Towards Efficient Deflectometry in Motion

Alexey Pak

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
alexey.pak@ies.uni-karlsruhe.de

Technical Report IES-2012-01

Abstract:
Despite years of research, the reliable shape reconstruction of highly specular objects is still a largely unsolved problem, especially for complex objects or worse-than-ideal observation conditions. In this report, we elaborate on a novel multi-view specular reconstruction method based on the consistency of normal vector maps (NVMs). In particular, this algorithm is applicable to complex moving objects, where most "standard" techniques fail. We start by demonstrating how NVMs represent the specular reflection data, then re-formulate the reconstruction problem in terms of an energy functional to be optimized. Finally, we suggest an efficient solution of the problem as a modification of the probabilistic voxel carving approach.

1 Introduction

In the recent years the tools to reconstruct 3D textured surfaces from multiple views (or video streams) have become powerful enough to enable numerous applications in industry and research (see e.g. [Liu11] for a review of the current techniques and applications). Similar solutions for surfaces exhibiting strong specularity would also have found multiple immediate applications: for instance, car producers would welcome an objective computer vision-based method to inspect the finished car bodies as they move through a light tunnel – a task that is presently done by humans.

However, the approaches that appear in the literature require very demanding measurement conditions or make strong assumptions about the reconstructed surfaces. In particular, a deflectometric inspection requires that the object is fixed with respect to the camera and the calibrated pattern generator during multiple pattern

exposures. In addition, one also needs some “regularization”, i.e. external information about the location of the surface; the fusion of multiple measurements is also non-trivial [WMHB09].

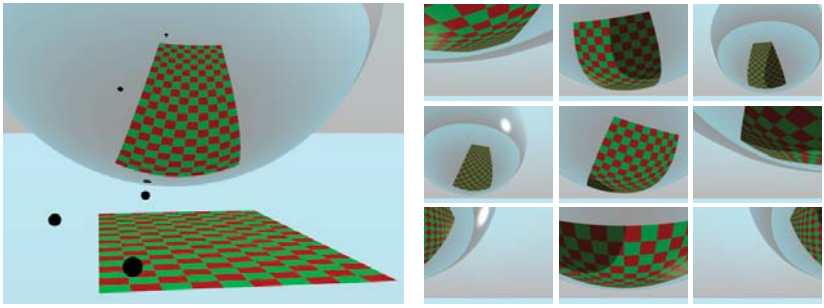
Nevertheless, the state-of-the-art deflectometric measurements now compete in accuracy with interferometry [FOKH12] due to extreme sensitivity of specular reflection to surface gradient changes. The ultimate solution would combine the flexibility of triangulation-based methods with the accuracy of deflectometry, possibly incorporating (but not crucially depending on) any additional information not contained in the camera images.

An alternative method of fringe reflectometry [HNA11] operates with single shots, and thus applies also to dynamic scenes. However, the reconstructed surfaces may only slightly deviate from a plane, and the scene geometry cannot be independently determined from the reconstruction itself. In addition, the method as described by Huang et al. is sensitive only to a narrow band of surface feature scales, which may potentially limit its general applicability.

In another recent work [WASS12], a moving surface is scanned by a laser ray constrained to a plane. The rays reflected from the surface draw a line on a diffusive screen, which is observed by a camera. The surface is reconstructed from the shape analysis of that line. While reportedly fast and accurate, this method also needs regularization, and utilizes in each camera snapshot only a fraction of the information potentially available in a series of deflectometric images.

Finally, the method of voxel carving based on normal vector consistency [BS03] enjoys potentially rather broad applicability. The volume containing the reconstructed object is divided into small regions (voxels), which can be occupied or empty. The camera images are processed to identify the (distorted) reflections of the unique point neighbourhoods of the calibrated pattern screen. The found correspondences then are used to reconstruct possible normal vectors of a surface under the assumption that it pass through a given voxel; finally, the voxels with incompatible reconstructed normals are labeled as empty.

