Teaching Stereo Perception to YOUR Robot

Marcus Wallenberg
http://users.isy.liu.se/cvl/wallenberg

Per-Erik Forssén
http://users.isy.liu.se/cvl/perfo

Computer Vision Laboratory
Linköping University
Linköping, Sweden

Abstract

This paper describes a method for generation of dense stereo ground-truth using a consumer depth sensor such as the Microsoft Kinect. Such ground-truth allows adaptation of stereo algorithms to a specific setting. The method uses a novel residual weighting based on error propagation from image plane measurements to 3D. We use this ground-truth in wide-angle stereo learning by automatically tuning a novel extension of the best-first-propagation (BFP) dense correspondence algorithm. We extend BFP by adding a coarse-to-fine scheme, and a structure measure that limits propagation along linear structures and flat areas. The tuned correspondence algorithm is evaluated in terms of accuracy, robustness, and ability to generalise. Both the tuning cost function, and the evaluation are designed to balance the accuracy-robustness trade-off inherent in patch-based methods such as BFP.

1 Introduction

Object recognition on robot platforms is greatly facilitated by dense stereo [5, 11, 18]. Stereo can both provide a rough outline of objects before they have been identified [5, 18], and give priors on feature sizes [11].

Over the years there has been a steady progress in the quality of stereo algorithms. This is evident from the Middlebury datasets [2] against which new algorithms are compared [20]. While this has ensured a steady progress in quality, it has also lead the stereo research field to focus on improving performance on this particular dataset.

In this paper we introduce a technique that allows generation of arbitrary new datasets, without the need for expensive equipment such as a LIDAR. Instead our technique can use cheap consumer depth sensors such as the Microsoft Kinect. Such datasets allow researchers and engineers to adapt algorithms to their own stereo rigs. We demonstrate this by putting our dataset generation to use in wide angle stereo learning.

Wide-angle stereo provides an overview of a scene, even at very short range. The large *field of view* (FoV) also ensures that the visual fields from different points of view have a high degree of overlap [19]. For these reasons, wide-angle lenses are popular in navigation and mapping using visual features, see e.g. [21] for an early example. Wide angle stereo is also potentially useful for visual object search on robot platforms [5, 18] as it could provide object outlines in a wider FoV.

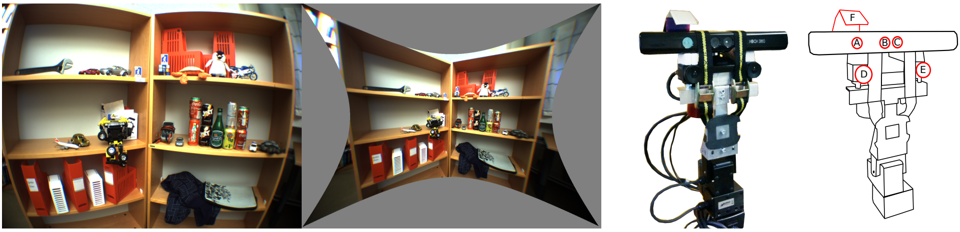

Figure 1: Left: In wide-angle images, angular resolution is near uniform. If they are rectified to preserve straight lines [10], most of the image is spent representing the periphery.
Right: Pan-tilt stereo rig with Kinect. (A) - SLP projector, (B) - RGB camera, (C) - NIR camera, (D) - Left wide-angle camera, (E) - Right wide-angle camera, (F) - Diffusor (raised).

1.1 Related work

In narrow-angle stereo, state-of-the-art stereo systems first rectify the images to remove lens distortion and bring them into fronto-parallel alignment [16]. This approach is impractical on wide angle lenses, as shown in figure 1 (left). There are two main problems with this approach. First, rectification of wide-angle stereo-pairs introduces shape distortions in the periphery. Second, after removal of lens distortion, the angular resolution becomes very high in the periphery of the images. This means that most pixels will describe the periphery, instead of the central region of the pair. This is wasteful, since the central region is where the images have common content, and disparity computation is actually possible. In contrast, the algorithm introduced in this paper works directly on wide-angle stereo images.

