UltraStereo: Efficient Learning-based Matching for Active Stereo Systems

Sean Ryan Fanello* Julien Valentin* Christoph Rhemann*
Adarsh Kowdle* Vladimir Tankovich* Philip Davidson Shahram Izadi

perceptiveIO

Abstract

Efficient estimation of depth from pairs of stereo images is one of the core problems in computer vision. We efficiently solve the specialized problem of stereo matching under active illumination using a new learning-based algorithm. This type of 'active' stereo i.e. stereo matching where scene texture is augmented by an active light projector is proving compelling for designing depth cameras, largely due to improved robustness when compared to time of flight or traditional structured light techniques. Our algorithm uses an unsupervised greedy optimization scheme that learns features that are discriminative for estimating correspondences in infrared images. The proposed method optimizes a series of sparse hyperplanes that are used at test time to remap all the image patches into a compact binary representation in $\mathrm{O}(1)$*. The proposed algorithm is cast in a PatchMatch Stereo-like framework, producing depth maps at* $500Hz$*. In contrast to standard structured light methods, our approach generalizes to different scenes, does not require tedious per camera calibration procedures and is not adversely affected by interference from overlapping sensors. Extensive evaluations show we surpass the quality and overcome the limitations of current depth sensing technologies.*

1. Introduction

Depth cameras have become key in tackling challenging computer vision tasks including human capture [10], 3D scene reconstruction [28], object recognition [51] and robotics [13]. Many different depth sensing techniques and technologies have been proposed: from gated [12, 2] or continous wave (e.g. Kinect V2) *time-of-flight* (ToF), to *triangulation-based* spatial [19, 52] or temporal [21] structured light (SL) systems.

ToF cameras have recently gained a lot of attention due to wide spread commercial availability (e.g. products from Intel, Microsoft, PMD and Sony). ToF sensors capture multiple images of the same scene under different active illumination. Using these images, a single depth map is estimated. For instance, the Kinect V2 sensor captures raw infrared (IR) frames at 300Hz in order to produce depth maps at 30Hz [48]. This requires custom high-speed image sensors that are expensive and low-resolution. The produced depth maps can suffer from motion artifacts, as they are constructed over a temporal window. Another significant limitation of ToF cameras is multipath interference (MPI). This occurs when, during a single camera exposure, emitted light is scattered and reflected from multiple surfaces, and collected as a single convolved sensor measurement. Despite the significant efforts on MPI [29, 20, 5, 37], there are no commercially-viable solutions for this problem if precise depth is required for general scenes.

SL systems [43, 21], fall into two groups: spatial or temporal. Temporal systems (e.g. Intel SR300) are computationally efficient but require multiple captures across a temporal window, causing motion artifacts. Moreover the maximum range is very short (up to 120cm). Most systems also require custom MEMs-based illuminators to produce dynamic rather than static patterns which can be costly and require more power. Spatial SL systems are simpler and either use a diffractive optical element (DOE) [19] or mask-based [22] illuminators to generate pseudo-random patterns, and off-the-shelf imaging sensors. These however suffer from robustness concerns: the pattern needs to be known a-priori, so any modifications of the pattern will cause issues. This can be caused if the wavelength of the emitted light drifts, e.g. if the laser is not thermally stabilized or if two or more projectors overlap onto the same scene, causing interference between sensors.

One way to overcome these challenges is to use an *active stereo* setup [31]. These systems use two calibrated cameras and project a pattern into the scene to provide texture for matching. Although the technique has been described in seminal work from the 1980s [38], widely available commercial products leveraging this approach only recently emerged [1]. Active stereo systems are compelling since: (1) they can operate in texture-less regions, which is a major shortcoming of passive stereo techniques [7]; (2) they mitigate the challenge of multi-path reflections, inherent in ToF [20]; and

*Authors equally contributed to this work.

(3) they offer more robustness over traditional triangulation-based structured light systems, by removing the need to project a known structured light pattern as well as avoiding interference between sensors [8].

However, this type of depth sensing carries with it a key challenge: the computational overhead to find correspondences across stereo images for disparity (and hence depth) estimation. Mainstream techniques usually take a matching window around a given pixel in the left (or right) image and given epipolar constraints find the most appropriate matching patch in the other image. This requires a great deal of computation to estimate depth for every pixel.

Significant efforts from the community have been spent on overcoming this computational bottleneck. *Local* stereo matching techniques have proposed solutions that are independent of the matching window size e.g. [42, 34, 30] or the range of disparities [7, 4]. However, these methods are still computationally expensive, especially when considering commercial viability or comparing to look-up based approaches used in ToF or temporal SL. Therefore, most algorithms have to trade resolution and accuracy for speed in order to reach real-time scenarios. Even then, these systems barely reach 30Hz. Recent methods have been proposed that can scale independently of the window size and the range of disparities. However, these methods rely on expensive steps such as computing superpixels [35, 33], making them prohibitive for real-time scenarios.

In this paper, we solve this fundamental problem of stereo matching under active illumination using a new learning-based algorithmic framework called *UltraStereo*. Our core contribution is an unsupervised machine learning algorithm which makes the expensive matching cost computation amenable to $\mathrm{O}(1)$ complexity. We show how we can learn a compact and efficient representation that can generalize to different sensors and which does not suffer from interferences when multiple active illuminators are present in the scene. Finally, we show how to cast the proposed algorithm in a PatchMatch Stereo-like framework [7] for propagating matches efficiently across pixels.

To demonstrate the benefits of our approach, we built a prototype system with off-the shelf hardware capable of producing depth images at over 200Hz. We also prove that the algorithm can be run on single camera-projector systems such as Kinect V1, surpassing the quality of the depth estimation algorithm used by this commercial system. Exhaustive evaluations show that our algorithm delivers state of the art results at high framerate and high image resolution. We show how the learned representation can generalize to different sensors and scenes.