While simple to implement, this method has several weak points. First, a sufficiently curved surface may distort the point neighborhoods so strongly that a reliable detection becomes impossible, and the resulting set of sparse constraints becomes too small. Second, as the authors of the above reference mention, the reconstruction accuracy depends sharply on the tolerance for normal vector deviations per voxel. This parameter is set globally and requires careful fine-tuning. Third, the naive voxel carving implemented in that paper gives no respect to voxel occlusions that are more than possible for any real-life objects.



(a) Simulated scene with the studied object (specular sphere) at the initial position, three cameras (marked with the black spheres), and the pattern screen.

(b) Images from the three cameras (one row per camera) corresponding to the three different object positions (one position per column).

Figure 2.1: Sample deflectometric setup with the object, pattern, and the cameras.

Our approach, first suggested in Ref. [Pak12] and further developed in this report, also employs voxel carving, but builds on a more general consistency condition. Instead of a single normal vector per voxel, we consider a set of all vectors consistent with multiple observations. The (non-unique) identification of color-coded pattern areas allows to build such sets without reliance on the fragile neighborhood analysis. Invalid voxels then receive inconsistent observations that lead to the empty set. The cones of candidate normal vector directions may be stored as e.g. two-dimensional maps on a unit sphere, and one effectively has to operate with two-dimensional binary images, as opposed to single vectors as in the method of Bonfort et al.

2 Normal vector maps

In order to discuss the construction of NVMs, we consider the synthetic scene in Fig. 2.1(a). We simulate several object positions, and capture (render) images from the three cameras (marked by black spheres) such as in Fig. 2.1(b). We assume that the position of the object's bounding box is available[1], and thus each observation also includes the camera projection parameters, and camera and pattern screen position and orientation with respect to the object.

[1] In a realistic measurement, this information can be obtained by e.g. tracking non-specular markers applied on or co-moving with the object; another possibility would be to treat these coordinates probabilistically and determine them by maximizing the likelihood. We postpone the detailed study of this question to further publications.

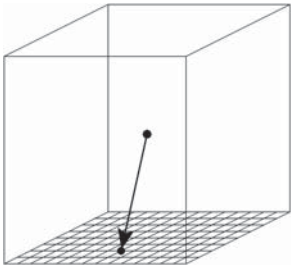


Figure 2.2: Rasterized cube surface as the parameterization of solid angles as viewed from the center of the cube.

Figure 2.3: Unfolded normal vector map. Black pixels mark the allowed, light gray – the excluded directions.

In order to perform voxel carving, one has to model the observations of small volumes of space. For simplicity, in all further examples we limit ourselves to considering the voxels being axis-aligned cuboids (more complex shapes, needed by complex voxel-carving algorithms, can be treated similarly).

Given a voxel, one first scans the camera pixels where it projects and identifies the observed pixel colors with those projected by the pattern generator. Discarding the underexposed (black) and the overexposed (white) pixels, the pattern colors can most easily be identified in the HSV-space (for hue-saturation-value) by comparing the hue component. Since the rendered images are anti-aliased, it is also important to properly attribute the transition pixels with intermediate color values.

The following step is to build the cones of the normal vectors compatible with the observation. Since the end result is an arbitrarily shaped "fan of rays", we parameterize it by rasterizing the complete 4π solid angle on a unit cube surface, as in Fig. 2.2. Each pixel on each face of the cube corresponds to a narrow cone originating from the center of the cube, and requires one bit of storage for labeling it either "allowed" or "excluded". Given sufficiently high resolution, the inhomogeneous density of the covered solid angle per pixel over the planar face is not important. In our implementation, each cube face contains 64x64 binary pixels (one bit per pixel), so that the complete net requires 3072 bytes.

In order to visualize the maps of the allowed directions, we unfold the cube surface into a net and interpret is as a planar image, such as in Fig. 2.3. The example in that figure displays a single cone of the allowed vectors that is directed primarily downwards (i.e. in the negative z direction), is slightly tilted forward (in the positive x direction), and does not intersect with the back and the top cube faces.