A common approach to wide-angle stereo is to remove only the radial distortion [23], and then run a descriptor-based wide-baseline stereo algorithm such as e.g. DAISY [24]. An alternative approach is to use simpler matching metrics, and instead leverage *correspondence propagation*. One such algorithm is the *best-first propagation* (BFP) algorithm [14], and a recent addition is the *generalised PatchMatch algorithm* (GPM) [4]. At the basic level, correspondence propagation algorithms are more general than stereo algorithms, but they are readily adaptable to the stereo problem. This was done for BFP in [15] and for GPM in [6] by complementing the correspondence propagation with correspondence pruning using an epipolar constraint.

The original BFP algorithm [14], grows correspondences around seed correspondences in a recursive fashion. In this paper we extend and improve BFP by adding a coarse-to-fine search, thereby removing the need for seed correspondences. We also limit propagation along linear structures. The parameters of our new algorithm are then tuned using dense ground-truth correspondences from our Kinect-generated dataset (see sections 4 and 5). A take-home message here is that such a dataset generation can be used to tune *any* correspondence algorithm (not just BFP) to a *specific* stereo rig. A different extension of the BFP algorithm for wide-baseline stereo and deformable object recognition is given in [13]. There however, matching of affine covariant regions and the adaptation of an affine transformation is needed for every correspondence, which significantly increases computational complexity.

1.2 Contributions and organisation

In this paper, we describe how to fuse inverse depth maps from the Microsoft Kinect into ground-truth disparity maps that can be used for tuning and evaluation of stereo algorithms. Such disparity maps allow tuning to a specific setting or a specific stereo rig. We also derive weights for range scan fusion, by propagating image plane noise back into 3D. Finally, we give a practical example of how a stereo system may be automatically tuned using ground-truth disparity maps.

This paper is organised as follows: In section 2 we describe our ground-truth generation procedure. In section 3, we describe our stereo algorithm. In section 4, we describe the dataset, and our automatic tuning and evaluation procedure. The paper ends with results (section 5) and a summary of our conclusions (section 6).

2 Ground-Truth Generation

Acquisition of wide-angle stereo images and generation of corresponding ground-truth is made using the pan-tilt rig shown in figure 1 (right). The rig has two wide-angle cameras and a Kinect *structured light pattern* (SLP) range sensor [1]. The pan-tilt unit is a Directed Perception PTU D46-17.5 with a pointing accuracy in pan and tilt of ±0.0514°. The left and right wide-angle cameras are Point Grey Flea2 cameras, with a resolution of 1280×960 pixels, and 2.5 mm wide-angle CCTV lenses from Bowoon Optical. The Kinect sensor has 1280×1024 pixel RGB and NIR cameras, and outputs 640×480 pixel inverse depth images. The NIR frames depict the scene as illuminated by the SLP projector which we optionally cover with a diffusor (see figure 1) to remove the SLP when needed.

As the Kinect has a narrow field of view, multiple scans are fused to obtain ground-truth disparity maps. These are obtained by varying the pan and tilt. At every pan-tilt position, each camera captures 10 images, which are then averaged to reduce pixel noise. For the inverse depth maps, only pixels successfully measured in *all* images are retained.

2.1 Calibration

For calibration of the cameras, a chessboard is placed in the scene at six different distances, and imaged by all four cameras at 11 different pan-tilt angles. To ensure the highest accuracy of chessboard corner locations, the image acquisition is performed in two passes: first the inverse depth maps are acquired, then a plastic diffusor covers the SLP projector (see figure 1) to remove the SLP, and NIR images are acquired.

2.1.1 Camera geometry and lens distortion

We employ the most common formulation of camera pose $\mathbf{P}$, consisting of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$, where $\mathbf{P} = \left[\mathbf{R}^T, -\mathbf{R}^T\mathbf{t}\right]$ describes the rigid transformation to the camera-centred coordinate system. In order to map a 3D point $\mathbf{x}$ to a pixel position $\mathbf{u}$ in an image, the intrinsic camera matrix $\mathbf{K}$, and the lens distortion parameters θ must also be known [9]. For the intrinsic parameters, we use a four-parameter model, which assumes zero skew. For the two cameras on the Kinect (with relatively narrow fields of view) we use the 8-parameter rational distortion model implemented in OpenCV 2.3.1 [3], which models both radial and tangential distortion. For the two wide-angle cameras, we instead use the wide-angle distortion model described in [10] which has three parameters: a principal point (d_u, d_v) and a radial distortion parameter γ.

