Local detection of stereo occlusion boundaries

Jialiang Wang and Todd Zickler
Harvard University
jialiangwang@g.harvard.edu, zickler@seas.harvard.edu

Abstract

Stereo occlusion boundaries are one-dimensional structures in the visual field that separate foreground regions of a scene that are visible to both eyes (binocular regions) from background regions of a scene that are visible to only one eye (monocular regions). Stereo occlusion boundaries often coincide with object boundaries, and localizing them is useful for tasks like grasping, manipulation, and navigation. This paper describes the local signatures for stereo occlusion boundaries that exist in a stereo cost volume, and it introduces a local detector for them based on a simple feedforward network with relatively small receptive fields. The local detector produces better boundaries than many other stereo methods, even without incorporating explicit stereo matching, top-down contextual cues, or single-image boundary cues based on texture and intensity.

1. Introduction

Precisely localizing object boundaries is important for grasping, navigation, and other visual tasks. In a stereo image pair, many object boundaries show up as stereo occlusion boundaries, which are 1D curves in the visual field that separate parts of the scene that are visible to both eyes (*binocular regions*) from those that are visible to only one eye (*monocular regions*, also called *half-occluded regions*).

Despite the importance of localizing stereo occlusion boundaries, the performance of modern stereo algorithms at these boundaries and their adjacent monocular regions is relatively poor. Figure 1 shows a breakdown of errors of the best-performing stereo algorithm on the Middlebury 2014 benchmark [24] for each of the past five years, with errors measured by percentage of "bad pixels" (disparity error ≥ 2) in the benchmark's "evaluation training dense set" using the provided disparity and occlusion maps. While there has been almost a three-fold decrease of error in binocular regions (20% to 7%), the error rate in monocular regions has decreased only slightly (66% to 52%) and remains much higher than the 10.5% that we conservatively estimate as the achievable error rate in monocular regions[1].

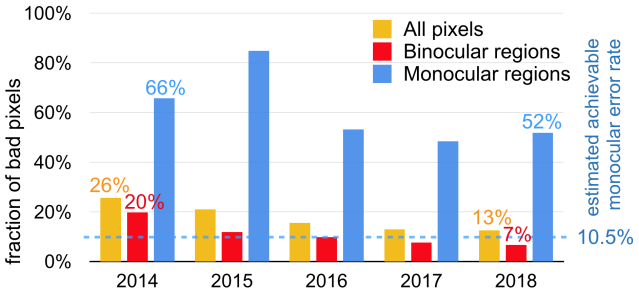

Figure 1. Motivation for improving the localization of stereo occlusion boundaries: Breakdown of errors for the best-performing algorithms on Middlebury 2014 stereo benchmark [24] during the past five years. Errors adjacent to stereo occlusion boundaries—in monocular/half-occluded regions—have not been substantially reduced, and they remain much higher than our estimate (10.5%) of the achievable error rate in those regions.

Stereo algorithms typically try to localize stereo occlusion boundaries by relying heavily on their co-occurrence with texture or intensity boundaries, or by using deep networks with large receptive fields that can incorporate top-down contextual cues by internalizing the non-local disparity patterns that occur in a given dataset. These approaches quickly break down in situations like Figures 2 and 8, where texture and intensity boundaries are absent, or where the input is very different from the dataset used for training.

We propose a different, direct approach to localizing stereo occlusion boundaries. We approach it as a local detection task, where each local 3D region of a stereo cost volume is independently classified as containing a stereo occlusion boundary or not. As we show, this is possible because there are local signatures near stereo occlusion boundaries in a stereo cost volume, even in cases like Figure 2 where texture and intensity boundaries are absent.

[1]Using the benchmark's disparity and occlusion maps, we simulate a vision system that achieves perfect matching in binocular regions and infers perfect occlusion maps, and then we calculate the disparity error in the monocular regions that results from naive, constant-disparity extrapolation from the true disparity of the background binocular region immediately adjacent to each monocular region.

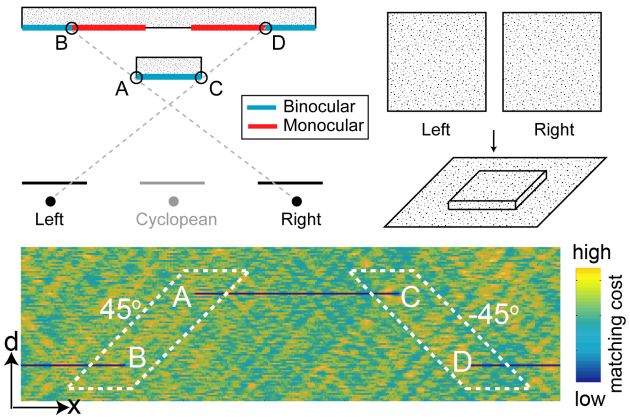

Figure 2. Stereo occlusion boundaries versus intensity or texture boundaries. There are no intensity or texture boundaries in a random-dot stereogram, but stereo occlusion boundaries exist and are locally-detectable in the stereo cost volume. In an epipolar slice of the cyclopean cost volume (*i.e.*, disparity space image) shown at bottom, points A&C are stereo occlusion boundaries and points B&D are associated monocular region boundaries on the background. They all separate low-cost matched regions from high-cost unmatched regions. With proper rectification [5], the foreground-background pairs (A/B, C/D) always lie on $\pm 45^o$ lines. White dashed boxes show detectable local signatures.

Our work is motivated by Anderson and Nakayama's long-standing hypothesis that such local detectors exist in the visual cortex [2], and by the substantial evidence from vision science that humans can accurately infer stereo occlusion boundaries even when binocular matching cues are absent or very weak [15, 2, 1, 30, 14, 16, 29].

We begin by introducing a taxonomy of stereo occlusion boundaries characterized by the local signatures that they induce in a stereo cost volume. Based on this, we design a detector using a multiscale feedforward network with receptive fields that are small enough to be able to localize boundaries around thin foreground structures. In order to understand what can be achieved using the detector alone, we intentionally exclude the direct use of single-image texture and intensity boundaries, and we train our detector network using simple synthetic data of piecewise planar scenes. Despite these restrictions, we find that the detector provides better boundaries than many other stereo algorithms, and that it succeeds for a variety of scene types, including Middlebury images [24], Sintel images [8], and many perceptual stimuli proposed by vision scientists.

2. Related Work

Our work is distinct from, but related to, methods for detecting boundaries using the local intensity and texture cues in a single 2D image (*e.g.*, [3, 22]). We draw particular inspiration from Xie and Tu [34], who train a multiscale feedforward network to detect these boundaries. The critical difference is that we operate on a 3D cost volume, which allows detecting object boundaries even when single-image texture and intensity cues are completely absent (Figure 2).

Our work is different from most existing approaches to find stereo occlusion boundaries, which are not purely local and instead rely on some sort of global reasoning. The most common approach is to find stereo occlusion boundaries through secondary processing, after an initial disparity map has been formed by committing to a single disparity value at each spatial location. A popular example is bidirectional verification — so-called "left-right consistency" in stereo [33, 13, 27] and "forward-backward consistency" in optical flow [17, 19]—that computes two separate disparity maps (or dense flow fields) from the two viewpoints and then reasons about occlusions based on their inconsistency. Another common approach is to infer occlusions and binocular-region disparities at the same time, by optimizing a global, spatially-regularized energy that includes a binary occlusion variable for each pixel [7, 6, 32, 38] or that incorporates occlusion constraints into binocular matching [20].

