Real-World Normal Map Capture for Nearly Flat Reflective Surfaces

Bastien Jacquet[1], Christian Häne[1], Kevin Köser[12*], Marc Pollefeys[1]

ETH Zürich[1]
Zürich, Switzerland

GEOMAR Helmholtz Centre for Ocean Research[2]
Kiel, Germany

Abstract

Although specular objects have gained interest in recent years, virtually no approaches exist for markerless reconstruction of reflective scenes in the wild. In this work, we present a practical approach to capturing normal maps in real-world scenes using video only. We focus on nearly planar surfaces such as windows, facades from glass or metal, or frames, screens and other indoor objects and show how normal maps of these can be obtained without the use of an artificial calibration object. Rather, we track the reflections of real-world straight lines, while moving with a hand-held or vehicle-mounted camera in front of the object. In contrast to error-prone local edge tracking, we obtain the reflections by a robust, global segmentation technique of an ortho-rectified 3D video cube that also naturally allows efficient user interaction. Then, at each point of the reflective surface, the resulting 2D-curve to 3D-line correspondence provides a novel quadratic constraint on the local surface normal. This allows to globally solve for the shape by integrability and smoothness constraints and easily supports the usage of multiple lines. We demonstrate the technique on several objects and facades.

1. Introduction

Reflective surfaces are a key design element in modern architecture and facades of skyscrapers and office towers are often dominated by glass and metal. Also many shops and even some particular private houses show large reflective surfaces, sometimes designed specifically to generate a certain impression.

The appearance of such buildings depends largely on what is reflected and in which way the surrounding scenery is distorted by the reflection. Many of the reflective surfaces are flat or nearly flat, but small deviations of the surface normals can nevertheless generate very strong distortions in the reflections. The reason for the small variations of the normals can be imperfections of the glass manufacturing, mechanical fixation, or can be part of the intended design. This results in many reflecting surfaces having their own specific distortion patterns, *e.g.* Fig. 1 shows a characteristic example. It has been shown very recently that considering reflections can significantly improve the realism of

Figure 1. Real-world glass reflection. Notice that reflection in different windows on the same facade can appear very different due to minor deformations and normal variations. Our goal is to capture normal maps of real windows to faithfully reproduce this effect.

image-based rendering techniques [22], however this was restricted to objects that behave like perfectly planar mirrors. One reason for this restriction might be that no practical way is known to capture the surface normals of big real world objects like shop windows or entire glass facades. Consequently, when digitally modeling distorted windows as perfectly flat this will dramatically hurt realism of the virtual 3D building or virtual city model. It is well known that realism can be added to the glass reflections by using a normal map [5] and the current state of the art practice for modeling windows in such models consists of using a plausible, generic bumpmap template (*e.g.* bulge), potentially perturbed with some low-frequency random noise pattern. Such a 'randomized surface' approach avoids the immediate synthetic impression created by perfect mirrors. However, it cannot easily be used to mimic the specific visual reflection patterns characteristic for a particular building or window. In contrast, our goal with this paper is to propose a practical approach to capture these small, but visually significant normal variations of real-world reflective surfaces outside the lab. As we will discuss in the related work section, to the best of our knowledge, this is the first approach demonstrated to be able to capture normal maps of outdoor windows, in place.

Our work should be seen as fitting in a major trend in computer vision and graphics that consists of developing techniques for capturing 3D models [19, 17] and measuring surface properties [6, 4] of surfaces to generate faithful copies of the real-world. More specifically, there has recently been a lot of interest in capturing visual models of cities [9, 18, 11]. It should be noted that existing techniques for capturing buildings struggle with windows as these do not reflect sufficient laser-light back to the range sensor or

*This work was done while this author was employed by ETH Zürich.

due to their reflective nature violating the Lambertian assumption underlying most image-based techniques.

The key idea we follow is to track the (curved) reflections of real world straight lines (*e.g*. typically a line observed on the opposing facade) when moving a video camera (handheld, or mounted on a capture vehicle) in front of the window of interest. Under the assumption that the window is a plane at a certain distance, we attribute the deformation of the straight line to variation of the local normal. Essentially, first we estimate the pose of the camera and the position of the 3D line before we track the curved reflection in the video. As we have to estimate the two degrees of freedom of the normal and each observation provides a single constraint, two passes of different lines (*e.g*. a horizontal and a vertical one) would be required and sufficient to determine each normal locally. However, similarly as in shape-from-shading, when we consider only physically plausible, global solutions for the set of normals forming a smooth surface, a single pass combined with integrability and smoothness assumptions are sufficient to recover the whole normal map for the window. This normal map can be used *e.g*. in ray-tracers to realistically render the reflections of the window that was captured. Before describing our approach more in detail, we discuss previous work with respect to normal and shape estimation of reflective surfaces.