In contrast to standard structured light methods, our approach does not require tedious camera-illuminator calibration and is not adversely affected by interference when multiple sensors are operating in the same space. Whilst related learning-based algorithms such as HyperDepth [16] have overcome quality and speed limitations in structured light systems, these methods require tedious calibration and expensive data collection phases. More fundamentally these cannot scale to the stereo (two or more) camera matching case. This is because methods like HyperDepth makes the assumption that the matching occurs across one camera observation and a fixed pattern that remains constant independent of scene depth, whereas the problem of active stereo inherently assumes that a pattern will be observed *differently* across two cameras, and both observed patterns *will* change based on depth.

2. Related Work

Depth estimation from stereo is one of the most active topics in computer vision of the last 30 years. The simplest setup involves two calibrated and rectified cameras, where the problem is to establish correspondences for pixels belonging to the same scanline in both images. As described in [45], the main steps of stereo algorithms are: matching cost computation, cost aggregation, disparity optimization followed by a disparity refinement step. Methods can be categorized in *local* [54, 42, 7, 35], *global* [17, 4, 32, 33] or *semi-global* [26, 6], depending on the techniques used to solve each step of the pipeline.

The correspondence search problem is the most demanding part of the system. Given a disparity space of L possible labels, simple brute force methods (e.g. block matching) have a complexity per pixel of $\mathrm{O}(LW)$, where W is the window size used to compute the correlation among patches (typically 9×9 for a 1.3M resolution image). In order to reduce this computational burden, most of the methods in literature try to remove the linear dependency on the window size W or on the number of disparities L.

Many correlation functions can be implemented as a filter with a computational complexity independent of the filter size. For instance, the sum of absolute differences (SAD) corresponds to a simple box filter [45] that can be optimized for computational performance. Recent real-time stereo approaches focus on filters that set a weight to each pixel inside the correlation window based on image edges, e.g. based on bilateral filtering [50, 40] or guided image filtering [24]. In addition, several methods have been proposed for guided cost volume filtering [42, 34, 30]. Since their runtime cost is $\mathrm{O}(L)$ per pixel, these approaches show good computational performances only if the number of disparities is small.

Other recent work leverage the framework of PatchMatch Stereo [7, 4]. PatchMatch Stereo alternates between random depth generation and propagation of depth. However, the runtime performance of the algorithm depends on the correlation window size W, with a total runtime cost of $\mathrm{O}(W \log L)$. A recent attempt to remove the linear dependency on both the window size and label space is presented

in [35]. This method strongly relies on superpixels, which require a non negligible amount of compute to estimate. Moreover superpixels are well defined for RGB images, but it is not straightforward to characterize in IR images.

Machine learning methods such as [56] use deep neural networks to compute the matching cost, increasing the overall complexity. Furthermore, those methods still need multiple disparity hypothesis to be evaluated. Others [11, 44] try to predict depth from a single image, but their applicability is limited to very specific scenes. [14] use diffuse infrared light to learn a shape from shading mapping from IR intensity to depth, but the technique works only for hands and faces in a very limited range. More recently, [16] uses machine learning to learn the reference pattern used in structured light system such as Kinect V1 and demonstrates state of the art results. However, the method cannot be applied in stereo setups as it requires per camera training and fails in the presence of pattern interference (see Fig. 8).

In contrast to previous work, we use an unsupervised machine learning technique to compute the matching cost in $\mathrm{O}(1)$, removing the dependency on the window size W. Leveraging the PatchMatch framework [3] to perform the disparity optimization and refinement leads the proposed method to have a total complexity of $\mathrm{O}(\log L)$ per pixel. In the experiments, we demonstrate that the method does not require a per camera training, generalizes to multiple scenes and does not suffer from interference.

3. UltraStereo Algorithm

Our method is designed to work with commercially available spatial structure light systems (e.g. Kinect V1) as well as with active stereo camera setups. The Kinect V1 hardware consists of a DOE projector and a single camera which is a particular case of active stereo, where one camera is a fixed reference image. This particular setup requires a tedious calibration procedure, since the relative position between the reference pattern and the camera-projector needs to be estimated (see [36] for details). To demonstrate the full potential of our method, we built a hardware prototype consisting of two IR cameras in a stereo configuration. We use monochrome Ximea cameras with a 1280×1024 spatial resolution, capable of capturing raw images at 210Hz. The cameras are calibrated and rectified using a standard multi view calibration approach [23].

In the remainder of the paper, we assume the pair of images to be rectified. Therefore, each pixel $\mathbf{p} = (x, y)$ in the left image L, has a correspondence match in the right image R which lies on line y, but at a different coordinate $\hat{x}$. The difference $d = x - \hat{x}$ is known as disparity and it is inversely proportional to the depth $Z = \frac{bf}{d}$, where b is the baseline of the stereo system and f the focal length. The main computational challenge is to solve the correspondence problem (finding $\hat{x}$) without any dependency on the window size W and the size of the disparity label space L. The latter can be removed using a variant of the PatchMatch framework [41]. With UltraStereo, we show how the dependency on the window size can be removed.