A notable exception that detects stereo occlusion boundaries locally is the work of Wang *et al*. [31], who use a spatial gradient operator in each epipolar slice of the cost volume. Our work can be viewed as an extension that applies to full 3D cost volumes and that uses more sophisticated nonlinear filters. Another exception is the optical flow work by Stein and Hebert [25], who first detect intensity and texture boundaries and then apply tests in their local neighborhoods to determine which ones are also occlusion boundaries. Our work is different because it succeeds even when no texture and intensity boundaries are present. Sundberg *et al*. [28] also explore local detection in optical flow, and they present a score based on three frames (instead of two) that fires at both object boundaries and intensity/texture boundaries.

Our local detector takes as input a multiscale stereo cost volume, meaning a four-dimensional data structure $C(x, y, d, s)$ storing the stereo matching cost associated with disparity d at spatial location (x, y), as computed using (square) window size s. We choose to parameterize the cost volume in the rectified cyclopean coordinate system [5] (Figure 2), from which it can be sheared to the left and right views as needed for efficient processing (see Section 4.1).

We can leverage recent data-driven approaches that improve the quality of this cost volume. In particular, we use the fast Siamese architecture of Zbontar and LeCun [37] but a multiscale variant, which reflects a long history in stereo [21, 36] and relates to multiscale networks for other tasks [26, 10]. We find that including small windows ($s = 5$) in the cost volume allows finding precise boundaries, even when the foreground structures are very thin.

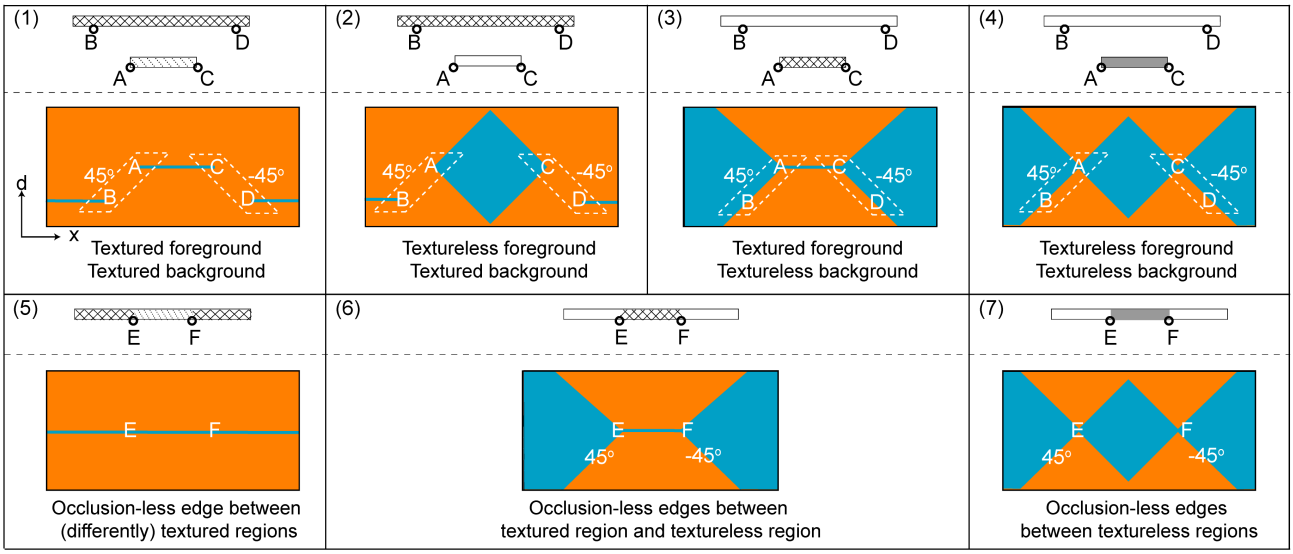

Figure 3. Seven cyclopean disparity space signatures, each for the scene shown above it. The idealized disparity space images use blue for "low-cost" regions and orange for "high-cost" regions. Scenarios (1–4) have distinctive local signatures at left and right occluding points (A and C) and at the associated background points (B and D). For comparison, scenarios (5–7) are planar textured scenes that have occlusion-less edges (E and F). Scenarios (6 & 7) produce signatures that are locally indistinguishable from their occlusion counterparts (3 & 4, respectively), so any local detector that fires at one will also fire at the other.

3. Taxonomy of Stereo Occlusion Boundaries

Stereo occlusion boundaries are detectable because of the local signatures they induce in the cost volume. The signatures will vary according to the size of the depth discontinuity and the textures that exist on the adjacent foreground and background surfaces, and a good detector should succeed in spite of these variations. This section describes four basic categories of occlusion boundary signatures ((1–4) in Figure 3) and provides a foundation for the design of our detector in Section 4.

One important fact we discover is that some types of occlusion boundary signatures cannot be distinguished from those of certain occlusion-less texture boundaries (*e.g.* (3) vs. (6) in Figure 3), implying that any detector of one will also detect the other. Below we discuss how this unavoidable "confusion" relates to human perception [29], and argue that it is just as much a feature as a bug since the detections occur at the correct location and depth in both cases.

For clarity, we present the taxonomy using 2D (x, d) epipolar slices of a cost volume, with targets of interest being single stereo occlusion boundary points. In reality, the boundary points will chain together across epipolar slices, forming detectable 1D structures at various locations and orientations within the cost volume. These are the structures that we actually detect in subsequent sections.

Consider a pair of rectified stereo cameras and a virtual cyclopean camera all with equal focal lengths and parallel optical axes (top-left of Figure 2). Let x, y index the two spatial dimensions of the cyclopean image plane along the epipolar direction x and its perpendicular y. Suppose a cyclopean cost volume $C(x, y, d, s)$ is tabulated by measuring the matching cost, using some cost function, between a left image patch of size s centered at left pixel $(x + d, y)$ and a right image patch at right pixel $(x - d, y)$. The bottom of Figure 2 shows an example of a 2D epipolar (x, d) slice of a cost volume for a particular choice of cost function and patch size. This figure also shows the four critical points related to stereo occlusions [2, 31]: the left and right stereo occlusion boundary points on the foreground A,C, and the associated background points B,D that are defined by the right-camera and left-camera occluding rays through A and C. These four points exist both in scene space (figure top) and in disparity space (figure bottom), and in this example, each of the four points co-occurs with a rapid spatial change in cost, from either high to low, or low to high. In disparity space, the length of $\overline{\text{AB}}$ (resp. $\overline{\text{CD}}$) grows with the size of discontinuity of scene depth, but due to the rectified camera geometry [5], it always has slope equal to $45°$ (resp. $-45°$).

Figure 3 depicts idealized epipolar slices for seven categories of local scene structure. The epipolar slices are idealized in the sense that cost is abstracted as being either "high" (orange) or "low" (blue). We identify four basic occlusion categories (1–4) that each induce a distinct signature in the cost volume. For comparison, we also show epipolar slices for textured occlusion-less scenarios (5–7) that induce similar left and right images and/or similar cost signatures.

Category (1) is an idealized version of Figure 2 and can always be distinguished from occlusionless texture bound-

aries (5) in the cost volume even though they can appear the same in a single image. Different local signatures occur (2-4) when the foreground or background lacks texture. Notably, we observe that when the background lacks texture (3 & 4), occlusion boundary points A,C in the cost volume cannot be locally distinguished from their planar cousins (points E,F in (6 & 7)). This effect is consistent with the perceptual study by Tsirlin *et al*. [29] that, among other things, introduced the stimuli in the second and third rows of Figure 8. The only difference between these two stimuli is in the left image: the left white rectangle is slightly wider in row two. This causes a change in depth perception, from two separate foreground planes to one. In the language of our taxonomy, the third row contains an E-type point (scenario (6)) while the second row contains an A-type point (scenario (3)) at the same spatial location. We contend that any local detector that fires at one of these points will also fire at the other and, moreover, that the distinction between the firing being caused by depth event (3) versus a texture event (6) cannot be made without additional non-local reasoning. Indeed, this is what we see in our results, including for the perceptual stimuli of Figure 8.