In order to convert 2D pixel coordinates (u, v) and inverse depth $d(u, v)$ (output by the Kinect) to the coordinates of a 3D point $\mathbf{x}$, we follow [22] and [12]. We define the camera-centred coordinate system and model the mapping from inverse depth to distance along a forward-pointing z axis such that

$$z = (\alpha d(u,v) + \beta)^{-1}, \quad \text{and} \quad \mathbf{x} = z\mathbf{K}^{-1}\mathbf{f}^{-1}\left([u,v,1]^T, \theta\right) \tag{1}$$

in the camera observing $d(u, v)$ (the NIR camera on the Kinect). Here, $\mathbf{f}^{-1}([u,v,1]^T, \theta)$ denotes the inverse of the distortion function at (u, v), with distortion parameters θ.

2.1.2 Error variance propagation

Since we make use of both the inverse depth and pixel coordinates in our calibration procedure, the effect of errors in these measurements must be taken into account during calibration. Assuming that pixel coordinates and inverse depth are each affected by independent additive zero-mean errors, we derive the corresponding error variance in 3D and also in the image plane of an arbitrary camera. To make the calculations more manageable, we disregard the effects of lens distortion on the resulting error variances. The perturbation in z resulting from a perturbation $\tilde{d}(u, v) = d(u, v) + \varepsilon_d$ in the inverse depth can be written as

$$\varepsilon_z = \tilde{z} - z = (\alpha\tilde{d}(u,v) + \beta)^{-1} - (\alpha d(u,v) + \beta)^{-1} \approx -\alpha\varepsilon_d z^2 \tag{2}$$

using a first-order Taylor expansion of (2) about $d(u, v)$ with respect to ε_d. The propagated variance σ_z^2 is then

$$\sigma_z^2 = \mathrm{E}\left[\varepsilon_z^2\right] = \alpha^2 z^4 \mathrm{E}\left[\varepsilon_d^2\right] = \alpha^2 z^4 \sigma_d^2, \tag{3}$$

where σ_d^2 is the variance of the inverse depth error. The error covariance of a 3D point $\mathbf{x} = [x, y, z]^T$ calculated from perturbed pixel coordinates $[u + \varepsilon_u, v + \varepsilon_v]^T$ and the perturbed z can (under the zero-mean and independence assumptions) then be expressed as

$$\mathbf{C}_{\mathbf{xx}} = \mathrm{E}\left[(\tilde{\mathbf{x}} - \mathbf{x})(\tilde{\mathbf{x}} - \mathbf{x})^T\right], \tag{4}$$

a diagonal matrix with elements σ_x^2, σ_y^2 and σ_z^2 (the error variances in x, y and z, respectively). Disregarding lens distortion, the coordinates of the points are

$$\mathbf{x} = z\mathbf{K}^{-1}[u,v,1]^T \quad \text{and} \quad \tilde{\mathbf{x}} = (z + \varepsilon_z)\mathbf{K}^{-1}[u + \varepsilon_u, v + \varepsilon_v, 1]^T. \tag{5}$$

From these expressions, σ_x^2 and σ_y^2 can be determined as

$$\sigma_x^2 = f_x^{-2}\left(\sigma_u^2 z^2 + \sigma_u^2\sigma_z^2 + \sigma_z^2(u - c_x)^2\right), \quad \sigma_y^2 = f_y^{-2}\left(\sigma_v^2 z^2 + \sigma_v^2\sigma_z^2 + \sigma_z^2(v - c_y)^2\right), \tag{6}$$

where f_x, f_y denote the focal lengths of the camera, (c_u, c_v) the principal point and σ_u^2 and σ_v^2 the error variances in u and v. When 3D points are projected to other cameras, the resulting image plane variances $[\sigma_{u'}^2, \sigma_{v'}^2]^T$ can be calculated by simply applying the appropriate projection operation to $[\sigma_x^2, \sigma_y^2, \sigma_z^2, 1]^T$ (described by $\mathbf{K}$ and $\mathbf{P}$ pertaining to the target camera). This means that the error variance in both 3D position, and 2D projection can be estimated for every measured pixel and corresponding inverse depth value, allowing us to implement a weighting scheme that seamlessly combines error measurements in both 3D and 2D.