2. Previous Work and Contribution

Specular surfaces, although present everywhere in man-made environments, still pose challenges for camera tracking, 3D reconstruction (cf. to Ihrke *et al*. [12] for a recent survey) or view interpolation. Towards this end, recently Sinha *et al*. [22] have presented an image-based rendering approach that addresses the problem of realistic mixing of reflected light and transmitted light using proxy geometries. In terms of 3D reconstruction of the specular surface itself, *e.g*. Zisserman *et al*. [29] infer surface properties from a specular reflection of a single light source. Later, Savarese *et al*. [21] use a calibrated camera observing a planar pattern and analyze which surface properties could be inferred from local measurements of a point plus a variable number of local directions (*e.g*. a point-to-point correspondence and two local directions can be used at each corner position when clicking correspondences on a chessboard pattern). Roth and Black [20] formalize the concept of specular flow and show that for a distant reflected scene it can in principle be used to recover shape. The theoretical properties of the specular flow are further studied by Vasilyev *et al*. [25] who also propose reconstruction algorithms under the assumption that the specular flow is given. In fact, this is a major stumbling block as the specular flow is much more complicated to estimate than the standard (diffuse) optical flow and no algorithms exist for doing this from real-world imagery. This is also the reason why we have resorted to track the reflection of lines instead of points as this is more feasible. However, in this case we only obtain a single constraint per point traversed by the reflection of the line. Notice also that the reflection of a line can consist of multiple curves as the topology of the reflection can change (as we will see later), which makes optical flow virtually impossible and even the tracking of lines very challenging. The shape from specular reflection problem, as we pose it, creates a constraint on the surface normal at each position that is quadratic, but, apart from this last fact, it is closely related to the shape from shading problem [28] as only a single constraint on the two degrees of freedom of the normal is obtained directly from the measurements.

The novelty of our work lies in the fact that we do not require a specifically designed calibration target like a checkerboard [7] or a pattern on a calibrated monitor [8, 24, 14, 2, 15]. To our knowledge, our paper is the first to present practical results on outdoor reconstruction of specular surfaces. As we focus in particular on near-flat surfaces like windows, we discuss related work addressing this case more in detail. Here, Ding and Yu [7] also reconstructed properties of almost flat surfaces. However, they require a calibration object to be held in front of the surface. The object they use is a color-coded checkerboard pattern that allows to establish point correspondences between the reflection and known 3D coordinates on the checker board. In a later work Ding *et al*. [8] also use reflections of lines, however, they require many known 3D lines that intersect the specular surface, which in practice makes the approach impossible to use in uncontrolled environments. Recently Liu *et al*. [15] uses dense (or sparse with B-spline parametrized smooth surface) correspondences between a known calibration pattern and a single image to reconstruct the surface. The need to know exactly the calibration pattern to compute correspondences makes also this method impracticable in uncontrolled environments. Consequently, we consider the main contributions of our work :

- the exploitation of existing scene structures only (in practice big windows would require unrealistically big calibration targets)
- a novel constraint on the surface normal derived from point-to-line correspondences rather than point-to-point. The line reflection passing over the surface allows to scan very densely instead on a few sparse feature positions
- camera tracking in reflective scenes where patch-feature-based approaches fail, allowing to rectify video frames with respect to the window frame
- robust, efficient line reflection tracking by 3d ortho-video cube segmentation that supports line topology changes
- both direct local estimation of normals (by using more than one 3D line) or global optimization by incorporating integrability constraints and shape priors

This altogether results in - to the best of our knowledge - the first system that is able to capture normal maps in real-world conditions, at physically inaccessible facades, on top of being very practical as all it needs is a video camera.

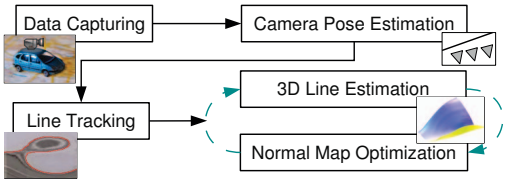


Figure 2. Overview of our normal map capturing pipeline.

3. Normal Map Capture

In this section we will present the general system (see also Fig. 2). After having moved a camera such that a 3D line's reflection passes over a window, we compute the camera path. We then rectify the video and track the reflection of the line in order to first approximate the 3D line's parameters (using reflection on the boundaries). Finally, we estimate a normal map for the window that in turn can be used to improve the 3D line's estimate and vice versa.

3.1. Tracking the Camera

When using a capturing vehicle for acquiring panoramic street-level data, the trajectory is often already available or could be computed reliably from omni-directional images and additional sensors eg. GPS/INS or wheel odometry. Using images only, for big reflective facades often everything is reflective but a grid of lines. This fools state of the art Structure-from-Motion algorithms like [26]. Consequently, in the following we briefly outline a technique based on vanishing points and chamfer matching to track the camera motion in planar, highly reflective scenes like the one shown in Fig. 1.

For each frame of the video that scans *e.g.* a window on a facade we estimate vanishing points using the software published[1] by Baatz *et al.* [1]. Although orthogonality of vanishing points can also reveal the focal length, we assume a camera with known intrinsics. After rectifying with respect to vanishing points we obtain a video with constant camera orientation (aligned with facade directions). We set our world coordinate system origin into the plane such that the first image's camera center has coordinates (0,0,-1).

In order to obtain the relative camera motion, we track the edges that support the vanishing points (i.e. we remove curved reflections). Rather than tracking all edges at once, we exploit the fact that we have stable orientation and subdivide the remaining (horizontal and vertical) edges into the four directions according to their gradient (north, south, east, west). For each of the four sets of edges, Chamfer matching[3] between subsequent frames is performed, however with a joint set of parameters for x,y and z translation (z-translation corresponding to scaling). To compensate for potential drift and small jitter from vanishing point estimation, we finally register all images to the first one using a full 6 degree of freedom pose parameterization also using Chamfer matching, where we can also optionally enforce straight motion, *e.g.* for sequences taken from a vehicle.

[1] available at http://people.inf.ethz.ch/gbaatz/software/

Figure 3. Ortho-rectified input image (left) and oriented edge image (right): Vertical edges are in green and blue (positive and negative gradient), horizontal ones are in red and magenta.

