Efficient Hybrid Tree-Based Stereo Matching With Applications to Postcapture Image Refocusing

Dung T. Vu, Benjamin Chidester, Hongsheng Yang, Minh N. Do, *Fellow, IEEE*, and Jiangbo Lu, *Member, IEEE*

***Abstract*— Estimating dense correspondence or depth information from a pair of stereoscopic images is a fundamental problem in computer vision, which finds a range of important applications. Despite intensive past research efforts in this topic, it still remains challenging to recover the depth information both reliably and efficiently, especially when the input images contain weakly textured regions or are captured under uncontrolled, real-life conditions. Striking a desired balance between computational efficiency and estimation quality, a hybrid minimum spanning tree-based stereo matching method is proposed in this paper. Our method performs efficient nonlocal cost aggregation at pixel-level and region-level, and then adaptively fuses the resulting costs together to leverage their respective strength in handling large textureless regions and fine depth discontinuities. Experiments on the standard Middlebury stereo benchmark show that the proposed stereo method outperforms all prior local and nonlocal aggregation-based methods, achieving particularly noticeable improvements for low texture regions. To further demonstrate the effectiveness of the proposed stereo method, also motivated by the increasing desire to generate expressive depth-induced photo effects, this paper is tasked next to address the emerging application of interactive depth-of-field rendering given a real-world stereo image pair. To this end, we propose an accurate thin-lens model for synthetic depth-of-field rendering, which considers the user-stroke placement and camera-specific parameters and performs the pixel-adapted Gaussian blurring in a principled way. Taking ∼1.5 s to process a pair of 640 x 360 images in the off-line step, our system named Scribble2focus allows users to interactively select in-focus regions by simple strokes using the touch screen and returns the synthetically refocused images instantly to the user.**

***Index Terms*— Stereo matching, depth estimation, cost aggregation, depth of field, post-capture refocusing.**

I. Introduction

DESPITE the long history of research in stereo matching, or stereo correspondence, this well-researched area continues to inspire new methods and to attract the attention of researchers due to its continued and extensive relevance within computer vision. Examples of applications that can benefit from fast and reliable depth-estimation from stereo matching exist in a diverse array of fields, ranging from computational photography, to robotics, to augmented reality, among others. For computational photography, depth information can enable the creation of novel artistic effects, such as depth-of-field rendering and depth-guided filtering, allowing for more creative and meaningful expressions of image content. To make these applications practical, the challenge remains to acquire accurate depth information in a computationally efficient manner, avoiding sophisticated optimization methods.

The significance of depth information for computational photography in particular has been demonstrated by the active search for novel depth-acquisition methods within this area of research. Examples of proposed methods include the work of Green *et al.* [2], which proposed a system of camera sensors to capture multiple images in a single exposure at different apertures and thereby estimate the depth of a scene. Levin *et al.* [3] and Bando *et al.* [4] proposed the addition of a special filter within the aperture of camera lenses (a patterned occluder, RGB color filter) to estimate depth. The recently developed Lytro camera [5] houses a microlens array integrated on a digital image sensor to capture the ray directions of the entire light field and is one of the first attempts to commercialize a computational technique of depth acquisition. Despite these recent advances, and even the development of the Lytro camera, these techniques have not yet reached widespread consumer adoption. Furthermore, some of the recent techniques and hardware, such as the PiCam [6] developed by Pelican Imaging, cannot avoid the need for efficient correspondence search to align the images captured from its camera array and to generate depth maps.

Though many algorithms that emphasize time-sensitivity have been proposed to solve the stereo correspondence problem, it still remains open to advances in estimation reliability and computation reduction. Additionally, along with the traditional difficulties of occlusion and depth discontinuities that must be handled by such an algorithm, an algorithm must also handle the imperfections of real-world images. Many proposed algorithms perform well on undistorted images that have been captured in controlled illumination, such as the standard Middlebury test cases. Nevertheless, in a real scenario, they are either too slow or fail to handle slight radiometric distortions from the camera, especially in large textureless regions, resulting in unsatisfactory depth estimation

Manuscript received October 1, 2013; revised March 10, 2014; accepted May 21, 2014. Date of publication June 5, 2014; date of current version July 1, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Janusz Konrad.

D. T. Vu and J. Lu are with the Advanced Digital Sciences Center, Singapore 138632 (e-mail: johan.vu@adsc.com.sg; jiangbo.lu@adsc.com.sg).

J. Lu thanks NVIDIA Corporation for providing the Tegra 3 prototype tablet. *(Corresponding author: J. Lu.)*

B. Chidester and M. N. Do are with the University of Illinois at Urbana-Champaign, Champaign, IL 61820 USA (e-mail: chidest1@illinois.edu; minhdo@illinois.edu).

H. Yang is with the University of North Carolina, Charlotte, NC 28223 USA (e-mail: yhs@cs.unc.edu). This work was mainly done when H. Yang was working at ADSC.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2329389

for some applications, particularly within computational photography.