3.1. Compact Representation of Image Patches

In order to evaluate a single disparity hypothesis, traditional stereo algorithms compute the cost between image patches of size W by analyzing all the per pixel differences. Common correlation functions include sum of absolute differences (SAD) and normalized cross evaluation (NCC). Even when a small patch is used (e.g. $W = 5 \times 5$), these functions require a significant amount of operations to obtain the matching cost [27] between two patches. The main intuition behind UltraStereo is that standard correlation functions such as SAD and NCC are unnecessarily expensive to capture the discriminative data contained in the actively illuminated pattern. Indeed given a high frequency pattern such as the one projected by a DOE, the information required for establishing robust matches belongs to a space that is much lower dimensional than the space of IR patches. Therefore, we design a function $\mathbf{y} = f(\mathbf{x})$ that maps an image patch $\mathbf{x} \in \mathrm{R}^W$ in a compact binary representation $\mathbf{y} \in \{0, 1\}^b$ with $b << W$. In particular, typical values are $W = 11 \times 11$ and $b = 32$. For compute reasons, we propose to use a linear mapping to transform the data from R^W to $\{0, 1\}^b$:

$$\mathbf{y} = f(\mathbf{x}) = sign(\mathbf{x}^{\mathrm{T}} \cdot \mathbf{Z} - \theta) \tag{1}$$

with $\mathbf{Z} \in \mathrm{R}^{W \times b}$ and $\theta \in \mathrm{R}^b$. In order to remove the dependency on the window size W, only $k << W$ non-zero elements are allowed in each column of the mapping $\mathbf{Z}$. In practice, using dense hyperplanes has very little impact on the final results. Similarly, using $y \in R^b$ (by dropping the sign in eq. 1) produces similar results compared to using $y \in \{0, 1\}^b$. Motivated by these observations and by the fact that Hamming distances are computed in $\mathrm{O}(1)$ in many modern GPUs, we naturally choose to perform the cost computation in the feature space $y \in \{0, 1\}^b$.

3.2. Unsupervised Binary Representation

The mapping function $\mathbf{f}$ has to be data dependent in order to learn the subspace of patches generated by the active illuminator. Our goal is to find a mapping $\mathbf{Z}$ that preserves the similarities of the input signal. To do so we resort to a greedy method that computes, at each step, the best hyperplane $\mathbf{z}$. This naturally leads to the optimization scheme popularly used when training decision trees. Binary trees are suitable models for making predictions that are both sparse and binary and they have demonstrated state of the art results for numerous problems including pixel-wise labeling [46, 14], regression tasks [47, 15] and correspondence search problems [53] by using very simple and sparse split functions. Each node in a decision tree contains a set of

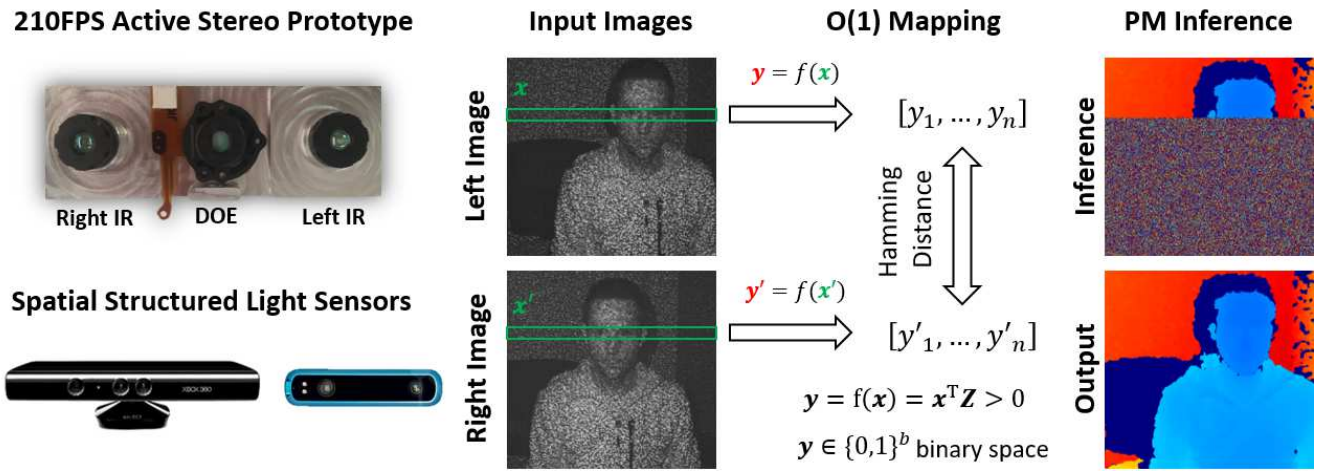

Figure 1. **UltraStereo framework**. We built an active stereo prototype (top left) capable of acquiring raw images at 210Hz. We use two IR cameras and a Kinect DOE for active illumination. The algorithm also works in spatial structured light systems such as KinectV1 and Primesense sensors. Given two rectified input images, we use an efficient $\mathrm{O}(1)$ mapping to transform image patches to a new binary space $\mathbf{y}$. The matching cost is then computed using this compact representation and a PatchMatch Stereo inference is used to infer disparities. See text for details.

learned parameters $\delta = (\mathbf{z}, \theta)$ which define a binary split of the data reaching that node. Based on the sign of $\mathbf{x}^{\mathrm{T}}\mathbf{z} - \theta$, samples are either routed to the left or the right child of the current node. The total number of binary splits b is equal to the number of nodes in the tree. In our case we set $b = 32$, which is equivalent to a single tree of height 5.

