Received August 22, 2016, accepted September 27, 2016, date of publication June 5, 2017, date of current version July 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2712154

Feature-Based Resource Allocation for Real-Time Stereo Disparity Estimation

ERIC HUNSBERGER[1], VICTOR REYES OSORIO[1], JEFF ORCHARD[2], AND BRYAN P. TRIPP[1], (Member, IEEE)

[1]Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[2]Cheriton School of Computer Science, University of Waterloo, ON N2L 3G1, Canada

Corresponding author: Bryan P. Tripp (bptripp@uwaterloo.ca)

This work was supported in part by a Mitacs and in part by CrossWing Inc.

ABSTRACT The most accurate stereo disparity algorithms take dozens or hundreds of seconds to process a single frame. This timescale is impractical for many applications. However, high accuracy is often not needed throughout the scene. Here, we investigate a ''foveation'' approach (in which some parts of an image are processed more intensively than others) in the context of modern stereo algorithms. We consider two scenarios: disparity estimation with a convolutional network in a robotic grasping context, and disparity estimation with a Markov random field in a navigation context. In each case, combining fast and slow methods in different parts of the scene improves frame rates while maintaining accuracy in the most task-relevant areas. We also demonstrate a simple and broadly applicable utility function for choosing foveal regions, which combines image and task information. Finally, we characterize the benefits of defining multiple individually placed small foveae per image, rather than a single large fovea. We find little benefit, supporting the use of hardware foveae of fixed size and shape. More generally, our results reaffirm that foveation is a practical way to combine speed with task-relevant accuracy. Foveae are present in the most complex biological vision systems, suggesting that they may become more important in artificial vision systems, as these systems become more complex.

INDEX TERMS Stereo vision, disparity, convolutional neural networks, Markov random fields, belief propagation, fovea, navigation, grasping.

I. INTRODUCTION

Many stereo vision algorithms [18], [45] take so long to run that they are impractical for some of their main applications, including robotics and autonomous driving. It has long been recognized (e.g. [3], [16], [30], [44]) that an effective way to work around limits of computing power is to use a ''fovea''. A fovea (analogous to a fovea in biological vision) is a small image region that is processed more intensively than the rest of the image, often at higher resolution. It is typically oriented to an important or useful feature. Rapid growth in computing power has not diminished the need for foveae, due to parallel increases in camera resolution and algorithmic complexity.

A. FOVEATION IN PRIMATE VISION

Foveae are central to the organization of primate visual systems. In primate vision, a large fraction of the visual cortex is dedicated to the central 1° of the visual field, with progressively less cortical area used for more peripheral parts of the visual field [7]. The eyes move frequently, typically jumping to a new target several times per second, to sense and analyze detail from different parts of the scene in series. Vision uses about half of the primate brain, so this strategy is essential for having good vision without an extremely large head. Artificial visual systems are much less power-efficient than primate vision systems, suggesting that foveae may be even more important for advanced autonomous robots than for primates.

Primates decide where to move the fovea through sophisticated systems for directing visual attention and eye movements (reviewed by [19], [27], [38]). Computational models of these systems, and applications to computer vision and robotics, are reviewed by [4], [12], [24], and [42]. As discussed in these reviews, human eye movements are occasionally directed to salient visual features (such as the onset of motion), but in most situations are overwhelmingly determined by task demands (see also [46]). Examples of

task-dependent targets include the next word while reading [36], the edge of an obstacle while navigating [37], an object a person wants to pick up [23], etc. The visual target can even be a completely featureless region. For example, people often glance at a spot where they intend to put something, even though the spot may be visually indistinct [40]. However, eye movements are very often determined by a combination of bottom-up and top-down factors.

A simple example of bottom-up/top-down interaction occurs in visual search tasks. Viewing an image of many small shapes, humans can rapidly find shapes with a distinctive feature of their choice (e.g. yellow objects, or horizontal objects, etc.) Interestingly, visual search for more complex conjunctions of features (e.g. horizontal yellow shapes) is slower and less automatic [41].

B. SCOPE OF THE PRESENT STUDY

In this study, we experiment with foveation to allow stereo disparity estimation at high frame rates, with high accuracy in task-relevant parts of the scene. Past work on foveated stereo (e.g. [2], [17], [25]) has mainly considered foveated hardware, but we consider software approaches that operate on different image regions in different levels of detail (see [15] for related work). We explore this approach in two contexts, focusing on dense disparity estimation due to its importance in robotics. First, we consider stereo images of small items on a table. This scenario is relevant to robotic grasping. Disparity is an important cue for three-dimensional object shape, but good disparity estimates are computationally expensive, and many state-of-the-art grasping systems (e.g. [28], [29]) fall back on monocular color images. For this application, we used a recent convolutional neural network (CNN) [45] that has fast and slow variants.

We also explored the same general approach in a second context, with navigation-relevant results on KITTI 2012 benchmark data. For this scenario, we adapted a multi-scale loopy belief propagation method on a Markov random field [9], so that it calculated the finest scales only in the fovea. In each of these cases, we substantially improved frame rates while maintaining accuracy in the most important image regions.

Taking inspiration from primate visual search, we defined ''important regions'' using simple, task-relevant visual features that can be computed quickly. Task-relevance is inherently task-specific [8]. However, for concreteness, we propose a class of features that we believe is fairly widely applicable in disparity estimation. The features are based on protrusion from a background. For example, in a grasping context, we modeled a support surface as a plane, and oriented the fovea to things above the plane. In a driving context, the background was the time-averaged disparity, which takes on the shape of the road and surrounding buildings. This directs the fovea to regions in which surfaces seem to be closer than normal for their part of the visual field, and which may therefore be important in short-term decisions. (See [31], [34] for related uses of depth as attention cues.)

Given a fixed processing budget, what is the best strategy for stereo vision? Our hypothesis is that foveation can improve stereo vision, over the alternative of uniform processing over the entire field of view. As such, we focus our experiments on how the judicious allocation of processing to the fovea can improve the accuracy of goal-relevant stereo information.

II. METHODS

We propose a disparity estimation approach to reduce computation time while maintaining accuracy in regions of interest. To achieve this, we use fast and less accurate disparity estimation across most of the image, and only apply a slower, more accurate disparity estimation in the regions of interest. These regions of interest are task-specific; the number of regions, their size, and where they are located over time, all vary depending on the task. Due to the analogue with the human eye, where high-resolution information is only collected from the fovea (the central region of the retina) and some parts of the brain predominantly process foveal information, we call our region of interest the fovea (or foveae in the case when we have more than one such region at a time). See [15] for a closely related approach.

Our method can be used to achieve a balance of accuracy and speed between fast and slow variants of the same disparity estimation algorithm, or between different disparity estimation algorithms where one is faster but less accurate than the other. Here, we apply our method to two scenarios: a navigation scenario, and a grasping scenario. The navigation scenario uses the KITTI dataset [14], and we apply fast and slow variants of a multi-scale belief propagation (BP) algorithm [9]. The grasping scenario uses our own data collected from various objects on a tabletop, meant to simulate a situation typically encountered when controlling a robot arm to grasp an object. For that scenario, we apply fast and slow variants of a deep-learning based disparity estimator [45].

Figure 1 motivates our approach in the navigation scenario. Navigation, particularly navigation with fast moving vehicles such as cars, requires depth maps that are updated many times a second to be able to avoid obstacles in a quickly changing scene (for example, a child running onto the road). Therefore, there is a limited amount of time to process each frame. As the figure shows, running the iterative BP algorithm on a higher-resolution frame takes almost 0.5s, even with a single iteration. Running the algorithm on a lower-resolution frame is much faster, but the accuracy is limited by the resolution, even with high numbers of iterations. By running the algorithm at low resolution for most of the image, and high resolution only in the task-important parts of the image, we move our plot closer to the bottom-left area of the figure, i.e. faster runtimes than the slow variant of the algorithm but better accuracy than the fast variant.

