Reflection Modeling for Passive Stereo

Rahul Nair[1] Andrew Fitzgibbon[2] Daniel Kondermann[1] Carsten Rother[3]

[1]Heidelberg Collaboratory for Image Processing, Heidelberg University, Germany
{first.last}@iwr.uni-heidelberg.de

[2]Microsoft Research, Cambridge, UK
awf@microsoft.com

[3]Computer Vision Lab Dresden, Technical University Dresden, Germany
carsten.rother@tu-dresden.de

Abstract

Stereo reconstruction in presence of reality faces many challenges that still need to be addressed. This paper considers reflections, which introduce incorrect stereo matches due to the observation violating the diffuse-world assumption underlying the majority of stereo techniques. Unlike most existing work, which employ regularization or robust data terms to suppress such errors, we derive two least squares models from first principles that generalize diffuse world stereo and explicitly take reflections into account. These models are parametrized by depth, orientation and material properties, resulting in a total of up to 5 parameters per pixel that have to be estimated. Additionally wide-range non-local interactions between viewed and reflected surface have to be taken into account. These two properties make optimization of the model appear prohibitive, but we present evidence that it is actually possible using a variant of patch match stereo.

1. Introduction

The traditional approach to stereo matching frequently models image formation as a world consisting of Lambertian surfaces observed through a perfect pin hole camera. While these assumptions, together with the right regularization, do suffice in many settings, there remain real-world situations where this is not the case. Recent benchmarks and challenges [12, 32] have shown that there are often situations where the imaging model is violated, either geometrically or radiometrically. Reflective surfaces violate the Lambertian world assumption and cause the observed color of a surface point to depend on the viewpoint. In turn, this leads to false minima in stereo matching data terms that depend on some form of brightness constancy (cf. Figure 1). The traditional approach to handle specular surfaces is either by robust data terms or by using strong regularization techniques. The work presented here has a different goal and was guided by the following questions: Is it possible to derive a data term that explicitly takes reflective materials into account? And if so, is this model of any use? Can we estimate scene parameters using this model?

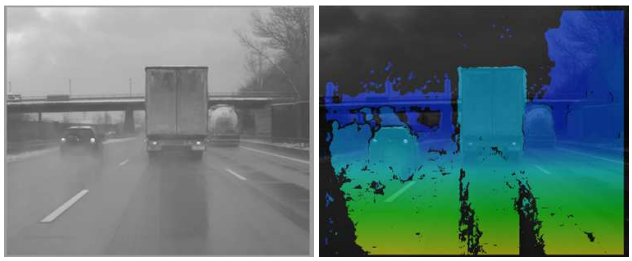


Figure 1. Illustration of Errors Caused by Reflective Surfaces. **Left:** Stereo image. **Right:** Typical Disparity Map. The color observed on the road at the reflection of the silhouette depends on the position of the camera. The algorithm therefore assigns erroneous disparity values at this location.

Our findings on handling stereo with reflections are summarized in Figure 2. By additionally modeling up to two bidirectional reflectance distribution function (BRDF) parameters, it is not only possible to remove errors due to reflective surfaces, but it is also possible to obtain material information from the two images that can potentially be used for segmentation purposes or view synthesis. Finally, while not explicitly estimated, the separation of diffuse and reflection components falls out of the box. Note that the work presented does not include any form of global regularization or post-processing on top of the presented results. This derived from the goal to give insights upon the utility of the proposed data-terms. The estimated parameters are a sole result of local optimization of the proposed model parameters.

In Section 3, we revisit the roots of stereo matching as a least squares problem and from this derive simplified models that take reflections into account. Also, we show that traditional diffuse world stereo is in fact just another special case. The models are parametrized by per pixel depth, normals and up to two surface material parameters which encode strength of the reflection component and roughness

of the reflecting surface. All in all, there are up to 5 parameters per pixel (cf. Fig. 2). The resulting optimization problem is high dimensional and requires that the surface belonging to each pixel 'knows' which surfaces it reflects from, thus yielding wide-range non-local interactions. We demonstrate that the optimization still is tractable using Stereo PatchMatch [7] with extensions that enable efficient reflection computation and more accurate normal estimation (cf. Section 4). While the computation of accurate surface orientation is usually not the main goal in stereo matching, it turns out that accurate normal estimation is the key to handling reflective surfaces. These insights and the properties of the resulting algorithms are further discussed in Section 5.

2. Related Work

Early approaches in handling specular surfaces involved the detection of specular highlights [3, 13, 21] and subsequent exclusion of the detected areas. Another approach with a similar goal is the usage of a cost function that is more robust towards specular highlights, for example the image gradient [7] or rank-based costs [5, 15, 16]. Both of these cost types achieve robustness towards image-wide or low frequency radiometric differences in the input images, but still have issues with strong specular highlights or high frequency reflections. For handling stronger highlights, Jin et. al. [18] make use of a rank-based cost, though in a multi-view setting. All these methods have in common that they do not change the diffuse world model but rather limit the reconstruction to the diffuse parts implicitely or explicitely. The apparent displacement of specular highlights provides information on the normals and surface curvature of object surfaces [6]. These displacements have been used to recover mirror surfaces [27, 1] given controlled lighting conditions. Another approach is *layer separation*, where the world is treated as a set of semitransparent depth layers mixing the color with each other. These methods either require user interaction [22] or operate in a multi-view setting [8, 10, 31, 30]. These previous models still restrict the correspondence of reflected objects to the horizontal scan line. In reality however this does not have to hold[6].

The model presented here is different in that the physical properties of the observed surfaces are modeled and these implicitly define a second observable layer. By doing so, it is possible to use reflection information in the image wherever it is available and thus obtain a parameter map akin to a material segmentation of the image.

An alternative might seem to be an example-based approach to material modeling [29], but such techniques would require a prohibitively large number of examples to learn any thing more than specular highlights.