An intriguing property of the local occlusion signatures is that they are unaffected by thin foreground surfaces that violate the so-called ordering constraint [4]. This constraint is commonly enforced by stereo algorithms to increase efficiency, but it prevents such algorithms from being able to recover the depth of thin foreground objects. In contrast, any occluding boundary detector that is based solely on the local signatures should avoid this problem, and should succeed regardless of "ordering". Figure 4 shows one example of a thin textured foreground and textured background that violate ordering. While the cost volume is slightly different from Figure 3(1), the local signatures are the same.

In what follows, we introduce a local stereo occlusion boundary detector that fires at A and C points of scenarios (1–4) as well as E and F points of scenarios (6 & 7). In all cases the detector is designed to identify the correct spatial location and disparity (and thus depth) of the detected boundary point, even though in some cases it is equivocal about whether the detected boundary is an occlusion event (*e.g*. point A in (3)) or a texture event (*e.g*. point E in (6)).

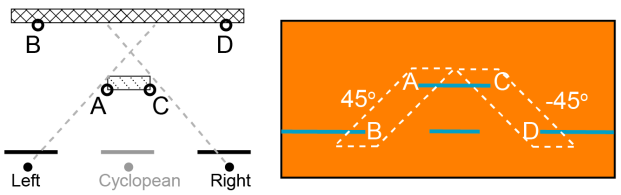

Figure 4. Local signatures remain unchanged when thin foreground structures violate the "ordering constraint". This example of a textured background and thin textured foreground that violates ordering has the same local signatures as Figure 3(1).

4. Stereo Occlusion Boundary Detector

Our detector is a feedforward network that effectively applies a non-linear filter around each point (x, y, d) in a multiscale cost volume to produce a boundary score $B(x, y, d) \in [0, 1]$ for that point. The network is designed to exploit the signatures described above while also providing enough capacity to account for textural variations and for variations in the local orientations of 1D boundaries within the cost volume.

Due to the inherent symmetry of left and right occlusion boundaries, it is unnecessary to train two separate detectors. Instead, as depicted in the right of Figure 5, we can train a detector for only left boundaries and then use the same detector for right boundaries simply by inputting a left/right-reflected copy of cost volume and again reflecting the output. The two left and right boundary maps can be maintained separately or, as we do here, can be combined into a single boundary map using an element-wise maximum.

The middle of our detection network is a 3D variant of the multiscale Holistically-Nested Edge Detection (HED) architecture of Xie and Tu [34], which includes a handful of convolutional and pooling layers. We precede this with a specialized transformation layer that applies geometric and morphological operations to allow supporting evidence from background B-type points to contribute the detection of occluding A-type points without non-local reasoning and regardless of the size of the depth discontinuity (*i.e*. length $\overline{\text{AB}}$ in disparity space). One of our design goals is to make the receptive fields as small as possible, both to detect thin foreground structures and to improve generalization to many types of stereo images (Figures 6, 8).

The remainder of this section describes the details of the network and how it is trained. At the end, we also describe the particular cost volumes that we use as input.

4.1. Transformation Layers

To help the boundary detector use supporting evidence from B-type points without having to reason about the size of $\overline{\text{AB}}$, we incorporate simple geometric and morphological processing before and after the core detection layers. Specifically, as shown in the left of Figure 5 for the case of left-side occlusions, the cost volume is sheared $C'(x, y, d, s) = C(x + d, y, d, s)$ to axis-align the occlusion rays $\overline{\text{AB}}$, and then it is morphologically processed by a cumulative minimum operation along the disparity dimension, $C_{cm}(x, y, d, s) = \min_{d' < d} C'(x, y, d', s)$, that pools supporting evidence from each background B-type point to within the local receptive field of its associated A-type point. The two transformed cost volumes C', C_{cm} are concatenated along the scale dimension to produce a data structure $\tilde{C}$ of size $W \times H \times D \times 2S$ that feeds into the main detection layers. After the detection layers produce boundary scores $\tilde{B}_L$, they are transformed back to the original

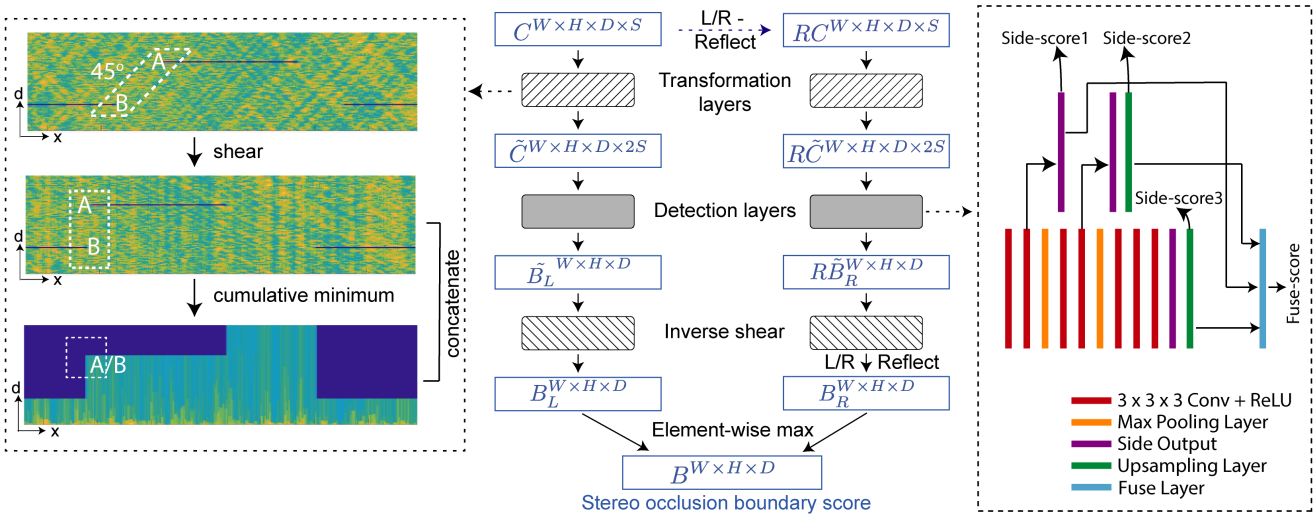

Figure 5. Feedforward stereo occlusion boundary detector. Symmetry of left and right boundaries means that right-side occlusions (C-type) can be detected using the same detector as for left-side occlusions (A-type) by separately inputting a left/right-reflected copy of the cost volume, and reflecting it back after the detector. In either case, there are shear, cumulative minimum, and concatenation operations that pool evidence from two sources: the local neighborhood of an A-type point and supporting evidence from any B-type point that exists along the 45° ray through A. The transformed input $\tilde{C}$ proceeds to detection layers (an HED-inspired 3D deep supervision network, right figure) that produce boundary scores $\tilde{B}_L$ which are then inverse-sheared to the original (x, y, d) coordinate system.

coordinate system by an inverse shear, $B_L(x, y, d, s) = \tilde{B}_L(x - d, y, d, s)$.