Therefore, to achieve aesthetic results for such applications, we propose an efficient stereo matching technique that estimates depth more robustly and reliably than other competing methods, even in challenging, real-life cases. Our method is based upon a recently popular Minimum Spanning Tree (MST) representation [1], but extends it to a hierarchical, coarse-to-fine representation, with non-local cost aggregation and color-guided depth refinement.

Furthermore, to demonstrate the effectiveness of our algorithm and also motivated by the growing interest in depth-induced photo effects, we have evaluated it in an application within computational photography of post-capture refocusing. With the advent of mobile devices and the current popularity of mobile photography and image editing, this application is highly relevant. To this end, we also propose an accurate thin-lens model for synthetic depth-of-field rendering. Additionally, given the effectiveness of our stereo algorithm, we propose a system, called *Scribble2focus*, for interactive refocus-rendering using simple cues, either on the touch screen of a mobile device or on a PC. The system can operate on any setup that can capture stereoscopic images, such as a PC or a mobile device equipped with a pair of consumer webcams. Depending on the computational power of capturing devices, depth estimation can either be run on the device, or carried out on a remote server before sending back the estimated depth map for interactive refocusing rendering. In fact, the proposed algorithms could also be used to process a stereoscopic pair of images captured from two viewpoints by a mobile device with only a single camera, once the image pair is rectified with the recovered epipolar geometry.

The paper structure is organized as follows. Section II reviews existing depth estimation methods, including methods using stereoscopic images and methods using hardware modification. Section III discusses our depth map estimation and refinement algorithm based on a multi-MST construction. Section IV discusses how we model and render a real-life, depth-of-field effect. Section V explains the overview of our interactive, mobile photo-refocusing system. Finally, Section VI evaluates our algorithm on both the standard Middlebury benchmark for stereo matching and real-life stereoscopic image pairs captured using an Android tablet. We also discuss our proposed application, Scribble2focus, in this section.

II. Related Work

As previously mentioned, various methods of depth inference are currently available for computational photography applications, and we now provide a more thorough consideration.

Among many proposed algorithms for stereo matching, each one is optimized for different criteria, but still struggling to strike a desired balance between speed and inference accuracy. Generally, they can be grouped into one of two categories: global methods or local methods. Global methods enforce global consistency constraints upon the estimated depth map using regularization. The matching pixel pairs of the entire image are estimated simultaneously by minimizing a global energy function with a certain smoothness condition. To solve this minimization, various approaches exist. Among the most popular are dynamic programming [7], Markov networks [8] and graph cut [9]. Though they do not solve the minimization exactly, they perform well in many challenging region types, such as textureless regions and along depth discontinuities. However, since this work is concerned with applications for which fast processing speed is necessary, these methods are not considered, as they are computationally prohibitive.

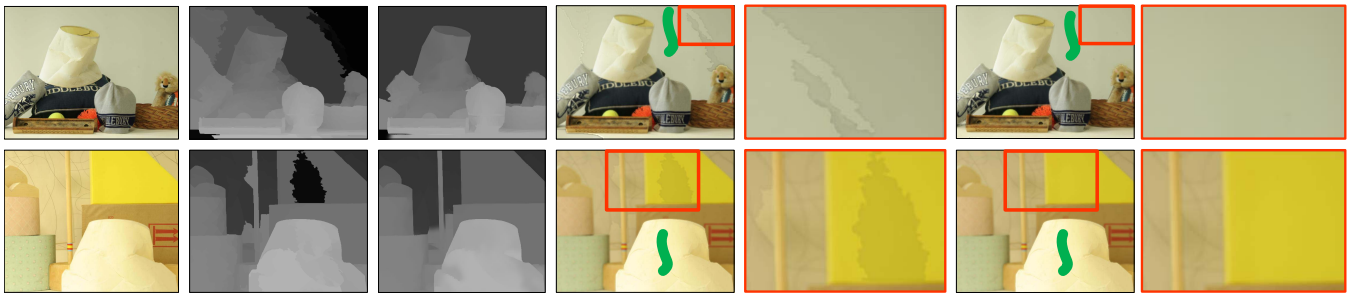
Dynamic programming methods attempt to reduce the computational burden of global methods while maintaining some level of global connectedness for inference by reducing the support of the smoothness constraint to individual scan-lines of the image, but this 1D relaxation usually suffers from the "streaking" effect due to the lack of enforced consistency between horizontal and vertical scan-lines.

Given the requirement of speed, local methods [10]–[12] are the most promising methods and are therefore the most comparable methods to that proposed. Local methods infer the disparity of each pixel independently, usually by comparing windowed regions around the reference pixel and the candidate matching pixel of the corresponding matching image. This approach results in faster, but often less reliable, depth inference. In particular, these methods struggle significantly in textureless regions due to matching ambiguity. The local methods that produce the best inference are those that are able to adaptively weight the support of the windows during matching to handle object boundaries. However, generating the weights is costly, as this amounts to translation-varying filtering. Although the guided filter [13] was recently proposed as an efficient method for computing adaptive support weights [14], it still struggles with these notoriously difficult textureless regions.