Let us consider the situation when the voxel color is identified as originating from some region of the pattern. Having chosen a point $\vec{V}$ inside the voxel, and a point

$\vec{P}$ on the corresponding area in the pattern, the reflection condition is

$$\vec{n} \sim \frac{\vec{C}-\vec{V}}{\left|\vec{C}-\vec{V}\right|} + \frac{\vec{P}-\vec{V}}{\left|\vec{P}-\vec{V}\right|},$$

where $\vec{C}$ is the camera location. We may now add the unit normal vector $\vec{n}$ to the map (i.e. set the corresponding pixel on one of the cube faces to true), and continue with some other combination of $\vec{V}$ and $\vec{P}$.

The above strategy is not extremely efficient and can be improved in multiple ways. We have implemented algorithms that work efficiently when the pattern contains a few uniformly filled polygons; depending on the pattern complexity, the final speedup compared to the above "naive" filling may be $\mathcal{O}(10^3 - 10^4)$, the details to be reported in a following publication.

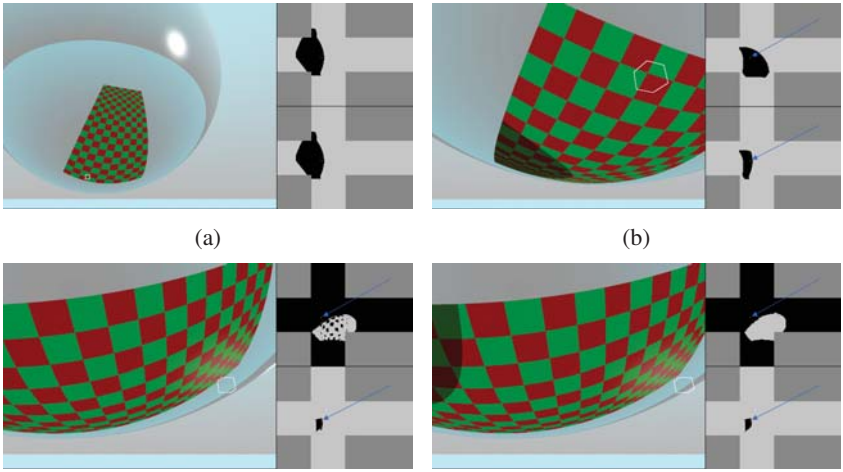
3 Single-voxel simulation

It is easy to see that the bitwise AND-fusion of NVMs for several views leaves only the directions that satisfy all conditions simultaneously. In Fig. 3.1 we present a few camera images and the corresponding NVMs, generated by the sequential AND-fusion for a single voxel. The frame in Fig. 3.1(a) is the first in the sequence, and thus the cumulative and the instantaneous NVMs are identical. In Fig. 3.1(a) and 3.1(b), the voxel image contains pixels of both pattern colors, which means that the reflected rays may have originated anywhere inside the entire pattern screen. In this case, we only include the normals compatible with the reflection towards the pattern.

In Fig. 3.1(c), the voxel contains one recognized and some unrecognized colors (but not the second pattern color); the corresponding pattern areas have to be excluded from the fully populated NVM. Finally, in Fig. 3.1(d) only unknown colors are observed. All these situations contribute to the cumulative NVMs, which in the end (i.e. after the fourth scene) matches quite closely the ground truth (the arrow endpoint).

A simulation of a single voxel tracked over 200 scenes, where each camera image has the dimensions of 512 x 384 pixels, takes a few seconds on a laptop PC. The memory requirements are relatively modest: each NVM occupies 3072 bytes, and one needs one current and one cumulative NVM per voxel.

The simplest reconstruction as in this example contains multiple opportunities for parallelization. Each voxel and each scene can be processed independently; color



(a) (b)

(c) (d)

Figure 3.1: Four subsequent (but not consecutive) steps from the reconstruction sequence for a single voxel that does enclose real surface. The voxel is denoted on the camera image with the white outline. In each frame, the upper NVM represents the instantaneous, the lower – the cumulative limitation of the recovered normal vectors for the voxel. The arrows indicate the locations of the actual normal (i.e. the ground truth). Frames (a) and (b) are taken with the first camera (the nearest to the observer in Fig. 2.1(a)), (c) and (d) – from the second camera (the leftmost camera in Fig. 2.1(a)).

identification can be done off-line and re-used for different voxels; polygon-filling and NVM fusion can be run on graphic cards etc.

