Intrinsic Depth:
Improving Depth Transfer with Intrinsic Images

Naejin Kong and Michael J. Black
Max Planck Institute for Intelligent Systems
Spemannstrasse 41, 72076 Tübingen, Germany
{naejin.kong,black}@tuebingen.mpg.de

Abstract

We formulate the estimation of dense depth maps from video sequences as a problem of intrinsic image estimation. Our approach synergistically integrates the estimation of multiple intrinsic images including depth, albedo, shading, optical flow, and surface contours. We build upon an example-based framework for depth estimation that uses label transfer from a database of RGB and depth pairs. We combine this with a method that extracts consistent albedo and shading from video. In contrast to raw RGB values, albedo and shading provide a richer, more physical, foundation for depth transfer. Additionally we train a new contour detector to predict surface boundaries from albedo, shading, and pixel values and use this to improve the estimation of depth boundaries. We also integrate sparse structure from motion with our method to improve the metric accuracy of the estimated depth maps. We evaluate our Intrinsic Depth *method quantitatively by estimating depth from videos in the NYU RGB-D and SUN3D datasets. We find that combining the estimation of multiple intrinsic images improves depth estimation relative to the baseline method.*

1. Introduction

As laid out by Barrow and Tenenbaum [2] and elaborated over the years, intrinsic images correspond to physical properties of the scene such as depth, reflectance, shadows, optical flow, and surface shape. Barrow and Tenenbaum emphasize that the recovery of such intrinsic images is difficult and that the solution should recover them together, exploiting consistency between them. Here we take a step in that direction. Given a video sequence, which may contain camera motion and independently moving objects, we estimate the following intrinsic images at each frame: depth, albedo, shading, optical flow, and surface contours. As predicted by Barrow and Tenenbaum, we find that these different intrinsic images provide complimentary information and that estimating them in a synergistic way improves our estimation of scene structure. In doing so, we combine several lines of work including example-based depth estimation, sparse structure from motion, optical flow, contour detection, and reflectance and shading analysis. We refer to our method as *Intrinsic Depth* estimation (Fig. 1).

There have been recent successes in directly inferring the depth structure of images and video sequences from pixel values. In particular, our method builds on the framework of *Depth Transfer* [12], which is a non-parametric, data-driven, method for estimating scene depth using a database of images (or videos) and corresponding depth images. Given a new query image Depth Transfer has several steps. First it finds similar images in a database using gist matching [22]; the gist features are computed from image pixels and optical flow. It then uses label transfer [18] between the query image and the matched images to create a set of possible depth values for the scene. A final stage performs spatio-temporal regularization in an MRF formulation. Given sufficient training data, the method performs well at extracting plausible, dense, 3D surface structure. The output is neither metrically accurate nor faithful to the object boundaries in the scene. Here, however, we show that we can do better by integrating depth estimation with the extraction of other intrinsic images.

Gist features computed from pixel values may include confounding effects of illumination and reflectance. By mixing together reflectance, illumination, motion, and surface shape, pixel values obscure the physical processes that give rise to them. If the database contains very similar images (as it does in [12]) good matches will be found. A query image, however, may look very different due to different illumination and having a database that covers all reflectance and illumination conditions may be prohibitive to construct. Consequently we hypothesize that albedo and shading, instead of RGB values, provide a more physically motivated foundation for depth transfer. To that end, we use the *Intrinsic Video* method [13], which extracts temporally