4.2. Detection Layers

The detection layers accept the transformed cost volume $\tilde{C}$ and produce per-voxel boundary scores $\tilde{B}_L^{W\times H\times D}$ that approximate the veridical binary boundary map $Z^{W\times H\times D} \in \{0, 1\}$ of an observed scene. We use a 3D variant of the HED architecture [34]. As shown in the right of Figure 5, it has seven convolutional layers, each with $3 \times 3 \times 3$ filters, separated by two max pooling layers . The number of channels increases by a factor of two after each pooling layer, beginning with 64 channels. Our best results are obtained using $2 \times 2 \times 2$ pooling followed by $2 \times 2 \times 1$ pooling, perhaps because the geometric processing eliminates the need to look far along the disparity dimension.

Similar to [34], outputs are extracted from before each pooling layer and from the final layer. At the i^{th} output, there is a deconvolution to upsample to the original size $W \times H \times D$ and then a classifier with sigmoid activation that produces the i^{th} boundary score $\tilde{B}_{side}^{(i)} \in [0, 1]$. We calculate the i^{th} side loss $l_{side}^{(i)}(Z, \tilde{B}_{side}^{(i)})$ using class-balanced cross entropy, and we compute their subtotal:

$$\mathcal{L}_{side}(Z, \tilde{B}_{side}^{(i)}) = \sum_{i=1}^{3} l_{side}^{(i)}(Z, \tilde{B}_{side}^{(i)}). \quad (1)$$

In addition, the three side losses are linearly combined with trainable weights h to produce a fourth "fused" score $\tilde{B}_{fuse}$ for which we also compute class-balanced cross entropy loss $\mathcal{L}_{fuse}(Z, \tilde{B}_{fuse})$. The total loss is the sum $\mathcal{L}_{\tilde{B}} = \mathcal{L}_{side} + \mathcal{L}_{fuse}$. At test time, the output boundary score is simply the average scores of the three side layers and the fuse layer:

$$\tilde{B}_{out} = \frac{1}{4}(\tilde{B}_{fuse} + \sum_i \tilde{B}_{side}^{(i)}). \quad (2)$$

4.3. Training

We render a synthetic dataset to train our detection network from scratch. This allows us to systematically cover all possible orientations of the occluding surface. We use simple two-plane scenes. The background plane is fronto-parallel and covers the entire visual field. (We find it unnecessary to also include slanted background planes, since the output of the cumulative minimum operation in Figure 5 is very insensitive to the background's orientation.) The foreground plane is square and slanted, with orientation parameterized by the normal direction $\mathbf{n}(\theta, \phi)$ and an azimuthal rotation α. We uniformly sample the upper hemisphere to obtain 136 different normals, and we uniformly sample 16 azimuthal angles in $[0, \pi/4]$. For each orientation of the foreground plane, we render 7 stereo pairs with randomly-selected combinations of background and foreground textures from a pre-determined dataset that consists of a grayscale version of the Describable Textures Dataset [11] plus 35 uniform-intensity "textures" that have different intensities. We force 5 scenes to be uniformly textured in both planes (with different intensities). We build a cost volume for each stereo pair, crop it into smaller

$256 \times 64 \times 128 \times S$ sub-volumes, and discard the sub-volumes that do not contain any positive training examples. In total there are $15,232$ stereo pairs of resolution 600×600.

We have an imbalanced training dataset with many more negatives in the cost volume. To effectively train our detector, we perform hard-negative mining where we identify the five highest-scoring negatives along the disparity dimension at each pixel (x, y) of the cost volume and train the network using only these negative examples. We train one epoch using a batch size of one and with the Adam optimizer. The learning rate is initially set to 10^{-4} and decreases by an order of magnitude every $10,000$ iterations after the first $20,000$ iterations.

4.4. Input Cost Volume

For our tests, we compute a multiscale cost volume in the following way. Let $I_l^{W \times H}$ and $I_r^{W \times H}$ be rectified grayscale stereo images, which are normalized to have zero mean and unit standard deviation. We construct the input cyclopean cost volume $C^{W \times H \times D \times S}$ using $S = 3$ Siamese networks. These networks have 2, 4 and 6 convolutional layers respectively, and 128, 64, and 64 channels. All filters are 3×3, and each layer except the last is followed by ReLU operation. We do not pad, stride or pool. The output from the final layer of each network is then normalized to have unit length and the normalized output can be considered as feature embedding of the center pixel with different patch sizes, denoted as $f_{l_s}^{W \times H \times K_s}$ and $f_{r_s}^{W \times H \times K_s}$. We build the cost volume using inner products between feature vectors: $C(x, y, d, s) = \langle f_{l_s}(x + d, y), f_{r_s}(x - d, y) \rangle$.

We use Middlebury 2014 dataset [24] to train the Siamese networks. There are 23 scenes with semi-dense ground truth data. We hold out the last five alphabetically (Storage, Sword1, Sword2, Umbrella and Vintage) for testing our stereo occlusion boundary detector, and use the remaining 18 examples (including the different lighting conditions) to train the Siamese networks. We use the Hinge loss as in [37], and we use similar training methods and hyperparameters except that (1) we exclude patches that span stereo occlusion boundaries and (2) we do not augment data.

5. Experiments

We test our detector on the Middlebury [24], Sintel [8] and Perceptual Stimuli [31] datasets. All experiments are done using the same network weights without fine-tuning. Experiments show that the detector succeeds despite being trained on rendered, abstract images.

We compare to occlusion boundaries that we extract from the depth maps produced by three stereo algorithms: (1) semi-global matching with left-right consistency check (SGM-LR) [18]: a global algorithm that explicitly outputs occlusion maps (and thus stereo occlusion boundary maps); (2) Consensus [9]: a message-passing algorithm with partial occlusion handling; and (3) PSM-Net [10]: an end-to-end network (we use the Stacked Hourglass model trained on KITTI 2015 [23] by the authors).

For our detector, we post-process the final score map B to produce thinner boundaries as follows. For every $\pm 45^\circ$ ray, $x = d$ or $x = -d$ in $B(x, y, d)$, we keep the location with the maximum score and suppress other locations' scores to 0, since there can be at most one stereo occlusion boundaries along each ray. We then apply a one-dimensinal non-maximum suppression along the x-axis. Finally, we convert our boundary score map from cyclopean coordinates (x, y, d) to the left view $(x + d, y, 2d)$ which is the native view for the other techniques.

For SGM-LR, we use the implementation by Yamaguchi *et al*. [35], which outputs both left and right occlusion maps. Theoretically, the pixels adjacent to the right (left) boundaries of the occluded regions in the left (right) occlusion map are the stereo occlusion boundaries. Empirically, we found the occlusion boundaries derived from the above step are noisy, so we further post-process to only keep the ones with correct occlusion polarities (*i.e*. the occluding surface has higher disparity than the background surface based on the output disparity map). We call this SGM-LR-Plus.

For Consensus and PSM-Net, we use the following steps to extract occlusion boundaries from the output left disparity map $\hat{d}(x, y)$. We first find all boundary candidate pixels $\hat{B}(x,y) = (\hat{d}(x,y) - \hat{d}(x-1,y) > 1) \vee (\hat{d}(x,y) - \hat{d}(x+1,y) > 1)$, where $\vee$ is the element-wise logical OR operator. We observe that $\hat{B}(x, y)$ is often thickened, and for each connected component along an epipolar scanline, the candidate closest to the foreground (the side with higher disparity) is usually the closest candidate to the actual boundary. Thus, we only keep these candidates.