4 Results of naive voxel carving

As discussed above, a sequence of camera observations in principle contains all the necessary information to recover the shape and the position of the object. However, the actual reconstruction results from a limited sequence depend strongly on the details: the voxel size, the chosen pattern, the camera resolution etc. In Fig. 4.1(right) we present the final NVMs corresponding to a single layer of voxels inside a small volume enclosing the boundary of a sphere (Fig. 4.1, left). The NVM fusion was performed "naively" as described above, and the volume and observations were chosen in order to avoid possible occlusions.

Approximately half of the voxels in the chosen layer lay inside the sphere or on the surface, the other half is outside. In the ideal situation, only a subset of NVMs in the upper-left corner of the resulting grid would remain non-empty. In reality, the remaining cones of normal vectors in the complementary region are narrow but non-zero, and the exclusion power of NVMs per se happens to be insufficient to reliably distinguish between the occupied and the empty voxels.

There exists several possible remedies to this situation. First, we may notice that the camera positions were too close to each other to reliably determine the distance to the object. A wider stereo base would solve this problem, but it would also give rise to occlusions. In a naive carving scheme, a voxel's NVM would then be affected by the occluded views, resulting in the wrong reconstruction.

Employing a priori information could also improve the result. For example, continuous objects tend to have similar occupancy values for the adjacent points. Similarly, smooth surfaces have strong correlations between the normal vectors at the close surface locations.

5 Probabilistic voxel carving

In order to merge the specular information with the a priori knowledge, one would ideally use some sort of a probabilistic framework. The approach that we chose as a model has been successfully employed in the state-of-the-art voxel carving tools, reconstructing diffuse objects [Liu11]. Below we re-formulate our problem following the notation of the above reference and suggest the possible solutions.

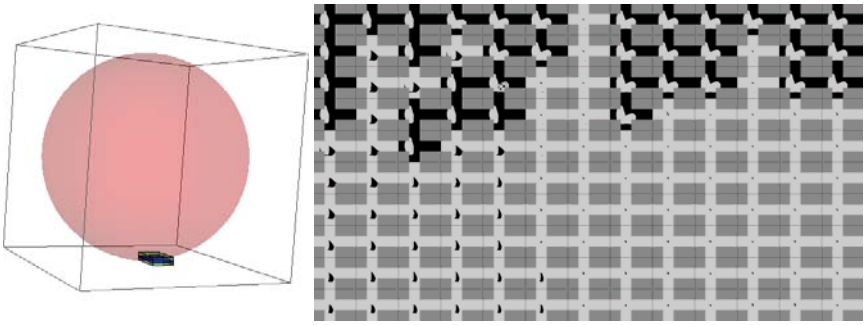


Figure 4.1: Right: location of voxelized volume and a "slice" of reconstructed voxels; left: resulting NVMs reconstructed from a sequence of 200 observations.

Let us consider a set $X = \{x_i\}$ of voxels, $i = 1, ..., N$. Each voxel is placed at position $\vec{r} = (x, y, z)$ and features the binary occupancy $o \in \{0, 1\}$ and the unit normal vector $\hat{n}$, i.e. $x_i = (\vec{r}_i, o_i, \hat{n}_i)$. The reconstruction then can be reformulated as an optimization problem:

$$X^* = \arg\min_X E(X), \tag{5.1}$$

where the energy functional $E(X)$ is

$$E(X) = \sum_i E_u(x_i) + \sum_{i,j} E_p(x_i, x_j) + \sum_{r \in R} E_R(\{x\}_r). \tag{5.2}$$

Here the second sum is performed over the pairs of the adjacent voxels, and $\{x\}_r$ in the last sum denotes the ordered set of voxels traversed by the ray r taken from the set of all rays R.

The unit energy E_u describes our a priori preference for a specific occupancy/normal vector combination in each point in space. In the simplest form, it may simply encode the desired fraction of the volume occupied by the object:

$$E_u^{\text{simple}}(x_i) = w_u(1 - o_i).$$

A more sophisticated function can encode a detailed information such as a CAD-model or a suitable simple basic shape.