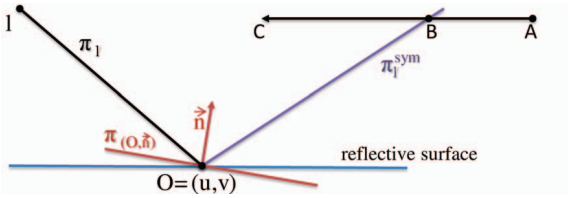


Figure 4. Monotonicity of traversal. For a point O with normal n, the reflection of the line l would intersect a straight path from A to C in a unique point B. Therefore l is only seen at O from B.

3.2. Tracking the reflected lines

As a first step the facades and windows need to be detected in the input video. While recent advances on facade parsing (*e.g.* Martinović *et al.* [16]) have shown the possibility to detect facades and find facade elements such as windows fully automatically, in the present work windows were hand-selected in one reference frame.

In order to estimate the normal map of a detected window, the position of the 3D line's reflection inside that window has to be known. A difficulty of this task is that, due to the distortions, this reflection is no longer a line in the image, it is not even guaranteed to be one connected curve as depicted in Fig. 5. Therefore standard line tracking methods cannot be applied.

We will use the fact that the camera moves approximately along a straight line to reformulate the problem of tracking the line as a volumetric segmentation problem.

Observation 1. *Having a straight camera path, the reflection of a static 3D line will be observed at a certain position $O = (u, v)$ of a static window only in one frame t (or in no/all frames).*

Proof. Let $\pi_{(O,\vec{n})}$ be the plane going through O with normal $\vec{n}$, π_l be the plane going through O containing l, and π_l^{sym} the reflection of π_l with respect to $\pi_{(O,\vec{n})}$. π_l^{sym} is the only location where l can be seen at O. The straight camera path will hence intersect π_l^{sym} in one single frame (or in all/no frames). See Fig. 4 for a 2D illustration. □

Obs. 1 means that once the reflection has passed over a particular window position (u, v), it cannot be seen there again from a later camera position and this allows to formulate the reflection tracking as a binary segmentation problem: before and after the reflection.

The input video is first rectified to a u-v-t video volume per window. Given that the camera positions and the facade plane are known, we can warp the images to fronto-parallel views (see Fig. 3). Stacking the fronto-parallel views of the extracted windows along the t direction leads to a video

volume per window. (u, v) are the pixel coordinates in the local window coordinate frame. According to Obs. 1 at each point on the window the line can only pass once and therefore the surface that separates "before" and "after" the line forms a height field (in temporal direction). The height for a particular (u, v) position tells when the line has passed this position.

As the segmentation boundary should be at the position of the reflected line we expect a high gradient in the input video volume at this position. Furthermore the normal of the segmentation boundary should align with the gradient inside the volume.

There are many volumetric segmentation methods published that can handle this situation. We chose to use the method described in Zach *et al*. [27]. The segmentation is described as a convex optimization problem in the continuum. It is particularly applicable for this problem as it allows to directly align the normal of the segmentation boundary with the gradient orientation inside the volume. This is in contrast to graph based methods where only the weights of the graph's edges can be adapted. Additionally, the height map constraint can be directly enforced.

To bootstrap the optimization we learn the appearance of the reflected line as follows. Even though the distortions can be quite strong and the reflected line might even disconnect, there are in general always a few frames where the line is strong enough to be segmented. Therefore we treat each frame individually in a first step. A cut through the window along pixels with strong gradient is optimized, which gives us candidate line reflections. For each candidate two boosted decision tree classifiers [10] are trained. The first one outputs the likelihoods that a patch is centered on the line and the second one outputs whether a patch is likely to lie right before or right after the reflected line (*c.f*. Fig. 12). The classification is done pixel-wise. For each pixel $\mathbf{q}$ we use as feature vector the color values of all the pixels in a window centered around $\mathbf{q}$. Finally we apply [27] by weighting the gradient lengths in spatial (respectively temporal) direction by line (respectively before/after) classifier based values.

At this point we want to make a few remarks:

- For difficult cases hard constraints can be added by annotating some before and after areas in a few of the input frames.
- Using a vehicle-mounted omni-directional camera it would also be possible to train the appearance of the reflection from a direct image of the 3D line.
- Filtering out candidates resulting in impossible segmentations or normal maps and keeping only the consensus validated normal map leads to a fully automatic system to track the reflected lines.
- Overall, using volumetric segmentation instead of local line tracking, we can naturally handle complex topology changes of the reflected line as depicted in Fig. 5.

3.3. Estimating the 3D line position in the world

For the subsequent normal estimation steps we need to know the position of the 3D line in the world. The parameters for this line are obtained in a two step process, first by an approximate solution improved afterwards using nonlinear refinement.

Initial 3D Line estimate In practice, it is often quite easy to obtain a sufficiently good initial guess of the 3D line parameters *e.g*. for a vertical line at the opposite side of the street, just by comparing the camera distance to the facade and the 3D line distance to the facade. However, the initial 3D line parameters can also be obtained by an image-based method: When having a closer look at Fig. 1 it can be seen that the reflection at the boundaries between adjacent windows behaves consistently. This is a general observation that is also true for the overwhelming majority of facades that we saw:

Observation 2. *All curved line intersections with their frame boundaries lie on a straight line and this is approximately the perfect mirror reflection of the 3D line on the facade.*

An explanation to this observation would be that the windows have the canonic normal close to the boundary and we will use this for the initial estimate of the 3D line that is now derived:

First, we observe that given the camera pose $\boldsymbol{P}$ and the normal $\boldsymbol{n}$, a particular point $\boldsymbol{X}_l$ on the 3D line in space is projected to image position $\boldsymbol{c} \simeq \boldsymbol{P}\boldsymbol{M}_{\boldsymbol{n}}\boldsymbol{X}_l$. Herein, $\boldsymbol{M}_{\boldsymbol{n}}$ is the 4×4 matrix that encodes the reflection on the plane with normal $\boldsymbol{n}$. Now given two endpoints of the curve (*e.g*. top and bottom for a vertical line) in the image and given the expected canonic normal $\tilde{\boldsymbol{n}}$ close to the boundary ($\tilde{\boldsymbol{n}} = (0, 0, 1)^\top$, due to observation 2) we can construct a 2D line $\boldsymbol{l}$ through these endpoints and backproject the line into 3D space, resulting in a plane π_i for the ith frame of the video:

$$\boldsymbol{\pi}_i = \boldsymbol{M}_{\tilde{\boldsymbol{n}}}^\top \boldsymbol{P}_i^\top \boldsymbol{l}_i \tag{1}$$

We bring all planes to normal form and stack the m 4-vectors to construct a matrix $\boldsymbol{\Pi}$. All points $\boldsymbol{X}_l$ on the 3D line must be on all these planes:

$$\boldsymbol{\Pi}\,\boldsymbol{X}_l = 0 \quad \text{with} \quad \boldsymbol{\Pi} = (\boldsymbol{\pi}_1\ \boldsymbol{\pi}_2\ \ldots\ \boldsymbol{\pi}_m)^\top \tag{2}$$

The 3D line is then determined as the 2D null space of the above matrix Π (through singular value decomposition).

3D Line Optimization Once an initial 3D line estimate is available and given the camera poses and the canonic normals at the boundaries, we can project the initial 3D line into image space and minimize the difference between projected and measured curve for each frame using the Levenberg-Marquardt method. This allows also to constrain the line to being vertical or horizontal. Later, once the whole normal field is recovered, one can go back to this stage and even reoptimize the line using all normals and all observations.

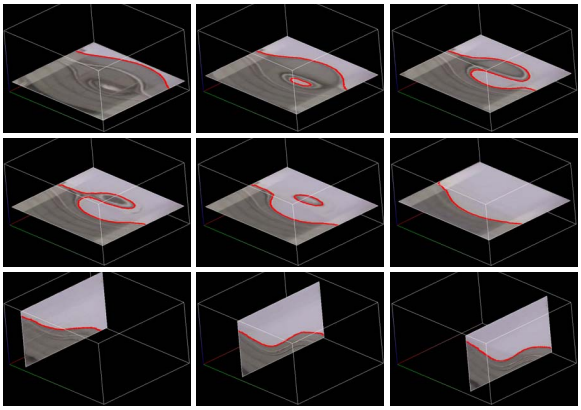


Figure 5. Cross sections of the video cube with overlayed tracked line in red. The first two rows show the tracked line on different frames of the input video. The occuring topology change of the line is handled properly. In the last row the video cube is cut along another direction, along which the tracked line forms a height map.

3.4. Estimating the Normal Field

At this point we know the camera poses for each image, the 3D line position and the reflection of the 3D line in each of the images. To start reconstructing the normals, we assume that the window is geometrically planar (i.e. all window points are in the $z = 0$ plane) but that there are small variations which we capture in the local normals.

We will now first look locally at a single point in the image, where we see some points of the 3D line being reflected. Tracing back the ray from the camera onto the reflective surface, it must be reflected in a way such that it meets the 3D line in 3D space. However, we do not know which point on the 3D line we are observing. Each of these points on the 3D line would give rise to a different local normal, since the normal is the bisection of the incoming and the outgoing ray of the reflective surface. We will first derive the set of normals compatible with a local observation and show that we obtain a quadratic constraint on the normal. Given this local constraint, we will discuss how to disambiguate this by exploiting the fact that the normal components are actually the slopes of the surface's height field and thus cannot vary freely.

Local quadratic constraint on normal The two degrees of freedom of the local normal are not fully determined by observing the line's reflection at some local position. Rather, depending on which point of the 3D line we are seeing, this leads to a different geometric configuration.

Let us take the position of the reflective surface point $\boldsymbol{O}$ in Fig. 6 (left). We wish to find the normal $\boldsymbol{n}$ and for this consideration we assume that $\boldsymbol{O}$ is the origin of the coordinate system. We also assume we observe this reflective surface in a camera with center $\boldsymbol{S}$, and the (normalized) direction of the ray from $\boldsymbol{O}$ to $\boldsymbol{S}$ is $\hat{\boldsymbol{S}}$. The light from a point $\boldsymbol{X}$ on the line $\boldsymbol{l}$ comes in from a (normalized) direction $\hat{\boldsymbol{X}}$. It is easy to verify that the normal $\boldsymbol{n}$ is aligned with the sum of the two unit vectors $\hat{\boldsymbol{X}} + \hat{\boldsymbol{S}}$ (as it is the bisector of the angle $\angle(\hat{\boldsymbol{X}}, \hat{\boldsymbol{S}})$). When $\boldsymbol{X}$ takes different positions on the line, $\hat{\boldsymbol{X}}$ describes the half-circle $\boldsymbol{\mathcal{C}_0}$ because of its unit