The work presented here is also closely related to various *inverse rendering* problems[25]. Common problems [25] require that a certain subset of variables has to be known e.g. inverse lighting [19] or inverse reflectometry [23]. Finally, we note that the *optimization* problem that is required to be solved here has a high-dimensional state vector at every pixel and an energy with long-range interactions between pairs of pixels. Moreover, the variables participating in the interactions are themselves a function of the unknown parameters. Until recently, this would have appeared tractable only for very simple greedy algorithms. However, recent work on the PatchMatch algorithm [17] has shown that it is an effective optimizer even for very high-dimensional state vectors.

3. Reflections on Stereo

The scene parametrization is depicted in Figure 2. The world is assumed to be representable on image grid Ω, where each pixel $i \in \Omega$ represents a surface element parametrized by radial distance d_i from the primary (left) camera center, surface normal orientation (θ_i, ϕ_i), diffuse color f_i and additional material parameters (μ_i, σ_i). Next, define V as the set of cameras expressed by their extrinsic and intrinsic parameters. For stereo $V = \{L, R\}$. Note that though d_i is a scalar, it implicitly corresponds to a 3D point and also a ray direction by a function $\mathbf{x}^v(d_i)$, defined only by the (known) camera parameters $v \in V$. When the superscript v is omitted, we refer to a 3D point in the primary (L) camera system. Similarly, (θ_i, ϕ_i) define the normal $\mathbf{n}^v(\theta, \phi)$ and d_i and $\mathbf{n}^v$ together define a plane $\mathbf{p}^v(d_i, \mathbf{n}^v)$. Wherever it eases readability, we simply refer to these derived values as $\mathbf{x_i}$, $\mathbf{n_i}$ and $\mathbf{p_i}$ respectively. The color vector f_i is required only for the derivation of the model. With the simplifications that will be made, we will see that f_i can be implicitly recovered from the observed color using the other parameters (cf. Eqs. 8,16). For convenience, we define $\mathbf{s}_i = \{d_i, f_i, \theta_i, \phi_i, \mu_i, \sigma_i\}$ as the set of all unknown parameters at $i \in \Omega$. With bold face capital letters we refer to the set of a single parameter over all pixels, e.g. $\mathbf{D} = \{d_i\}$, $\mathbf{F} = \{f_i\}$, $\mathbf{S} = \{\mathbf{s}_i\}$, ..., $i \in \Omega$. Given $v \in V$, define $\pi^v : \mathbb{R}^3 \to \mathbb{R}^2, v \in V$, as the mapping that projects 3D world points into view v. When applied to scalar d_i, $\pi^v(d_i) = \pi^v(\mathbf{x}_i(d_i))$, Finally, let C be a color space and let $I^v : \mathbb{R}^2 \to C, v \in V$ map the position on the 2D image plane of camera v to the observed color at this point with bilinear interpolation for non-integer coordinates. The least squares stereo data term can then be expressed as the sum of pixel-wise costs $LSQ(\mathbf{S}) = \sum_{i \in \Omega} E(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i})$, where the pixel-wise cost $E(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i})$ is defined as

$$E(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i}) = \sum_{v \in V} ||I^v(\pi_i^v(d_i)) - m(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i}, v)||_2^2. \quad (1)$$

The model function in Eq. 1 computes the observed color of i in view v as a function of the parameters $\mathbf{s}_i$, the set of all other world parameters $\mathbf{S}_{\setminus \mathbf{s}_i}$ and view v. Note that $d_i \in \mathbf{s}_i$.

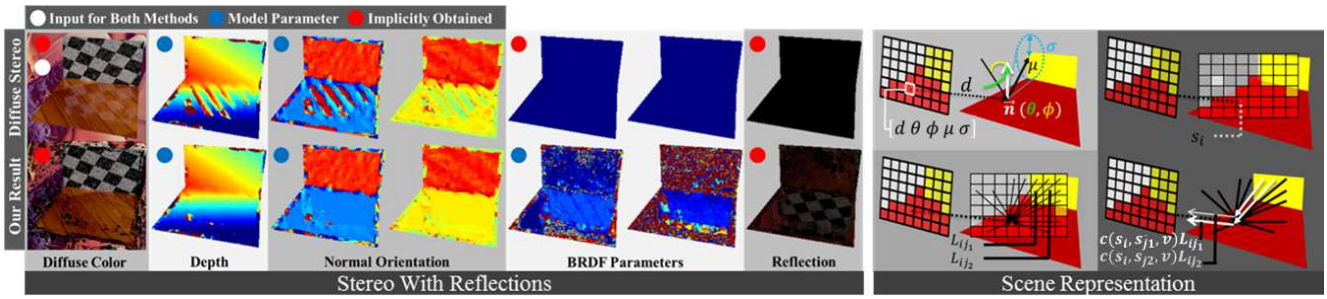


Figure 2. **Left:** Stereo with Reflections. By explicitly parameterizing the BRDF as well as geometry (a total of 5 parameters per pixel), and exploiting the ability of PatchMatch to efficiently optimize high-dimensional energies, it is possible to obtain the BRDF *and* a better geometry. **Right:** Scene representation and illustration of screen-space render equation. Per pixel, surface geometry is parametrized using the depth r along the pixel ray and the surface normal represented by the Euler angles (θ, ϕ). Materials are represented by a mixing parameter μ and optionally the specular roughness σ. Larger μ corresponds to stronger reflections. Larger σ means more diffuse reflections.