To have better representation of color patches and texture, some researchers have proposed the use of segmented color-images for stereo matching [15]–[17]. These region-based algorithms have smooth disparity estimates inside homogeneous regions, and color segments can help reduce the computation time. However, trusting in color segmentation alone is unreliable, as it may fail to provide a good representation of regions in the image, resulting in erroneous depth estimation.

Another important direction has been the adoption of tree-based structures for stereo matching. Veksler [18] and Deng and Lin [19] proposed to use tree-based graphical models to improve the performance of dynamic programming to solve the stereo correspondence problem. While Veksler [18] directly uses the image pixels as nodes in the tree, Deng and Lin [19] proposed that pixels of similar color should be grouped as line segments and a tree is constructed to connect all the line segments together. The tree-based dynamic programming algorithms are much faster than MRF-based global methods, however, the resulting accuracy is comparably weak and also outperformed by leading local methods [14], [20].

The most closely related work to the proposed algorithm is that of Yang, who proposed a non-local filter method [1]. This method uses a local pixel dissimilarity cost with a non-linear,



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

Fig. 1. Comparison between the non-local filter-based stereo matching algorithm [1] and our method on two challenging cases. (a) Original images. (b) Depth estimation result of the non-local filter algorithm (left) and our method (right). Depth-of-field rendering using (c) the non-local method's disparity map, and (d) our method's disparity map.

tree-based aggregation scheme to produce similar results to global optimization algorithms without incurring similarly excessive computational complexity. The dissimilarity cost between stereo image pairs is computed at each pixel, as in standard local methods. However, the image is represented as a planar graph, with each pixel being a node. Like in dynamic programing methods, the fully connected graph is reduced to a tree structure. In Yang's method, the chosen structure is a Minimum Spanning Tree (MST), which preserves connections between pixels of similar intensity. The aggregated cost at each pixel is computed by traversing the MST, so that every pixel contributes to the depth estimation of every other pixel, unlike standard local methods. Yang's algorithm produces competitive depth estimation with minimal computational cost. However, even this approach still has difficulty with some regions of typical, real-life images taken by commodity cameras, as shown in Fig. 1. In particular, the non-local filter fails to provide the correct depth estimates of large, textureless regions like the wall or the box of uniform color. Due to the particular illumination, the color intensity of the wall changes slightly from patch to patch, resulting in patches of different estimated disparity. Our method improves upon that of [1] to handle these challenging regions by introducing an additional, region-level MST, which significantly improves estimation quality with the computation time only slightly increased.

Other depth inference methods that require hardware modifications of the camera have also been proposed, and this area of research has recently received more attention, due in part to the popularity of the Microsoft Kinect camera [21]. The Kinect camera uses active infra-red illumination to find the scene's depth map in real time. However, the produced depth map has poor resolution, and furthermore, Kinect's active light system is susceptible to interference from other light sources and therefore might not work well with certain materials or in outdoor lighting. Coded aperture methods [3], [4] use a specially designed filter attached behind the camera's aperture to estimate the depth map of a scene from the depth-dependent, ray-diffusion characteristics of the filter. Zhou *et al.* [22] proposed the use of photometric cues by turning on and off the flash during capture of stereoscopic images to improve the quality of inference at depth boundaries and to overcome the challenge of occlusion. These methods, however, still require significant further development before they might be ready for widespread use. Whereas our method only requires two stereo cameras, and even a single camera, such as a camera from a mobile phone, could be used by simply capturing images from two unique perspectives, creating a parallax.

III. Depth From Stereoscopic Image Pairs

In this section, we present our method for depth inference from calibrated stereoscopic images. For applications in such areas as computational photography, robotics, or augmented reality, a stereo matching algorithm must effectively handle the challenge of real-world images while meeting computational constraints. For the specific application to image refocusing and editing, it must also preserve the edges of the scene in the depth map, as the sharpness of object edges highly influences the perceived visual quality of the edited image. Inspired by the strengths of both the MST cost aggregation method of [1] and region-based stereo matching, we propose a region-based enhancement to the MST. We extend the method of Yang [1] by adding a second MST, that is created from a segmented version of the image, where each node is a superpixel from the segmented image. The region-level MST enables aggregation over a coarse scale of the image, which is helpful for large regions of uniform color and texture, while the pixel-level MST enables aggregation over a finer scale, which is helpful for edge boundaries. In summary, our stereo algorithm performs the following steps: pixel-level cost initialization, pixel-level and region-level MST construction, adaptive cost aggregation on both MSTs, a Winner-Take-All strategy to estimate the disparity map, and disparity map refinement using a non-local disparity refinement method [1] followed by Cross-based Local Multipoint Filtering (CLMF) [23].