Middlebury. We test our detector on the Middlebury 2014 dataset using the 5 scenes (with perfect rectification and lighting) held out from training the cost volume Siamese networks. We hand-label stereo occlusion boundaries on these images with the help of the semi-dense ground truth disparity maps. We only evaluate on the spatial dimensions (x, y) due to missing ground truth disparities. We use half resolution for the Sword1, Sword2, Umbrella scenes and quarter resolution for the Storage and Umbrella scenes since some comparison methods (*e.g*. PSM-Net [10]) cannot handle large disparities. We use the BSDS correspondence algorithm [22] to evaluate our stereo occlusion boundary accuracy, but require a stricter threshold for finding correspondences since our images are larger and our motivation is to get the boundaries as precisely as possible. We use 0.003 of the image diagonal length, which roughly equals to 5 pixels in Middlebury half-size images, and 2 pixels in quarter-size images. This is also a minimum reasonable distance in practice considering possible human labelling errors and

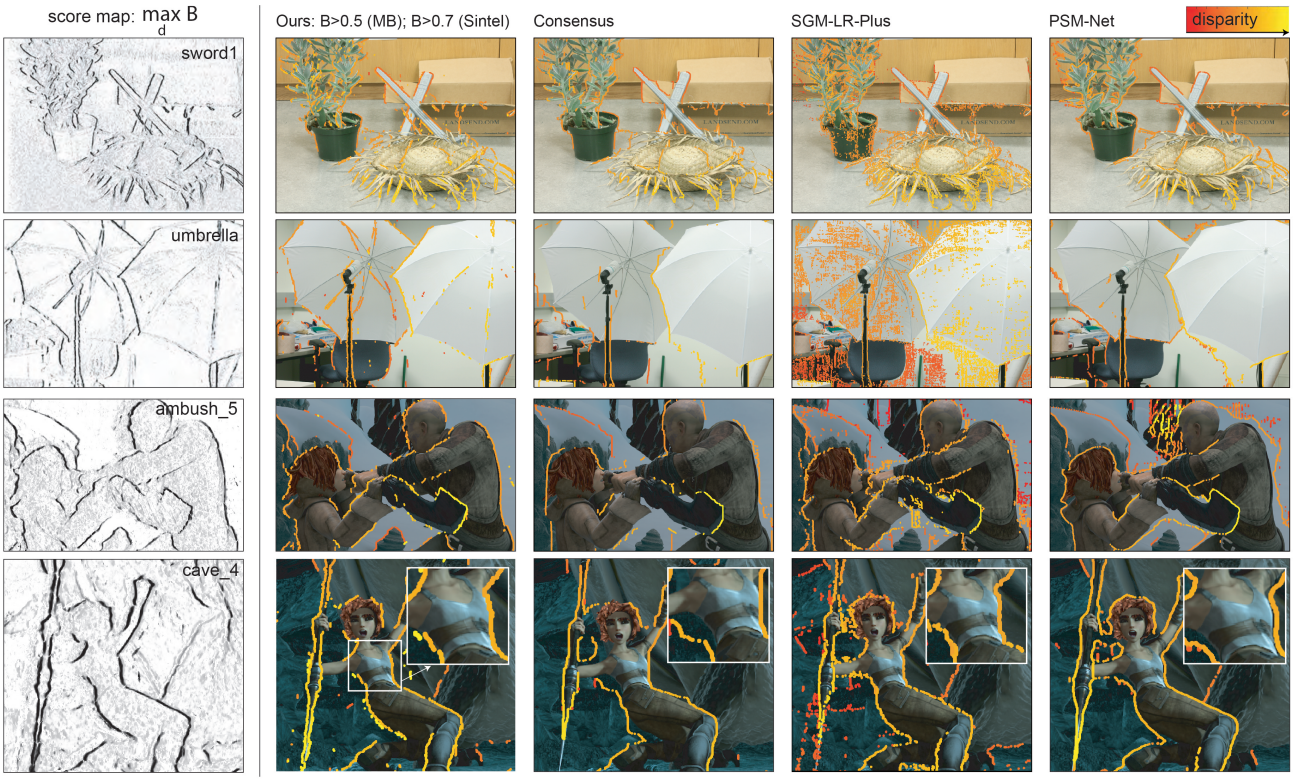

Figure 6. Qualitative results on Middlebury and Sintel. First column is a spatial map of our boundary score, $\max_d(B)$, and other columns are detected boundaries (ours with $B > 0.5$ for Middlebury and $B > 0.7$ for Sintel) colored by the disparity of their detection. Zoom in to see that our detector's boundaries are localized more precisely in (x, y) whereas the boundaries extracted from the comparison methods are consistently offset from the true ones, especially the left boundaries. In some cases (e.g. cave_4), our purely-local detector could produce salt-and-pepper-like false positives. See supplementary materials for more results. Note that in quantitative evaluations (Figs. 7 & 9), our detections of E and F-type points are counted as "false positives" even though they occur at the correct location and disparity.

lens blur. As shown in Figure 7, our detector clearly outperforms other methods by a big margin. Figure 6 shows some qualitative results with disparities encoded in color. The boundaries predicted by our detector are very precise

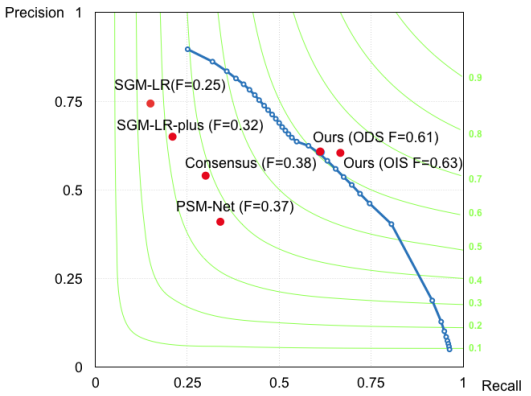

Figure 7. Middlebury precision-recall curve and F-scores evaluated on (x, y). Our detector achieves ODS F-score=0.61 with a strict "true positive" criteria (see text).

whereas many predicted boundaries in other methods offset from the actual boundaries. Notice some of the "false positives" of our method are approximately E, F types of points, thus our actual F-score could be higher.

Sintel. We also test our detector on Sintel clean pass [8], using the first frame of each image sequence. We use Sintel clean because many Sintel final images have high levels of blur or fog which make it hard to define a meaningful notion of ground truth. We exclude "ambush_7", "bandage_2", "mountain_1", "shaman_2" and "shaman_3" when computing the scores[2], resulting in a total of 18 examples. We evaluate the accuracy in (x, y, d), allowing 5 pixels in each dimension for finding "true positives". We use half-pixel resolution in the cyclopean d dimension in our method by interpolating the cost volume by $C(x, y, d + 1/2) = \max(C(x, y, d), C(x, y, d + 1))$, so that all methods have the same disparity resolution in left-view coordinates. Figures 6 and 9 show the results. Again, our detector locates more precise stereo occlusion boundaries than comparison

[2]These images either have no stereo occlusion boundaries or large disparity values that over the limit of some comparison algorithms.