We begin by presenting the general framework of our method (Section II-A), and the details of how we estimate task-specific cost and choose the fovea position both generally and for the two example tasks (Section II-B). We then go

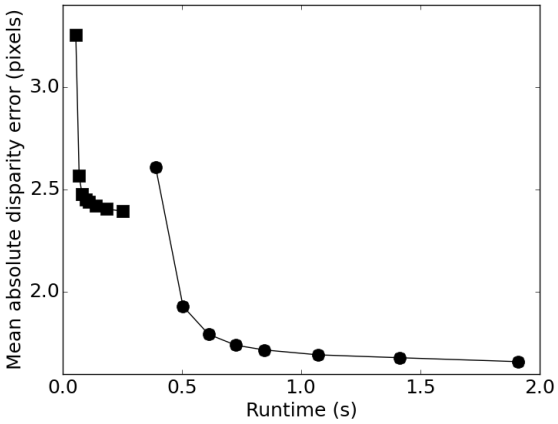

FIGURE 1. Performance of multi-scale belief propagation on example data, illustrating the motivation for our approach. Disparity estimation error is plotted vs. runtime. The right curve (circles) were obtained with a high-resolution stereo image pair, and the left curve (squares; higher error) with lower-resoluton versions of the same images. Within each curve, the runtime varies with numbers of iterations at each scale (1, 2, 3, 4, 5, 7, 10, and 15 iterations). Given a time budget of, for example, 0.25s/frame, it would not be possible to process these images at high resolution. However, if certain areas in the images were of greater practical interest, then results that are nearly as useful might be achieved by processing just those areas at high resolution. The data are 50 frames from the KITTI dataset [14], downsampled by a factor of two (high resolution) and four (low resolution) in each dimension.

into more detail on the grasping (Section II-C) and navigation (Section II-D) scenarios, including details of the disparity estimation algorithms, and specifics on the task-specific cost and fovea selection as it pertains to each scenario.

A. FRAMEWORK

Figure 2 is a schematic of the general structure of our approach. The input to the system is a stereo image pair. First, the fast stereo method is applied to the input, creating a ''rough'' (low accuracy) disparity image. Applying a task-specific cost function, the rough disparity image generates an importance image, indicating which areas of the image are most in need of more accurate disparity estimation. One or more fovea locations are then selected to cover these important areas, and the more accurate (slower) disparity method is applied in these areas. These improved disparity estimates are then pasted into the rough disparity image, resulting in a global disparity estimate with improved accuracy in the foveated regions.

B. FOVEA PLACEMENT

The fovea position is chosen in an attempt to minimize task-specific cost functions of the form,

$$C_{x,y} = \sum_{x,y} w_{x,y} \min\left(\left|d_{x,y} - d^*_{x,y}\right|, C_{\text{sat}}\right), \tag{1}$$

where $d_{x,y}$ is the estimated disparity at pixel (x, y), $d^*_{x,y}$ is ground-truth disparity, $w_{x,y}$ is a per-pixel weight map, and pixel-wise errors saturate at C_{sat}. We do not evaluate $C_{x,y}$ at runtime, rather we reduce this cost indirectly by using more accurate stereo estimation methods in high-weight regions.

The weights $w_{x,y}$ are task-specific, and may be static, or vary over time based on a preliminary disparity estimate $d^0_{x,y}$. In our examples, this preliminary disparity estimate comes from the fast variant of the chosen disparity estimation algorithm. (We have also obtained these estimates by remapping depth estimates from previous frames in video).

In navigation, one possible weighting scheme would be to emphasize regions in the direction of travel, because the risk of collisions with obstacles is greatest in this direction. Another would be to emphasize regions in which surfaces are relatively close (i.e. disparity is high). However, both of these schemes emphasize the close region of the ground, which is always present and often clear of obstacles. Instead, we define

$$w^d_{x,y} = \max(d^0_{x,y} - \bar{d}_{x,y} - d_{\text{th}}, 0), \tag{2}$$

where $\bar{d}_{x,y}$ is the mean of $d_{x,y}$ over time, and d_{th} is a small threshold. This approach emphasizes parts of the image in which surfaces are closer than usual. We used $w_{x,y} = w^d_{x,y} w^s_{x,y}$, where $w^s_{x,y}$ is a static weight template that is higher in the horizontal centre of the image (the direction of travel) than the edges, and also lower at the top of the image, which typically contains sky. Figure 3 shows an example frame from the KITTI dataset [14] in which a cyclist is strongly emphasized by this method. To avoid startup transients, we calculated $\bar{d}_{x,y}$ over full benchmark videos before processing. This is unrealistic for deployment on a robot, but a pixel-wise recursive low-pass filter would have a similar effect with minimal computational cost.

We chose $d_{\text{th}} = 1$ to ensure that the method focuses on regions that are substantially different than expected. If it is set to zero, the method may choose regions where the preliminary disparity estimate is very slightly larger than expected over a large region. For example, if the preliminary estimate predicts that all parts of the road surface are one pixel closer than expected, it may choose to put the fovea on that location, rather than a location where a smaller object (e.g. a pedestrian) is significantly closer than expected. By choosing $d_{\text{th}} > 0$, we force the algorithm to ignore small, widespread disparity differences, in favour of larger disparity differences in local regions. This can also help draw the fovea to areas where the disparity estimator has predicted an erroneous ''phantom'' object at high disparity. We do not have a formal rationale for setting $d_{\text{th}} = 1$, but heuristically, it should be greater than most of the noise and smaller than the relevant obstacles.

The cost, shown in equation (1), depends on the ground-truth disparity $d^*_{x,y}$, which is obviously not available to the algorithm in the real-world. Rather, we must have a model of the types of errors made by the disparity estimation algorithm. In this work, we assume that errors are equally likely in all image regions, and are equal in magnitude at all disparities. This reduces the task of fovea placement to choosing a region that covers the greatest total $w_{x,y}$, calculated efficiently using an integral image. This approach minimizes the cost if the error is statistically uniform over

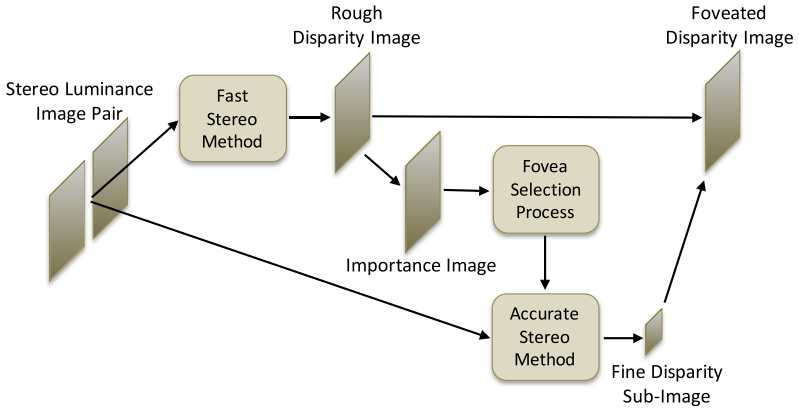

FIGURE 2. General structure of our approach. The shaded boxes correspond to single-channel images (luminance, disparity, and importance).

the image and lower for the slow method (fovea) than the fast method. A more accurate model of the disparity estimation errors could account for effects like larger errors near the image edges (due to fewer nearby points to provide context), or larger absolute errors at higher disparities. Such a disparity model could provide expected disparity errors at each point, $\mathbf{E}\left[\min\left(\left|d_{x,y}-d^*_{x,y}\right|, C_{\text{sat}}\right)\right]$. In that case, we would choose the fovea position such that it covered the region with the highest weighted difference in expected cost between the slow and fast methods.

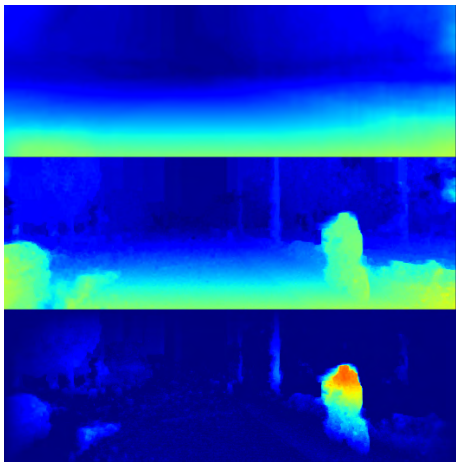

FIGURE 3. Weighting by disparity in excess of the average. Top: Average disparity $\bar{d}_{x,y}$ estimated over a long image sequence. Centre: Disparity $d_{x,y}$ estimated in a single frame. Bottom: The weight $w = w^s w^d$, where w^s is a static weight template that emphasizes the direction of travel, and $w^d_{x,y} = \max(d_{x,y} - \bar{d}_{x,y}, 0)$.