In order to learn the split parameters of a tree, one must define a suitable objective function. In [16], authors used a classification based objective function aiming at minimizing the entropy over all the labels. However, our goal is to be as general as possible in order to avoid per-camera training, therefore we rely on an unsupervised objective function, similar to the one used in density forests [9]. Given N unlabelled image patches $\mathbf{x}_i$ collected in arbitrary scenes, we aim at approxminating the underlying generative model of infrared patches. Starting from the root node that contains the set S of all the examples $\mathbf{x}_i \in \mathrm{R}^W$, we randomly sample multiple split parameters proposals δ. In order to enforce sparsity and remove the dependency on W, only k elements in $\mathbf{z}$ are forced to be non-zero. For each candidate δ, we evaluate the information gain:

$$I(\delta) = H(S) - \sum_{d \in L,R} \frac{|S_d(\delta)|}{|S|} H(S_d(\delta)) \qquad (2)$$

where the set $S_d(\delta)$ is induced by the particular split function δ. The entropy $H(S)$ is assumed to be the continuous entropy of a W-dimensional Gaussian, which is equal to:

$$H(S_d) = \frac{1}{2} \log((2\pi e)^W |\Lambda(S_d)|) \qquad (3)$$

where $\Lambda(S)$ is the $W \times W$ covariance matrix of the current set S and $|\cdot|$ indicates its determinant. The candidate δ that maximizes Eq. 2 is selected for each node. The training procedure continues greedily until depth 5 of the tree is reached. Notice that we are not storing any model in the leaves since we only want to exploit the binary splits induced by the δ parameters. At the end of the training phase we concatenate all the split parameters in order to form our linear mapping $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_b]$ and $\theta = [\theta_1, \dots, \theta_b]$.

Note that other binary mappings schemes like Rank and Census have been proposed in the past [55]. Others used random sparse hyperplanes [25] to perform this binary mapping. However, these methods are not data driven and they are a particular case of the proposed framework, which learns a more general mapping function. More importantly, due to their hand-crafted or random nature, previous methods require an output binary space to be of the same magnitude or larger than the patch size W. On the contrary, our method is also able to significantly reduce the dimensionality of the data while retaining the discriminative information they carry for the task of establishing robust correspondences. Hence, UltraStereo encodes patches of size $11 \times 11 = 121$ in 32 binary values which are efficiently stored in a single integer.

3.3. Matching Framework

Once every patch in the left and right images is projected in $\{0, 1\}^b$, we use a variant of the PatchMatch Stereo framework [41] to conduct the inference. The main steps of this framework consist of initialization, propagation and post-processing.

In the initialization step, 5 random disparities are sampled per pixel and their matching cost is evaluated in parallel. In order to achieve subpixel precision, these random disparities

are real-valued. The candidate with the lowest matching score is kept.

Due to its inherently iterative nature, the propagation proposed in [7] can not directly leverage the full potential of massively parallel architectures like GPUs. In order to speedup this step, in our reimplementation we subdivide the image in 64×64 blocks. We run the PatchMatch propagations inside these local blocks, where rows and column are processed in parallel and independently on dedicated GPU threads. In total we run 4 propagations: left to right, top to bottom, right to left, bottom to top. Every time a disparity gets propagated we also use a standard parabola interpolation on the matching cost to further improve subpixel accuracy.

In our GPU post-processing step, we invalidate disparities which are associated with large hamming distances (bigger than 5) and we run connected components followed by a minimum region check to remove outliers. Finally a median filter is run on 3×3 patches to further reduce noise while preserving edges.

3.4. Computational Analysis

We consider an image with N pixels, L possible discrete disparities and images patches of size W. Note that the PatchMatch stereo framework has a complexity of $\mathrm{O}(NW \log L)$ [35]. We now study the complexity of UltraStereo. The mapping to the binary space as described in Eq. 1 is independent of the window size W, since each column of $\mathbf{Z}$ has only k non-zero elements. In practice, we optimized for $k = 4$ due to our computational budget, however empirically we noticed that the number of non-zero elements is not a crucial parameter. The complexity of this mapping is therefore $\mathrm{O}(1)$. Similarly to [41], we don't perform the 'bisection search' over the label space. This further reduces the complexity and removes the dependency on L. The complexity of UltraStereo is then linear with respect to the image size N, and it has only two small constants involved: k non-zero pixels for the mapping computation and the binary space b for the hamming distance. We can count the number of different bits between two binary strings in $\mathrm{O}(1)$, thanks to the $\_\_popc()$ function implemented in most of the latest GPU architectures.

To give more concrete evidence on the speed of the proposed algorithm, we used 1280×1024 images, a window size $W = 11 \times 11$, where only 4 non-zero pixels are sampled per column of $\mathbf{Z}$, and a binary space of size $b = 32$. We tested UltraStereo on a Nvidia Titan X GPU. The initialization step is performed in $130\mu s$, each disparity propagation requires $350\mu s$, and the post-processing takes $400\mu s$. The total algorithm runs in $2.03ms$ per frame, corresponding to 492Hz. From a memory perspective, our method requires to store only the left and right image. As comparison, the method presented in [16] requires around 1.5GB of GPU memory. This makes their computation memory bound, which slows down the overall speed of the system: as reported in [16] the most accurate configuration requires $2.5ms$ per frame, which is slightly more than the proposed method.

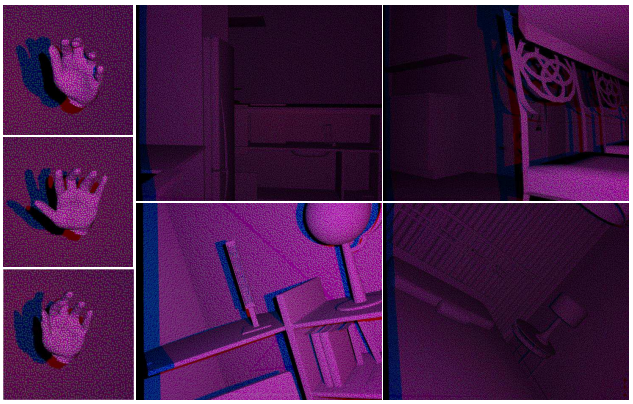

Figure 2. **Synthetic Data.** Representative examples of our synthetic dataset which comprises several indoor environments and a hand sequence.