2.1.3 Calibration procedure

Calibration of the rig is done using a standard chessboard calibration pattern, and is carried out in several steps. First, the intrinsics and lens distortion parameters are estimated for all cameras from a series of 100 images using OpenCV, and the wide-angle distortion model [10] is estimated for the two wide-angle cameras. As has been previously observed [22], the inverse depth map obtained from the Kinect device does not correspond to the sensor grid of the NIR camera, as it is cropped, subsampled by a factor of two and also shifted in relation to the NIR image. We estimate this shift, and compensate for it using a similar technique to the one proposed in [22]. The main steps are then (in order):

- **Rotation data collection:** Several sets of images are captured using the procedure described in section 2.1.
- **Inverse depth conversion estimation:** The α and β parameters for inverse depth to range conversion are fitted by aligning the calculated 3D positions of the observed calibration pattern so that they differ from the ideal calibration pattern by a rigid-body transformation only.
- **Extrinsic parameter estimation (camera pose):** The relative camera poses of the four cameras are estimated by minimising the 2D reprojection error of the reconstructed 3D calibration pattern obtained in the previous step.
- **Rotation axis estimation:** The position and orientation of the PTU rotation centre relative to the NIR camera is estimated. This is done using the reconstructed 3D planes, their 2D projections and the recorded pan-tilt angles, by minimising the pairwise 3D *and* 2D alignment errors between all pan-tilt poses at all distances.

Once all these parameters have been estimated, it is possible to create a 3D reconstruction in a common coordinate frame from multiple Kinect range scans at arbitrary pan and tilt angles. The error variances described in section 2.1.2 are used to properly balance the 3D and 2D residuals minimised in the calibration steps. The weight of dimension l for a point-point residual between 2D or 3D points $(\mathbf{p}_j, \mathbf{p}_k)$ is defined as

$$w_{l,jk} = \left(\sigma_{l,j}^2 + \sigma_{l,k}^2\right)^{-\frac{1}{2}} . \tag{7}$$

For an (ideal) point given by the calibration pattern, all $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 0$. For a 2D pixel coordinate found using the OpenCV's chessboard detector, $\sigma_u^2 = \sigma_v^2$. For all other points, the variances are calculated as described in the previous section. The values of σ_u^2, σ_v^2 and σ_d^2 are intrinsic to the measurement process, and are regarded as constant.

2.2 Data acquisition and fusion for ground-truth generation

In order to calculate disparities covering the full field of view of the wide-angle stereo rig using Kinect inverse depth maps, images are captured from a number of different pan-tilt poses (we use 51 poses spread evenly in pan-tilt space). The individual inverse depth maps taken from all different Kinect poses must then be combined so as to cover the fields of view of the wide-angle stereo cameras. This is done by first backprojecting all pixels containing measurements in the inverse depth images into 3D, using (1). The recorded angles from the PTU, and the estimated rotation axes corresponding to the pan and tilt motions are then used to transfer all points to a common coordinate frame. We choose this coordinate frame to be one of the recorded poses, for which we have corresponding wide-angle images. We then project the 3D point cloud to both the left and right cameras, and for each pixel in the left image find the resulting disparity distribution.

In order to fuse the disparity distribution into a single value, we use *mean-shift* [8] to find one or several modes, each with a variance below a set threshold (we use $t_\sigma = 1$ pixel). The measurements are also spatially weighted in the image, using Gaussian weights ($\sigma = 1$ pixel), centred on the pixel where a disparity estimate is desired. This is similar to [25], but in case of multiple modes we select the mode closest to the camera, on the grounds that the occluding surface will be the one visible. In contrast, [25] selects the farthest mode (which may be preferable with their motion stereo data). For each pixel location, the number of measurements used in reconstruction and the variance of their disparity magnitudes are recorded to be used as a confidence measure.

3 Wide-Angle Stereo using Coarse-to-Fine BFP

Given that ground-truth disparity maps can now be generated, we turn to the practical application: tuning of a stereo algorithm. In this section, we review the *best-first propagation* (BFP) dense correspondence algorithm [14] and describe how we have extended BFP with a coarse-to-fine search, a structure dimensionality parameter and a sub-pixel refinement step.