Contrary to the example of Figure 3, the high-weight pixels are not always co-located. Therefore we also experimented with dividing the fovea into multiple sub-foveae with the same total number of foveal pixels. Whereas optimal placement of a single fovea with given height and width simply corresponds to finding the maximum in a two-dimensional image, it is not practical to find the global optimal placement of multiple foveae. Instead, we used a greedy approach in which we placed a single sub-fovea optimally, set the weight within it to zero, then placed another single sub-fovea optimally, etc. We calculated the total remaining weight with different numbers of sub-foveae, and used the number of sub-foveae with the lowest weight. These computations were all performed with images that were downsampled several times, to reduce the run time.

For grasping, we wish to have the best depth estimates of objects that are potential grasp targets, and ignore objects that are part of the background. In our grasping scenario, the main background object is the table. After determining the preliminary depth estimate, $d^0_{x,y}$, using the fast variant of the depth estimation algorithm, we used RANSAC [11] to estimate the plane of the table. We then defined the background distance $\bar{d}_{x,y}$ as the expected depth of the table at each pixel, and employed the same weighting scheme as in navigation (equation (2)), in this case emphasizing points that were nearer than the table. For simplicity, we set the static weight $w^s_{x,y}$ uniformly across the image. Other choices of $w^s_{x,y}$ are possible, e.g. higher weighting of regions closer to the camera, or closer to the current location of the gripper.

C. GRASPING SCENARIO

Grasping is usually not highly time-sensitive, but it requires accurate shape estimates. In this scenario, we used a pair of relatively slow and accurate stereo methods based on convolutional neural networks (CNNs).

1) CNN-BASED DISPARITY ESTIMATION

Zbontar *et al.* [45] trained CNNs to calculate the matching cost across two rectified images. They developed two kinds of networks: a faster (less accurate) network and slower (more accurate) network. Both of the networks attempt to find the similarity between two image patches at a particular disparity. At each disparity level to be tested, an image patch is grabbed from both the left and right image. One of those patches

is transposed some number of pixels to the side equal to the current disparity level, and the similarity is computed. The two networks differ in how they compute the similarity between two image patches. The fast network computes the dot product between feature vectors produced from the left and right images, while the slow network includes additional layers that are trained to produce a better similarity metric than the simple dot product. This results in a 3D tensor of dimensions $disparitylevel \times height \times width$, where each element represents the probability that particular pixel should be classified at that particular disparity. The number of disparity levels to be checked highly impacts the running time of the neural nets. On 1200×1000 pixels images with 350 disparity levels, the fast method takes about 5.67 seconds to run, while the slow method takes 467 seconds, even though both CNNs are GPU accelerated. We chose to check for the disparties of up to 350 pixels since the objects in our scene were fairly close to our camera, given its baseline of 120mm.

The fast method can have substantial artifacts that can extend the appearance of an object well beyond its physical boundaries, which would impair grasp planning. The slow method does not suffer from these artifacts, but it has an excessive runtime for grasp planning.

2) METHOD

We were not aware of standard grasping-related stereo datasets, so we collected new images with a ZED stereo camera from StereoLabs. The grasping-related scenario involves household objects on a table. We assume the support surface can be approximated as a 2D plane.

The fast method was used over the entire image to arrive at an initial estimate for $d^0_{x,y}$. These pixel values were passed into a linear regressor that used RANSAC to find a suitable plane equation to model the table. This plane defines $\bar{d}_{x,y}$, the background disparity at each pixel, from which we calculated $w^d_{x,y}$ (equation (2)). We then found objects of interest as follows: the image was segmented into candidate objects (using SciPy's *label* function), ignoring small groups of pixels (typically including outliers that were actually part of the table). Finally the object with the highest total weight was chosen. Once it was located, we extracted left and right object images from the full images. Before passing them to the slow network, we used $\bar{d}_{x,y}$ to estimate the disparity that pixel (x, y) would have if the object were not present. This was used to set a new lower bound on the disparity, further reducing the number of disparity levels that had to be checked. The object images were then passed through the slow network. The final disparity map consisted of the slow estimate of the object, and the fast estimate of the rest of the scene. Figure 4 shows some of the stages in this process.

D. NAVIGATION SCENARIO

For the navigation scenario, we applied multi-scale loopy belief propagation (BP) to the KITTI driving dataset. We had two reasons for using a different stereo method here than in the grasping scenario. First, we wanted to test our general approach with multiple stereo algorithms. Second, navigation is typically more time-sensitive than grasping, so we wanted to use a method that runs at a more realistic frame rate in this case.

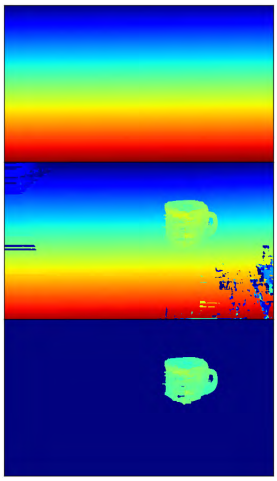

FIGURE 4. Weighting by disparity in excess of the background. Top: Background disparity $\bar{d}_{x,y}$ estimated using RANSAC to model the table. Centre: Disparity $d_{x,y}$ measured using the fast network. Bottom: The weight $w^d_{x,y} = \max(d_{x,y} - \bar{d}_{x,y} - d_{\text{th}}, 0)$.

1) DATASET

The methods were tested on data from the 2012 KITTI Vision Benchmark Suite [14]. This dataset includes high-resolution rectified stereo images (1242×375 pixels) taken at ten frames/s from the roof of a car during city driving. Rectified images and odometry are available for a number of long sequences. Ground truth depth (but not odometry) is available for single frames within a number of short 20-frame sequences. Ground truth is from a LIDAR sensor. The ground-truth points are fairly dense, because points were combined from several sweeps of the sensor using a semi-manual registration process. The scenes include static obstacles such as buildings and parked cars, and moving obstacles such as cars, pedestrians, and people on bicycles.

2) MARKOV RANDOM FIELDS AND LOOPY BELIEF PROPAGATION

In order to explain our combination of fast and slow methods in this scenario, we must first briefly review Markov random fields (MRFs) and belief propagation (BP).

Pixel-by-pixel stereo correspondences are frequently ambiguous, so disparity estimation is improved by considering spatial correlations. The full correlation matrix is unmanageable, because disparity images of practical interest have tens of thousands of pixels or more. A successful approach [39] has been to model correlations in disparity images with Markov random fields. In a Markov random field (MRF), each region is independent of the rest of the field, conditional on values in the region's boundary. That is,

$p(x_i|x_b, x_o) = p(x_i|x_b)$, where x_i is the part of the field inside the boundary, x_b is the part that comprises the boundary, and x_o is the part outside the boundary [10]. There are various ways to estimate the maximum *a posteriori* disparity from the MRF.

One such method is belief propagation (BP), a general algorithm for inference on graphical models (including Bayesian networks as well as MRFs). BP provides exact solutions on trees. Loops in the statistical relationships of Markov random fields prevent exact inference, but "loopy" BP typically produces good approximations after running for a few iterations [33].

The starting point for our work in the navigation scenario is a BP implementation by Felzenszwalb and Huttenlocher [9] that has several optimizations, which together accelerate the algorithm by several orders of magnitude. One of these optimizations is a multi-scale method that reduces the number of iterations needed to propagate information to distant parts of the image. Our modification consists simply of executing finer scales (which take up most of the computation time) only in sub-images rather than over the whole image.