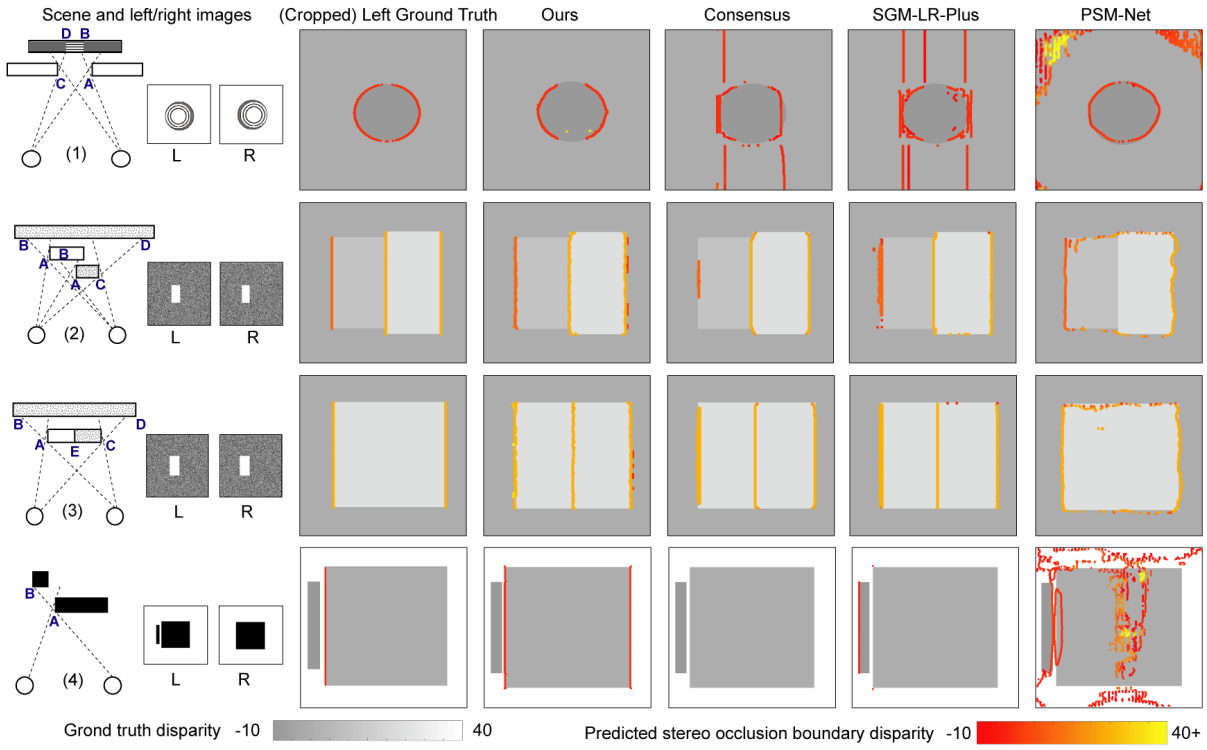

Figure 8. Results for perceptual stimuli. Detected boundaries (ours with $B > 0.7$) are colored according to the disparity of their detection and superimposed on the true disparity map. As per Fig. 3, local detectors like ours can successfully localize the location and depth of both E-type events (row 3) and A-type events (row 2) without distinguishing their cause. See the supplementary material for more results.

methods, even along very thin objects (*e.g.* in cave_4).

Perceptual Stimuli. Wang *et al.* [31] collected a dataset of 12 perceptual stimuli that lack matching and/or monocular cues, many of which are used by vision scientists to show human can use occlusion cues alone to identify depth discontinuities. Figure 8 shows some selected results. Our detector detects all stereo occlusion boundaries as well as E-type points in (3) as expected. Methods that explicitly handle occlusion (SGM-LR-Plus and Consensus) locate relatively good stereo occlusion boundaries when there is enough matching information (stimuli (2) and (3), random dots). However, they fail completely when matching information is unavailable (stimuli (1) and (4)).

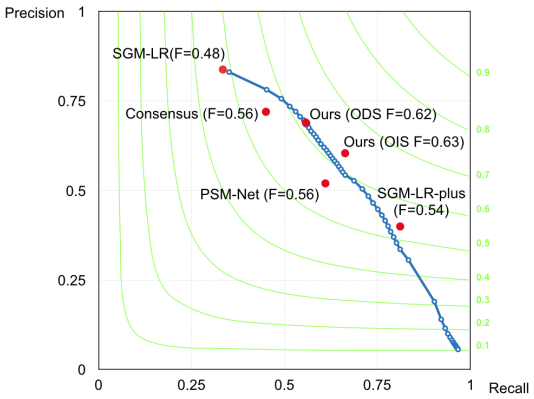

Figure 9. Sintel precision-recall curve and F-scores evaluated on (x, y, d). Our detector achieves the best result.

6. Conclusion

In many cases, stereo occlusion boundaries can be detected and localized with high precision without computing a dense disparity map, and without incorporating top-down contextual cues or single-image texture and intensity cues. These detections are therefore an additional cue that can be incorporated into stereo vision systems. Future work should explore extensions to these detectors that are selective with respect to orientation in (x, y, d). This might provide a mechanism to explain the perceptual phenomenon of illusory stereo contours [12].

Acknowledgement: This work was supported by National Science Foundation award IIS-1618227. The research computing was partially supported by the AWS Cloud Credits for Research program.

References

[1] Barton L. Anderson. The role of partial occlusion in stereopsis. *Nature*, 1994. 2

[2] Barton L. Anderson and Ken Nakayama. Toward a general theory of stereopsis: binocular matching, occluding contours, and fusion. *Psychological Review*, 1994. 2, 3

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 2

[4] Henry H. Baker. Depth from edge and intensity based stereo. *Stanford University Department of Computer Science Technical Report*, 1982. 4

[5] Peter N. Belhumeur. A Bayesian approach to binocular steropsis. *International Journal of Computer Vision*, 1996. 2, 3

[6] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 1999. 2

[7] Aaron F. Bobick and Stephen S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 1999. 2

[8] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. *European Conference on Computer Vision*, 2012. 2, 6, 7

[9] Ayan Chakrabarti, Ying Xiong, Steven J. Gortler, and Todd Zickler. Low-level vision by consensus in a spatial hierarchy of regions. *Computer Vision and Pattern Recognition*, 2015. 6

[10] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. *Computer Vision and Pattern Recognition*, 2018. 2, 6

[11] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition*, 2014. 5

[12] Walter H. Ehrenstein and Barbara J. Gillam. Early demonstrations of subjective contours, amodal completion, and depth from half-occlusions:"stereoscopic experiments with silhouettes" by Adolf von Szily (1921). *Perception*, 1998. 8

[13] Pascal Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine vision and applications*, 1993. 2

[14] Barbara Gillam. The influence of monocular regions on the binocular perception of spatial layout. *Vision in 3D Environments*, pages 46–69, 2011. 2

[15] Barbara Gillam and Eric Borsting. The role of monocular regions in stereoscopic displays. *Perception*, 1988. 2

[16] Julie M. Harris and Laurie M. Wilcox. The role of monocularly visible regions in depth and surface perception. *Vision research*, 2009. 2

[17] Xuming He and Alan Yuille. Occlusion boundary detection using pseudo-depth. In *European Conference on Computer Vision*. Springer, 2010. 2

[18] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 2008. 6

[19] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. *European Conference on Computer Vision*, 2018. 2

[20] Hiroshi Ishikawa and Davi Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European conference on computer vision*. Springer, 1998. 2

[21] David G. Jones and Jitendra Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 1992. 2

[22] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004. 2, 6

[23] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. *Computer Vision and Pattern Recognition*, 2015. 6

[24] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. *German Conference on Pattern Recognition*, 2014. 1, 2, 6

[25] Andrew N. Stein and Martial Hebert. Local detection of occlusion boundaries in video. In *British Machine Vision Conference*, 2006. 2

[26] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *Computer Vision and Pattern Recognition*, 2018. 2

[27] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. *Computer Vision and Pattern Recognition*, 2005. 2

[28] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbeláez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *Computer Vision and Pattern Recognition*, 2011. 2

[29] Inna Tsirlin, Laurie M. Wilcox, and Robert S. Allison. Monocular occlusions determine the perceived shape and depth of occluding surfaces. *Journal of Vision*, 2010. 2, 3, 4

[30] Rüdiger Von Der Heydt, Hong Zhou, and Howard S. Friedman. Representation of stereoscopic edges in monkey visual cortex. *Vision research*, 2000. 2

[31] Jialiang Wang, Daniel Glasner, and Todd Zickler. Toward perceptually-consistent stereo: A scanline study. *International Conference on Computer Vision*, 2017. 2, 3, 6, 8