The BFP algorithm [14, 15] starts off from a sparse set of seed correspondences (obtained using e.g. SIFT [17]). It recursively grows the disparity map around these seeds, using *zero-mean normalised cross-correlation* (ZNCC), in 5×5 correlation windows. First, all potential correspondences near a seed are found (within a disparity gradient limit of one pixel). These are then scored, and the best one is selected, and added to the list of seed correspondences. The set of potential correspondences is then updated accordingly.

The *Coarse-to-Fine Best First Propagation* (CtF-BFP) algorithm is our extension of the BFP algorithm to multiple scales. This removes the need for initialisation with seed correspondences. Instead, the pixel-to-pixel correspondence at the coarsest scale is initialised to an identity mapping. Thus, starting from the assumption that all displacements are zero at the very coarsest scale, we use BFP to propagate these correspondences across progressively finer scales, while rejecting those that have low ZNCC scores. A subpixel refinement step can also be employed at each scale.

For each subsequent scale, the result from the previous one is used as initialisation. The procedure is to (1): re-scale correspondences from the previous scale to the current one. Then (2): refine correspondences in the right camera image by evaluating all potential one-pixel shifts of the current correspondences, selecting those with highest ZNCC, and (3): propagate these matches in best-first order at the current scale. When this is done, (4): optionally, perform subpixel refinement by fitting a symmetric quadratic polynomial to the ZNCC scores within a 3×3 pixel neighbourhood using least squares.

At each scale, propagation is controlled by three parameters: the *correlation window size* (the integer-valued size of the square correlation window), a *ZNCC threshold* and a *structure threshold*. The structure threshold is applied to a structure measure defined as $s = \lambda_2/\sqrt{\lambda_1}$, where λ_1 and λ_2 are the largest and second largest eigenvalues of the estimated autocovariance matrix in the correlation window. This measure is used to limit propagation in areas with one-dimensional structure ($\lambda_1 >> \lambda_2$) or little to no structure (where both λ_1 and λ_2 are small). We have added this threshold, as we found that one-dimensional structures are prone to aperture problems and drift, in contrast to what is claimed in [14].