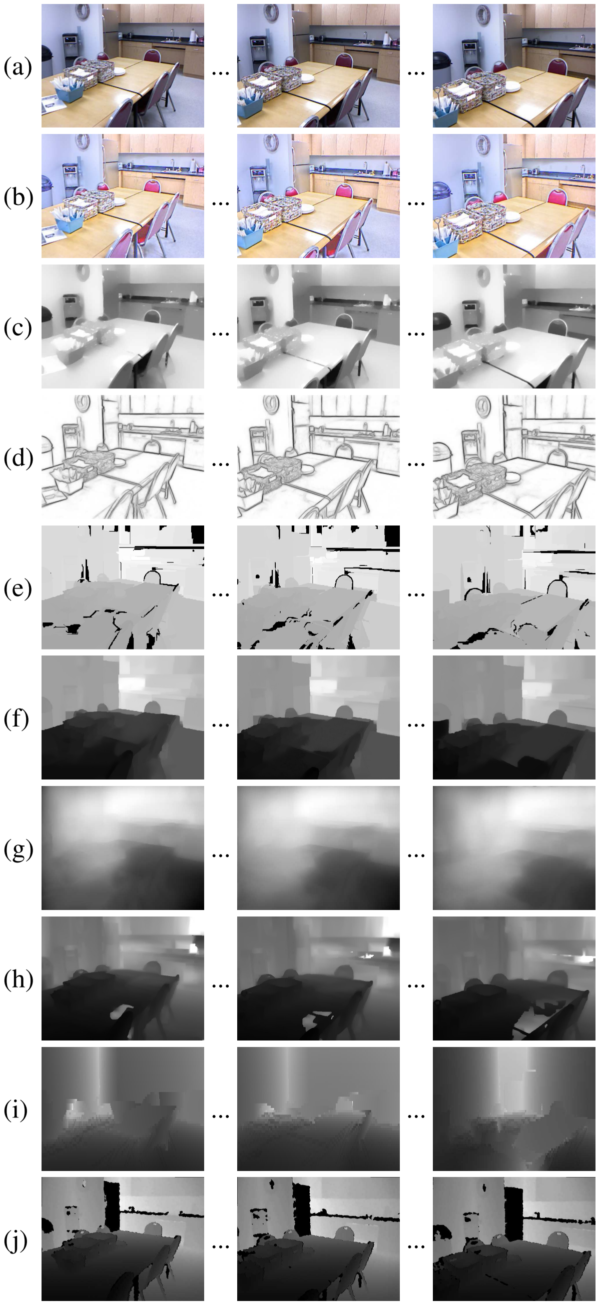


Figure 1. **Intrinsic Depth.** (a) Input video. (b),(c) Albedo and shading estimated by the intrinsic video method [13]. (d) Surface contours from [8] modified to combine RGB, albedo and shading information. (e) Proxy depth by propagating sparse SfM [28] depth using video segments from [9]. (f) Depth estimated by our method, which combines the previous two methods. (g) Depth from the original Depth Transfer method [12]. (h) Depth from the fully-metric method [32]. (i) Depth from the example-based single image method [24, 25]. (j) Ground truth depth. Note that integrating information from different intrinsic images improves the estimation of the depth structure. In (e) and (j), black pixels indicate that no valid depth values are provided.

coherent albedo and shading from a video sequence by exploiting optical flow (Fig. 1(b,c)). We then use the estimated albedo and shading to compute gist features separately on albedo, shading, RGB, and flow and use these features for generating candidate image matches.

We use albedo and shading in another way as well. Depth Transfer uses spatial regularization and ideally such smoothing should be disabled at surface boundaries. It is well known that edges in images are a poor proxy for surface boundaries because they combine surface markings with shape and illumination. Again we hypothesize that albedo and shading can provide important information to help disambiguate what are surface markings and what are object boundaries. In particular, surface boundaries in the depth map are likely to correspond to discontinuities in the shading images. However, shading edges are affected by illumination, thus simply relying on shading alone is insufficient. Consequently we train a new contour detector using RGB values, shading, and albedo to predict contours at surface boundaries. We use the decision forest method in [8] and train it on the synthetic 3D Sintel database [5] in which surface boundaries are known. We modify Sintel to create a training set with ground truth albedo and shading by simplifying the lighting conditions and making all surface materials Lambertian. We find that the resulting detector makes better predictions about surface boundaries (Fig. 1(d)) and we use these in regularizing our depth estimates.

Better scene matching and better surface contour detection improve depth estimation compared with Depth Transfer. We improve metric accuracy as well by integrating structure from motion estimation (SfM) [28] into the framework. SfM computes camera poses and sparse 3D points that are metrically accurate but that need to be densified to become an intrinsic "image." Many methods have been used for densification, but here we integrate sparse matches within our Intrinsic Depth framework. We first obtain semi-dense proxy depth maps by computing segmentation volumes from [9] and estimating the depth of each segment from the depth of the sparse 3D points projected into the image (Fig. 1(e)). We then use these proxy maps as priors in estimating our depth, replacing the use of average depth data in [12].

We find that these changes produce markedly more realistic depth maps with more precise depth boundaries and better metric accuracy (Fig. 1(f,g)). By combining Depth Transfer with intrinsic image decomposition, Intrinsic Depth makes a step towards an integrated treatment of intrinsic image extraction.

2. Previous Work

Depth estimation from image cues. The estimation of depth from a single image may use many well-studied cues such as texture gradients, atmospheric effects, vanishing points, etc. Progress has accelerated due to the recent availability of training data with depth sensors and corresponding color imagery.

One class of approaches learns a probabilistic model from training data and poses the estimation problem as inference. Saxena et al. [24, 25] predict depth from monocular image features using an MRF. A more efficient learning strategy for this approach is proposed in [3]. Performance improves by incorporating semantic labels [16] and even more by jointly inferring depth and other cues such as segmentation, scene category, saliency, etc. [15].

Example-based methods assume that appearance and depth are correlated. Hassner and Basri [10] combine known depth values of patches from similar objects to produce a plausible depth estimate of a query image of a single object. Konrad et al. [14] extend this idea to deal with the whole scene by simply fusing candidate depth maps. The spirit of the *Depth Transfer* method in [12] is similar, but it combines the candidate depth maps on a per-pixel basis using label transfer [18] by warping every pixel based on SIFT flow [18]. In addition, their method is not limited to single images, but rather exploits temporal information to obtain temporally coherent depth estimates. While Depth Transfer gives impressive results, the resulting depth maps are blurry and do not precisely correspond to the scene structure.

Most recently, Liu et al. [19] train a method to estimate depth from one image using a combination of a convolutional neural network (CNN) and a conditional random field. Their results look very natural and suggest that the CNN features are useful for this task. If perceptual quality is more desirable than metric accuracy, estimated depth can be transformed as in [7].

Structure from motion. There is a long history of work on structure from motion estimation (SfM). Very briefly, if the video involves a static scene with sufficient camera motion, current SfM methods work well (e.g. [21, 32]). While there are solutions for dealing with independently moving objects (e.g. [31]) this case remains a challenge. Karsch et al. [12] compare their method with [32] and demonstrate that, as expected, [32] works only for videos with sufficient parallax, while [12] produces results for any video regardless of the camera motion or object motion. The results of [12], however, are of much lower fidelity.

Intrinsic image estimation. The idea of extracting image-registered "intrinsic images" dates back to Barrow and Tenenbaum [2]. Recently this term has been taken to mean only "albedo" and "shading" but more generally includes the estimation of physically relevant properties such as depth, normals, optical flow, surface boundaries, etc.

Most recent work has focused on estimating albedo and shading from a single image. The most successful recent approaches require additional depth information, e.g. from an RGB-D sensor [1, 6, 11]. These methods essentially use depth to estimate shading and albedo while our method takes the opposite approach; that is, we start by estimating albedo and shading and then use this to estimate depth. Note that our method does not require an RGB-D sensor at test time, though we use RGB-D data for training as in other depth transfer approaches.