A. Depth From Multiple Minimum Spanning Trees

Fig. 2 shows the flow chart of our depth estimation and refinement process. Given the calibrated stereo image pair, I_0 and I_1, a disparity map D is recovered such that a pixel $I_0(p)$ at location $p = (u, v)$ in the reference image I_0 and a pixel $I_1(p_d)$ at location $p_d \equiv p + (d, 0)$, a d horizontally displaced pixel of p, in the matching image correspond to the same 3D point. A discrete disparity range, $H = [d_{min}, d_{max}]$,

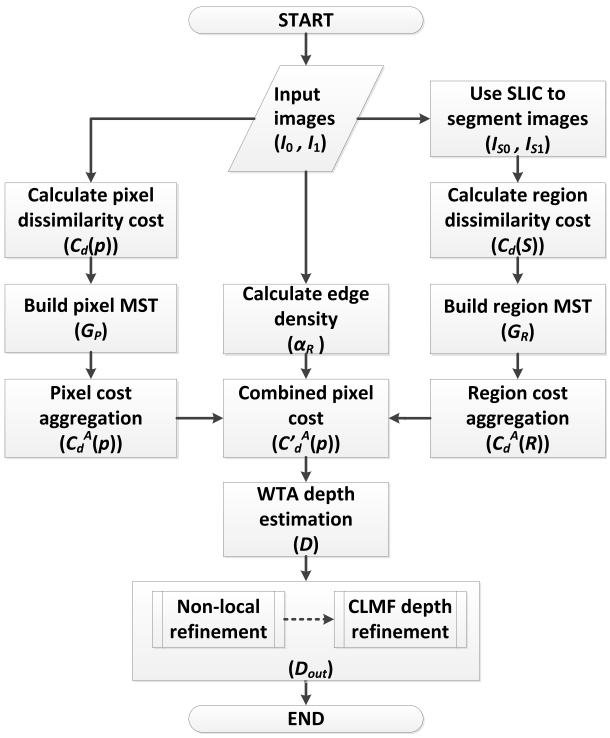


Fig. 2. Flow chart of the proposed stereo matching algorithm.

is specified based on the focal length of the cameras, the baseline between the stereoscopic images, and the desired depth resolution.

1) Pixel-Level Cost Computation and MST Construction: First, the matching cost between I_0 and I_1 is computed at each pixel for each disparity level. The common cost measurement, truncated absolute differences (TAD), is adopted to minimize the impact of outlier pixels. Similar to [20], the dissimilarity of pixels $I_0(p)$ and $I_1(p_d)$, denoted by $C_d(p)$, is given by a convex combination of the color dissimilarity e_i (measured in L1 distance) and the gradient difference e_g,

$$C_d(p) = \beta e_i(p, p_d) + (1 - \beta) e_g(p, p_d), \tag{1}$$

where e_i and e_g are defined as follows,

$$e_i(p, p_d) = \min\left(|I_0(p) - I_1(p_d)|, T_i\right),$$
$$e_g(p, p_d) = \min\left(|I'_0(p) - I'_1(p_d)|, T_g\right),$$

with $I_0(p)$ and $I_1(p_d)$ denoting the color vector and $I'_0(p)$ and $I'_1(p_d)$ denoting the horizontal gradient at the corresponding pixel. The two truncation parameters, $T_i = 8$ and $T_g = 2$, are set empirically to limit the negative impact of outliers. In experiments, the weight β is set to 0.11,

Yang [1] showed that cost aggregation on a MST produces quality disparity estimation. The MST connects all the vertices of the graph so that each pixel has support from every other pixel in the image, depending on their similarity, without the computationally expensive calculation of adaptive windows as in some accurate local methods. Following Yang's method [1], we construct the pixel-level MST by creating a planar graph $G_P = (V_P, E_P)$ and applying to it Kruskal's algorithm [25]. In the graph, each vertex represents a pixel and is connected via edges to its eight neighboring pixels. Each edge weight $\omega_P(p, q)$ between the two connected vertices (p, q) is given as

$$\omega_P(p, q) = |I(p) - I(q)|.$$

Applying Kruskal's algorithm on G_P generates a fully-connected MST. Since a tree has no cycles, it admits an efficient, recursive computation of the aggregated cost. Additionally, the MST in particular ensures that cost for a particular pixel is only aggregated over neighboring pixels of similar color. This results from the definition of the edge weights, since the MST has a total weight less than or equal to the total weight of every other possible spanning tree of G_P. Fig. 5 shows the connections, but not the edge weights, of the generated pixel-level MST for a patch from the Lamp Shade image.

2) Region-Level Cost Computation and MST Construction: As we discussed in Section II in motivation of our method, although the pure pixel-level MST method of [1] improves upon local methods in troublesome textureless regions and texture with large boundaries, it still produces unreliable depth estimates that are especially problematic for our considered applications, as seen in Fig. 1. To overcome these problems, we propose our modification of Yang's non-local filter method [1], which combines pixel-level and region-level cost computation and MST construction. We use the Simple Linear Iterative Clustering (SLIC) method of [24] to find the superpixel segmentations I_{S0} and I_{S1} for the respective input images, I_0 and I_1. The resulting superpixels, or regions, from SLIC adhere well to image boundaries. Additionally, SLIC provides us with the freedom to tweak the compactness of the generated superpixels, which is important, as the desired superpixel size in SLIC depends on the size of the input images. In our experiment with the NVIDIA tablet, the stereo images have a resolution of 640×360, for which we chose the superpixel size to be 150 pixels. Fig. 3b shows the superpixel segmentation result using SLIC. This size leads to a good balance between large coverage of color patches in textureless regions and minimal incorrect segmentation across object boundaries. The region cost $C_d(S)$ for each superpixel S, corresponding to a set of pixels in the original image, for each disparity d is given by the following expression,

$$C_d(S) = \sum_{p \in S} C_d(p), \tag{2}$$

where $p \in S$ are all the pixels inside the superpixel S.