4. Evaluation

In an effort to provide for rigorous and extensive experiments that allow to appreciate the different trade-offs present in depth algorithms, we developed and implemented a synthetic rendering pipeline and we designed a wide variety of experiments using both real and artificially generated data. We first show that our method performs very favorably compared to state-of-the-art techniques, not only in terms of average error but also using other useful metrics such as edge fattening and invalidation. Then, we qualitatively show that our algorithm also surpasses other approaches like Census and LSH. Note that for all the experiments, we set the parameters of our method to be $W = 11 \times 11$ and $b = 32$, the active range of depths is $[500, 4000]\ mm$.

4.1. Quantitative Evaluation

Our synthetic dataset is composed of 2500 training images and 900 test images (500 articulated hand images, 400 interior images from 5 different environments). We estimated the structured light reference pattern from a Kinect sensor using the technique presented in [36]. The extracted pattern is then defined as an ideal emitter in Blender and projected in hand-crafted indoor scenes. Similarly to our prototype, the virtual stereo system uses a baseline of $9cm$ between the virtual infrared cameras, and the emitter is placed between the two sensors. During rendering, the light fall-off is estimated based on the inverse-square law and (read + shot) noise [18] is added. For all the interior sequences, the 6 d.o.f. camera pose associated to each frame is randomly sampled from an uniform distribution. Fig. 2 contains representative examples of the rendered images. Note that for

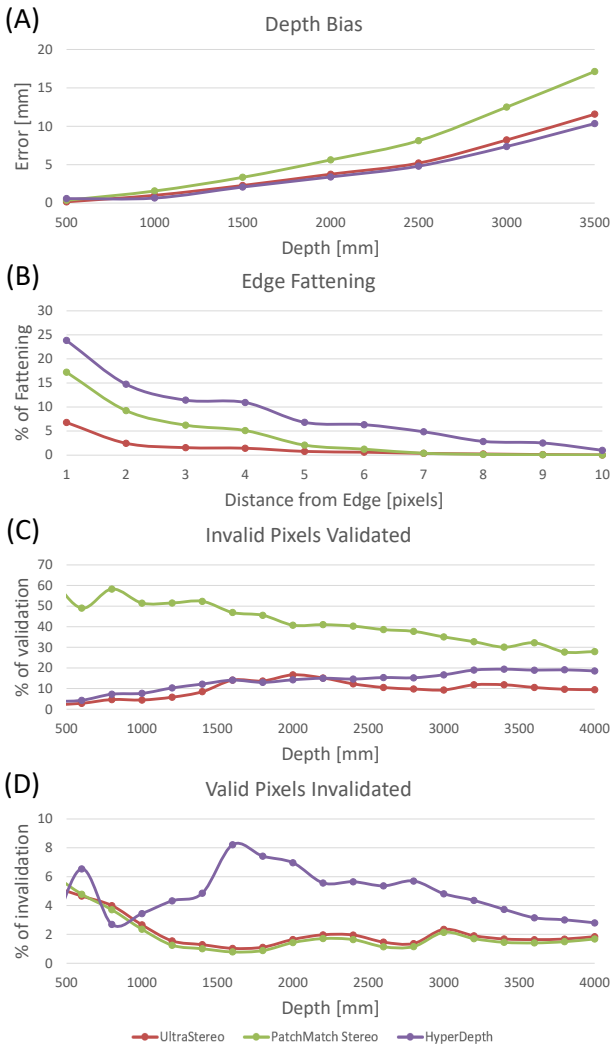

Figure 3. **Quantitative results on synthetic data.** (A) captures the average error that the different methods are making as a function of depth; lower is better. Note how UltraStereo provides estimates that are on par with HyperDepth. (B) quantifies the percentages of pixels that are fattened as a function of the distance to the edge of the foreground; lower is better. UltraStereo significantly outperforms the baselines on that metric as it provides at least two times less edge fattening compared to the competing baselines in the 1-6 pixels range. (C) and (D) respectively illustrate that UltraStereo offers the best trade-off in terms of validating invalid pixels (C; lower is better) and invalidation of valid pixels (D; lower is better).

each image, the red channel corresponds to regions that are visible from the other camera, the green channel contains the projected pattern and the blue channel encodes regions that are visible for the illuminator. Using these different channels, we can define in advance which pixels should be invalidated (either not visible by the emitter and/or one camera) and those which should be validated (visible by the emitter and both cameras).

Bias It is easy to prove [49] that the expected depth error ϵ follows $\epsilon = \frac{\Delta d Z^2}{bf}$, where Δd is the disparity error, Z the current depth, b the baseline and f the focal length of the system. The bias is defined as the average absolute depth error present in the whole test set. Fig. 3 reports results on synthetic data demonstrating that UltraStereo is better than PatchMatch Stereo and on par with HyperDepth in terms of bias. Achieving superior results compared to PatchMatch Stereo indicates that the learned sparse representation is effective and it is more robust against outliers and noise when we use a Kinect DOE which contains only 10% of bright spots. Indeed dark spots have a lower SNR which could adversely affect the matching cost, whereas the UltraStereo binary mapping is more robust. This is aligned with the findings in [27].

To quantify the bias in real data, we recorded images of a flat wall at multiple known distances: from $50cm$ up to $350cm$. We repeated this test for different algorithms and sensors, in particular we selected those technologies which use active illumination and those algorithms that are triangulation based such as: Kinect V1, RealSense R200, PatchMatch Stereo [7], HyperDepth [16] and UltraStereo. For PatchMatch Stereo and HyperDepth we reimplemented the methods following the original papers [7, 16], whereas for commercially available sensors we use the depthmaps generated by the cameras. In Fig. 4 we report the results, which are aligned to the ones showed in [16]. Notice that UltraStereo is again on par with HyperDepth and achieves state of the art results with very low quantization errors. Some methods degrade very quickly after $100cm$ due to higher noise (R200) or high quantization artifacts (Kinect V1). The R200 is an active stereo algorithm like ours, however it shows very high error after 100cm, this is probably due to the compromises made between accuracy and speed. Additional qualitative results of UltraStereo on real data are shown in Fig. 5 and Fig. 6. Notice how our method exhibits low quantization effects, low jitter and very complete depth maps.