Recent work has addressed the problem of intrinsic image estimation in video sequences by exploiting temporal information to reduce the uncertainty of the problem. Kong et al. [13] exploit motion to extract temporally coherent albedo and shading. Ye et al. [30] use optical flow to propagate an initial albedo decomposition of the first frame over the video sequence. Bonneel et al. [4] separate image gradients into albedo and shading gradients based on scribbles provided by the user, and propagate the strokes to subsequent frames using optical flow. We used the method in [13] since this method is fully automatic and generates shading that is piecewise smooth while well capturing overall surface structure.

3. Formulation

Given a new query video, our goal is to estimate a dense depth map at every frame. We briefly summarize the original Depth Transfer method [12] and overview our modifications. While the original method can deal with both single images and videos, our method focuses only on videos with camera motion, possibly including moving objects. Therefore we only describe the video-based procedures here.

Overview. The system initially obtains similar looking video frames in the database by matching a set of gist descriptors of the query video to every video clip in a database. We find that better candidates are selected if each descriptor is further decomposed into *albedo gist* and *shading gist*.

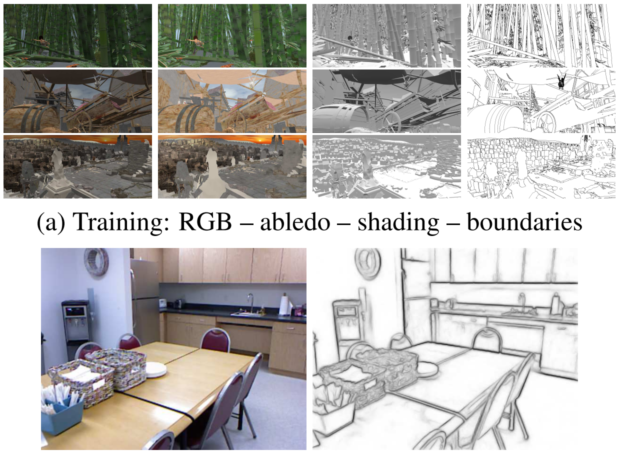
Next, the system warps the stored depth maps associated with the candidate frames onto each frame of the query video using SIFT flow [18].

The final step enhances the warped depth maps using image boundaries and optical flow. We replace image boundaries with surface contours predicted using pixel RGB, albedo and shading. In addition, we use sparse points and camera poses from structure from motion estimation [28] in regularizing the estimated depth.

3.1. Exacting intrinsic images

Intrinsic video for database and input. Our database is composed of RGB-D sequences and their corresponding estimated albedo, shading, and optical flow. We create this using time-varying raw sequences from the NYU RGB-D dataset[1], in which every clip is composed of a long image sequence of a moving camera, possibly including moving

[1] http://cs.nyu.edu/∼silberman/datasets/nyu_depth_v2.html



(a) Training: RGB – abledo – shading – boundaries

(b) Example contour detection

Figure 2. **Surface contours estimated from albedo and shading.** (a) A few frames from our contour training dataset: RGB, albedo, shading, and boundaries from left to right. (b) An RGB image and its surface contours predicted by our method modified from [8].

objects, and illumination variation. Note that this is different from the typical NYU RGB-D dataset [20], which is composed of single frames. For each video frame, we decompose it into albedo and shading

$$I_t(\mathbf{x}) = A_t(\mathbf{x}) \cdot S_t(\mathbf{x}), \tag{1}$$

where t is frame index, I_t is an RGB image, A_t is an albedo image, S_t is a shading image, and $\mathbf{x}$ is pixel position. Note, importantly, that we do not use the depth for estimating the albedo and shading. Our goal is to be able to extract intrinsic images, including depth, directly for video observations.

In order to extract temporally coherent albedo and shading from challenging RGB videos, we chose the intrinsic video method in [13], since this method does make any assumptions about the scenes if the videos have enough motion throughout the sequences; for example, they can include independently moving objects. The shading sequences from this method convey piecewise smooth structure, whose discontinuities overall align with the true shape of the scenes. We estimated optical flow from each of the sequences using the method of [17]. We tried other state-of-the-art flow algorithms [23, 26, 27], but this consistently performed the best on this database. We use the same methods to compute albedo and shading from a query video.

Surface contours. Shading provides a good cue about the location of surface boundaries, but shading boundaries are easily affected by illumination variation and thus not perfectly reliable. In [8] it is shown that surface contours can be predicted better by combining pixel values with extra information from known depth maps. We find that a similar approach works well by substituting the extra depth channel with albedo and shading. Specifically we retrain their decision forests on ground truth combinations of RGB, albedo, shading, and corresponding boundaries using the Sintel dataset [5]. See Fig. 2 and Section 6 in **Sup. Mat.**

Sparse depth and segmentation. We compute sparse SfM using VisualSFM[2], which implements multicore bundle adjustment [28]. We apply this to the test sequences to compute the depth at sparse points as well as camera poses. We then densify these as described in Section 1 of **Sup. Mat.** using segmentation volumes extracted by [9]. This provides semi-dense, metric, depth that acts as a prior and improves accuracy.

3.2. Modified Depth Transfer

We describe details of the modifications made to the original Depth Transfer method, then show and reason about the improvement over the original method.

Candidate frame selection. For each video sequence, the system computes a set of gist descriptors that are composed of the gist of each video frame (image gist), gist of each flow field (flow gist), and gist of the full video sequence (video gist). We further decompose the image and video gist using albedo and shading. According to the gist numbers, the system first chooses the 7 best matching videos and then the best matching frame from each of the videos.