The BP implementation of [9] results in a labelling f, which assigns a label $f_p \in \mathcal{L}$ to pixel p, where $\mathcal{L}$ consists of all possible disparities. The algorithm minimizes an energy function

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{p,q \in \mathcal{N}} V(f_p - f_q) \tag{3}$$

where D_p (the "data cost") is the cost of labelling pixel p as f_p, and $V(f_p - f_q)$ (the "discontinuity cost") is an additional cost of labelling neighboring pixels p and q as f_p and f_q. Because the disparity of neighboring pixels is strongly correlated, a larger discontinuity cost is assigned for larger differences between neighbors (here the immediately adjacent pixels, horizontally and vertically.) Specifically,

$$V(f_p - f_q) = \min\left\{\left|f_p - f_q\right|, V_{\max}\right\}, \tag{4}$$

where $V_{\max}$ is a saturation value that limits the cost of large discontinuities. The saturation value is used because while most discontinuities are expected to be small, some (at object boundaries) are expected to be larger, with no particular expectation about how much larger. Minimizing $E(f)$ corresponds to maximum *a posteriori* estimation in a probabilistic context [9].

BP involves iterative computation and exchange of messages between pixels. In iteration t, the message $m^t_{p\to q}(f_q)$ from pixel p to pixel q for disparity f_q is [9]

$$m^t_{p\to q}(f_q) = \min_{f_p}\left(V(f_p - f_q) + D_p(f_p) + \sum_{s \in \mathcal{N}(p)\backslash q} m^{t-1}_{s\to p}(f_p)\right) \tag{5}$$

where $\mathcal{N}(p)\backslash q$ consists of the neighbors of p other than q. After T iterations, final labels f^*_p are assigned as

$$f^*_p = \arg\min_{f_p}\left(D(f_p) + \sum_{q \in \mathcal{N}(p)} m^T_{q\to p}(f_p)\right). \tag{6}$$

In stereo estimation, the first step in this process is to calculate a data cost volume in terms of image coordinates $p = (x, y)$ (where x and y are the horizontal and vertical image coordinates, respectively), and disparity (in pixels) d. The data cost volume is

$$D_{x,y,d} = \min\left\{\left|I^l_{x,y} - I^r_{x-d,y}\right|, D_{\max}\right\}, \tag{7}$$

where I^l and I^r are luminance of the left and right images, and $D_{\max}$ is a saturation value.

Prior to calculating the data cost, we processed the images with a Laplacian filter to emphasize edges.

3) LOCAL MULTI-SCALE BELIEF PROPAGATION

We expand on [9] by using different numbers of scales in different parts of the image. The number of operations in a single scale of [9] is order $\mathcal{O}(nl)$, where n is the number of pixels and l is the number of disparities. The majority of the computational cost is incurred at the finest scale, which has four times as many pixels as the next-finest scale. Our system processes the finest one or two scales only in the foveal region. The resulting depth estimate is similar to that of full multi-scale BP in the fovea, but (depending on fovea size) it can run nearly as fast as if the finest scales are omitted.

Importantly, messages from coarser scales are used to initialize messages in finer, foveal scales. This allows information from coarse scales to propagate from other image regions across the fovea, which makes depth estimates in the fovea continuous with those in the surrounding areas, and also makes them similar to those of full-resolution BP. (They differ somewhat around the fovea border, due to propagation across the border in the fine scale of full BP.) This fits into the general structure provided in Figure 2, with the important addition that the accurate stereo method is initialized with values (i.e. messages) from the fast stereo method, rather than being an independent calculation at specific regions (as in the grasping scenario).

TABLE 1. System parameters and values used in performance tests.

Laplacian filter width	3 pixels
Data-cost weight	.014
Data-cost saturation value (C_{max})	112
Discontinuity cost saturation value	12.1
BP iterations per scale	3

The system parameters and their values for different tests are listed in Table 1. Parameter optimization was performed using the package Hyperopt [1].

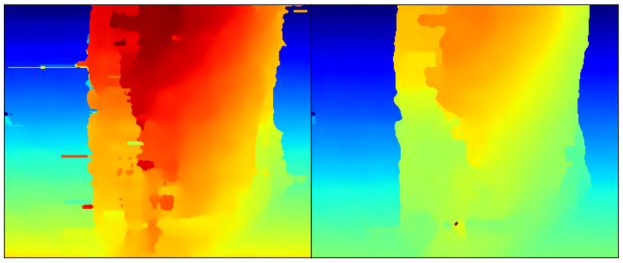

FIGURE 5. Example disparity images of a graspable object produced with CNNs. Left: Processed with the fast network, many artifacts. Right: Processed with the slow network.

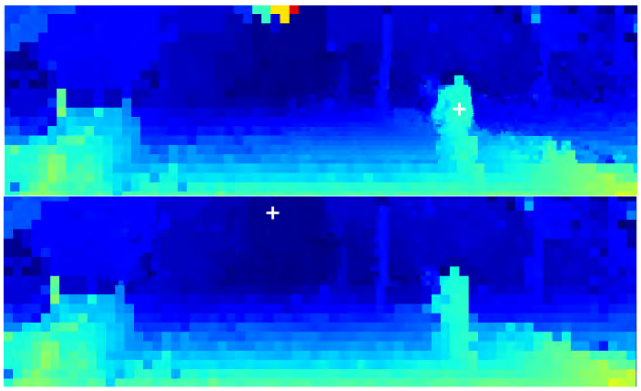

FIGURE 6. Examples of stereo disparity estimation with foveal multi-scale belief propagation. The white + marks indicate fovea centers. These examples were processed at multiple resolutions, with equal computational demands at each of these resolutions (e.g. a parafoveal region twice the size of the fovea had half the resolution). In the top panel, the fovea resolves a cyclist at high resolution. There is a large artifact in the top-center of the image. In the bottom panel (same frame), the fovea is instead centered on this artifact, and corrects it.

III. RESULTS

We evaluated the effectiveness of our approach in both the grasping and navigation scenarios. In this section, we first present a number of motivating examples that illustrate how our method helps improve disparity estimates in a number of specific cases (Section III-A). We then present qualitative results (as ground truth is not available) for the grasping scenario (Section III-B). For the navigation scenario, ground-truth is available, so we present a more detailed quantitative analysis of the accuracy and runtime of our method (Section III-C). Finally, we examine the effects of splitting up the foveal area into multiple independent regions (Section III-D).

A. MOTIVATING EXAMPLES

Figure 5 shows two disparity images obtained from the same inputs. The left image was processed using the fast network. There are many artifacts that could be interpreted as protrusions from the object. These protrusions would affect grasping decisions on an autonomous robot. The right image was processed with the slow network. Protrusion-like artifacts were reduced substantially, and gradients were smoother and more distinct. This representation would be more suitable for grasping applications, but the run time of the slow method is at least an order of magnitude longer than that of the fast method.

Figure 6 shows two disparity estimates from a single frame in the navigation scenario, with the fovea in different places (the centers are marked with white + signs). These examples were processed with multiple levels of resolution, with a small high-resolution region at the fovea center and multiple levels of decreasing resolution with greater distance from the fovea center, analogous to the human eye and visual cortex. The total runtime of BP in these examples is only a few times as great as BP at the lowest resolution. In the top frame, the fovea clearly resolves the cyclist, while in the bottom frame the fovea correctly interprets a low-disparity region in which the top frame has an artifact.

FIGURE 7. Example foveation sequence. The fovea boundary is shown as a white square, and the frames are ordered from top to bottom.

Figure 7 shows a sequence of foveations generated by our weight-covering method (Section II-B) while driving between parked cars. The sequence is from top to bottom. The cars on the right are foveated in turn. The cars on the left

have slightly lower weights than those on the right, and are never foveated.