Invalidation The percentage of invalidated pixels defines the number of pixels for which the final depth image won't contain any estimations. Ideally, depth would be estimated for all the pixels, but unfortunately occlusions, saturation of the infrared sensors and low SNR can make the disparity estimation quite ambiguous, often resulting in gross errors. To limit these errors, an invalidation pass is usually performed during the post-processing step. As described in Sec. 3.3, our invalidation scheme relies on pruning unlikely matches (large hamming distances) followed by a minimum region check. Figure 3 illustrates that our algorithm outperforms the baselines by invalidating less valid pixels, but also invalidating more invalid pixels. Similarly to the results obtained on synthetic data, HyperDepth [16] and PatchMatch Stereo

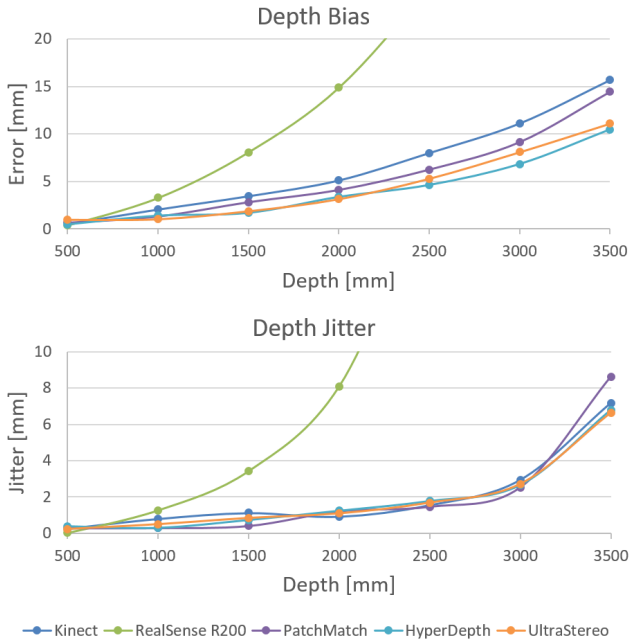

Figure 4. **Quantitative Results.** We computed the depth bias and jitter at multiple distances comparing UltraStereo with many state of the art technologies. Results show the accuracy of the proposed algorithm with respect to the competitors.

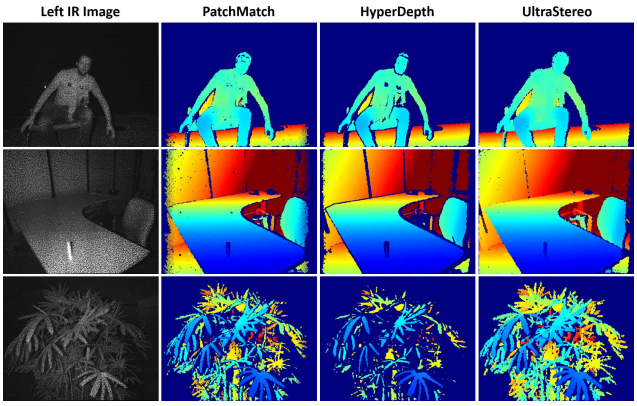

Figure 5. **Qualitative Evaluation.** Examples of depth-maps produced with UltraStereo and state of the art competitors. Notice how the method in [16] shows high invalidation in regions where the texture changes, the method in [7] is offline and still it fails delivering complete depth-maps especially in thin structures like the the plant.

[7] wrongly invalidate much more than UltraStereo on real data. This can be observed in Fig. 5. Notice how boundaries between two different textures are always invalidated by [16], whereas UltraStereo provides more complete depth maps. This especially clear for thin structures like plants (c.f. third row).

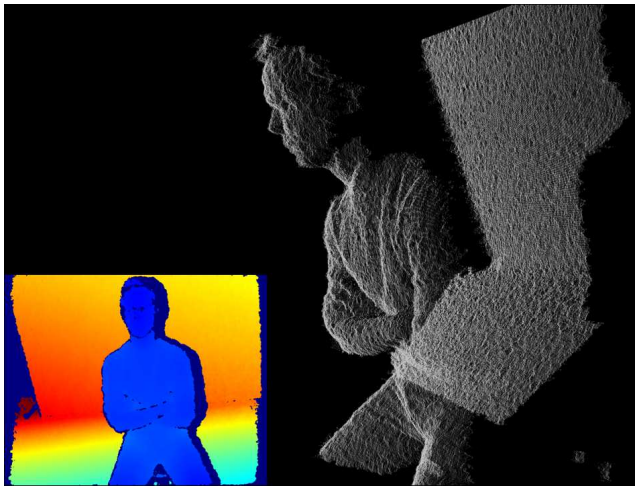

Figure 6. Example of pointcloud produced with our algorithm. Notice the absence of quantization and flying pixels.

Edge fattening Edge fattening is a common problem for most local methods and it is particularly visible for thin structures (e.g. fingers). To define the amount of edge fattening among the various baselines and our method, we synthesized images of an articulated hand in front of a wall, making it simple to define the edges of the hand in the depth image. Hands are very complex objects with high variability along the boundaries, making them a perfect candidate to evaluate edge fattening. To generate realistic sequences, we defined key hand poses and interpolation between them was performed to provide for a different hand pose every single frame. The hand has been placed at approximately $100cm$ from the sensor. Figure 3 depicts how our method outperforms the baselines and is less prone to fatten objects.