After segmentation, the resulting superpixel image does not form a regular grid, so to represent the image as a graph $G_R = (V_R, E_R)$, each superpixel is represented by a node, and a connection is made between every node. However, as shown in Fig. 4, this results in connections between nodes that are not truly "neighbors", such as the red superpixel that is incorrectly connected to the superpixel R. For superpixels, we consider two nodes to be neighbors only if some pixel from one superpixel is a neighbor with a pixel from the other superpixel. To prevent the error in aggregation that would be caused by connections between non-neighbors, we penalize these

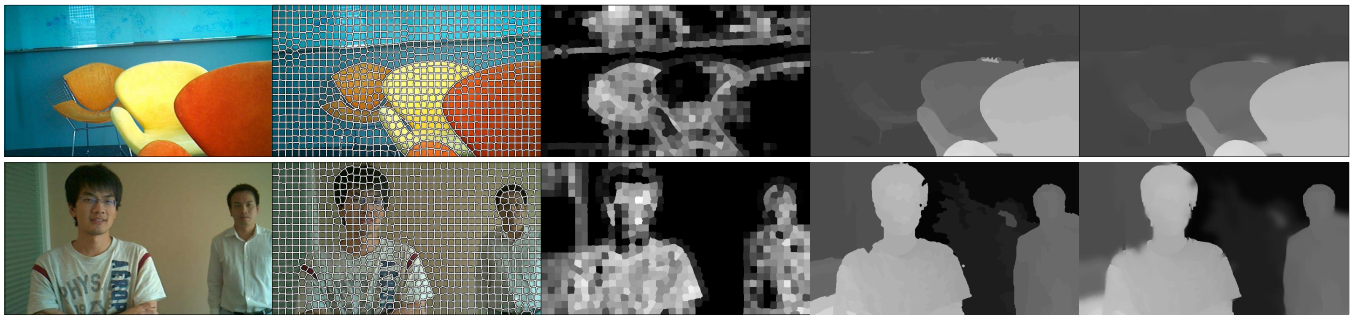


Fig. 3. Result from each step of disparity map estimation. (a) Original images. (b) Segmentation images from the SLIC algorithm [24]. (c) Edge density images. (d) Disparity maps without the CLMF-1 refinement [23]. (e) Disparity maps after the CLMF-1 refinement [23].

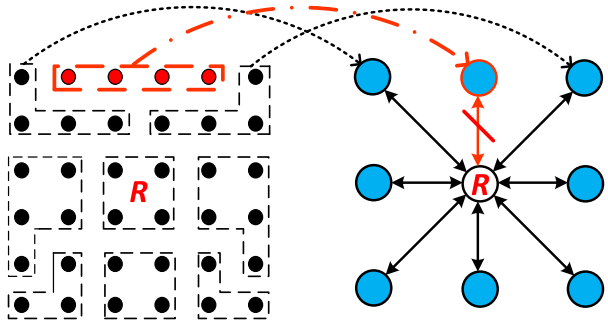


Fig. 4. Region graph builder. Superpixels resulting from the SLIC over-segmentation method are treated as nodes on an eight-connected undirected regular graph. The red colored superpixel does not share any neighbour pixel with superpixel R therefore the edge between them is penalized.

connections by setting their edge weights to the maximum value. For neighboring nodes, S and T, the edge weight between them is calculated based on the difference of the color distribution within each node. There are a variety of metrics that could be used to define the distance between color distributions. We compute the color histogram and use the difference of the *dominant* colors, I_S and I_T, also known as the modes, of each region as the metric. This metric is simple to compute and is more robust than the difference of the *mean* colors of the regions, as SLIC sometimes generates segments that slightly straddle regions of different color. The edge weight is then computed as

$$\omega_R(S,T) = |I_S - I_T|. \tag{3}$$

Finally, as in the construction of the pixel-level MST, we apply Kruskal's algorithm [25] on G_R to obtain the resulting region-level MST. Fig. 5 shows how the region-level MST looks like in a patch of the segmented Lamp Shade image.