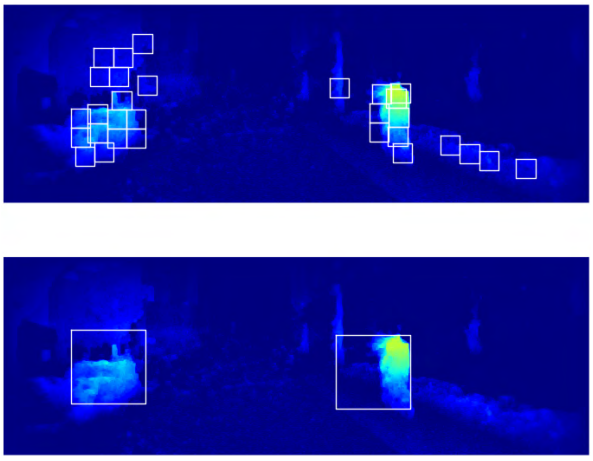

FIGURE 8. An example of greedy-optimal placement of 30 sub-foveae (top) and 2 sub-foveae (bottom), with the same total area, overlaid with the corresponding weight image $w = w^s w^d$. In the weight image, darker blue corresponds to low weight. In this example, two sub-foveae provided the best coverage of weighted pixels, compared to options of between 1 and 30 sub-foveae.

Figure 8 shows examples of breaking the foveal area into multiple sub-foveae. In the top image, the fovea has been broken into 30 sub-foveae that are placed individually. A greater number of smaller foveae can cover irregular shapes more precisely. However, because our greedy method of sub-fovea placement is not globally optimal, more sub-foveae are not guaranteed to cover a larger cost. In this example, numbers of sub-foveae from 1-30 were compared (with the same total foveal pixels in each case), and two sub-foveae provided the best coverage (bottom image).

B. GRASPING SCENARIO RESULTS

The following results were obtained using an NVIDIA Titan X to run the CNNs trained by Zbontar *et al.* [45] and using the ZED stereo camera from StereoLabs. StereoLabs provides their own stereo algorithms, but we have not used those in this analysis. Figure 9 shows estimated disparity for four different scenarios. The foveated method was able to remove a number of artifacts in the disparity estimate, as compared with the fast method, without the full computational cost of the slow method. In the center-left column, the bottle has a bar-like artifact on left hand side and uneven edges when using the fast method. Both the slow and foveated methods removed the artifact and smoothed out the edges. However, the slow method took more than five times as long to achieve those results (see times in Figure 9). In the last column we see a more complicated grasping scenario. In this case it is possible to devote resources to both objects in the image. However, it is more likely that only one of the object is currently relevant. In this scenario we added a weight term that corresponded to the distance from the point to a (phantom) gripper. The bottle was then chosen as the object to foveate in order to minimize $C_{x,y}$. Resources were then allocated there even though it is farther away from the camera.

Improvements in runtime were even greater in the other two examples: the foveated cup and can were roughly ten times faster than using the slow method. Figure 10 shows a close up of the cup from Figure 9. In this case, accuracy improved in the body and lip of the cup. The fast method shows an uneven surface, with some holes and discontinuities, but the slow and foveated methods estimate a smooth, continuous surface.

In summary, using the foveated method allowed greater accuracy in the task-relevant sections of the image, while reducing run time. A limitation is that the speed improvements depend on the area subtended by the objects of interest. Thus in the limit as a large or close object fills the image, the foveated time will approach the slow time.

C. NAVIGATION SCENARIO RESULTS

Because ground truth is available for the KITTI data, we examined accuracy in more detail for this scenario. We tested the dependence of both runtime and accuracy on the total foveal area. In general, our expectations were that runtime would increase almost linearly with the foveal area, and that accuracy would improve with increasing foveal area. For example, with zero foveal area, speed and accuracy should be roughly equal to that of full BP with down-sampled images. On the other hand, setting the fovea area equal to the image area should result in speed and accuracy equal to that of full multi-scale BP. We further anticipated that weighted error would drop sharply with increasing foveal area, since we tried to align the fovea with the highest-weight regions. For these experiments, we used the best of 1-5 sub-foveae (see Section III-D).

Figure 11 shows an example of results that are in keeping with these expectations. In order to achieve a range of runtimes consistent with the frame rate of the KITTI data (10Hz), the full images (roughly 1242 $\times$ 375 pixels with minor variations) were downsampled by a factor of two in each dimension, the finest scale of belief propagation was omitted in the fovea, and the two finest scales were omitted in the periphery.

The horizontal axis is the total foveal area as a fraction of image area. When we held the fovea at the centre of the image (blue line), error dropped roughly linearly as a function of foveal area. In contrast, weighted error (centre panel) dropped more steeply when a single fovea was aligned with high-weight regions, quantifying the benefit of moving the fovea. It dropped more steeply still when several sub-foveae were allowed. The run time with zero foveal area includes both the calculation of data cost, and belief propagation at the coarser scales. To reduce the time needed to calculate the data cost in the periphery, the cost was calculated on subsampled pixels, at the resolution of the peripheral belief propagation.

Performance was generally poorer in high-weight regions, i.e. around nearby obstacles, than in the more

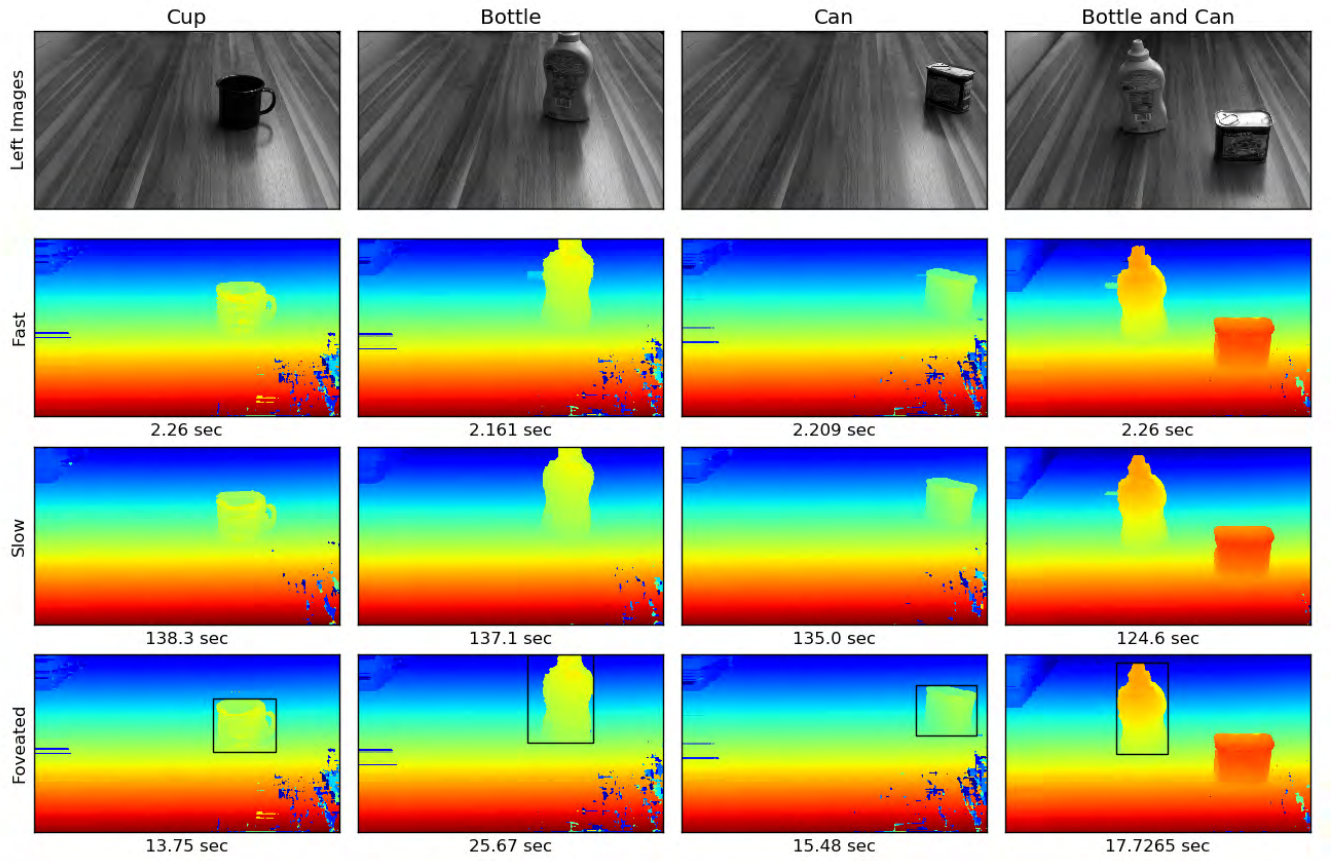

FIGURE 9. Four example images from the grasping scenario. Each column presents a different scenario for grasping, with the rows showing the left image from the stereo pair, the disparity image from the fast and slow disparity estimators, and the disparity image from our foveal method. The foveal method is able to localize the potential target object and place the fovea there for an improved estimate of the object shape.