The observed color from any viewpoint is most generally explained by the rendering equation [20], which, modified to our notation and assuming isotropic light sources is given by (cf. Fig. 2, Right)

$$m(\mathbf{s_i}, \mathbf{S}_{\setminus \mathbf{s}_i}, v) = e_i + \sum_{j \in \Omega \setminus i} c(\mathbf{s}_i, \mathbf{s}_j, v) L_{ij}. \tag{2}$$

In essence, this equation states that the color observed at a location (the pixel) from a surface point corresponds to the amount of light e_i that the surface patch emits itself and the fraction $c(\mathbf{s}_i, \mathbf{s}_j, v)$ of light L_{ij} received from another surface point j that is reflected into the camera v. The function c corresponds to a discrete version of the BRDF, which, as a reminder, is a material specific property that governs how surfaces appear under different lighting and viewing angles. Note that in general, the light L_{ij} transported from one surface to another depends on the light that the transmitting surface receives from all other surfaces in the scene etc. There is no analytical solution for the forward problem so that renderers have to employ Monte Carlo or Finite element techniques to compute the full global light transport. W.l.o.g. we assume that the BRDF c decomposes into a diffuse, viewpoint independent part (i.e. a constant part) and a viewpoint dependent specular part

$$c(\mathbf{s}_i, \mathbf{s}_j, v) = c_i^{\text{diff}} + c^{\text{spec}}(\mathbf{s}_i, \mathbf{s}_j, v), \tag{3}$$

such that Eq. (2) can be written as

$$m(\mathbf{s_i}, \mathbf{S}, v) = e_i + \sum_{j \in \Omega \setminus i} c^{\text{diff}} L_{ij} + \sum_{j \in \Omega \setminus i} c(\mathbf{s}_i, \mathbf{s}_j, v) L_{ij}. \tag{4}$$

Since the amount of light received from the other surfaces is viewpoint independent L_{ij}, we define the diffuse color f_i of the surface point as

$$f_i = e_i + \sum_{j \in \Omega \setminus i} c^{\text{diff}} L_{ij}. \tag{5}$$

Finally, we make a **single-bounce assumption**: the light received from another surface position only corresponds to its diffuse color. Obviously, the model now cannot explain multiple reflections, but this is an approximation required to make the model tractable. Using this approximation, Eq. (2) can be rewritten as

$$m(\mathbf{s_i}, \mathbf{S}, v) = f_i + \sum_{j \in \Omega \setminus i} c^{\text{spec}}(\mathbf{s}_i, \mathbf{s}_j, v) f_j. \tag{6}$$

The actual model that is now obtained depends on the definition of c^{spec}. In the following, we will show that the standard stereo model is a special case of Eq. (6) with a diffuse BRDF. Furthermore, we will present two other models that are of interest and which arise by plugging in other BRDF models.

Diffuse World Stereo (DN)[1] For $c^{\text{spec}}(\mathbf{s}_i, \mathbf{s}_j) \equiv 0$, we obtain

$$E(\mathbf{s}_i, \mathbf{S}) = \sum_{v \in V} \| I^v(\pi^v(d_i)) - f_i \|_2^2. \tag{7}$$

The solution for $\mathbf{F}$ given $V = \{L, R\}$ and depth map $\mathbf{D}$ is

$$f_i^{\text{diffuse}} = \frac{I^L(\pi^L(d_i)) + I^R(\pi^R(d_i))}{2}. \tag{8}$$

Inserting f_i^{diffuse} into Eq. (7), we obtain

$$E(d_i) = \frac{1}{2} \sum_{i \in \Omega} \left\| I^L(\pi^L(d_i)) - I^R(\pi^R(d_i)) \right\|_2^2, \tag{9}$$

which coresponds to the standard least squares stereo matching term.

Delta-BRDF Model (DNM)[2] Consider

$$c^{\text{spec}}(\mathbf{s}_i, \mathbf{s}_j, v) = \begin{cases} \mu_i & \text{if } H(\mathbf{n}_i^v)\mathbf{x}_i^v \times \left(\mathbf{x}_j^v - \mathbf{x}_i^v\right) = 0 \\ 0 & \text{otherwise}, \end{cases} \tag{10}$$

[1] **D**epth, **N**ormals. The latter only used with cost aggregation
[2] **D**epth, **N**ormals, **Mu**

where $H(n) = I - 2nn^{\mathrm{T}}$ is the Householder transform that describes mirror reflection.

This BRDF model corresponds to a perfect mirror reflection which is only weighted by the parameter μ_i. The sum in Eq. (6) reduces to

$$m_{\mathrm{DNM}}(\mathbf{s_i}, \mathbf{S}, v) = f_i + \mu_i f_{\rho(i,\mathbf{S},v)}, \tag{11}$$

where the function $\rho(i, \mathbf{S}, v)$ finds the pixel corresponding to the intersecting surface point. In practice ρ has to be implemented by some form of ray tracing. In the next section, we show how this is done efficiently in screen space. Also note how this model is extremely sparse in the number of interactions for a fixed choice of surface normals.

Rough Gloss Model (DNMS)[3] Finally, we consider a specular term of the form

$$c^{\mathrm{spec}}(\mathbf{s}_i, \mathbf{s}_j, v) = \begin{cases} \frac{\mu_i}{M(\mathbf{S},i,v)} & \text{if cd } < \sigma_i \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

$$\text{cd} = \left\langle \frac{H(\mathbf{n}_i^v)\mathbf{x}_i^v}{||H(\mathbf{n}_i^v)\mathbf{x}_i^v||}, \frac{\mathbf{x}_j^v - \mathbf{x}_i^v}{||\mathbf{x}_j^v - \mathbf{x}_i^v||} \right\rangle, \tag{13}$$

where $M(\mathbf{S}, i, v)$ is a normalizing factor corresponding to the number of pixels for which the condition cd is true. This type of BRDF implies a constant value if the angle between mirror reflection direction and direction toward $\mathbf{s}_j$ is smaller than a certain threshold defined by the fourth parameter σ. With this kind of BRDF model, we emulate the roughness parameters observed in common BRDF models such as Phong, Gaussian or Ward BRDF models. The corresponding model is

$$m_{\mathrm{DNMS}}(\mathbf{s_i}, \mathbf{S}, v) = f_i + \frac{\mu_i}{M(\mathbf{S}, i, v)} \sum_{j \in \tilde{\Omega}_i^v} f_j. \tag{14}$$

The differences to Eq. (6) are quite subtle. c^{spec} can be eliminated by reducing the support of the inner sum to those pixels that lie in the valid range. The number of entries in the sum can still get quite large with a larger distance between viewed surface and reflected object, thus making the evaluation of the objective very time-consuming. In the next section a constant time screen space approximation for the computation is presented.