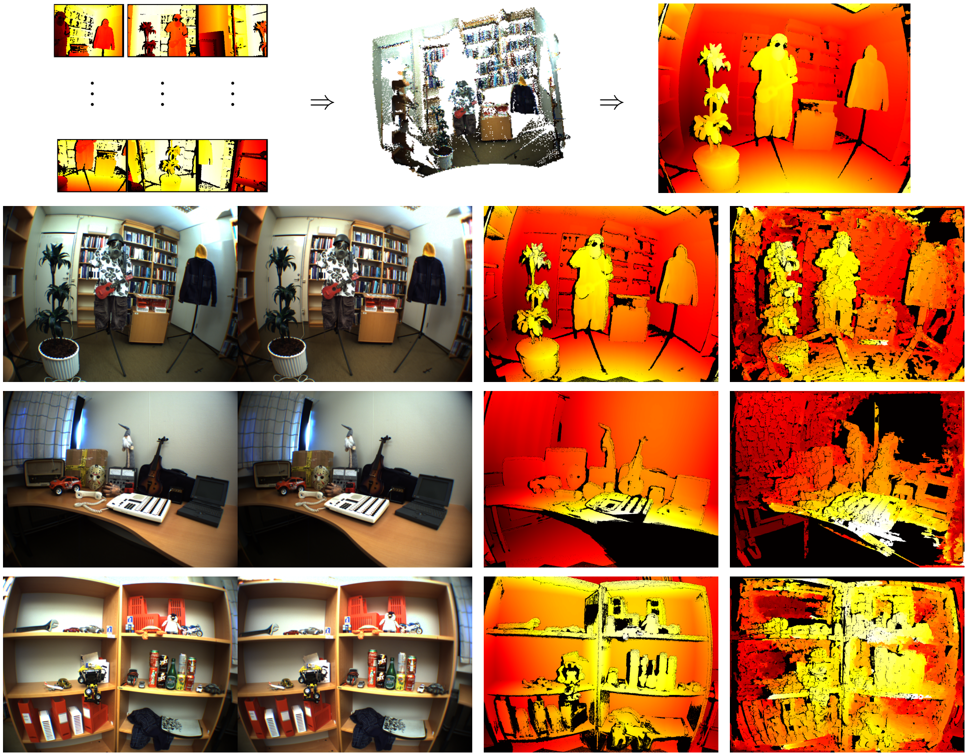

Figure 2: Top: Example of ground-truth generation. Left: Individual Kinect range scans. Centre: several range scans projected into 3D. Right: The magnitude of the fused ground-truth disparity map.
Bottom: One of the 51 example views of each scene. Columns left to right: Left wide-angle image, right wide-angle image, magnitude of disparities deemed reliably reconstructed. The final column shows magnitude of disparity estimate obtained using tuned CtF-BFP.

4 Datasets and performance measures

In order to evaluate the performance of the stereo algorithm, we have imaged three indoor scenes using the procedure described in section 2.2. For each scene, the dataset has 51 wide-angle stereo pairs and 51 Kinect inverse depth maps. From these, each of the 51 wide-angle disparity maps corresponding to the stereo pairs has been calculated using the procedure described in section 2.2. Examples of individual range scans, an intermediate point cloud and the final disparity map are shown in figure 2 (top row).

When measuring performance of the stereo algorithm, we denote the estimated disparity map to be evaluated by $\mathbf{D}(u,v)$ defined on the domain $\mathcal{V}$ (pixels where disparities have been estimated). Similarly, the ground-truth disparity image is $\mathbf{D}^*(u,v)$, and the set of valid ground-truth pixels is $\mathcal{V}^*$. If a ground-truth pixel is to be considered valid, it must be both reliably calculated and *possible* to match using the method being used. Reliability of the calculated disparity is based on the number of measurements used in fusion, and the variance of the estimated disparity distribution (see section 2.2). We choose to require at least 9 measurements and a maximum variance of 1 pixel for a reliable disparity estimate. Also,

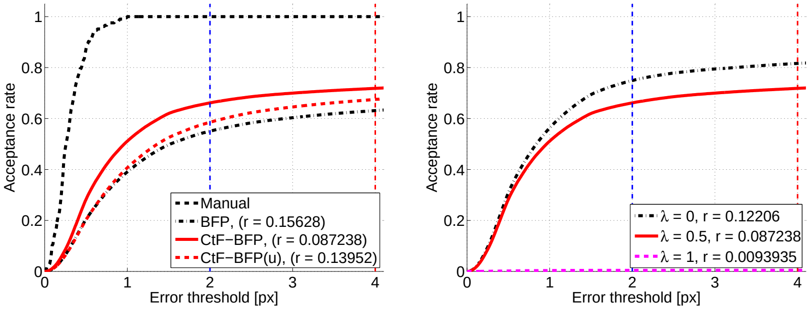

Figure 3: Left: Average acceptance curves over all data sets for automatically tuned CtF-BFP with $\lambda = 0.5$, BFP with original parameters, CtF-BFP(u) (before tuning). Errors on manually selected correspondences included as a best case. Right: Average acceptance curves over all data sets for parameters tuned using $\lambda = 0, 0.5, 1$.

since CtF-BFP is ZNCC-based, correspondences that are *not* local ZNCC maxima will not generate correct matches. Instead, these unmatchable points will drift and introduce fluctuations in the performance measures. Therefore, we use the ground-truth disparity maps to find and disregard all such unmatchable correspondences. The remaining reliable *and* matchable pixels make up the set of valid ground-truth pixels $\mathcal{V}^*$ in our performance measures below.

We then define the *acceptance rate*, as the relative portion of disparities with an error below a threshold t_a:

$$a(t_a) = \frac{1}{|\mathcal{V}^*|} \sum_{(x,y)\in\mathcal{V}\cap\mathcal{V}^*} I(||\mathbf{D}^*(u,v) - \mathbf{D}(u,v)|| \leq t_a) \tag{8}$$

where $|\mathcal{V}^*|$ is the number of valid ground-truth pixels, and $I(s)$ is an indicator function that returns 1 if statement s is true, and 0 otherwise. We also define a *rejection rate*, as the relative portion of estimated pixels that differ from ground-truth by more than a threshold t_r:

$$r(t_r) = \frac{1}{|\mathcal{V}\cap\mathcal{V}^*|} \sum_{(x,y)\in\mathcal{V}\cap\mathcal{V}^*} I(||\mathbf{D}^*(u,v) - \mathbf{D}(u,v)|| > t_r)\,. \tag{9}$$

These measures are similar in nature to the *recall* and *precision* measures [7] commonly used in classification tasks.