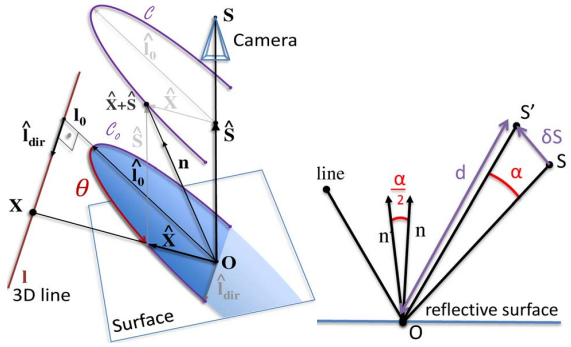


Figure 6. **Left :** Illustration of the cone constraint on the normal $\boldsymbol{n}$ when a point $\boldsymbol{X}$ on the 3D line $\boldsymbol{l}$ is reflected at the origin $\boldsymbol{O}$ and observed in the camera with center $\boldsymbol{S}$. $\boldsymbol{n}$ must be the angular bisector between normalized rays $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{S}}$, forcing it somewhere on a cone $\boldsymbol{\mathcal{C}}$. The true position on the cone depends on the actually observed point along the line, an information that is not available. **Right :** Normal change implied by a missestimation of the camera center $\boldsymbol{S}$: If $\boldsymbol{S}$ is missestimated by some $\boldsymbol{\delta S}$, this causes some angular error $\alpha/2$ that is approximately the ratio of the error $\boldsymbol{\delta S}$ and the (usually very large) distance d to the surface point.

length and $\hat{\boldsymbol{S}}$ stays fixed, therefore $\hat{\boldsymbol{X}} + \hat{\boldsymbol{S}}$ is also bound to be on the half-circle $\boldsymbol{\mathcal{C}}$ ($\boldsymbol{\mathcal{C}_0}$ lifted by $\hat{\boldsymbol{S}}$). Finally, $\boldsymbol{n}$ is bound to be on the half-cone with vertex $\boldsymbol{O}$ that contains $\boldsymbol{\mathcal{C}}$. Algebraically, we start from ($\simeq$ meaning proportional to) :

$$\boldsymbol{n} \simeq \hat{\boldsymbol{X}} + \hat{\boldsymbol{S}} \tag{3}$$

Let us name $\boldsymbol{l}_0$ the closest point to $\boldsymbol{O}$ on the line $\boldsymbol{l}$, $\hat{\boldsymbol{l}_0} = \boldsymbol{l}_0/||\boldsymbol{l}_0||$, $\hat{\boldsymbol{l}}_{\text{dir}}$ the unit-length direction of the line $\boldsymbol{l}$, and $\theta = \angle(\hat{\boldsymbol{l}_0}, \hat{\boldsymbol{l}}_{\text{dir}})$ (notice that $\hat{\boldsymbol{l}_0} \perp \hat{\boldsymbol{l}}_{\text{dir}}$). It derives :

$$\hat{\boldsymbol{X}} + \hat{\boldsymbol{S}} = \quad \cos\theta\, \hat{\boldsymbol{l}_0} + \sin\theta\, \hat{\boldsymbol{l}}_{\text{dir}} + \hat{\boldsymbol{S}} \tag{4}$$

Then, by using the half-angle substitution $t = \tan\frac{\theta}{2}$ (*i.e.* $\cos\theta = \frac{1-t^2}{1+t^2}$ and $\sin\theta = \frac{2t}{1+t^2}$), we obtain :

$$\hat{\boldsymbol{X}} + \hat{\boldsymbol{S}} = \quad \frac{1}{1+t^2}\left(\hat{\boldsymbol{S}} + \hat{\boldsymbol{l}}_0 + 2t\,\hat{\boldsymbol{l}}_{\text{dir}} + t^2\hat{\boldsymbol{S}} - t^2\hat{\boldsymbol{l}}_0\right) \tag{5}$$

$$\boldsymbol{n} \simeq \underbrace{\begin{pmatrix}\hat{\boldsymbol{S}} + \hat{\boldsymbol{l}_0} & 2\hat{\boldsymbol{l}}_{\text{dir}} & \hat{\boldsymbol{S}} - \hat{\boldsymbol{l}}_0\end{pmatrix}}_{\boldsymbol{D}} \begin{pmatrix}1 & t & t^2\end{pmatrix}^\top \tag{6}$$

Here, the matrix $\boldsymbol{D}$ constructed from the position of the camera center and the line relative to the surface point fully encodes the geometric setting. Since the explicitly parametrized conic $\begin{pmatrix}1 & t & t^2\end{pmatrix}^\top$ has the implicit equation :

$$\begin{pmatrix}1 & t & t^2\end{pmatrix}\begin{pmatrix}\;\;1\\ -2\\ \;\;1\end{pmatrix}\begin{pmatrix}1\\ t\\ t^2\end{pmatrix} = 0, \tag{7}$$

we obtain :

$$\boldsymbol{n}^\top \underbrace{\boldsymbol{D}^{-\top}\begin{pmatrix}\;\;1\\ -2\\ \;\;1\end{pmatrix}\boldsymbol{D}^{-1}}_{\boldsymbol{C}}\boldsymbol{n} = 0 \tag{8}$$

which describes an (oblique) cone of normals (of which half is feasible due to the observation that $|t| < 1$).

Degenerate Cases The degenerate cases arise when $\boldsymbol{D}$ does not have full rank. This happens when the point of reflection (the origin in this derivation), the line and the camera center are all in one plane (*e.g.* frontal to a facade and a vertical line right behind the camera).