Eliminating first bounce f_i The two Eqs. (11) and (14) both have the structure

$$m_{\dots}(\mathbf{s_i}, \mathbf{S}, v) = f_i + \mu_i r_i^v(\mathbf{S}), \tag{15}$$

where $r_i^v(\mathbf{S})$ (abbr. r_i^v) is the reflection component. Following the same arguments as in the diffuse case, the least square solution for f_i for fixed geometry is given by

$$\frac{I^L(\pi^L(d_i)) + I^R(\pi^R(d_i)) - \mu_i \left(r_i^L + r_i^R\right)}{2}, \tag{16}$$

yielding the following per pixel cost for both models:

$$E(\mathbf{s}_i, \mathbf{S}) = \frac{1}{2} \left\| I^L(\pi^L(d_i)) - I^R(\pi^R(d_i)) - \mu_i \left(r_i^L - r_i^R\right) \right\|_2^2. \tag{17}$$

While $\mathbf{F}$ has not been completely eliminated from the cost, it now only appears in the reflection term. In the next section, this is further simplified to compute reflections in screen space.

Offscreen bounces Our model assumes that specular bounces touch parts of the scene that are visible in the image. In a similar vein, some readers may wonder why light sources outside the scene were not mentioned at all. The straight forward answer to this is model tractability. Some form of prior information has to be introduced to estimate effects from outside the scene. In the present work, the main interest is in what can be derived only from information available in the image. Therefore, instead of additionally modeling lights and shading, the diffuse shading and diffuse reflection of lights as well as surface emissivity are just part of the diffuse color f_i. Similarly, if the diffuse color is not limited to lie in the [0, 1], f_i can also model observed light sources. The case when the model is still violated is when a specular surface introduces the off screen bounces. As mentioned, handling these is subject to future work.

4. Inference

The DNM and DNMS models still have wide-range non local interactions as the reflected color observed in a certain pixel still depends on the geometry of the scene and the diffuse color of the reflected pixels. In our experience, trying to directly minimize the energy is slow and does not yield any useful results. With variational techniques we found that a good initial guess or a scale space approach was needed. On most surfaces that are not purely glossy, the reflected part of the signal bears similarities to a low-pass version of the perfect mirror reflection. Therefore, cues for the actual surface and the reflected surface appear on different scales, thus violating the basic assumption of scale space approaches that the estimated depth is consistent over all scales. An observation made on many stereo results that suffer from reflection is that erroneous depth is only measured when texture or silhouette edges are reflected. Therefore, the geometry next to reflection artifacts are often correct. On the other hand, the only areas that can be used to reliably measure the material properties are precisely these edges, as they contain the required information on μ and σ. A heuristic to account for the latter a patch based aggregation strategy of costs lends itself while for the former regions with correct geometry are required to somehow propagate their information into erroneous regions.

PatchMatch Stereo [7] displays this kind of behavior, so suggests itself as a framework to solve these models. In the following, we show that by making some further simplifi-

[3] **D**epth, **N**ormals, **M**u, **S**igma

cations to the model and some extensions to the framework, both DNM and DNMS models can be solved using stereo Patchmatch. The basic strategy is to first solve for standard diffuse stereo to obtain an initial guess for geometry. To achieve normal estimation of sufficient accuracy, PatchMatch with continuous refinement is required. This novel extension to PatchMatch optimization is described below. Using this initial guess, again two iterations of continuous PatchMatch are applied using the DNM or DNMS models respectively to obtain the final result.

PatchMatch Stereo revisited PatchMatch stereo operates on an extended cost volume

$$C : \Omega \times \mathbb{R}^N \rightarrow \mathbb{R}, (i, \mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i}) \rightarrow E^{pm}(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i}), \quad (18)$$

which outputs the cost for assigning parameter $\mathbf{s}_i$ to pixel location i. E^{pm} is usually defined as a basic pixel cost $E(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i})$ aggregated over a support neighborhood $N^{pm}(i)$ around i, with

$$E^{pm}(\mathbf{s}_i, \mathbf{S}_{\setminus \mathbf{s}_i}) = \sum_{j \in N^{pm}(i)} w\left(I^L(i), I^L(j)\right) E\left(\tau(j, s_i), \mathbf{S}_{\setminus \mathbf{s}_i}\right). \quad (19)$$

Here, $w(I^L(i), I^L(j))$ is an optional weighting term that can either be constant or an color adaptive support weight

$$w\left(I^L(i), I^L(j)\right) = \exp(\gamma^{-1} ||I^L(i) - I^L(j)||_1). \quad (20)$$

The mapping τ transforms the $\mathbf{s}_i$, which is represented according to pixel i into a representation according to pixel j. For standard fronto-parallel stereo where disparities d_i are estimated ($\mathbf{s}_i = d_i$), the mapping is

$$\tau^D : (j, d_i) \rightarrow d_i. \quad (21)$$

In [7] it is assumed that the patch geometry can be described by a slanted plane, therefore we get

$$\tau^{DN} : (j, \{d_i, n_i\}) \rightarrow \{||\mathbf{x}_j \cap \mathbf{p}_i||, \mathbf{n}_i\}. \quad (22)$$

The $\cap$ symbol denotes the intersection of the direction given the pixel ray $\mathbf{x}_j$ and the plane $\mathbf{p}_i$ defined by $(d_i, \mathbf{n}_i)$. This maps the normals as they are, but transforms the depth such that it belongs to the same plane as the surface described by $\mathbf{s}_i$ in pixel i.

The algorithm then operates as follows: for initialization, the $\mathbf{s}_i$ are drawn randomly from the feasible set of parameters. Then two steps are alternated for each pixel and all pixel is traversed in some order:

In the **propagation** step, the current parameter set in i is replaced by

$$\mathbf{s}_i^{new} = \underset{\{s_j | j \in N(i)\}}{\arg\min} E^{pm}(\tau(i, \mathbf{s}_j), \mathbf{S}_{\setminus \mathbf{s}_i}), \quad (23)$$

where $N(i)$ describes some neighborhood of i. Note that this is the same τ as defined above, but instead of applying the same s_i to several neighboring pixels, we are now choosing the neighboring $\mathbf{s}_j$ that gives the smallest cost when applied to the current pixel i.