4.2. Binary representation

We provide here evidence for the quality of the information captured by the learned representation. We tested UltraStereo against Census [55] and more recent work that use Locality Sensitive Hashing (LSH) [25] which are other binary representations used for depth estimations. We trained our sparse hyperplanes by collecting one thousand images coming from a Kinect sensor. Fig. 7 shows qualitative results for the three methods. LSH and Census show more incomplete and noisier depth-maps compared to UltraStereo. Note that here Census uses 121bits (equal to the window size W), where LSH and UltraStereo only use 32 bits.

To further assess the performances of the different binary representations, we use ground-truth depth coming from our synthetic pipeline and perform an exhaustive discrete search over the disparity labels. We compute the error as percentage of pixels with less than 1 disparity distance from the ground-truth. LSH achieves an overall precision of 51%,

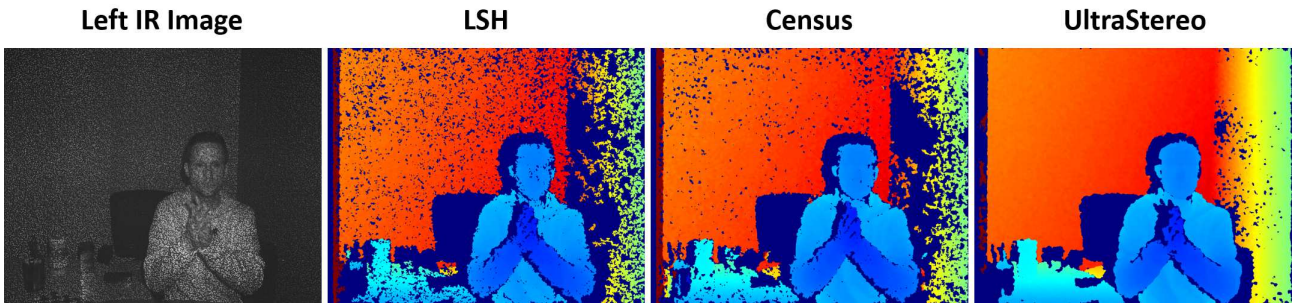

Figure 7. **Binary Representations.** We show qualitative results using LSH [25], Census [55] and our data-driven representation. Notice how UltraStereo provides a more complete and less noisy depthmap.

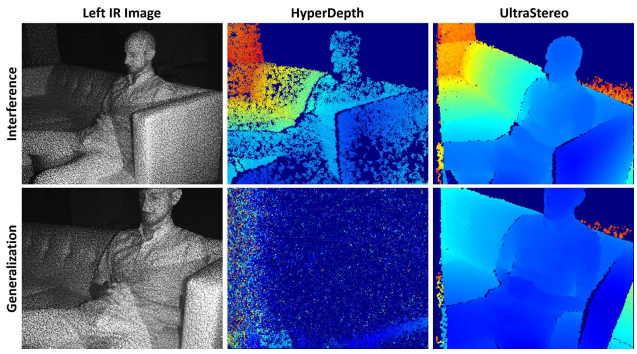

Figure 8. **Interference and Generalization.** UltraStereo does not suffer from interference (top row) and the learned binary mapping can generalize to different sensors (bottom row). On the contrary, the current state of the art [16] needs per-camera training and it is not robust to interference.

Census 61%, UltraStereo reaches an accuracy of 72%, showing again the effectiveness of the learned representation.

We also assessed the impact of the number of non-zero components k in the hyperplanes $\mathbf{Z}$. When using 8 non-zero elements we reached an accuracy of 74%, increasing this number to $k = 32$ we start to saturate with 76% accuracy, and finally using dense hyperplanes leads to a precision of 78%. This shows that this parameter is not crucial and it can be tuned according to the computational resources available.

4.3. Interference and Generalization

An important issue usually overlooked in the literature is the problem of interference caused by multiple sensors. Complex volumetric reconstruction systems such as [39] may require the use of multiple sensors with different viewpoints. Spatial Structured light systems like Kinect V1 suffer from interference, and the same limitation is present in [16]. In Fig. 8, we show how [16] is severely impacted when multiple active illuminators are present in the scene, where UltraStereo, implemented on our prototype active stereo setup, still produces high quality results.

Another important property of our method is its generalization capabilities. Since our learning algorithm is completely unsupervised, the resulting hyperplanes can be transfered from one sensor to another one without affecting precision. In Fig. 8, we show results when we use the learned binary hyperplanes applied to a different camera. Note how the HyperDepth algorithm [16] does not generalize at all and needs to be calibrated for each individual camera in order to provide high quality depth estimates.

5. Conclusion

In this paper we have presented UltraStereo, a breakthrough in the field of active stereo depth estimation. We showed how the stereo problem can be formulated to have a complexity that does not depend on the window size nor the size of the disparity space. To reach such a low complexity, we used an unsupervised machine learning technique that learns a set of sparse hyperplanes that projects image patches to a compact binary representation which preserves the discriminative information required for estimating robust correspondences. To perform the inference in the disparity space, we use a variant of the PatchMatch framework for which we described modifications to efficiently run on GPU. Through extensive experiments, we showed that UltraStereo outperforms the state of the art, not only in terms of speed but also in accuracy. Additionally UltraStereo does not suffer from per-camera calibrations nor interference problems which can prove problematic for some of the state-of-the-art techniques.

Acknowledgements

We thank the entire perceptiveIO team for continuous feedback and support regarding this work.