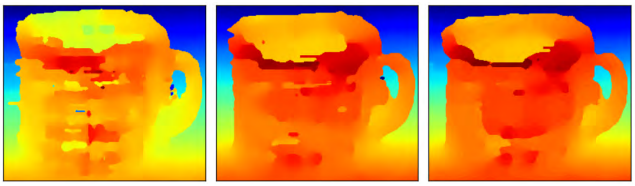

FIGURE 10. Comparison of the different disparity measurements in the area around a cup. The foveated method (right) provides a similar disparity estimate to the slow method (centre), and a much better estimate than the fast method (left).

smoothly-changing background. Performance in these regions was also quite variable from frame to frame (not shown).

The run times (bottom panel) are mean total times per frame (for calculation of the data cost, belief propagation, and inference) on a MacBook Pro with a 2.5GHz Intel Core i7 processor. Additional time for single fovea selection was about 50μs.

We found that the weighted errors were sensitive to the way we calculated the data cost. Superior results were obtained by calculating the data cost at each pixel (i.e. same resolution as the fovea), and downsampling by summing the results across windows of 2 $\times$ 2 pixels, with a slight increase in runtime.

TABLE 2. Results on the 2012 KITTI Stereo Benchmark. Out-Noc: percentage of unoccluded pixels with >3-pixel error. Out-All: same except including occluded pixels. Avg-Noc & Avg-All: mean absolute errors (in pixels) over unoccluded pixels and over all pixels, respectively. The top two rows are from the MRF method with the fovea size set to zero (first row) and the full frame (second row). The corresponding runtimes are per frame on a single core of a MacBook Pro (2.5 GHz Intel Core i7). For additional context, results in the last two rows are reproduced from [45] for the fast and slow CNN methods that we adopted for our grasping scenario. These runtimes were with a NVIDIA GTX Titan X.

Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime
MRF no fovea	14.16%	15.90%	2.9 px	3.4 px	0.062s
MRF full fovea	9.35%	11.21%	2.1 px	2.7 px	0.188s
CNN (fast)	2.82%	-	-	-	0.8s
CNN (slow)	2.43%	3.63%	0.7 px	0.9 px	67s

There are no benchmarks that specifically address our main goal (reduction of runtime while maintaining task-relevant performance). However, to put these results in context, we evaluated *unweighted* errors against the KITTI 2012 stereo benchmark test data. Average absolute errors, and percentages of pixels with >3 pixels error, are shown in Table 2 for both 0% fovea and 100% fovea. At the time of submission, the 0% and 100% fovea results correspond to 77th-place and 65th-place rankings on the KITTI benchmark in terms of the

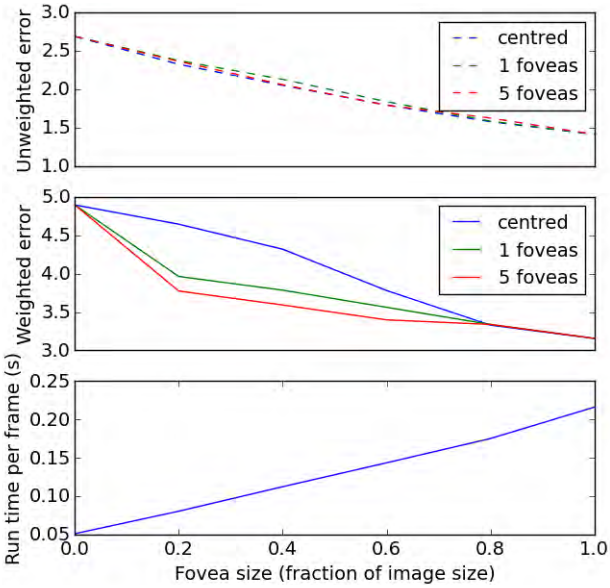

FIGURE 11. An example of performance as a function of fovea size. The fovea size (horizontal axis) varies as a fraction of the total image size, from 0 to 1. The top panel shows unweighted error based on LIDAR-based ground-truth disparity. The error measure is the mean absolute difference between estimate and ground truth, with the difference in each pixel clipped at a maximum of 20 pixels, to de-emphasize outliers. The blue line corresponds to a single fovea centred on the image (i.e. not moved to high-weight regions). The green line shows the same error with a single mobile fovea, and the red shows the same error with up to five mobile sub-foveae. The centre panel is identical except that the error is weighted, with mean weight normalized to one. Moving a single fovea to high-weight regions made the error drop more sharply with increasing foveal area (green vs. blue line), and more so when up to five sub-foveae were allowed (red line). The weighted error was higher overall than the unweighted error, reflecting the fact that our weighting scheme emphasized more challenging regions. The bottom panel shows how run time increased with fovea size.

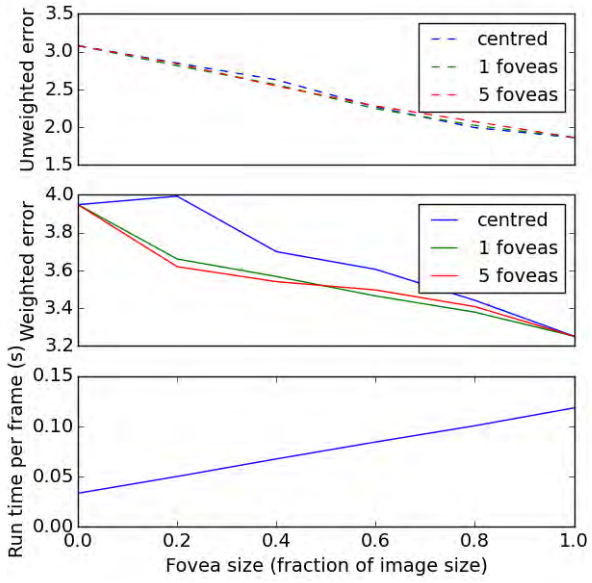

FIGURE 12. As figure 11, with two differences: 1) In this case, the peripheral data cost was calculated at each pixel and averaged to the peripheral resolution, whereas in figure 11 the data cost was only calculated once per pixel at the peripheral resolution, by skipping some of the pixels at the full resolution; 2) an additional belief propagation scale was omitted in both the fovea and the periphery.

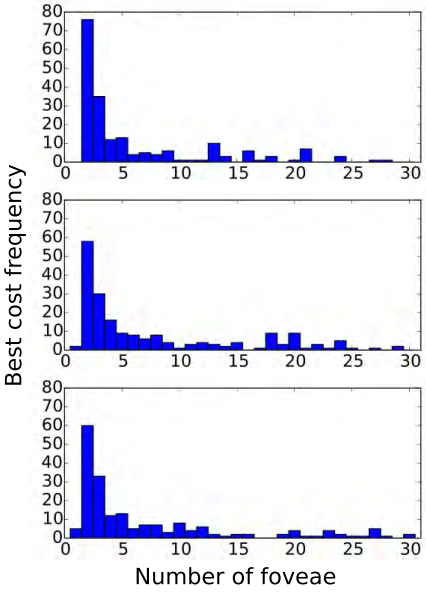

FIGURE 13. Frequency with which different numbers of foveae covered the greatest integrated weight, in weight images $w_{x,y} = w^s_{x,y} w^d_{x,y}$. This histogram is based on all 194 stereo image pairs from the KITTI training dataset. The histograms correspond to different total foveal areas. Top: 10% of frame; middle: 20% of frame; bottom: 30% of frame.

percentage of unoccluded pixels with greater than 3-pixels error. However, higher-ranked methods have an average runtime of 80s per frame, rather than < 0.2s. These results are not directly related to task-weighted error (our main interest), but they provide context in terms of a standard benchmark. Slight improvements on these results were obtained by gaussian filtering of the disparity estimates.