[32] Yichen Wei and Long Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. *Computer Vision and Pattern Recognition*, 2005. 2

[33] Juyang Weng, Narendra Ahuja, and Thomas S. Huang. Two-view matching. In *International Conference on Computer Vision*, 1988. 2

[34] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Conference on Computer Vision*, 2015. 2, 4, 5

[35] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. 2014. 6

[36] Yibing Yang, Alan Yuille, and Jie Lu. Local, global, and multilevel stereo matching. In *Computer Vision and Pattern Recognition*, 1993. 2

[37] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 2016. 2, 6

[38] C. Lawrence Zitnick and Takeo Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000. 2

Supplementary material: Local detection of stereo occlusion boundaries

Jialiang Wang and Todd Zickler
Harvard University
jialiangwang@g.harvard.edu, zickler@seas.harvard.edu

1. Additional Details Regarding Middlebury Experiments

We test our detector on five scenes from the Middlebury 2014 training set [6] using stereo occlusion boundaries that we manually label (in (x, y)) as ground truth. These five scenes are held out when training the Siamese matching network that provides input cost volumes C to our detector. Figure 1 shows the results using Middlebury "half resolution" images.

Figure 1. First column: Manually-labelled ground truth stereo occlusion boundaries in (x, y). Second column: Score $max_d B(x, y, d)$ with B being our detector's output before non-maximum suppression. Third column: Detection results in (x, y, d), using $B > 0.5$, superimposed on the stereo left image. For visualization, ground truth labels are dilated to five-pixel wide.

2. Additional Results: Sintel

Figure 2 shows additional results on the Sintel [1] "clean pass" dataset, along with the labelled ground truth stereo occlusion boundaries in (x, y, d). We require a pixel to be mutually-visible (*i.e.* in a binocular region) and to occlude at least two adjacent pixels in the cyclopean view in order for it to be considered as a stereo occlusion boundary.

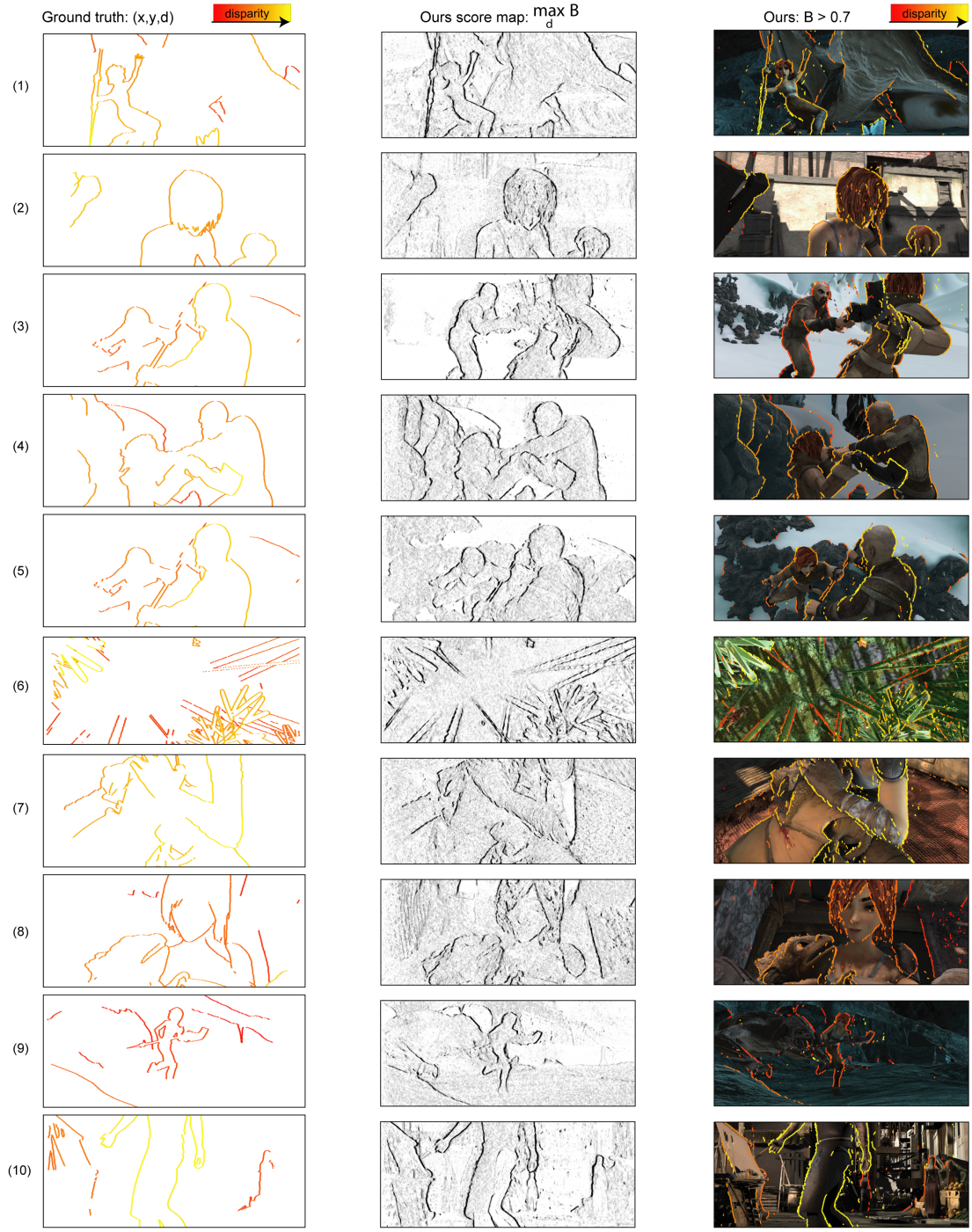

Figure 2. First column: Labelled ground truth stereo occlusion boundaries in (x, y, d), with d encoded using color. Second column: $max_d B(x, y, d)$ where B is our detector's output before non-maximum suppression. Third column: three-dimensional detection results (x, y, d) (using $B > 0.7$) superimposed on the left image. For visualization, ground truth labels are dilated to five-pixel wide.

3. Additional Results: Perceptual Stimuli

There are twelve stimuli in the Perceptual Stimuli Dataset [7]. Here we show the remaining eight stimuli that are not included in Section 5 of our paper, along with comparisons to three other stereo methods [5, 3, 2]. All methods work well for stimuli with sufficient matching cues (*e.g.* (1), (7) and (8)). Some other methods break down when ambiguities arise (*e.g.* (2) and (3)). All other methods fail when there is no matching information. Our method detects all stereo occlusion boundaries, as well as some E, F-type points, as discussed in Section 3 of the paper. Since our detector is a local one, we have some noise in some examples, especially (4), where the repeated textures are ambiguous and nearly E or F-type points in our taxonomy.