References

[1] Intel real sense r200. http://software.intel.com/en-us/realsense/r200camera. 1

[2] Zcam. http://en.wikipedia.org/wiki/ZCam. 1

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. PatchMatch: A randomized correspondence algorithm for

structural image editing. *ACM SIGGRAPH and Transaction On Graphics*, 2009. 3

[4] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: Patchmatch belief propagation for correspondence field estimation. *IJCV*, 110(1):2–13, 2014. 2

[5] A. Bhandari, A. Kadambi, R. Whyte, C. Barsi, M. Feigin, A. Dorrington, and R. Raskar. Resolving multi-path interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *CoRR*, 2014. 1

[6] M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *VISAPP*, 2008. 2

[7] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *BMVC*, 2011. 1, 2, 5, 6, 7

[8] A. Butler, S. I. andOtmar Hilliges, D. Molyneaux, S. Hodges, and D. Kim. Shake'n'sense: reducing interference for overlapping structured light depth cameras. In *Human Factors in Computing Systems*, 2012. 2

[9] A. Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013. 4

[10] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *SIGGRAPH*, 2016. 1

[11] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 3

[12] O. Elkhalili, O. Schrey, W. Ulfig, W. Brockherde, B. Hosticka, P. Mengel, , and L. Listl. A 64x8 pixel 3-d cmos time-of flight image sensor for car safety applications. In *European Solid-State Circuits Conference*, 2006. 1

[13] S. Fanello, U. Pattacini, I. Gori, V. Tikhanoff, M. Randazzo, A. Roncone, F. Odone, and G. Metta. 3d stereo estimation and fully automated learning of eye-hand coordination in humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, 2014. 1

[14] S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. Kang, and T. Paek. Learning to be a depth camera for close-range human capture and interaction. *ACM SIGGRAPH and Transaction On Graphics*, 2014. 3

[15] S. R. Fanello, C. Keskin, P. Kohli, S. Izadi, J. Shotton, A. Criminisi, U. Pattacini, and T. Paek. Filter forests for learning data-dependent convolutional kernels. In *CVPR*, 2014. 3

[16] S. R. Fanello, C. Rhemann, V. Tankovich, A. Kowdle, S. Orts Escolano, D. Kim, and S. Izadi. Hyperdepth: Learning depth from structured light without matching. In *CVPR*, 2016. 2, 3, 4, 5, 6, 7, 8

[17] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 2006. 2

[18] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on IP*, 2008. 5

[19] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns, Apr. 3 2012. US Patent 8,150,142. 1

[20] D. Freedman, E. Krupka, Y. Smolin, I. Leichter, and M. Schmidt. SRA: fast removal of general multipath for tof sensors. *ECCV*, 2014. 1

[21] J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011. 1

[22] E. Gordon and G. Bittan. 3d geometric modeling and motion capture using both single and dual imaging, 2012. 1

[23] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision (2nd ed.)*. Cambridge University Press, 2003. 3

[24] K. He, J. Sun, and X. Tang. Guided image ltering. In *Proc. ECCV*, 2010. 2

[25] P. Heise, B. Jensen, S. Klose, and A. Knoll. Fast dense stereo correspondences by binary locality sensitive hashing. In *ICRA*, 2015. 4, 7, 8

[26] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008. 2

[27] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *PAMI*, 2009. 3, 6

[28] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *ACM UIST*, 2011. 1

[29] D. Jimenez, D. Pizarro, M. Mazo, and S. Palazuelos. Modelling and correction of multipath interference in time of flight cameras. In *CVPR*, 2012. 1

[30] M. Ju and H. Kang. Constant time stereo matching. pages 13–17, 2009. 2

[31] K. Konolige. Projected texture stereo. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 148–155. IEEE, 2010. 1

[32] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011. 2

[33] Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu. Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs. In *ICCV*, 2015. 2

[34] J. Lu, K. Shi, D. Min, L. Lin, and M. Do. Cross-based local multipoint ltering. In *Proc. CVPR*, 2012. 2

[35] J. Lu, H. Yang, D. Min, and M. Do. Patchmatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *Proc. CVPR*, 2013. 2, 3, 5

[36] P. McIlroy, S. Izadi, and A. Fitzgibbon. Kinectrack: 3d pose estimation using a projected dense dot pattern. *IEEE Trans. Vis. Comput. Graph.*, 20(6):839–851, 2014. 3, 5

[37] N. Naik, A. Kadambi, C. Rhemann, S. Izadi, R. Raskar, and S. Kang. A light transport model for mitigating multipath interference in TOF sensors. *CVPR*, 2015. 1

[38] H. Nishihara. Prism: A practical real-time imaging stereo matcher mit ai memo no. 780. *Cambridge, Mass., USA*, 1984. 1

[39] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken,

J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi. Holoportation: Virtual 3d teleportation in real-time. In *UIST*, 2016. 8

[40] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–73, 2008. 2

[41] V. Pradeep, C. Rhemann, S. Izad, C. Zach, M. Bleyer, and S. Bathiche. Monofusion: Real-time 3d reconstruction of small scenes with a single web camera. 2013. 3, 4, 5

[42] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, pages 3017–3024, 2011. 2

[43] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8):2666–2680, 2010. 1

[44] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 31(5):824–840, 2009. 3

[45] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3), Apr. 2002. 2

[46] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *Proc. CVPR*, 2011. 3

[47] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. CVPR*, 2013. 3

[48] J. Stuhmer, S. Nowozin, A. Fitzgibbon, R. Szeliski, T. Perry, S. Acharya, D. Cremers, and J. Shotton. Model-based tracking at 300hz using raw time-of-flight observations. In *ICCV*, 2015. 1

[49] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., 2010. 6

[50] C. Tomasi. Bilateral filtering for gray and color images. In *Proc. ICCV*, 1998. 2

[51] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics (TOG)*, 34(5):154, 2015. 1

[52] P. Vuylsteke and A. Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(2):148–164, 1990. 1

[53] S. Wang, S. R. Fanello, C. Rhemann, S. Izadi, and P. Kohli. The global patch collider. *CVPR*, 2016. 3

[54] K.-J. Yoon and I.-S. Kweon. Locally adaptive support-weight approach for visual correspondence search. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 924–931. IEEE, 2005. 2

[55] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994. 4, 7, 8

[56] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. *arXiv preprint arXiv:1409.4326*, 2014. 3