To find a useful set of parameters, we minimise an objective function $J(t_a,t_r)$ based on $a(t_a)$ and $r(t_r)$, where

$$J(t_a,t_r) = \lambda r(t_r) - (1-\lambda)\int_0^{t_a} a(t)dt, \tag{10}$$

thus minimising the rejection rate and average error while maximising the number of points accepted. The parameter $\lambda \in [0,1]$ allows us to control the relative weight of the rejection and acceptance terms. Different values of λ produce different behaviour of $J(t_a,t_r)$ by influencing the trade-off between accuracy, coverage and robustness.

$_E\backslash^T$	*Before tuning*	*Training set 1*	*Training set 2*	*Training set 3*
Evaluation set 1	-0.184	**-0.253**	-0.245	-0.240
Evaluation set 2	-0.054	-0.099	**-0.131**	-0.106
Evaluation set 3	-0.055	-0.116	-0.099	**-0.121**

Table 1: Cross-validation values of $J(t_a, t_r)$ (see (10)) with $t_a = 2$, $t_r = 4$ and $\lambda = 0.5$. Rows show results for different evaluation sets, columns correspond to different parameter settings.

5 Experiments and Results

In this section, we present the results of ground-truth generation and automatic stereo tuning. Note that our stereo system normally works with images of size 640×480 pixels rather than the maximum resolution of the Flea2 for reasons of computational complexity. Therefore, this is also the resolution at which we have chosen to perform tuning and evaluation.

In order to verify the quality of the ground-truth disparity maps, one map from each scene was evaluated using a sparse set of 208 correspondences manually selected in the full-resolution images, and spread over the image plane. All of these were found to differ from the fused disparity map by less than one pixel (compared to around 1.5 pixels when not using the proposed weighting scheme). Considering the accuracy with which the manual correspondences can be determined, this suggests that the accuracy of the fused disparity estimate is comparable to the manual correspondences. Errors on these correspondences are included as a best-case performance in figure 3.

The parameters listed in section 3 (window size, ZNCC threshold, structure threshold and subpixel refinement toggle) were optimised for each of the 6 scales used. Window size was constrained to be an odd number between 3 and 21 (corresponding to 3×3 and 21×21 pixel windows), and the ZNCC threshold was constrained to $[0, 1]$. A coarse grid search in the parameter space was used as initialisation. Starting from the coarsest scale, parameters for each successive scale were then optimised independently. The parameters at all scales were then refined using the same objective function evaluated at the finest scale.

Accuracy vs. robustness: Tuning was performed using one image of each scene (three in total) with $t_a = 2$, $t_r = 4$ and $\lambda = 0.5$. The parameters obtained were then evaluated using a different set of images (also one of each scene). Results of this evaluation are found in figure 3 (left). As can be seen in the figure, the addition of the coarse-to-fine scheme, structure threshold and subpixel refinement brings about a slight increase in performance over the original BFP when keeping the other parameters as they are. Tuning CtF-BFP then brings further improvement in both accuracy and robustness, as indicated by the increased acceptance and reduced rejection rates. To investigate the effects of the λ parameter, tuning was then re-run using $\lambda = 0$ and $\lambda = 1$. This corresponds to either maximising the number of accepted pixels without regard for possible outliers ($\lambda = 0$), or minimising the number of outliers among the estimated disparities without regard for coverage or average inlier accuracy ($\lambda = 1$). Results of these experiments are shown in figure 3 (right). As expected, $\lambda = 1$ produces a very sparse disparity estimate (less than 0.5% of pixels accepted on average) but with a rejection rate of 0.9%. With $\lambda = 0$ on the other hand, average acceptance is 75.5% with an average of 12% rejection rate.