The original matching score [12] between a frame in the query video q and a frame of a clip c in the database is defined as

$$w_i \|G(Iq) - G(Ic))\|^2 + w_f \|G(Fq) - G(Fc)\|^2, \tag{2}$$

where w_i and w_f are blending weights ($w_i = w_f = \frac{1}{2}$), and G is a gist operator [22], Iq is a query video frame whose optical flow field is Fq, Ic is a video frame to compare with, whose flow field is Fc. Our matching score is modified as

$$\begin{aligned} & w_a \|G(Aq) - G(Ac)\|^2 + w_s \|G(Sq) - G(Sc)\|^2 \\ & + w_i \|G(Iq) - G(Ic))\|^2 + w_f \|G(Fq) - G(Fc)\|^2, \end{aligned} \tag{3}$$

where w_a, w_s, w_i and w_f are blending weights given as $w_a = w_s = w_i = w_f = \frac{1}{4}$, Aq and Sq are albedo and shading of a query video frame, respectively, and Ac and Sc are those of a frame to compare with.

The video gist is defined as the gist of a median image over all video frames. We further define the albedo gist and the shading video gist as the gist of a median albedo and that of a median shading image over the video, respectively. For video clip selection, we replace the original video gist with a blending of the video gist, albedo video gist, and shading video gist with even factors.

Figure 3 shows that our modified candidate selection performs better in that it chooses more similar looking frames.

[2] http://ccwu.me/vsfm/

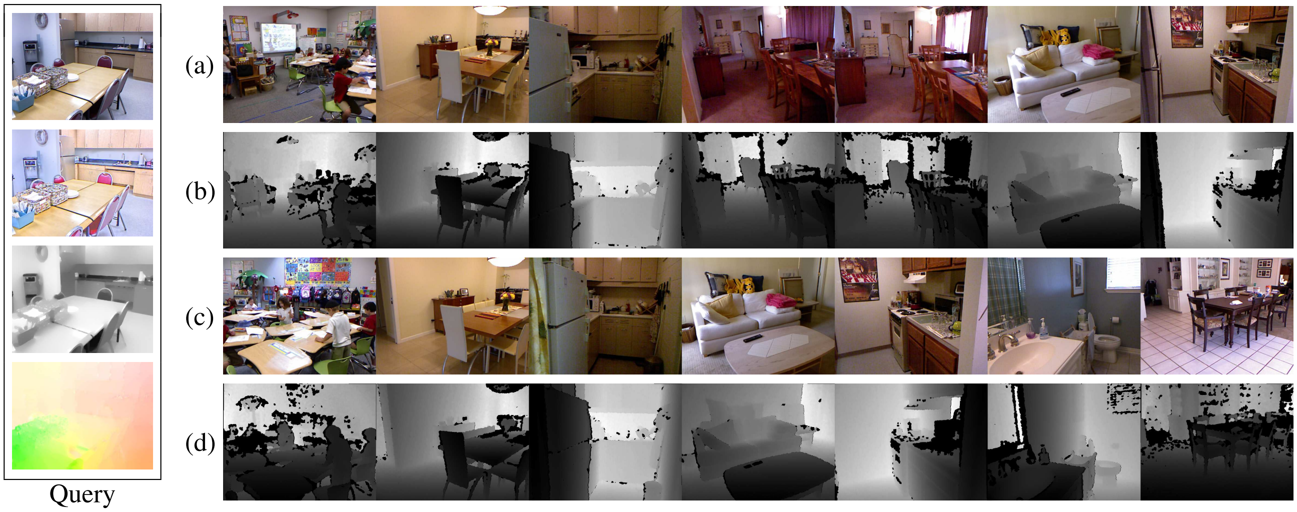
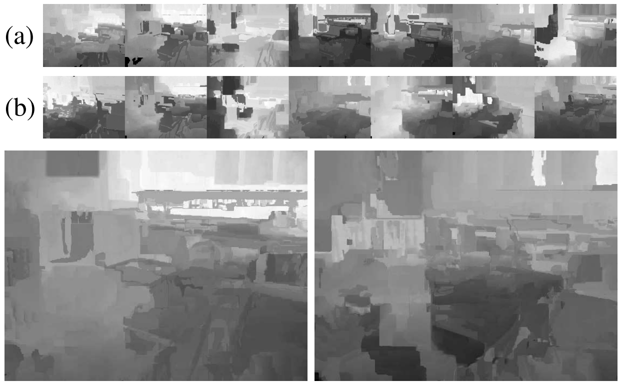


Figure 3. Candidate frame selection for a frame of the video in Fig. 1. The system chooses 7 candidate frames from the database. (a) shows candidates selected by pixel values and flow (original method) and (b) shows the corresponding depth maps. (c) shows candidates selected by pixels, albedo, shading, and flow, and (d) shows the corresponding depth maps. In (a) and (c), the images are sorted according to their matching scores in a descending order; the leftmost image is the best match for the query video (black pixels are unreliable measurements).

Image gist descriptors extracted from pixel values can give implausible matches if the scene of the query video looks very different from any of the training video clips. If two clips capture the same scene but the illumination is different, the image-based gist can get fooled and, in this case, albedo gist may perform better. If two clips are from two different scenes, but their color distributions are somewhat similar, then only the shape difference gives us a cue to choose the right one. In this case, shading gist may perform better. Thus shading and albedo gist compliment each other.

Warping candidate depth. Matching using SIFT flow [18] is a key component of Depth Transfer, which performs per-pixel warping between pixels with similar appearance. As in [12], we fill holes in the candidate depth map using spatio-temporal interpolation, and warp it to the query video frame using the SIFT flow. Our SIFT flow is computed using the albedo of the query image and that of the candidate frame instead of RGB values. Figure 4 compares warped depth maps from the same query, where (a) is from candidates using pixels and flow, and (b) is from candidates using extra albedo and shading information. We can see that our fused depth conveys more structural information.