The pairwise energy E_p describes the correlation between the two voxels. If the normal vector is not taken into consideration, only the correlation between the two occupancies remains:

$$E_p^{\text{simple}}(x_i, x_j) = w_p(o_i - o_j)^2 \|\vec{r}_i - \vec{r}_j\|^{-1}.$$

Likewise, a more complicated function would also involve normal-occupancy and normal-normal interactions.

The a priori terms give the optimization some "guidelines", and one may discuss the optimal functional form or the strength coefficients. However, the most interesting term in Eq. (5.2) is the ray energy E_R, which we suggest to take as follows:

$$E_R(\{x\}_r) = \min_{\hat{n} \in \text{NVM}(r, \vec{r}_i)} \|\hat{n}_{i^*} - \hat{n}\|^2. \tag{5.3}$$

The index i^* here denotes the first occupied voxel on the ray. For example, given the voxels on the ray $\{x\}_r = (x_{i_1}, ..., x_{i_K})$, i^* is defined as

$$i^* = \begin{cases} i_1, \ o_{i_1} = 1 \\ i_2, \ o_{i_1} = 0, o_{i_2} = 1 \\ \dots \ \dots \end{cases}.$$

The minimum in Eq. (5.3) is taken over all normals consistent with the color of the ray r at the position of the observed voxel $\vec{r}_{i^*}$, or over the members of the set of directions allowed by the NVM, denoted as $\mathrm{NVM}(r, \vec{r}_{i^*})$. Finding this minimum itself is an optimization problem, with the minimum energy of zero reached when the current normal vector is inside the allowed set, and growing quadratically with the distance to this set otherwise.

The optimization of the complete model in Eq. (5.1) (equivalent to maximum a posteriori probability inference in a higher-order Markov random field) is a formidable task, since the search space includes N discrete and $2N$ continuous parameters. However, the method of "deep belief propagation" of Ref. [Liu11] gives a recipe to efficiently compute the messages from the ray factors to variables, which allows it to successfully find solutions with tens of millions of voxels (for photo-consistency reconstruction). Our model has a very similar structure and we expect similar performance benefits also in the (explicitly non-linear) formulation of Eq. (5.2). As of writing this report, the development of the reconstruction program is still underway.

6 Conclusion

This report summarizes the current status of the suggested method of shape reconstruction from multiple views with the help of NVMs. We outline the basic ideas behind the algorithm, identify the weaknesses of a naive implementation, and suggest the mathematical grounds for the general probabilistic framework. As a solution method, we suggest to use the "deep belief propagation" algorithm that has been shown to successfully perform in similar problems.

Bibliography

[BS03] Thomas Bonfort and Peter Sturm. Voxel carving for specular surfaces. Proc. 9th IEEE International Conference on Computer Vision (ICCV '03), pages 691–696, 2003.

[FOKH12] Christian Faber, Evelyn Olesch, Roman Krobot, and Gerd Häusler. Deflectometry challenges interferometry: the competition gets tougher! Proc. SPIE 8493, Interferometry XVI: Techniques and Analysis, pages 84930R–84930R–15, 2012.

[HNA11] L. Huang, C. S. Ng, and A. K. Asundi. Dynamic three-dimensional sensing for specular surface with monoscopic fringe reflectometry. *Optics Express*, 19:12809–12814, 2011.

[Liu11] Shubao Liu. *Statistical Inverse Ray Tracing for Image-Based 3-d Modeling*. PhD thesis, Brown University, 2011.

[Pak12] Alexey Pak. Recovering shapes of specular objects in motion via normal vector map consistency. Proc. SPIE 8493, Interferometry XVI: Techniques and Analysis, pages 84930T–84930T–8, 2012.

[WASS12] R. D. Wedowski, G. A. Atkinson, M. L. Smith, and L. N. Smith. A system for the dynamic industrial inspection of specular freeform surfaces. *Optics and Lasers in Engineering*, 50:632–644, 2012.

[WMHB09] S. Werling, M. Mai, M. Heizmann, and J. Beyerer. Inspection of specular and partially specular surfaces. *Metrology and Measurement Systems*, 16, 2009.