In **random refinement** the current parameter set in i is then refined by drawing random parameters around the current parameter according to some probability distribution $\mathcal{D}(\mathbf{s}_i, \alpha)$ centered around $\mathbf{s}_i$ with additional parameter α that usually corresponds to the variance of the distribution

$$\mathbf{s}_i^{new} = \underset{\mathbf{s} \sim \mathcal{D}(\mathbf{s}_i, \alpha)}{\arg\min} E(\tau(i, \mathbf{s}), \mathbf{S}_{\setminus \mathbf{s}_i}). \quad (24)$$

In stereo PatchMatch this $\mathcal{D}$ corresponds to a double exponential distribution[4]. An intuition and a proof of why this technique works are given in [4]. To sum it up, the method works well if the scene consists of large homogeneous areas with the same or slowly varying parameter sets. This is to some extent true for general depth maps and more so for materials, as natural scenes often only consist of a few different materials.

Proposed PatchMatch Variant We make three modifications to the original PatchMatch implementation: First, the refinement step is extended to do gradient-descent based refinement after random refinement. This step significantly improves the quality of the estimated geometry (especially normals) even for diffuse PM. Next, the per pixel cost is modified in such a way that it does not depend on the parameters in the other pixels. This enables the application of continuous PM also to the DNM and DNMS models. Finally, data driven sampling routines are employed for the random refinement step. In the following, we limit ourselves to a general overview of each of these modifications and refer to the supplemental material for additional implementation details.

Continuous Refinement If the pixel-wise cost is defined in such a way that it is differentiable w.r.t $\mathbf{s}_i$, i.e. the Jacobian $J_E(\mathbf{s}_i)$ can be computed, it is possible to find the local cost minimum using gradient descent or trust region solvers. For the DN model this is evident if linear or higher order spline interpolation between pixels is employed. For DNM and DNMS this becomes a bit more challenging since the evaluation of the cost requires a ray-tracing step. It is also important to be able to compute the derivatives of the reflected color with respect to the change of normal orientation. In practice, the continuous part of the optimization is implemented using the Ceres-Solver [2] library, which computes exact derivatives using automatic differentiation techniques [26] given a differentiable objective function.

Screen Space Reflection Computation For continuous optimization of DNM(S) the reflected color at pixel i has to be computed in a way that is differentiable. We achieve this using a series of approximations illustrated in Fig. 3.

[4]The sampling strategy employed additionally stratifies the samples into quantile brackets.

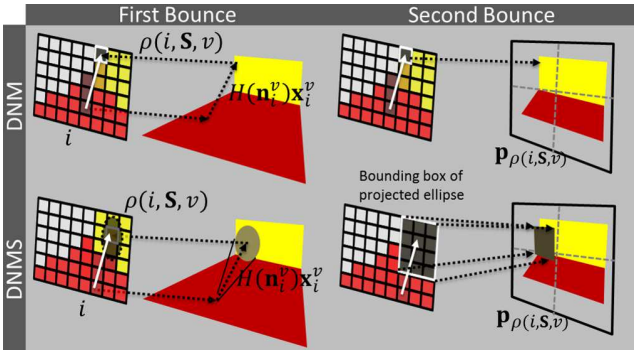


Figure 3. Illustration of screen space reflection computation for DNM and DNMS. First $\rho(i,\mathbf{S},v)$ is computed by projecting $H(\mathbf{n}_i^v)\mathbf{x}_i^v$ into the camera and searching for intersections along the resulting line. Next, the geometry of the reflected world is approximated using $\mathbf{p}_{\rho(i,\mathbf{S},v)}$ enabling differentiation w.r.t. parameters in i. For DNMS the projection of the conic intersection is approximated by a rectangle which is aggregated over using integral images [11].

First, we find the location $\rho(i,\mathbf{S},v)$ of the mirror reflection by intersecting the mirror ray $H(\mathbf{n}_i^v)\mathbf{x}_i^v$ at i given view v with the current geometry. To efficiently compute the intersection, we borrow ideas from recent work in computer graphics [14, 24]. $H(\mathbf{n}_i^v)\mathbf{x}_i^v$ is first projected into view v resulting in a line in image space. We subsequently search for the intersection along this line using [9].

Once $\rho(i,\mathbf{S},v)$ is found the DNM reflection color can be approximated as $r_i^v \approx I^v(\rho(i,\mathbf{S},v))$. This term is not differentiable w.r.t. the parameters at i. To obtain a differentiable term a second approximation has to be made: we assume that the geometry of the reflected world can be described by the plane $\mathbf{p}^v_{\rho(i,\mathbf{S},v)}$. Then $r_i^v = I^v(\pi^v(\mathbf{p}^v_{\rho(i,\mathbf{S},v)} \cap H(\mathbf{n}_i^v)\mathbf{x}_i^v))$. By setting $\mathbf{p}^v_{\rho(i,\mathbf{S},v)}$ constant during continuous refinement and only updating it during propagation and random refinement we now obtain a differentiable term.

For the DNMS model, the contributions of many pixels have to be taken into account. Evaluating this cost can therefore consume a large amount of time. We simplify this in a similar manner as above, yielding a constant computational overhead, irrespective of the area of integration. First, we compute the geometry of the reflected world in the same way as above. Next, we project the intersection between this plane and the cone given by Eq. 12 (cf. Fig. 3 bottom right) into view v to obtain an area of support over which we have to integrate the color. Finally, we approximate this region using a rectangular shape, such that the sum can be computed using bilinearly interpolated integral images [11] in $O(1)$ to obtain a differentiable reflection term for DNMS.

Data Driven Sampling Finally during random refinement of the DNM/S models, we replace $\mathcal{D}(\sigma,\theta)$ with a screen space sampler. Given current reflected position $\mathbf{s}_j$, the sampler uniformly samples neighboring pixels as candidate reflection points $\tilde{\mathbf{s}}_j$. The orientation parameters are then computed such that they satisfy $\tilde{\mathbf{s}}_j$ to be the primary point of reflection. This sampling is done additionally to the standard exponential sampling of orientation to allow for searching the proximity of $\tilde{\mathbf{s}}_j$ more closely.