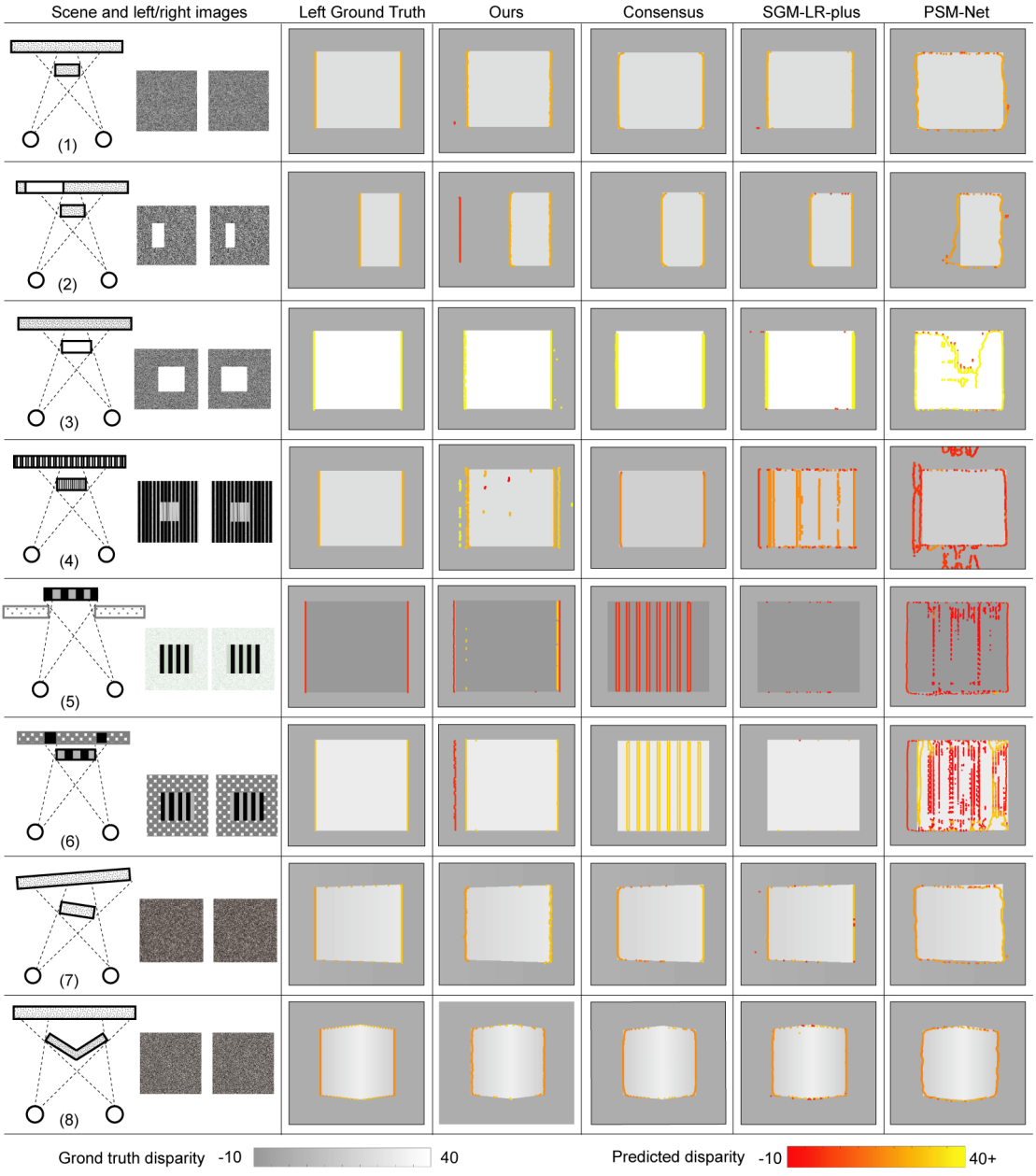

Figure 3. Perceptual Stimuli: comparison of our method's stereo occlusion boundaries (using $B > 0.7$) and those generated by four other stereo algorithms. Some results are cropped to show the region of interest.

4. Examples of Synthetic Training Images

Figure 4 shows some examples of the synthetic training images we rendered to train our stereo occlusion boundary detector. Notice that we randomly sample textures for each scene. In general, we found it helpful to include as many textures as possible. Our detector is robust and works well on all three datasets in test time despite only trained on these simple and abstract images.

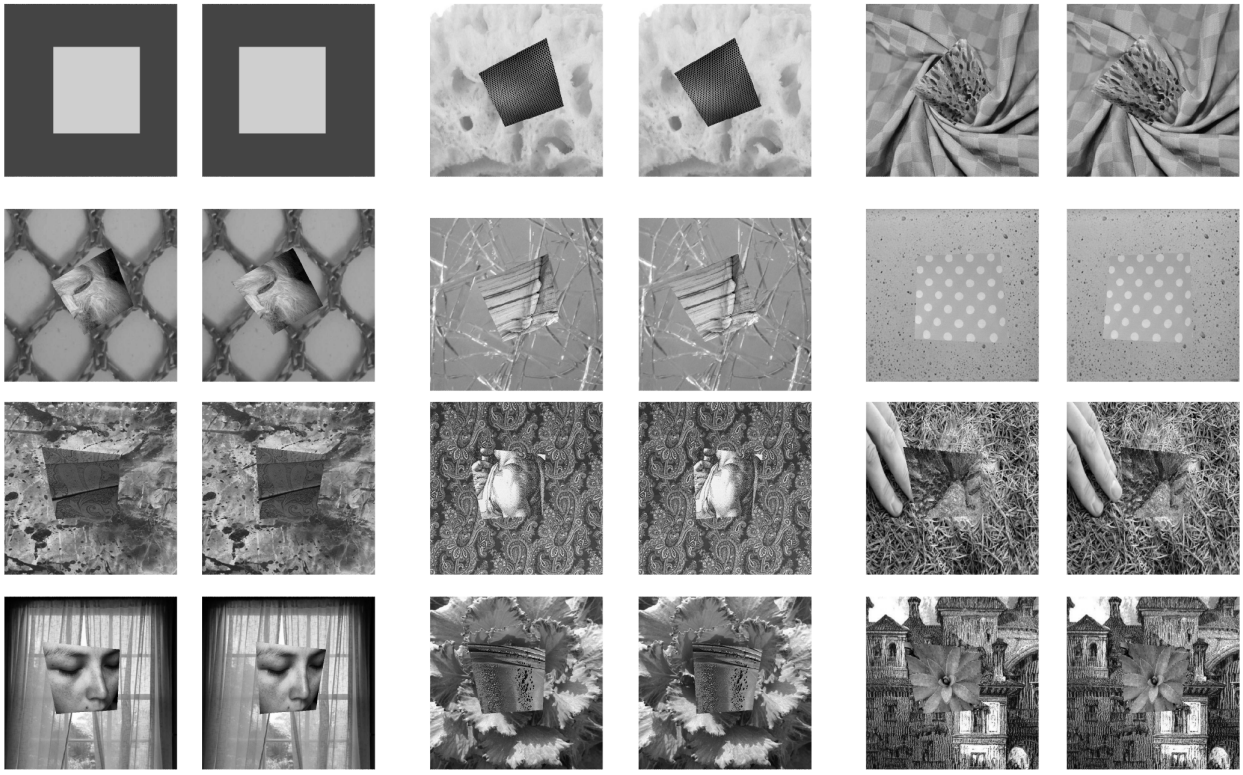

Figure 4. Selected synthetic images we use to train our feedforward stereo occlusion boundary detector. These images are rendered using a grayscale version of the Describable Textures Dataset [4] plus 35 uniform-intensity "textures" that have different intensities. The images are 600×600 resolution.

References

[1] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. *European Conference on Computer Vision*, 2012. 2

[2] Ayan Chakrabarti, Ying Xiong, Steven J Gortler, and Todd Zickler. Low-level vision by consensus in a spatial hierarchy of regions. *Computer Vision and Pattern Recognition*, 2015. 3

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. *Computer Vision and Pattern Recognition*, 2018. 3

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition*, 2014. 4

[5] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 2008. 3

[6] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. *German Conference on Pattern Recognition*, 2014. 1

[7] Jialiang Wang, Daniel Glasner, and Todd Zickler. Toward perceptually-consistent stereo: A scanline study. *International Conference on Computer Vision*, 2017. 3