(c) Baseline: fusing (a) (d) Ours: fusing (b)

(e)

case	rel	log10	RMS
Fused depth baseline	1.243	0.261	2.312
Fused depth ours	0.960	0.228	1.905

Figure 4. Warped depth maps of the candidates shown in Fig. 3. Known depth maps of 7 candidates are warped onto a query video frame using SIFT flow. (a),(b) Candidates in Fig. 3(b) and (d) warped to the query frame, respectively. (c),(d) Median of (a) and (b) over the candidates, respectively. Our fused map in (d) better captures the overall shape of the table compared to that in (c). (e) Errors of the fused depth videos (30 frames) compared with ground truth (see Section 4 for the error measures).

Regularization. The warping process in the previous step considers neither spatial smoothness nor temporal coherence in the warped depth values, thus the warped depth is inconsistent and noisy. The final step is very important to enhance consistency in the warped depth values. The original method performs spatio-temporal regularization on the intensity and gradients of warped depth values based on image boundaries and optical flow. It minimizes

$$\underset{D_t}{\operatorname{argmin}} \sum_t \quad E_{\text{data}}(D_t, C_t^{(1\ldots K)}) + \gamma E_{\text{prior}}(D_t, \mathcal{P}_t) + \alpha E_{\text{spat}}(D_t) + \beta E_{\text{temp}}(D_t, D_{t+1}, \mathbf{u}_t), \tag{4}$$

where D_t is an unknown depth map of the query frame at t that we wish to estimate. E_{data} is the data term that takes

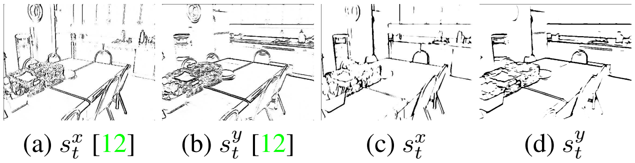

(a) s_t^x [12] (b) s_t^y [12] (c) s_t^x (d) s_t^y

Figure 5. **Spatial weight functions.** (a),(b) Original functions along with horizontal and vertical gradients of pixel values, respectively. (c),(d) Ours from the predicted contours in Fig. 2.

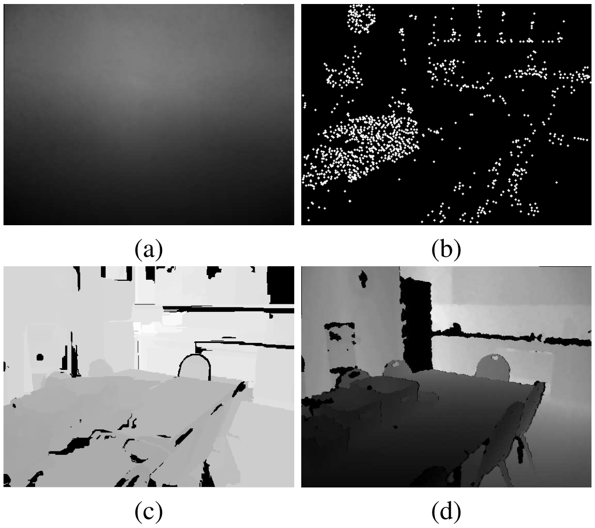

(a) (b) (c) (d)

Figure 6. **Prior.** (a) Depth prior as an average of all depth maps in the training data. (b) SfM points projected into the image (here simply visualizing projected pixel locations). (c) Out proxy depth map using the sparse depth in (b) and segmentation from [9]. (d) Ground truth. The original method simply replicates the same (a) throughout the video. Our prior is dense and metrically more accurate, and reflects depth variation over time.

all K candidate depth maps $C_t^{(1\dots K)}$ for the query frame at t. E_{prior} is a soft constraint to guide the estimation using a prior depth map, $\mathcal{P}_t$, at t. E_{spat} is a spatial smoothness term that uses image boundaries. E_{temp} is a temporal coherence term that uses optical flow, $\mathbf{u}_{t,t+1}$, between t and $t+1$. We modify E_{prior}, E_{spat}, and E_{temp} as discussed below. The default settings of the weights are $\alpha = 10$, $\beta = 100$, and $\gamma = 0.5$, while we use $\alpha = 100$, $\beta = 100$, and $\gamma = 10$. We define a sigmoid function here that is used below

$$sig(x, \nu, \mu) = \left(1 + e^{\nu\cdot(\mu - x)}\right)^{-1}, \tag{5}$$

where ν and μ are constants that shape of the soft threshold. See **Sup. Mat.** for the original forms of the terms above.

Data term. We keep this term the same as in [12]. This term measures how close the inferred depth map D_t is to each of the warped candidate depth maps. The weight is fixed to 1 relative to other weights α, β, and γ in Eq. (4).

Spatial smoothness. Our spatial term is defined as

$$E_{\text{spat}}(D_t) = \sum_{\mathbf{x}} s_t^x(\mathbf{x})\rho(\nabla_x D_t(\mathbf{x})) + s_t^y(\mathbf{x})\rho(\nabla_y D_t(\mathbf{x})),$$

where $\nabla_x D_t$ and $\nabla_y D_t$ are horizontal and vertical depth gradients, respectively, $\rho(x) = \sqrt{x^2 + \epsilon^2}$, and $\epsilon = 0.01$. The weighting functions s_t^x and s_t^y control the smoothness of the estimated depth map. It allows higher smoothing influence where contours do not arise in the image, so that the discontinuities are kept where the contours arise. In the original method these spatial weights are determined by image boundaries, but they may come from surface markings rather than surface boundaries.