5. Experiments and Results

We refer to standard PatchMatch with PM and, similarly, to continuous (data driven) PatchMatch with CPM and CDDPM. Additionally, we prefix the optimization method with the model that is to be inferred. DN-PM, DNM-PM, DNMS-PM therefore refer to standard PatchMatch optimization using the DN, DNM and DNMS models respectively. The algorithms utilized a patch window of size 13 px and an exponential color based adaptive support weight (ASW) [7] with parameter 0.08 for images normalized between 0 and 1.

The DNM and DNMS models are more sensitive to the choice of color-based ASW since strong reflected edges that give the primary cues for estimating the material properties also cause a strong down-weighting of pixels. The scenes used in the following experiments were modeled in Blender and rendered using the Blender-Cycles renderer that approximates global illumination. This allows for ground truth evaluation and verification of the reconstructed parameters. Experiments were run on a Intel i7-4470K @ 3.50 GHz. The baseline (DN-PM) requires 5s per PM-iteration. Continuous optimization (DN-CPM) induces a 3x overhead (15 s/iter.). The proposed models require an additional factor of 2.5 (DNM-CDDPM, 40s/iter) and 18 (DNMS-CDDPM, 270s/iter) compared to DN-CPM. The bottleneck here is the screen-space reflection computation. To project

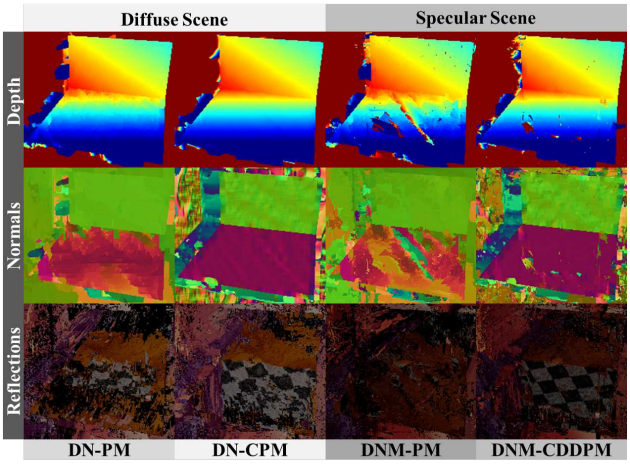


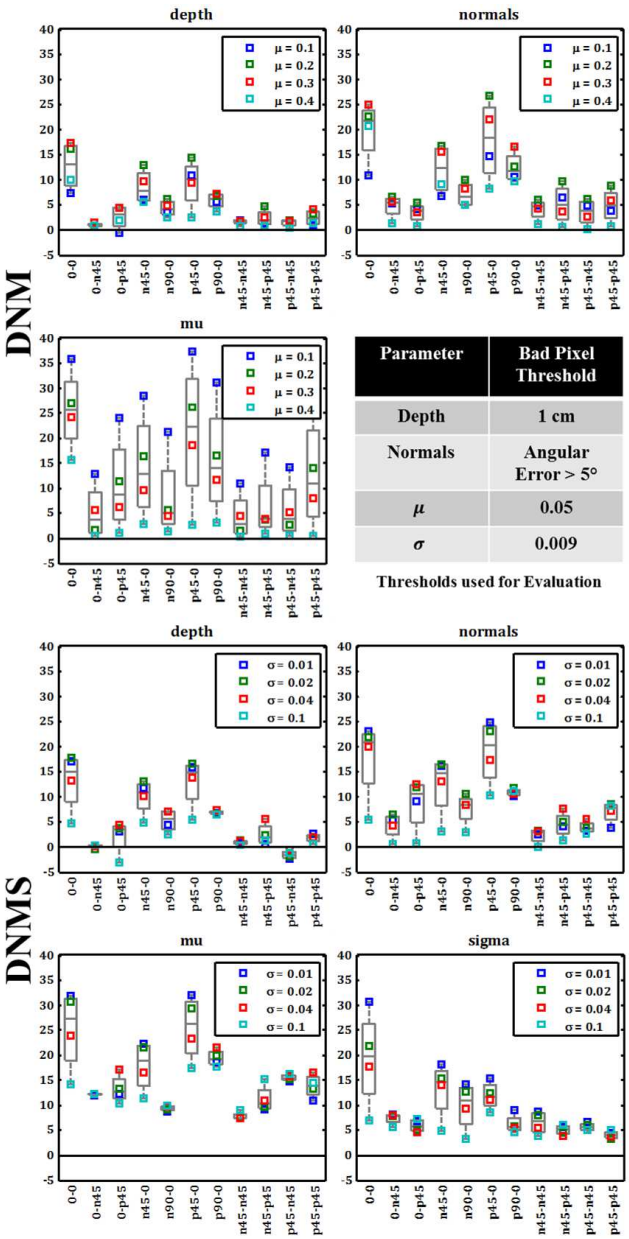
Figure 4. Quality of Normals. **Top to bottom**: depth map, RGB-coded normals and computed reflections. **From left to right**: a) and b) results of DN-PM vs DN-CPM on a diffuse scene. c) and d) result of DNM-PM vs DNM-CDDPM. Inputs are from Fig. 2

the speed-up that is currently possible using a faster implementation, we note that recent work on this topic report real-time frame-rates for both DNM (500fps)[24] and DNMS[14]. This indicates that it may be possible reduce the overhead of these models to the level of DN-CPM.

Quality of Normals The continuous data driven PatchMatch approach was motivated by the goal of achieving high quality normals. To verify the effect of normal estimation on handling reflections, we ran PM and CPM/CDDPM on a fully diffuse scene and a scene containing a specular surface. The results are illustrated in Figure 4. For the diffuse scene (left half), the reconstructed depth maps are nearly identical using DN-PM and DN-CPM (small deviations can be observed in detail though). Yet, the normal map reveals large differences. The computed mirror reflection using these normals also confirms these findings. For the specular scene (right half), we compare two iterations of DNM-PM after two iterations of DN-PM with 2 iterations of DNM-CDDPM after two iterations of DN-CPM. The differences here are more striking both in depth and normals. Notice the (erroneous) low frequency normal error of the lower surface in DN-CPM, which is no longer present in DNM-CDDPM wherever the surface reflected something else in the scene. This is a strong indicator that modeling reflection not only can correct errors due to reflections, it actually can aid in more accurate geometry estimation. The improved micro-structure of the lower surface is further evidenced by the quality of the computed reflections. Finally, some artifacts can still be observed in the DNM result. These are due to reflections of occlusion boundaries that have a similar effect as the ones normal occlusions have in standard stereo. This is not a shortcoming of the model per se, but a result of the simplifications made to compute the reflected color.

