVS initialization, and 30% over the next best method.

We end with some qualitative examples to illustrate where the proposed normal prediction can help. Overall, the experiments confirm the intuition that the prediction will fill in holes in homogeneous areas, where MVS struggles. A prime example is the bunny in Fig. 7. MVS does alright on the fur, but can only reconstruct the textured part of the earmuffs. Still, there are enough points on the earmuffs to learn the normal prediction, hence a good part of the untextured orange plastic gets filled in. The white stripes on the vases in Fig. 7, also challenge MVS. This is an example for a material with a non-lambertian shading component, nevertheless the prediction fills in a large part of the missing surface. The plastic packaging in Fig. 7 is even more challenging, with multiple colors as well as specularity. Note how the regression predicts adequate normals for different parts including the blue area at the bottom, the yellow/white area in the center, and even the shadow area on the red object behind the bag.

5. Conclusion

We have described a method to densify multi-view stereo reconstructions with the help of shading cues. Like some other recent methods, we sidestep analytic shading models. Instead, we view surface normal estimation as a discriminative regression problem and train a CNN to predict normal vectors from raw image patches. The basic insight is that the regression problem can be greatly simplified if one sac-

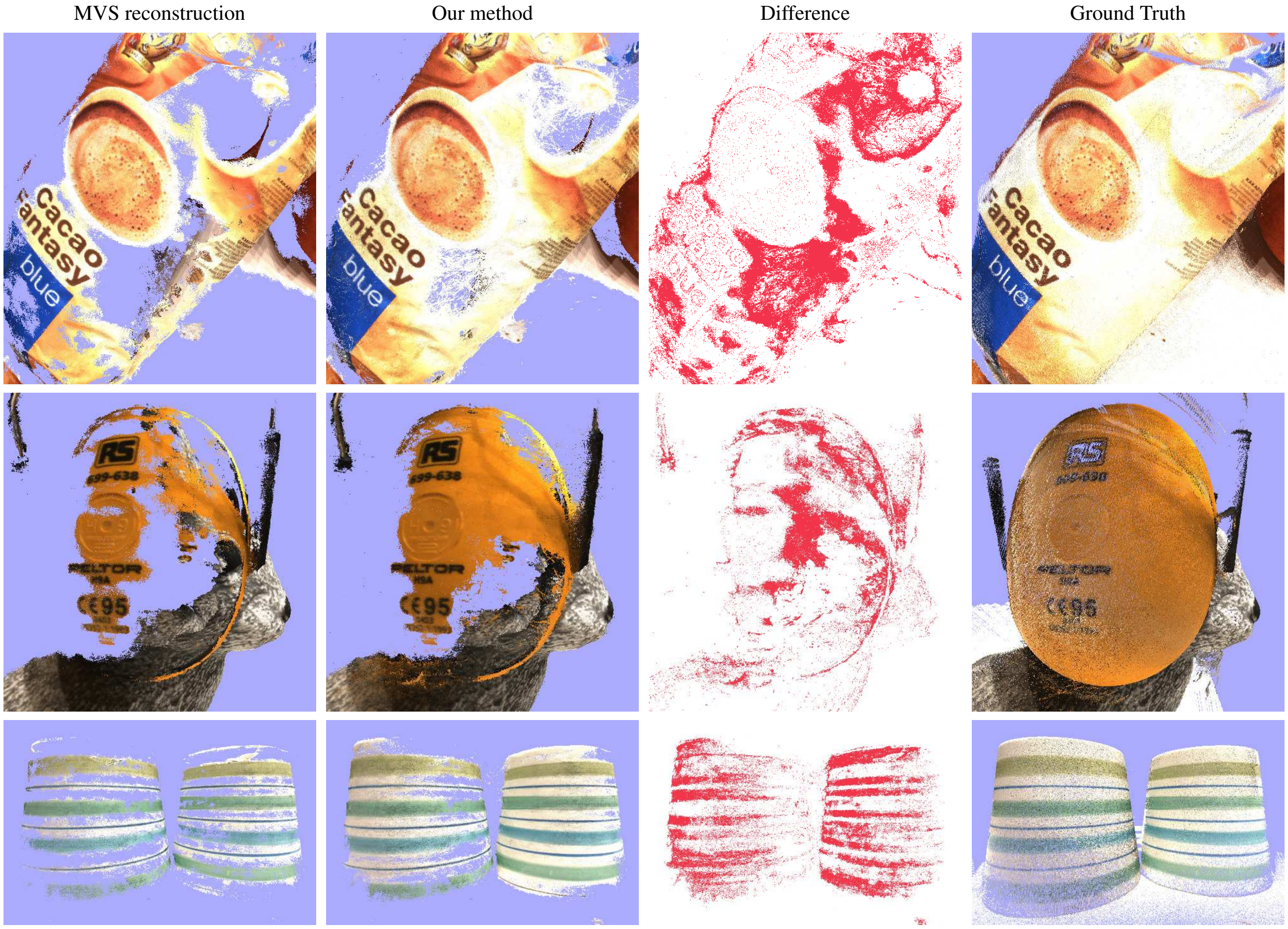

Figure 7. Reconstruction improvements of our method. **Top**: Challenging object with multiple colors and with specularities. **Middle**: Object with homogeneous colour. **Bottom**: Vase with over-exposed and homogeneous white areas.

rifices generality and learns an individual predictor for the fixed illumination, viewpoint and scene properties of each specific image. The prediction is embedded in a conventional multi-view reconstruction pipeline: point successfully reconstructed via multi-view correspondence form the training set for normal estimation, and the resulting dense normal maps are integrated to depth maps to improve the 3D model.