Counter-intuitively, with our improved calculation of data cost, belief propagation at the finest scale sometimes increased the *weighted* error. However, this quirk allowed us to improve runtime by omitting an additional scale of belief propagation. Figure 12 shows an example (with improved data cost) in which the three finest scales of belief propagation were omitted in the periphery, and the two finest were omitted in the fovea. Although the data cost calculation took longer in this case, processing could nonetheless be completed at 10 frames/s with a larger fovea (roughly 60% vs. 20%), and about 7% lower error.

D. SUB-FOVEA SELECTION

As shown previously in Figure 8, the number of foveae used can have a significant effect on how the importance region is covered, and thus greatly effect the results. Here, we explore in more detail how thoroughly the weight, $w_{x,y}$, is covered by different numbers of foveae.

Figure 13 shows histograms of the numbers of sub-foveae with the best weight coverage, on KITTI data, up to a maximum of 30 sub-foveae. The top, middle, and bottom histograms correspond to total foveal areas of 10%, 20%,

TABLE 3. Percent coverage of importance weight on KITTI data, for various fovea sizes and numbers. The left column gives the fovea size as a fraction of the full image. The centre and right columns give the % weight covered by an optimally placed single fovea and optimal number of foveae, respectively.

Fovea Fraction	Single Fovea	Optimal # Foveae
0.1	34.7 ± 15.0%	38.5 ± 15.3%
0.2	45.1 ± 18.4%	48.8 ± 18.7%
0.3	51.3 ± 19.9%	54.2 ± 19.9%

and 30% of the images. Two sub-foveae were most frequently best.

Table 3 compares the weight coverage of the optimal number of sub-foveae with that of single sub-foveae. Consistent with Figure 13, the single sub-foveae have worse coverage on average than the optimal number. However, mean coverage by a single fovea is within 10% of optimal coverage in each case. Therefore, although multiple sub-foveae are generally better than one, the difference is generally not large.

The run time taken by our greedy method of sub-fovea placement is roughly linear in the number of sub-foveae, and the time to compare all of {1, 2, ..n} sub-foveae is quadratic. As an example, the average time to find the best arrangement of 1-30 sub-foveae in the top panel of Figure 13 was 125ms on a 2014 MacBook Pro, but checking only 1-5 sub-foveae (one of which was most often optimal in any case) reduced this time to 4ms.

IV. DISCUSSION

In this work, we experimented with foveated vision for real-time disparity estimation. This approach allowed us to process frames much more quickly than the most accurate stereo methods, but roughly as accurately in regions that we identified as being the most task-relevant.

This approach requires identification of the most task-relevant, or "important" parts of the image. As pointed out by [8], task-relevance depends entirely on the task. However, for concreteness, we developed a family of utility functions that is fairly widely applicable in the context of stereo vision. This approach involves combining a spatial weighting with an "excess disparity" estimate. In a grasping context, we defined excess disparity relative to a support surface, and in a navigation context we defined excess disparity relative to time-averaged disparity. This weighting scheme requires less computation than a more sophisticated saliency map [21], while ignoring features that contribute to saliency but have little relevance to grasping or obstacle avoidance. Our spatial weighting term emphasized the centre of the image, but other spatial weightings are possible. For example, in grasping, one alternative would be to emphasize points near the gripper, and in navigation the spatial weighting term could also consider the robot's future path. Importantly, it would be straightforward to switch between multiple such weighting schemes, or to mix them continuously, according to task and goal changes. With such variations, we believe this general approach may be applicable to improving accuracy for many disparity-related decisions.

Finally, we studied the use of multiple foveae. We developed an efficient greedy algorithm for placing multiple foveae. We also characterized the advantage of multiple foveae over single foveae. The primate visual system has a single fovea of fixed retinal position and shape per eye, and the foveae of each eye are typically oriented toward the same feature. Robots with hardware foveae (e.g. [2], [20]) have the same constraint. Using our utility metric with the KITTI data, we found that two square foveae were optimal more frequently than any other number from one to thirty, and that larger numbers of foveae (up to thirty) were occasionally optimal. However, total importance weight spanned by single foveae was >90% of that spanned by the optimal number of foveae. It is interesting that larger numbers of foveae (which are better able to fit complex shapes) were not typically much better than one or two. This suggests that a fixed hardware fovea has a minor cost in this context.

Accurate disparity estimation is particularly important for autonomous grasping in robotics, because the ideal gripper configuration depends on shape characteristics on a centimetre scale. We used one of the best-performing stereo methods currently available [45]. However, we have not attempted to control grasps using these results, and it is not obvious that the results are good enough for this purpose. State-of-the-art grasping systems often use monocular images [28], [29], despite the high relevance of depth information in principle. A slow, foveal stereo method is likely to be useful for grasping, but existing stereo methods may not yet be accurate enough.

The MRF method that we adapted ran on a CPU. If it ran instead on a large enough GPU, the image size would be less relevant, perhaps nullifying the benefits of foveation. However, in this case we expect that our approach would maximize the image size that a given device could process. For example, if each scale had the same number of pixels (e.g. halving field of view and doubling resolution with each step), then each scale could potentially use the whole device. The frame rates and image sizes we reported depend on the hardware used, but we showed useful results in two very different scenarios, suggesting that the approach may be more widely applicable.

More generally, despite increasing computing power, foveae are likely to become more important in the future. Vision systems in robotics are growing in sophistication (e.g. [26]), but are still much simpler than human vision, which integrates many cues to robustly estimate depth. Notably, even a fairly abstract real-time simulation of the full human vision would require several racks of specialized neuromorphic computers [13], and these requirements would increase by several orders of magnitude if the full field of view were processed in as much detail as the fovea. In short, good vision is inherently computationally intensive, and foveation helps to manage the cost. This suggests that there may be a long-term and growing need for this approach in robotics, as more sophisticated vision systems are developed.

A. RELATIONSHIP WITH OTHER WORK

Most previous work in foveated stereo (reviewed in [22]) has employed hardware foveation, in which the fovea is always at the image centre, and cameras are oriented and verged to important features. One such method [25] uses two vergent cameras, with resolution that drops off in steps as you move away from the image centre (from fovea to periphery). It constructs a depth map (composed of an array of planar patches) from multiple fixations. New peripheral patches overwrite older ones, and foveal patches overwrite peripheral patches. Other work (e.g. [2]) used images with a log-polar resolution profile.

In contrast with foveated hardware approaches, we developed software approaches that processed different parts of stereo images in different levels of detail, allowing arbitrary motion of the fovea at the frame-rate. This approach was previously taken by [15], but they reported only qualitative stereo results, which were relatively noisy (their code is not available, but the right column of their Figure 6 can be compared qualitatively with our examples). We extended this approach to more sophisticated stereo algorithms, including convolutional networks and Markov random fields, leading to improved results. We also considered multiple foveae per frame. In other related work, [6] developed an FPGA-based stereo camera system with multiple foveae. However, they did not attempt disparity estimation. The general approach of processing the most relevant parts of an image in more detail has been widely adopted for detection and recognition (e.g. [5], [8], [43]).

At the time of writing, two methods on the KITTI 2012 leaderboard outperformed our benchmark results (for unweighted error) in terms of both accuracy and runtime. The first (Toast2) is from [35]. This method uses block matching and the Census Transform, along with image shears, to infer the depth of slanted planar surfaces in the scene [35]. It would be interesting to see whether our importance-weighting approach could be combined with this stereo method. However, because this method fits image blocks to planes, it is possible that its better average performance corresponds to low error on large flat surfaces such as roads and walls, with lesser advantages for typical obstacles. The other method, DispNetC [32], is a recent convolutional network with faster runtime and somewhat greater error than that of [45]. It may be useful in future work to apply our approach to these methods, or future stereo methods.

ACKNOWLEDGEMENTS

The authors thank Hamid Tizhoosh, John Zelek, and John-Paul Gignac for helpful discussions.