We therefore removed the need of 3D to 2D correspondences of earlier methods by implicitly representing the set of 3D points possibly reflected at this pixel. The price to pay is to not directly have the normals but only a cone of possible normals. The ambiguity could be resolved by observing another line and intersecting the two (half-)cones to obtain the normal. Notice that this is different from using the 3D intersection of 2 lines (tracking a 3D point on a chessboard), since our 3D lines need not intersect each other (*e.g.* lamp post and facade line), and we also allow the other lines to be acquired independently (*e.g.* another day). This would allow for pointwise, independent normal estimates for the whole window, however due to the local nature of the measurements the estimates might be noisy. Instead, we propose to exploit constraints related to the physical properties of the material, i.e. that the normal field has to be smooth and integrable, therefore resolving the local ambiguity in a more global manner. As an additional benefit, this allows recovering normal maps with one single pass, *e.g.* by a vehicle driving down the road.

Global formulation In order to solve for the normal field, we design an energy using all the constraints we have on the normal map, and then minimize it with an iterative method.

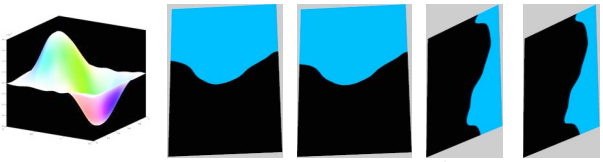
$$E(\overrightarrow{\boldsymbol{n}}) = \iint_{p\in\mathcal{W}} \left[\mathbb{1}_{p\in\mathcal{D}} \left({\boldsymbol{n}_p}^\top \boldsymbol{C}_p \boldsymbol{n}_p \right)^2 + \mu \left\| \frac{\partial n_p^x}{\partial y} - \frac{\partial n_p^y}{\partial x} \right\|^2 + \gamma \left(\frac{\partial {n_p^x}^2}{\partial x} + \frac{\partial {n_p^x}^2}{\partial y} + \frac{\partial {n_p^y}^2}{\partial x} + \frac{\partial {n_p^y}^2}{\partial y} \right) \right] \mathrm{d}p \quad (9)$$

Here, $\overrightarrow{\boldsymbol{n}}$ stands for the whole 2D vector field over the window, $\mathcal{W} \subset \mathbb{R}^2$ is the set of points of the window, $\mathcal{D} \subset \mathcal{W}$ is the set of points where we observed a reflected line, $\mathbb{1}_{p\in\mathcal{D}} = 1$ if $p \in \mathcal{D}$ else 0, $\boldsymbol{n}_p = [{n^x}_p, {n^y}_p, 1]^\top$ is the normal at point p (assuming a near-flat window), $\boldsymbol{C}_p$ is the quadratic constraint of Eq. (8) at point p.

This energy is made of 3 parts, in order of appearance :

1. the data term, which is the square of the signed algebraic distance of each normal to its corresponding allowed cone (note that the square of algebraic distance to a conic is 4th order and therefore in general not convex).
2. the integrability constraint, we use a soft enforcement as experiments with hard constraint were more prone to fall into a local minimum.
3. the smoothness constraint, enforcing neighbor normals to be close to each other or in other words the curvature of the height field to be low. This energy is also known as thin plate or bending energy of elastic materials in physical models. If h is the height map, it can be rewritten as $\frac{\partial^2 h}{\partial x^2}^2 + 2\frac{\partial^2 h}{\partial x \partial y}^2 + \frac{\partial^2 h}{\partial y^2}^2 = \operatorname{div}\operatorname{grad} h$.

Since the window is not infinitely large, we use Neumann boundary conditions, i.e. for the vertical boundaries we assume that the horizontal normal component stays the same and vice versa. In terms of window height field assumption that corresponds to linear extrapolation of height



Ground truth surface, 200× height-scaled | true horiz. line pass | rend. horiz. line pass | true vert. line pass | rendered (from horiz.)

Figure 7. Comparison of reflection on ground truth surface and estimated surface. The surface is estimated from a single horizontal line pass and then this pass is rerendered using the estimated data. Additionally, reflection of a vertical line on another camera path is synthesized using the estimated data demonstrating the faithful reconstruction. Normal error mean and median are 0.1°.

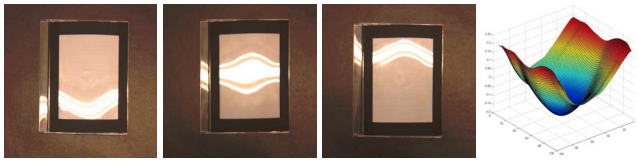


Figure 8. Three frames of the CD Case sequence, and estimated height field with z x1000

orthogonal to the boundaries. In our work, we used $\mu = 100$ and $\gamma = 5$.

The starting point of the optimization is chosen by assuming that each normal lies on the plane defined by the ray from its point to the camera center and the z direction (this is a naive per window point constraint, without any neighborhood or smoothness constraint). The advantages of the given formulation is its ability to handle missing data ($\mathbb{1}_{p\notin\mathcal{D}} = 0$) as well as multiple reflected 3D lines (it just adds more constraints), while at the same time limiting the sensitivity to noise due to global optimization. Once we have this first estimate for the normal map, it is possible to refine the position of the 3D line(s) by back-reflecting each reflected line using the newly computed normal map and vice versa. However, for the demonstrations of this paper we did not make use of this possibility and present results obtained from the first convergence of the normal map.

4. Evaluation

In this section we first evaluate the sensitivity of the approach with respect to noisy poses. Afterwards we compute the accuracy using a known ground truth surface (rendered) and prove that a single pass is sufficient to capture both degrees of freedom of the normal. Finally we show qualitative real-world results.