A main message of our paper is that even a rather small number of training examples are enough to learn normal estimation from raw intensities, if the problem is tightly constrained. For a particular view of a particular scene, it is indeed possible to infer shape by *just looking at the image*, with an accuracy similar to the one of MVS.

So far our method only fills in missing depth measurements. The original MVS points are not modified, and depth map fusion is done in a separate step. In future work we plan to investigate an early fusion, which directly reconstructs the 3D surface from multiple normal maps and sparse depth measurements.

References

[1] J. Ackermann, M. Ritz, A. Stork, and M. Goesele. Removing the example from example-based photometric stereo. In *Trends and Topics in Computer Vision*, pages 197–210. Springer, 2010.

[2] A. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? *ECCV 2006*.

[3] T. Beeler, D. Bradley, H. Zimmer, and M. Gross. Improved reconstruction of deforming surfaces by cancelling ambient occlusion. *ECCV 2012*.

[4] A. Blake, A. Zisserman, and G. Knowles. Surface descriptions from stereo and shading. *Image and Vision Computing*, 3(4):183–191, 1985.

[5] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo – stereo matching with slanted support windows. *BMVC 2011*.

[6] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. *ECCV 2008*.

[7] R. T. Collins. A space-sweep approach to true multi-image matching. *CVPR 1996*.

[8] J. M. Coughlan and A. L. Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. *NIPS 2000*.

[9] J. E. Cryer, P.-S. Tsai, and M. Shah. Integration of shape from shading and stereo. *Pattern Recognition*, 28(7):1033–1043, 1995.

[10] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV 2015*.

[11] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3d primitives for single image understanding. *ICCV 2013*.

[12] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. *ECCV 2014*.

[13] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE TPAMI*, 10(4):439–451, 1988.

[14] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 16(1):35–56, 1995.

[15] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. *CVPR 2009*.

[16] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI*, 32(8):1362–1376, 2010.

[17] S. Galliani, M. Breuß, and Y. C. Ju. Fast and robust surface normal integration by a discrete eikonal equation. *BMVC 2012*.

[18] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. *ICCV 2015*.

[19] T. S. F. Haines and R. C. Wilson. Integrating stereo with shape-from-shading derived orientation information. *BMVC 2007*.

[20] J. Ho, J. Lim, M.-H. Yang, and D. Kriegman. Integrating surface normal vectors using fast marching method. *ECCV 2006*.

[21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[22] B. K. Horn and M. J. Brooks. The variational approach to shape from shading. *CVGIP*, 33(2):174–208, 1986.

[23] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. *CVPR 2014*.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[25] H. Jin, D. Cremers, D. Wang, E. Prados, A. Yezzi, and S. Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *IJCV*, 76(3):245–256, 2008.

[26] I. Kemelmacher-Shlizerman and S. M. Seitz. Face reconstruction in the wild. *ICCV 2011*.

[27] Ľ. Ladický, B. Zeisl, and M. Pollefeys. Discriminatively trained dense surface normal estimation. *ECCV 2014*.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *Int'l J Computational Geometry & Applications*, 14(4/5):261–276, 2004.

[30] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE TPAMI*, 15(4):353–363, 1993.

[31] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.

[32] S. R. Richter and S. Roth. Discriminative shape from shading in uncalibrated illumination. *CVPR 2015*.

[33] D. Samaras, D. Metaxas, P. Fua, and Y. G. Leclerc. Variable albedo surface reconstruction from stereo and shape from shading. *CVPR 2000*.

[34] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *CVPR 2006*.

[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *ECCV 2012*.

[36] E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *MVA*, 23(5):903–920, 2012.

[37] C. Wallraven, V. Blanz, and T. Vetter. 3D-reconstruction of faces: Combining stereo with class-based knowledge. *DAGM 1999*.

[38] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. High-quality shape from multi-view stereo and shading under general illumination. *CVPR 2011*.

[39] Z. Wu and L. Li. A line integration based method for depth recovery from surface normals. *ICPR 1988*.

[40] S. Xu, A. Georghiades, H. Rushmeier, J. Dorsey, and L. McMillan. Image guided geometry inference. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 310–317. IEEE, 2006.

[41] D. M. Young. *Iterative solution of large linear systems*. Elsevier, 2014.

[42] B. Zeisl, C. Zach, and M. Pollefeys. Stereo reconstruction of building interiors with a vertical structure prior. *3DIMPVT 2011*.

[43] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. *CVPR 2015*.