REFERENCES

[1] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 115–123.

[2] A. Bernardino and J. Santos-Victor, "A binocular stereo algorithm for log-polar foveated systems," in *Biologically Motivated Computer Vision*, vol. 2525. Berlin, Germany: Springer, 2002, pp. 127–136.

[3] R. Bevec and A. Ude, "Object learning through interactive manipulation and foveated vision," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Oct. 2013, pp. 234–239.

[4] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[5] N. J. Butko and J. R. Movellan, "Optimal scanning for faster object detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, vol. 1. Jun. 2009, pp. 2751–2758.

[6] P. Camacho, F. Coslado, M. González, and F. Sandoval, "Multifoveal imager for stereo applications," *Int. J. Imag. Syst. Technol.*, vol. 12, no. 4, pp. 149–165, 2002.

[7] P. M. Daniel and D. Whitteridge, "The representation of the visual field on the cerebral cortex in monkeys," *J. Physiol.*, vol. 159, no. 2, pp. 203–221, 1961.

[8] M. Desnoyer, "Visual utility: A framework for focusing computer vision algorithms," Ph.D. dissertation, Dept. Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2015.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, Oct. 2006.

[10] P Fieguth, *Statistical Image Processing and Multidimensional Modeling*. New York, Ny, USA: Springer, 2010.

[11] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations," *ACM Trans. Appl. Perception*, vol. 7, no. 1, pp. 1–39, 2010.

[13] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[15] R. B. Gomes, L. M. G. Gonalves, and B. M. de Carvalho, "Real time vision for robotics using a moving fovea approach with multi resolution," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 2404–2409.

[16] S. Gould *et al.*, "Peripheral-foveal vision for real-time object recognition and tracking in video," in *Proc. IJCAI Int. Joint Conf. Artif. Intell.*, 2007, pp. 2115–2121.

[17] N. C. Griswold and C. F. Weiman, "A modification of the fusion model for log polar coordinates," *Proc. SPIE*, vol. 1192, pp. 854–866, Mar. 1990.

[18] F. Guney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *Proc. CVPR*, 2015, pp. 4165–4175.

[19] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends Cognit. Sci.*, vol. 9, no. 4, pp. 94–188, Apr. 2005.

[20] S. Huber, B. Selby, and B. P. Tripp, "Design of a saccading and accommodating robot vision system," in *Proc. 13th Conf. Comput. Robot Vis.*, Victoria, BC, Canada, 2016, pp. 350–357.

[21] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vis. Res.*, vol. 40, pp. 1489–1506, Jan. 2000.

[22] V. J. Traver and A. Bernardino, "A review of log-polar imaging for visual perception in robotics," *Robot. Auto. Syst.*, vol. 58, no. 4, pp. 378–398, 2010.

[23] R. S. Johansson, G. Westling, A. Bäckström, and J. R. Flanagan, "Eye–hand coordination in object manipulation," *J. Neurosci.*, vol. 21, no. 17, pp. 6917–6932, 2001.

[24] A. Kimura, R. Yonetani, and T. Hirayama, "Computational models of human visual attention and their implementations: A survey," *IEICE Trans. Inf. Syst.*, vol. 96, no. 3, pp. 562–578, 2013.

[25] W. N. Klarquist and A. C. Bovik, "FOVEA: A foveated vergent active stereo vision system for dynamic three-dimensional scene recovery," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 755–770, Oct. 1998.

[26] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 951–970, Jul. 2013.

[27] E. Kowler, "Eye movements: The past 25 years," *Vis. Res.*, vol. 51, no. 13, pp. 1457–1483, Jul. 2011.

[28] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," in *Proc. ICLR*, 2013, pp. 705–724.

[29] S. Levine *et al.* (2016). "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." [Online]. Available: https://arxiv.org/abs/1603.02199

[30] F. L. Lim, G. A. W. West, and S. Venkatesh, "Use of log polar space for foveation and feature recognition," *IEE Proc.-Vis., Image, Signal Process.*, vol. 144, no. 6, pp. 323–331, 1997.
[31] A. Maki, P. Nordlund, and J.-O. Eklundh, "Attentional scene segmentation: Integrating depth and motion," *Comput. Vis. Image Understand.*, vol. 78, no. 3, pp. 351–373, 2000.
[32] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.
[33] K. P. Murphy, Y. Weiss, and I. Michael Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. UAI*, 1999, pp. 467–475.
[34] G. Pasquale, T. Mar, C. Ciliberto, L. Rosasco, and L. Natale, "Enabling depth-driven visual attention on the iCub humanoid robot: Instructions for use and new perspectives," *Frontiers Robot. AI*, vol. 3, pp. 1–11, Jun. 2016.
[35] B. Ranft and T. Strauß, "Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1941–1947.
[36] K. Rayner, T. J. Slattery, and N. N. Bélanger, "Eye movements, the perceptual span, and reading speed," *Psychonomic Bull. Rev.*, vol. 17, no. 6, pp. 834–839, 2010.
[37] C. A. Rothkopf, D. H. Ballard, M. M. Hayhoe, and O. Regan, "Task and context determine where you look," *J. Vis.*, vol. 7, no. 14, pp. 1–20, 2007.
[38] A. C. Schütz, Doris I Braun, and Karl R Gegenfurtner, "Eye movements and perception: A selective review," *J. Vis.*, vol. 11, no. 5, pp. 1–30, 2011.
[39] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, Jul. 2003.
[40] B. W. Tatler, M. M. Hayhoe, M. F. Land, and D. H. Ballard, "Eye guidance in natural vision: Reinterpreting salience," *J. Vis.*, vol. 11, no. 5, pp. 1–23, 2011.
[41] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cogntive Psychol.*, vol. 12, no. 1, pp. 97–136, 1980.
[42] J. K. Tsotsos, *A Computational Perspective on Visual Attention*. Cambridge, MA, USA: MIT Press, 2011.
[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, pp. I-511–I-518.
[44] C. F. R. Weiman and R. D. Juday, "Tracking algorithms using log-polar mapped image coordinates," *Proc. SPIE*, vol. 1192, pp. 843–853, Mar. 1990.
[45] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 2, pp. 1–32, 2016.
[46] G. J. Zelinsky, W. Zhang, B. Yu, X. Chen, and D. Samaras, "The role of top-down and bottom-up processes in guiding eye movements during visual search," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18. 2006, pp. 1569–1576.

ERIC HUNSBERGER received the B.A.Sc. degree from the University of Waterloo in 2011. He is currently pursuing the Ph.D. degree in computational neuroscience with the University of Waterloo. His research focuses modelling and understanding object recognition, learning, and depth perception in the primate visual system.

VICTOR REYES OSORIO received the B.Sc. degree in physics from the University of Waterloo in 2016. He is currently pursuing the M.A.Sc. degree in neurorobotics with the University of Waterloo. He is interested on modelling primate visual and motor systems and replicating their performance in robotics.

JEFF ORCHARD received the B.Math. degree in applied mathematics from the University of Waterloo, in Canada, the M.Sc. degree from The University of British Columbia, and the Ph.D. degree in computing science from Simon Fraser University, Canada, in 2003. Since 2003, he has been a Faculty Member with the David R. Cheriton School of Computer Science, University of Waterloo. He has authored research in image processing and medical imaging. His research focuses on computational neuroscience, using mathematical models and computer simulations of neural networks in an effort to understand how the brain works, learning methods for deep perceptual networks, network dynamics, decision-making, spatial navigation, and neural population coding.

BRYAN P. TRIPP received the B.Sc. degree from the University of Waterloo in 1997, the M.Sc. from the University of Toronto in 2002, and the Ph.D. degree from the University of Waterloo in 2009. He was a Post-Doctoral Fellow with McGill University from 2009 to 2011. He is currently an Assistant Professor with the Department of Systems Design Engineering and the Centre for Theoretical Neuroscience, University of Waterloo. His research focuses on modeling of the primate visual and motor systems, using statistical models of neural activity, deep learning, spiking networks, and integration of biophysical neural models with rob.