Additional evaluation of the CPM vs PM performance were carried out on the Middlebury Stereo Datasets [28]. While this dataset was not intended to evaluate normals we do observe a consistent improvement of the bad-disparity rate between DN-PM and DN-CPM of up to 1%. Further analysis can be found in the supplemental material.

DNM/S Model Verification We compared DN-PM with DN-CPM, DNM-CDDPM and DNMS-CDDPM for 11 different scenes of varying curvature (foreground@ 4-6 m, background@ 60 m, 50° FOV) of the specular surface. The ground truth BRDF parameters of the specular surface are constant over the whole surface. For the evaluation of the DNM model, for each scene the μ parameter for the lower surface was varied between 0.0 and 0.4. The latter corresponds to a peak diffuse signal to reflection ratio of over 0.6 in this scene. Similarly we report results for DNMS with $\mu = 0.25$ and $\sigma = 0.01, 0.02, 0.04$ and 0.1 created using two iterations of DNM/S-CDDPM after two iterations of DN-CPM . We display example results on DNMS-CDDPM



Parameter	Bad Pixel Threshold
Depth	1 cm
Normals	Angular Error > 5°
μ	0.05
σ	0.009

Thresholds used for Evaluation

Figure 5. Summary of results for each scene for different ground truth parameters (markers) of the specular surface. Y-Axis: Decrease in number of bad pixels in percentage of total number of pixels. Larger values are better. X-Axis: Enumeration of Scenes (cf. Fig. 6). The box plot additionally mark median and quartiles. The inlet table contains the bad pixel thresholds utilized. We observe consistent improvements of the reconstructed parameters. **DNM:** The overall trend is a deterioration of results for stronger μ. **DNMS:** The overall trend is towards a smaller effect for larger σ consistent with large σ corresponding to more diffuse surfaces.

in Fig. 6. Further results used in the subsequent analysis can be found in the supplemental material. Fig. 5 quantifies the results over all tested parameters and scenes for the DNM and DNMS models respectively. In these plots, we

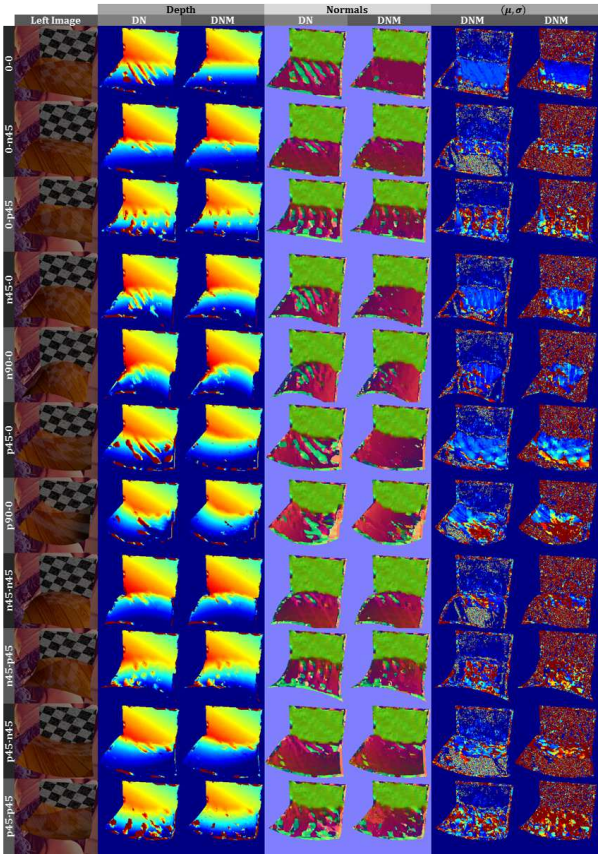


Figure 6. Example Scenes Used for the ground truth evaluation (μ = 0.25, σ = 0.01). From left to right: Left image, diffuse-depth, DNMS-depth, diffuse-normals, DNMS-normals, μ, σ.

report the decrease in the number of 'bad' pixels (cf. Table inset in Fig. 5) between results using DN-CPM and results using DNM/S-CDDPM for each of the parameters.

The metrics mostly correspond to the 3D-space version of the bad-pixel metric commonly used in Middlebury evaluations [28]. They were chosen as they are best suited for the multi-modal, heavy-tailed error distributions that are caused by reflections. Summarizing, DN-CPM consistently decreases the GT error over DN-PM, and DNM/S-CDDPM consistently further decreases the error. The scenes where the relative decrease is low, correspond to the situation where the actual area reflecting something is relatively small (e.g. Scene 0-n45 (second row) in Fig. 6). The remaining artifacts often correspond to the reflections of depth edges. Also consistent with the findings above are the normals that are improved upon in areas that reflect other parts of the scene. The proposed method is able to improve the geometry and recover meaningful parameters over a wide range of different surface curvatures. The most difficult situation happens to be a convex surface oriented towards the camera as lot of reflected rays bounce back into the direction of the camera. For larger values of μ,

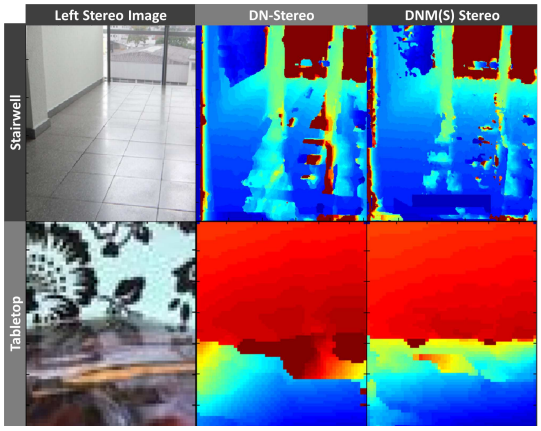


Figure 7. Real Images. From left to right: Left image, Diffuse depth, DNMS(top)/DNM(bottom) depthmaps. Note the reduction of errors caused by specular reflections.

the proposed optimization strategy starts to fail, while for larger values of σ the scene becomes indistinguishable from a completely diffuse scene such that the DN model does not produce artifacts.