3) Adaptive Fusion of Pixel-Level and Region-Level Costs: Once the pixel-wise cost for each disparity has been created, as well as the MSTs, the cost must be aggregated for each pixel. On each of the MSTs, we employ the non-local cost aggregation by Yang [1] to find the cost at each node. Let us consider the MST structure $T(V, E)$, where each $V_i \in V$ is a node and each $E_j \in E$ is an edge of T. We calculate the

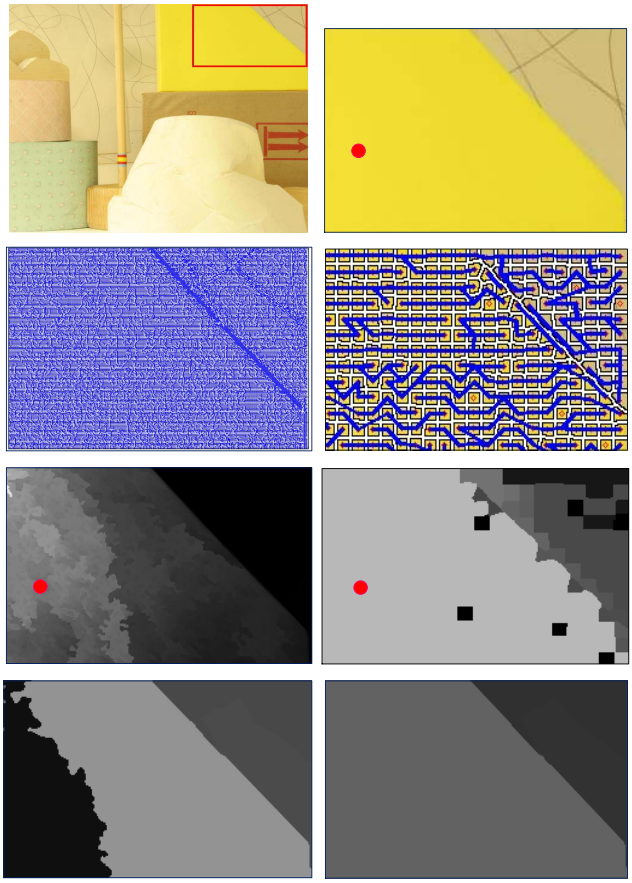


Fig. 5. Close-up examination of the multiple Minimum Spanning Trees. Row 1: (a) Original Lamp Shade image. (b) A close-up region from the original image with the red test point. Row 2: (a) Pixel-level MST. (b) Region-level MST. Row 3: the weighted contribution of all the pixels inside the close-up region to the test point according to (a) the pixel-level MST and (b) the region-level MST. Row 4: depth estimation result by (a) using the pixel-level MST only and (b) our method using adaptive fusion of the pixel-level and region-level MSTs.

distance $D(V_i, V_j)$ of the path $P(V_i, V_j)$ connecting V_i with V_j as the sum of all edge weights ω along the path,

$$D(V_i, V_j) = \sum_{\omega_E \in P(V_i, V_j)} \omega_E. \tag{4}$$

For each node V_i, $C_d(V_i)$ denotes the matching cost for disparity d and $C_d^A(V_i)$ denotes the aggregated cost. Based on the MST structure, we aggregate the cost for each node non-locally with weighted support from every other node in the tree $T(V, E)$

$$C_d^A(V_i) = \sum_{V_j \in T} W(V_i, V_j) C_d(V_j), \tag{5}$$

where weight $W(V_i, V_j)$ is calculated as an exponential function of the distance $D(V_i, V_j)$,

$$W(V_i, V_j) = \exp\left(-\frac{D(V_i, V_j)}{\sigma}\right). \tag{6}$$

We use σ to control the support in the cost aggregation process over the nodes. If we increase the σ value, distant nodes on the tree can provide a larger contribution. However, a large σ value has a trade-off. Large contribution of far distance nodes is good for low texture regions but it creates an error propagation problem for sharp edges and thin objects. Through experiments, we set the σ value of both pixel-level and region-level MST to be 0.1. Fig. 5 shows the weighted contribution of nodes on the pixel-level and region-level MSTs to a test point within a patch of the image.

In the cost aggregation process, Yang [1] utilizes the MST structure to efficiently compute the aggregated cost by two traversals of the tree. The aggregated costs are computed recursively by using the sum at previous nodes on the MST. Hence, the algorithm only requires a few operations per node. We apply this process on both the pixel-level and region-level MSTs to obtain $C_d^A(p)$ and $C_d^A(R)$ for each node.

After aggregating the cost for both the pixel-level and region-level MSTs, we can now see the shortcomings of the pixel-level MST when it alone defines the aggregation. Consider the case in Fig. 5. It can be seen that the pixel-level MST accurately estimated the weighted contribution of pixels near color and depth discontinuities. However, in the textureless region, the weighted contribution of neighboring pixels in the pixel-level MST was split among patches. This effect leads to the wrong depth estimation inside the textureless region. The region-level MST outperforms the pixel-level MST in this case, as it is able to correctly aggregate from support within the uniformly yellow region and more accurately estimate the depth for this region.