In order to predict contours that better obey true surface boundaries, we modify a contour detector in [8] so as to combine physical and structural information from albedo and shading. We find that our contours better correspond to surface boundaries than those from [8], thus improve the fine quality of the estimated depth maps. See Sections 6 and 7 in **Sup. Mat.** for more details. We define new spatial weights that encourage depth discontinuities along the relevant surface boundaries by extracting vertical and horizontal contours from the raw contour map δ_t as

$$\begin{aligned} s_t^x(\mathbf{x}) &= 1 - sig(G_v(sig(\delta_t(\mathbf{x}), 50, 0.3), \sigma), 50, 0.3) \\ s_t^y(\mathbf{x}) &= 1 - sig(G_h(sig(\delta_t(\mathbf{x}), 50, 0.3), \sigma), 50, 0.3), \end{aligned} \tag{6}$$

where G_v and G_h are 1D vertical and horizontal Gaussian filters ($\sigma = 2$), respectively. Figure 5 illustrates the weight functions.

Prior. Our prior term minimizes the difference between the estimated depth map and the prior depth map $\mathcal{P}_t$. The original method simply replicates an average depth map over the database over time, while we compute more accurate and temporally varying proxy depth maps using sparse points from SfM estimation.

$$E_{\text{prior}}(D_t, \mathcal{P}_t) = \sum_{\mathbf{x}} s_t^{\text{pxy}}(\mathbf{x}) \cdot \rho(a \cdot D_t(\mathbf{x}) - \mathcal{P}_t(\mathbf{x})), \tag{7}$$

where $\mathcal{P}_t$ is our proxy depth map and s_t^{pxy} is a binary mask that is set to 1 if $\mathcal{P}_t$ is valid at that pixel, 0 otherwise. $\rho(x)$ is the same as above and a is an unknown scale variable (see below). Since sparse points only guarantee their accuracy at a few projected pixels, they do not provide a sufficient prior. Simple extrapolation provided unsatisfactory results when the points were not well spread over the image. Instead we find that a recent video segmentation method [9] provides good over-segmentation volumes to densify these sparse depth values reasonably. See Fig. 6 and our proxy map (c), which provides a crude approximation to the solution.

Temporal coherence. This term encourages temporal coherence of the estimated depth maps by using the optical flow and the camera motion of the query video. In the original method, depth is considered to be strictly coherent over the correspondences. This assumption is violated when the camera moves and can be particularly bad with large motions. Thus we incorporate camera poses estimated from SfM into this term.

We define our temporal term as

$$
\begin{aligned}
&E_{\text{temp}}(D_t, D_{t+1}, \mathbf{u}_t) = \sum_{\mathbf{x}} s_t^{\text{temp}}(\mathbf{x}) \cdot \\
&\rho(S_{t+1}(\mathbf{x} + \mathbf{u}_t(\mathbf{x}), R_{t+1}, \theta) \cdot D_{t+1}(\mathbf{x} + \mathbf{u}_t(\mathbf{x})) \cdot a \\
&\quad - S_t(\mathbf{x}, R_t, \theta) \cdot D_t(\mathbf{x}) \cdot a + o_t(c_t, c_{t+1})), \qquad (8)
\end{aligned}
$$

where $\mathbf{u}_t$ is optical flow from t to $t+1$, and a is an unknown global scale factor to compensate for the scale ambiguity of the SfM output. $s_t^{\text{temp}}(\mathbf{x})$ is a weight function, measured by the following flow confidence to down weight occluded and dis-occluded pixels

$$
s_t^{\text{temp}}(\mathbf{x}) = 1 - sig\left(\left|G(J_t(\mathbf{x}), \sigma)\right|, 1000, 0.005\right), \qquad (9)
$$

where $J_t(\mathbf{x}) = \; I_{t+1}(\mathbf{x} + \mathbf{u}_t(\mathbf{x})) - I_t(\mathbf{x})$, G is a Gaussian filter ($\sigma = 1$), and I_t and I_{t+1} are the query frames at t and $t+1$, respectively. $\rho(x)$ is the same as above.

The motivation of Eq. (8) is that two corresponding pixels, $\mathbf{x}$ at t and $\mathbf{x} + \mathbf{u}_t(\mathbf{x})$ at $t+1$, should project to the same 3D position using S_t, S_{t+1} and o_t derived from camera poses at t and $t+1$:

$$
S_t(\mathbf{x}, R_t, \theta) = R_{(3,:)t} \cdot \begin{pmatrix} (x - p_x)/f_x \\ (y - p_y)/f_y \\ 1 \end{pmatrix} \qquad (10)
$$

$$
o_t(c_t, c_{t+1}) = c_{(3)t+1} - c_{(3)t}, \qquad (11)
$$

where $\mathbf{x} \;=\; (x, y)$, $[R_t|c_t]$ is an inverse extrinsic matrix (from camera to world) at t for which R_t is a 3x3 rotation matrix and c_t is a 3D camera position, and $\theta = (f_x, f_y, p_x, p_y)$ represents the intrinsic parameters for which f_x and f_y are focal lengths and (p_x, p_y) is a principal point. Subscripts $(3, :)$ and (3) indicate the third row of the matrix and the third component of the vector, respectively.

See Section 4 in **Sup. Mat.** for optimization details.

4. Experiments

We tested our method on static scenes captured with a significantly moving camera and non-rigid scenes with camera motion. Please watch our full supplementary video on the project homepage[3]. Qualitatively our method produces temporally coherent dense depth maps preserving strong surface boundaries that are metrically accurate. From the raw NYU RGB-D data, we take 223 sequences (scenes) corresponding to 31 semantically different indoor environments. We split each sequence into up to 4 non-overlapping sub-sequences (clips), each 30 frames long. We evaluate on these clips but use the full sequences for SfM.

We compare our results with those from the original Depth Transfer method [12] and the fully-metric method [32] that only relies on SfM and multi-view stereo, and the single image method [24, 25] (http://make3d.cs.cornell.edu). We adopt error measures from [12], including a relative (**rel**) error $\frac{|D - D^*|}{D^*}$, $\log_{10}$ (**log10**) error $|\log_{10}(D) - \log_{10}(D^*)|$, and root mean squared (**RMS**) error $\sqrt{\sum_{i=1}^{N}(D_i - D_i^*)^2/N}$, where D and D^* are estimated and ground truth depth maps, respectively, i is pixel index, N is the number of pixels in an image. All estimation is processed at the native resolution.