Generalisation vs. adaptation: To evaluate the ability of the learned parameter sets to generalise to different scenes, tuning was carried out three times with different training images. For each of the training scenes, five images from different poses were used to tune the algorithm. Cross-validation was then performed using a sixth image from each set. Results of cross-validation are found in table 1. Performance is significantly improved for all scenes. The best performance is obtained on the same scene as the one depicted in the training set, although variability across scenes is low. This indicates that the algorithm balances adaptation to a particular scene and generalisation to novel ones.

6 Concluding remarks

We have introduced a method for generating dense wide-angle stereo ground-truth using the Microsoft Kinect. To improve ground-truth accuracy, we make use of error propagation from image plane measurements to properly balance errors in 3D and across multiple cameras in calibration. We have also shown how the BFP algorithm benefits from an extension to multiple scales, eliminating the need for seed correspondences from *e.g.* SIFT. Using our ground-truth disparity maps, we have automatically tuned our BFP extension to produce high-quality disparity maps from wide-angle images. This is done *without rectification or epipolar constraints*, which could be added [15] to further improve the results. Our experiments demonstrate that high-accuracy ground-truth can be obtained using our method, that the tuning procedure is capable of balancing accuracy, coverage and robustness, and that the parameters obtained generalise well to novel data from the same stereo rig. Ground-truth generation and subsequent automatic tuning should be useful to anyone wishing to adapt a stereo system to a specific setting, such as wide-angle stereo on a robot platform.

Acknowledgements: This work was supported by the Swedish Research Council through a grant for the project *Embodied Visual Object Recognition*, and by Linköping University.

References

[1] Kinect – Xbox.com.
http://www.xbox.com/Kinect.

[2] vision.middlebury.edu/stereo/.
http://vision.middlebury.edu/stereo/.

[3] OpenCV – open source computer vision library.
http://opencv.willowgarage.com/.

[4] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized PatchMatch correspondence algorithm. In *European Conference on Computer Vision*, LNCS, September 2010.

[5] Mårten Björkman and Jan-Olof Eklundh. Vision in the real world: Finding, attending and recognizing objects. *Int. J. of Imaging Systems and Technology*, 5(16):189–209, 2006.

[6] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC'11*, pages 1–11, 2011.

[7] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *ICML06*, pages 233–240, 2006.

[8] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1), 1975.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] Johan Hedborg and Björn Johansson. Real time camera ego-motion compensation and lens undistortion on GPU. Technical report, Linköping University, Department of Electrical Engineering, Sweden, 2007.

[11] Scott Helmer and David G. Lowe. Using stereo for object recognition. In *ICRA10*, 2010.

[12] C. Daniel Herrera, Juho Kannala, and Janne Heikkilä. Accurate and practical calibration of a depth and color camera pair. In *CAIP11*, 2011.

[13] Juho Kannala, Esa Rahtu, Sami S. Brandt, and Janne Heikkila. Object recognition and segmentation by non-rigid quasi-dense matching. In *CVPR08*, 2008.

[14] Maxime Lhuillier and Long Quan. Match propagation for image-based modelling and rendering. *IEEE TPAMI*, 24(8):1140–1146, 2002.

[15] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE TPAMI*, 27(3):418–433, 2005.

[16] Charles Loop and Zengyou Zhang. Computing rectifying homographies for stereo vision. In *CVPR99*, 1999.

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60 (2):91–110, 2004.

[18] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems Journal*, 56(6):503–511, 2008.

[19] Tomás Pajdla, Tomás Svoboda, and Vacek Hlavac. *Panoramic Vision: Sensors, Theory and Applications*, chapter Epipolar Geometry of Central Panoramic Cameras. Springer Verlag, 2001.

[20] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[21] Shishir Shah and J. K. Aggarwal. Mobile robot navigation and scene modeling using stereo fish-eye lens system. *Machine Vision and Applications*, 10(4):159–173, 1997.

[22] Jan Smisek, Michal Jancosek, and Tomás Pajdla. 3d with kinect. In *ICCV 2011 Workshops*, 2011.

[23] C. Strecha, W. von Hansen, L. van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR08*, June 2008.

[24] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE TPAMI*, 32(5):815–830, 2010.

[25] Christian Unger, Eric Wahl, Peter Sturm, and Slobodan Ilic. Probabilistic disparity fusion for real-time motion-stereo. In *ACCV10*, 2010.