The key consideration is how to fuse the aggregation from each of the MSTs to produce more accurate depth estimation. Ideally, the region-level cost aggregation at a coarser level would act to complement the finer, pixel-level aggregation and vice versa. In a textureless region, which has no depth discontinuities, the region-level MST should dominate the cost aggregation and the resulting depth estimation. However, in a region of rich texture, the algorithm should rely more on the finer, pixel-level cost. In our algorithm, we use the edge density as a measure of the level of texture within a region. Then the pixel-level and region-level aggregated costs are adaptively blended in an unsupervised fashion, according to edge density, as follows

$$C_d'^A(p) = \alpha_R C_d^A(p) + (1 - \alpha_R) C_d^A(R), \tag{7}$$

where $p \in R$ and α_R is the edge density of the region R. To find the edge density as shown in Fig. 3c, we first apply the Canny edge detector [26] to find edges in the image and then calculate the density as the ratio of the number of edge pixels, N_e, to the number of pixels in the region, N_R,

$$\alpha_R = \frac{N_e}{N_R}. \tag{8}$$

Finally, a WTA optimization is applied to find the best disparity value at each pixel based on the combined aggregated cost $C_d'^A(p)$. Fig. 3d shows the resulting depth maps.

B. Disparity Map Refinement

For non-global methods, it is common practice to apply some form of post-processing refinement, usually relying upon smoothness or consistency constraints, to the generated disparity map D. We apply a two-step refinement process. We first employ the non-local refinement method of Yang [1]. In this method, we in turn consider the left and right images as the reference and find their respective disparity maps. A mutual consistency check is then employed to detect pixel pairs with consistent disparity values, and if pixels are found to be consistent, they are marked as stable. A new pixel dissimilarity cost is assigned to each pixel based on the its stability:

$$C_d^{new}(p) = \begin{cases} |d - D(p)| & p \text{ is stable and } D(p) > 0, \\ 0 & \text{else.} \end{cases} \tag{9}$$

We then run pixel-level aggregation on the pixel-level MST again. The cost of the stable pixels, and ultimately their depth values, are propagated to the unstable pixels, providing a more consistent depth map. However, this process can improve the quality mostly only for incorrect disparities that are caused by occlusion. In the case of large textureless regions, pixels might be wrongly classified as stable due to the ambiguity of matching. Therefore, we need a second refinement step that smooths large textureless regions and provides sharp depth discontinuities along object boundaries.

To accomplish this, we use the color image for guidance. The basic idea is to enforce the depth map to be coherent with the color image, based on the assumption that a local region of pixels that have similar color are likely to exist in the same disparity plane. The refined disparity map D_{out} is a locally filtered version of the original depth map, where the weights are based on the spatial and color similarity of pixels in a local neighborhood $\mathcal{N}(p)$,

$$D_{out}(p) = \sum\nolimits_{j \in \mathcal{N}(p)} w_{pj} D(j). \tag{10}$$

One commonly used weighting scheme is the bilateral filter. Here, we use both a faster $O(1)$ and more effective version called the cross-based multi-points filter (CLMF-1), which was recently proposed in [23]. The reason we choose CLMF-1 over another well-known $O(1)$ filtering technique–the guided-image filter (GF) [13]–is that the GF's kernel cannot handle more than two local color-line models. It has the tendency to blend them together to create a color blending effect when doing edge-preserving image smoothing, and a

fuzzy depth boundary is verified in the depth up-sampling experiment of [23]. The CLMF-1 explicitly adds a support region estimation procedure, leading to a sharply, well-aligned depth boundary in the filtering stage. Fig. 3e shows our disparity refinement results.

IV. Interactive Refocus Rendering

In the previous section, we presented our depth map estimation and refinement algorithm. The resulting depth map from our algorithm, D_{out}, has sharp edges that align well with objects. The disparity values of objects at the same depth level are smooth and consistent. The acquisition of such a depth map is of fundamental importance for various depth-based image editing processes in computational photography. To demonstrate the effectiveness of our stereo algorithm within this realm, we present an interactive refocus-rendering application. Provided with a pair of stereoscopic images captured by a user, our application processes these images to obtain the depth map of the scene. In the application, users are able to draw a simple stroke through an object or region in the scene to denote that it should be emphasized. If we were to use only color segmentation and Gaussian filtering, the application would easily fail, as non-connected objects at the same depth range would be rendered differently. In this application, we must intelligently infer an in-focus/out-of-focus effect, according to the user's input stroke and the scene's depth map, to render a real-life, depth-of-field (*d.o.f.*) image. We also develop the proper method that should be used to render the refocusing effect that is true to the real physical model. Our method for this interactive refocus-rendering application is discussed in two subsections: the physical model of the *d.o.f.* effect based on the physical model of thin lens; and the rendering of the *d.o.f.* effect using a Gaussian point spread function (PSF) with a pixel-adapted sigma value.