We measure errors after normalizing the estimated depth video (0-truncated negative values if any) and ground truth depth video separately; each video is scaled so that its minimum and maximum values, over all frames, stay within $[0.1, 10]$ meters (roughly the NYU RGB-D depth range). Note that we exclude the known invalid regions in the ground truth depth when normalizing it and when computing errors.

Test Cases from NYU RGB-D. These test cases are chosen from our database derived from raw NYU RGB-D data. We randomly take 10 test scenes (30 frames for each scene; resolution 561×427) at each time while leaving out the rest as a training set, and repeat this 5 times to measure average errors. Here we use VisualSFM [28] (http://ccwu.me/vsfm/) to estimate SfM cues. Figure 1 is chosen from these test cases. More examples are in **Sup. Mat.**

Table 1 shows that our depth estimated without SfM cues is already better than that from [12], and also better than depth from ours with contours from [8]. Table 2 shows that our warping step using albedo gist and shading gist works better than only using RGB gist. Table 3 shows that our depth estimates are more accurate than sparse depth from SfM at those points where SfM estimates are available.

Test Cases from SUN3D. We randomly take 50 scenes from the SUN3D dataset [29], which is composed of RGB-D videos and pre-computed SfM data. Each scene is compose of a 30-frame clip whose resolution is 640×480. Note that the SfM method to generate this database uses measured depth maps thus projected depth values from its SfM points are very accurate. Figures 7-12 in **Sup. Mat.** illustrate representative examples from these test cases.

In Table 4, we show that our method without SfM is consistently better than the baseline, while our full method, with SfM cues, is significantly better. Our full method also performs better than ours with contours from [8]. Table 5 shows that our warping step works better than the original.

Quantitative evaluation against [32] on the subsets of both test cases above is addressed in Section 5 of **Sup. Mat.**; [32] performs best on outdoor scenes but we found that it works poorly on indoor scenes for which ours does best.

Non-rigid Scenes with Camera Motion. We show that our

[3] https://ps.is.tue.mpg.de/research_projects/intrinsic-depth

method works reasonably on non-rigid scenes with camera motion, where standard SfM methods have trouble. We take 6 such scenes (17 clips, 30 frames each) from the raw NYU RGB-D data, and evaluate them in Table 6. One of those is presented in Fig. 7. See the caption for descriptions.

method	rel	log10	RMS
Baseline [12]	1.820	0.309	3.107
Ours w/o SfM	1.468	0.287	2.887
Ours w/ contours from [8]	0.720	0.235	2.102
Our full method	0.718	0.232	2.098

Table 1. Estimated depth maps for NYU RGB-D test cases.

method	rel	log10	RMS
Warped depth of [12]	1.414	0.271	2.384
Our warped depth	1.311	0.271	2.367

Table 2. Warped depth for NYU-D test cases.

method	rel	log10	RMS
SfM depth	1.028	0.252	2.068
Our depth at SfM points	0.685	0.181	1.581

Table 3. SfM cues for NYU RGB-D test cases. We evaluate at the points with SfM estimates.

method	rel	log10	RMS
Baseline [12]	2.003	0.333	3.593
Ours w/o SfM	1.406	0.291	3.071
Ours w/ contours from [8]	0.426	0.125	1.264
Our full method	0.398	0.119	1.215

Table 4. Estimated depth for SUN3D test cases.

method	rel	log10	RMS
Warped depth of [12]	1.315	0.272	2.605
Our warped depth	1.207	0.270	2.555

Table 5. Warped depth for SUN3D test cases.

method	rel	log10	RMS
Baseline [12]	1.496	0.250	2.444
Baseline [12]*	1.830	0.316	2.853
Baseline [12]†	1.849	0.302	2.890
Fully-metric method [32]	1.517	1.139	4.788
Our full method	0.875	0.244	2.208
SfM depth	1.025	0.269	1.951
Our depth at SfM points	0.766	0.216	1.666

Table 6. Estimated depth for non-rigid scenes with moving cameras. *: with basic motion segmentation. †: with motion segmentation using homographies. In the last two rows, we evaluate at the points with SfM estimates.

5. Conclusions and Future Work

We have demonstrated how the computation of several intrinsic images (depth, shading, albedo, flow, and contours) can work together synergistically. Shading and albedo can help the estimation of example-based depth estimation. Combining RGB, shading, and albedo can produce better surface contour detection. Flow helps link information in time providing consistency of albedo, shading, depth, and contours. Together these insights allow us to improve on Depth Transfer [12]. Additionally we show how integrating sparse SfM with an example-based depth method improves metric accuracy and how it can be seen as a form of densification. We demonstrate this visually and quantitatively on the NYU RGB-D and SUN3D datasets.