Expected accuracy on normal map The pose has to be estimated with respect to the facade and in Fig. 6, we analyze the sensitivity with respect to noise in the pose. In particular in planar pose estimation for cameras with limited field of view there is an almost-ambiguity between rotation (*e.g.* panning right) and translation (moving right). Fig. 6 (right) shows that if the error in the camera rotation angle is α then the normals are shifted by $\frac{\alpha}{2}$, which means that the reachable angular accuracy is in the order of magnitude

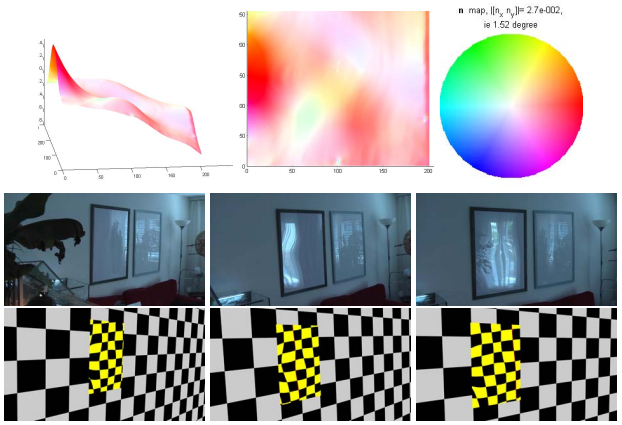


Figure 9. Estimated height field and normal map for the livingroom sequence (see attached video). 2nd row shows still frames from the original sequence, 3rd row with our reconstructed normal map and an artificial checkerboard pattern.

of $\delta\theta = \frac{1}{2}\operatorname{atan}\frac{\delta S}{d}$ where δS stands for the inaccuracy in the absolute camera centers, and d for the distance to the plane. Typical values for our sequences are $\delta S = 5$cm and $d = 20$m, giving an angular accuracy of $\delta\theta = 0.08$ deg. This also shows why we see bigger distortions when looking from further, and why the $z = 0$ assumption does not contradict the non orthogonal normal computation, since moving O has the same effect than moving S on angles.

Rendered data with ground truth information As a proof of concept and to analyze the accuracy of the method, a virtual surface model has been created that is essentially one period of a sine in one direction and one period of a cosine in the orthogonal direction plus some high frequency noise. The maximum difference from flat is 1.2°. Then, we used a raytracer to render a horizontal edge reflecting on this surface from a vertical camera path. The resulting movie has been processed using the complete pipeline including camera tracking, rectification and video cube segmentation, 3D line estimation and normal estimation. The geometric configuration has been chosen to mimick a real scene and camera motion (*i.e.* 1x1.5m window, 2m long hand-held camera path, 8m in front of the facade, 3D line 20m away). We obtain a mean and median error between estimated and ground truth normal of 0.1°. Additionally, we re-render the same movie just changing ground truth normals with estimated ones and obtain a visually almost indistinguishable result (see Fig. 7 and even better in the supplementary video).

To validate that this accuracy obtained from one pass only is good enough, we also render a reflection of a different vertical line from a completely different viewpoint using both the normals estimated from the horizontal line and the ground truth surface: The result looks very similar (see Fig. 7). This proves that the whole surface can be reconstructed with a single pass.

Real world surfaces We made several qualitative experiments, first with a simple case like the CD case shown in Fig. 8, and then with more challenging ones : a poster frame in a livingroom in Fig. 9, and several real facades, for example the one shown in Fig. 10 or 11. Fig. 12 is an example that demonstrates results with fully automatic tracking of reflected line. For a better overview of our results, we refer the reader to the video attached that better shows the distortion induced by the camera motion. On the real facade example, we compared the rendering with our computed normal maps with state-of-the art bulge ramp and we show still frames for comparison.

Figure 10. Three frames of another facade sequence and estimated normal map. The window surface contains several parallel visual discontinuities that are well described in the normal map

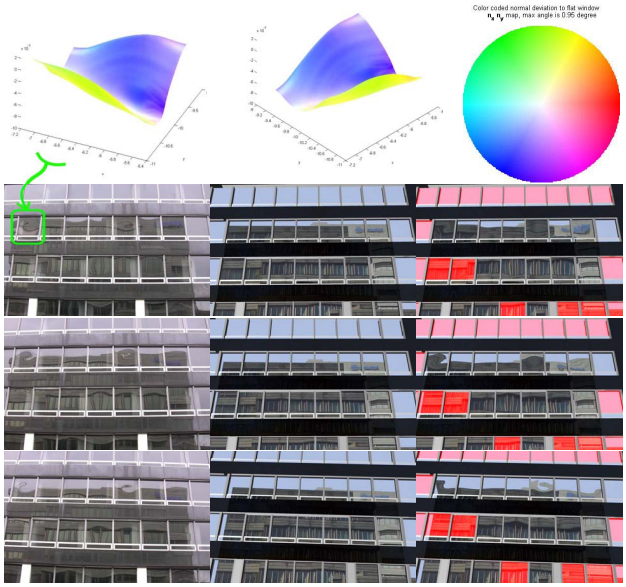


Figure 11. Estimated height field and normal map for a real facade sequence (see Sec. 4 and attached video for explanations). 1st column shows still frames from the original sequence, 2nd with perfect mirrors, 3rd our reconstruction (red windows do not have a computed normal map)

5. Conclusion and future work

We have presented a practical method to capture normal maps for almost-flat, reflective surfaces such as windows or glass/metal facades. In particular we use no more than a hand-held or vehicle-mounted video camera and track the reflection of a straight line, which is typically easy to find in urban environments. To enable this to be carried out efficiently, we have also proposed a practical method for 6 DOF tracking of the camera in front of reflective facades. Clearly, the main application of our work is realistic modeling of glass windows, therefore we also proposed how to enable automatic city-scale reconstruction using a vehicle-mounted camera. In this way, statistical models of window normal distributions can be established. Our approach can also easily be used to capture other reflective surfaces such