Real Images As a proof of concept, we ran the methods on a stairwell scene with diffuse specular reflections from a grating (and outside) as well as a tabletop scene with mirror reflections similar to the synthetic ones used above. While we observe improvements in many parts of both scenes (reflection of grating, reflection of black and white surface), there remain erroneous areas due to ambiguities and saturation effects. Yet, we note that these are the results achieved only using the local data term and expect further improvements if regularization is included. A discussion of further aspects important to the applicability to real images can be found in the supplemental material.

6. Conclusion

This work addressed the matter of specular reflections which violate the diffuse world model commonly used for stereo matching. By including the second order terms of the image formation model governed by the render equation, we derived two data terms that are capable of explaining specular reflections. We showed that the optimization of the resulting optimization problem is possible using CDDPM. In consequence, it was possible to estimate depth, normal orientation and material parameters in each pixel. Ground truth evaluation on synthetic datasets shows consistent improvement of estimated parameters and also indicates that by harnessing reflection as opposed to suppressing it, it is possible to estimate geometry with a higher accuracy.

Acknowledgements This work was partially funded by the HGS MathComp, Heidelberg University. Furthermore, we would like to thank our anonymous reviewers as well as Holger Heidrich, Florian Becker and Frank Lenzen for their invaluable comments.

References

[1] Y. Adato, Y. Vasilyev, T. Zickler, and O. Ben-Shahar. Shape from specular flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(11):2054–2070, Nov. 2010.

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org, 2014.

[3] R. Bajcsy, S. Lee, and A. Leonardis. Color image segmentation with detection of highlights and local illumination induced by inter-reflections. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, volume i, pages 785–790, June 1990.

[4] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–24:11, July 2009.

[5] D. N. Bhat, S. K. Nayar, and A. Gupta. Motion estimation using ordinal measures. In *CVPR 1997*, pages 982–987. IEEE, 1997.

[6] A. Blake and G. Brelstaff. Geometry from specularities. pages 394–403, Dec. 1988.

[7] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, pages 14.1–14.11. BMVA Press, 2011.

[8] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[9] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Syst. J.*, 4(1):25–30, Mar. 1965.

[10] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding*, 97(1):51 – 85, 2005.

[11] F. C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3):207–212, Jan. 1984.

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR 2012*, pages 3354–3361, June 2012.

[13] R. Gershon, A. D. Jepson, and J. K. Tsotsos. The use of color in highlight identification. In *IJCAI*, pages 752–754, 1987.

[14] L. Hermanns and T. A. Franke. Screen space cone tracing for glossy reflections. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 102:1–102:1, New York, NY, USA, 2014. ACM.

[15] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *CVPR '07*, pages 1–8, June 2007.

[16] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1582–1599, Sept. 2009.

[17] M. Hornacek, F. Besse, J. Kautz, A. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using patchmatch belief propagation. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision , ECCV 2014*, volume 8691 of *LNCS*, pages 220–234. Springer International Publishing, 2014.

[18] H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo beyond lambert. In *CVPR 2003*, volume 1, pages 171–178, June 2003.

[19] V. Jolivet, D. Plemenos, and P. Poulingeas. Inverse direct lighting with a monte carlo method and declarative modelling. In P. Sloot, A. Hoekstra, C. Tan, and J. Dongarra, editors, *Computational Science, ICCS 2002*, volume 2330 of *LNCS*, pages 3–12. Springer Berlin Heidelberg, 2002.

[20] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, Aug. 1986.

[21] S. W. Lee and R. Bajcsy. Detection of specularity using color and multiple views. In G. Sandini, editor, *Computer Vision ECCV'92*, volume 588 of *LNCS*, pages 99–114. Springer Berlin Heidelberg, 1992.

[22] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1647–1654, Sept. 2007.

[23] S. R. Marschner, S. H. Westin, E. P. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-based brdf measurement including human skin. In D. Lischinski and G. W. Larson, editors, *Rendering Techniques '99*, Eurographics, pages 131–144. Springer Vienna, 1999.

[24] M. McGuire and M. Mara. Efficient GPU screen-space ray tracing. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):73–85, December 2014.

[25] G. Patow and X. Pueyo. A survey of inverse rendering problems. volume 22, pages 663–687. Blackwell Publishing, Inc, 2003.

[26] L. B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *LNCS*. Springer, Berlin, 1981.

[27] S. Roth and M. Black. Specular flow and the recovery of surface structure. In *CVPR 2006*, volume 2, pages 1869–1876, 2006.

[28] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[29] A. Treuille, A. Hertzmann, and S. M. Seitz. Example-based stereo with general brdfs. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, volume 3022 of *LNCS*, pages 457–469. Springer Berlin Heidelberg, 2004.

[30] Y. Tsin, S. B. Kang, and R. Szeliski. Stereo matching with reflections and translucency. In *CVPR 2003.*, volume 1, pages 702–709, June 2003.

[31] S. Wanner and B. Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In J. Weickert, M. Hein, and B. Schiele, editors, *Pattern Recognition*, volume 8142 of *LNCS*, pages 1–10. Springer Berlin Heidelberg, 2013.

[32] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black. Lessons and insights from creating a synthetic optical flow benchmark. In A. Fusiello, V. Murino, and R. Cucchiara, editors, *Computer Vision , ECCV 2012. Workshops and Demonstrations*, volume 7584 of *LNCS*, pages 168–177. Springer Berlin Heidelberg, 2012.