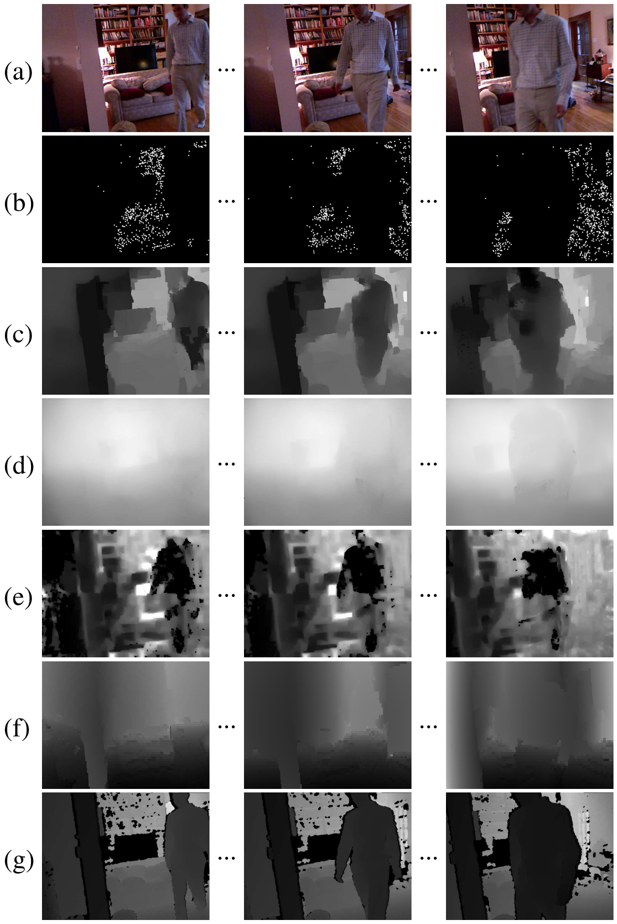


Figure 7. Living room scene with a walking person and camera motion. (a) Input RGB sequence. (b) SfM point projections. Note that SfM points are missing at most regions. (c) Our method estimates depth overall well regardless of missing SfM cues. (d) Depth from [12] with motion segmentation heuristics using homographies. (e) [32] has trouble due to unmodeled non-rigidity. (f) [24, 25] fails to capture the moving person and yields inconsistent depth across time. (g) Ground truth depth.

We see this a modest step towards a more integrated treatment of intrinsic images as laid out by Barrow and Tenenbaum. In particular, depth, surface normals, and shading are tightly coupled to image appearance, and a more integrated optimization of these all together should yield finer surface details and increased robustness.

Acknowledgments. We thank J. Anning for technical support and voice recording, J. Wulff for his help regarding the Sintel dataset, and A. O. Ulusoy for useful advice.

References

[1] J. T. Barron and J. Malik. Intrinsic scene properties from a single RGB-D image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24, 2013. 3

[2] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages 3–26, 1978. 1, 3

[3] D. Batra and A. Saxena. Learning the right model: Efficient max-margin learning in laplacian CRFs. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2136–2143, 2012. 3

[4] N. Bonneel, D. Sun, K. Sunkavalli, S. Paris, and H. Pfister. Reflectance and illumination video editing using fast user-guided intrinsic decomposition. Technical Report TR-02-14, Harvard University, February 2014. 3

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. European Conference on Computer Vision (ECCV)*, volume 7577 of *Part IV. LNCS*, pages 611–625, 2012. 2, 4

[6] Q. Chen and V. Koltun. A simple model for intrinsic image decomposition with depth cues. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 241–248, 2013. 3

[7] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, and H.-P. Seidel. A perceptual model for disparity. *ACM Trans. Graph.*, 30(4), 2011. 3

[8] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1558–1570, 2015. 2, 4, 6, 7, 8

[9] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2141–2148, 2010. 2, 4, 6

[10] T. Hassner and R. Basri. Example based 3D reconstruction from single 2D images. In *Beyond Patches*, page 15, 2006. 3

[11] J. Jeon, S. Cho, X. Tong, and S. Lee. Intrinsic image decomposition using structure-texture separation and surface normals. In *Proc. European Conference on Computer Vision (ECCV)*, volume 8695 of *LNCS*, pages 218–233, 2014. 3

[12] K. Karsch, C. Liu, and S. B. Kang. Depthtransfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2144–2158, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[13] N. Kong, P. V. Gehler, and M. J. Black. Intrinsic video. In *Proc. European Conference on Computer Vision (ECCV)*, volume 8690 of *LNCS*, pages 360–375, Sept. 2014. 1, 2, 3, 4

[14] J. Konrad, M. Wang, and P. Ishwar. 2D-to-3D image conversion by learning depth from examples. In *CVPR Workshops*, pages 16–22, 2012. 3

[15] C. Li, A. Kowdle, A. Saxena, and T. Chen. Towards holistic scene understanding: Feedback enabled cascaded classification models. *CoRR*, abs/1110.5102, 2011. 3

[16] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1253–1260, 2010. 3

[17] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *Ph.D. dissertation, MIT*, 2009. 4

[18] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011. 1, 3, 5

[19] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[20] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. European Conference on Computer Vision (ECCV)*, volume 7576 of *Part V. LNCS*, pages 746–760, 2012. 4

[21] R. Newcombe, S. Lovegrove, and A. Davison. DTAM: Dense tracking and mapping in real-time. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011. 3

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.*, 42(3):145–175, May 2001. 1, 4

[23] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4

[24] A. Saxena, S. H. Chung, and A. Y. Ng. 3-D depth reconstruction from a single still image. *Int. J. Comput. Vis.*, 76(1):53–69, Jan. 2008. 2, 3, 7, 8

[25] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, May 2009. 2, 3, 7, 8

[26] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vis.*, 106(2):115–137, 2014. 4

[27] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1385–1392, 2013. 4

[28] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3057–3064, 2011. 2, 3, 4, 7

[29] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using sfM and object labels. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1625–1632, 2013. 7

[30] G. Ye, E. Garces, Y. liu, Q. Dai, and D. Gutierrez. Intrinsic video and applications. *ACM Trans. Graph.*, 33(4), 2014. 3

[31] G. Zhang, J. Jia, W. Hua, and H. Bao. Robust bilayer segmentation and motion/depth estimation with a handheld camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):603–617, 2011. 3

[32] G. F. Zhang, J. Y. Jia, T. T. Wong, and H. J. Bao. Consistent depth maps recovery from a video sequence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):974–988, June 2009. 2, 3, 7, 8