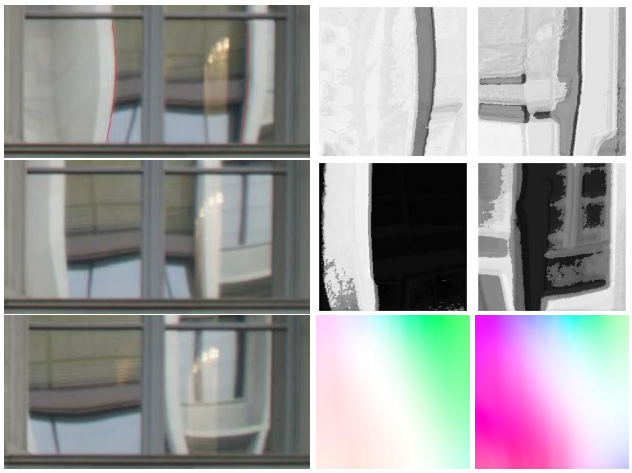


Figure 12. **Left**: cut out from three input frames containing two windows. **Right**, first row: Line classifier scores of both windows shown on left side at the respective frame (dark is high score). Second row: Before/after score (dark is after). Third row: Estimated normal maps. Note that no annotations were used.

as *e.g*. CD-boxes, picture frames, TV-sets and many other man-made objects found in our environment. It is in fact often surprising how irregular the reflection of many of these objects are and how applying generic normal maps reveals the synthetic nature.

Along the line of [23], we plan to investigate techniques to suppress or separate out fraction of light coming from transparency and multi-glazing.

We intend to explore the applicability of our technique to model distortions in transparent materials that can be observed *e.g*. in older glass with uneven thickness. There has already been some interesting work in this area, *e.g*. [13]. Another future direction is the possibility to extend our technique to non-flat reflective surfaces by making use of a proxy-geometry. This would enable to recover the detailed normal maps of cars or other reflective objects with carefully designed reflection patterns.

Acknowledgments This work was supported by the 4DVideo ERC Starting Grant Nr. 210806 and by the Swiss National Science Foundation under Project Nr. 143422. We thank Christopher Zach for publishing his Structure-from-Motion code, and for regularly providing helpful advices.

References

[1] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Leveraging 3d city models for rotation invariant place-of-interest recognition. *IJCV (S.I. Mobile Vision)*, 2012. 3

[2] J. Balzer, S. Holer, and J. Beyerer. Multiview specular stereo reconstruction of large mirror surfaces. In *CVPR*, 2011. 2

[3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. IJCAI, 1977. 3

[4] W. Chen, J. Bouguet, M. Chu, and R. Grzeszczuk. Light field mapping: efficient representation and hardware rendering of surface light fields. SIGGRAPH, 2002. 1

[5] J. Cohen, M. Olano, and D. Manocha. Appearance-preserving simplification. SIGGRAPH, 1998. 1

[6] K. Dana, B. Van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. SIGGRAPH, 1999. 1

[7] Y. Ding and J. Yu. Recovering shape characteristics on near-flat specular surfaces. In *CVPR*, 2008. 2

[8] Y. Ding, J. Yu, and P. Sturm. Recovering specular surfaces using curved line images. In *CVPR*, 2009. 2

[9] C. Frueh and A. Zakhor. An automated method for large-scale, ground-based city model acquisition. *IJCV*, 2004. 1

[10] S. Gould, O. Russakovsky, I. Goodfellow, P. Baumstarck, A. Y. Ng, and D. Koller. The STAIR Vision Library. http://ai.stanford.edu/~sgould/svl, 2010. 4

[11] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *CVPR*, 2013. 1

[12] I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Computer Graphics Forum*, 2010. 2

[13] K. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. In *ICCV*, 2005. 8

[14] J. Lellmann, J. Balzer, A. Rieder, and J. Beyerer. Shape from specular reflection and optical flow. In *IJCV*, 2008. 2

[15] M. Liu, R. Hartley, and M. Salzmann. Mirror surface reconstruction from a single image. In *CVPR*, 2013. 2

[16] A. Martinović, M. Mathias, J. Weissenberg, and L. Van Gool. A three-layered approach to facade parsing. ECCV, 2012. 3

[17] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 1

[18] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *IJCV*, 2008. 1

[19] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 2004. 1

[20] S. Roth and M. Black. Specular flow and the recovery of surface structure. In *CVPR*, 2006. 2

[21] S. Savarese, L. Fei-Fei, and P. Perona. What do reflections tell us about the shape of a mirror? In *APGV*, 2004. 2

[22] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. SIGGRAPH, 2012. 1, 2

[23] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, 2000. 8

[24] M. Tarini, H. P. A. Lensch, M. Goesele, and H.-P. Seidel. 3d acquisition of mirroring objects using striped patterns. *Graph. Models*, 2005. 2

[25] Y. Vasilyev, T. Zickler, S. Gortler, and O. Ben-Shahar. Shape from specular flow: Is one flow enough? In *CVPR*, 2011. 2

[26] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, 2010. 3

[27] C. Zach, L. Shan, and M. Niethammer. Globally optimal finsler active contours. In *DAGM*, 2009. 4

[28] R. Zhang, P.-S. Tsai, J. Cryer, and M. Shah. Shape-from-shading: a survey. *PAMI*, 1999. 2

[29] A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *Image and Vision Computing*, 1989. 2