A. Physical Model of Depth-of-Field Effect

In photography, light rays emanating from a point in 3D space pass through a lens and are focused onto an image plane, which is a given distance behind the lens. At a certain distance from the camera, these light rays will converge to an exact point in the image plane, and as a result, the object will appear sharp in the captured image. If the object is too close to or too far from the camera, the point of convergence of these rays will be either behind or in front of the image plane, and the object will appear blurred. This is caused by the diffusion of the light rays throughout a region of the image, which is usually modeled as a circle and is called the *circle of confusion*, although in actuality, the shape of this region depends on the shape of the aperture. These effects are visualized in Fig. 6. Although a point in the scene may project onto the image plane with a circle of confusion of non-zero diameter, so long as the diameter is below a specified threshold, denoted C_T, it is still considered to be in focus. The range of the distance from the camera for which an object is considered to be in focus is the *d.o.f.*

Based on the camera's intrinsic values and specifications (sensor size, focal length, baseline distance), we convert the

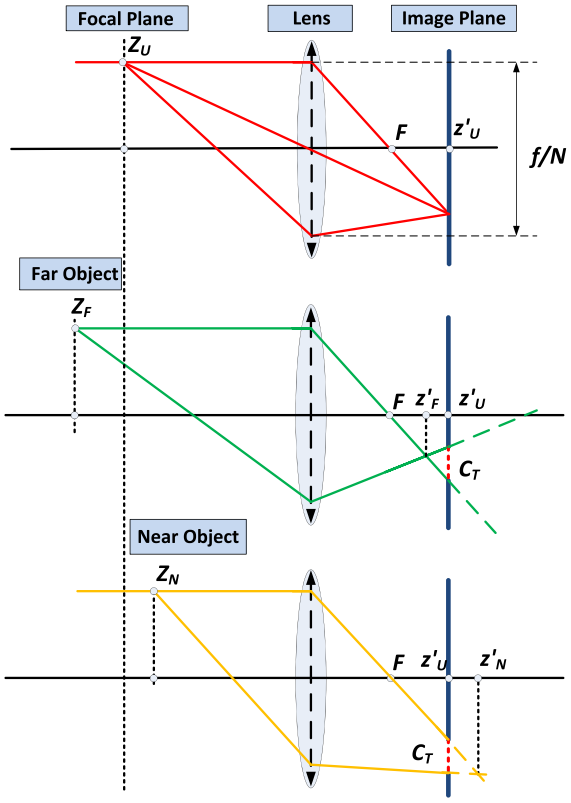


Fig. 6. A thin lens model of how the objects are captured on an image plane. Row 1: An object focuses on a image plane. Row 2: Farthest distance to the camera for objects to have a permissible circle of confusion size. Row 3: Nearest distance to the camera for objects to have the permissible circle of confusion size.

inferred disparity map into the real depth of the scene. From the input of a user's stroke and the depth map D_{out}, we interpret the user's selection depth Z_U and the desired in-focus region. If the user's stroke passes through multiple depth planes which are far from one another, then the user has defined the *d.o.f.* directly. The in-focus depth range will extend from the nearest depth plane Z_N to the farthest depth plane Z_F, and thus it is not necessary to define a permissible threshold C_T. In this case, Z_U is chosen to be

$$Z_U = \frac{1}{3}(Z_F - Z_N) + Z_N. \tag{11}$$

If the user's stroke lies in only one depth plane, the algorithm considers Z_U to be the depth of the user's selected stroke and the *d.o.f.* is calculated as follows (the derivation of which is presented in Appendix A),

$$d.o.f. = [Z_N, Z_F], \tag{12}$$

$$Z_N = \frac{f^2 Z_U}{C_T N(Z_U - f) + f^2}, \tag{13}$$

$$Z_F = \frac{f^2 Z_U}{C_T N(Z_U - f) - f^2}. \tag{14}$$

As previously mentioned, if a point p lies outside of the *d.o.f.*, then it will appear blurred in the image based on its distance from the camera sensor. To render the depth-based refocusing effect, we therefore need to calculate the diameter

TABLE I
SYMBOLS AND DESCRIPTIONS FOR *d.o.f.* CALCULATION

Symbol	Description
Z_U	User selection depth
Z_N	Nearest distance within depth-of-field range
Z_F	Farthest distance within depth-of-field range
N	f-number (camera intrinsic value)
C_T	Permissible size of circle of confusion
f	Focal length (camera intrinsic value)

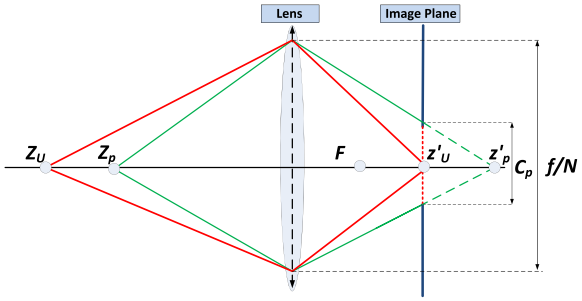


Fig. 7. If an object is out of focus range, it will create a circle of confusion on the image plane.

TABLE II
SYMBOLS AND DESCRIPTIONS FOR CIRCLE OF CONFUSION SIZE CALCULATION

Symbol	Description
z'_p	Behind lens focus distance of point p
z'_U	Image plane to lens distance
C_p	Size of circle of confusion of point p
f/N	Aperture size at focal length f with f number N

of the circle of confusion of each point that lies outside of the *d.o.f.* using the similar triangle formula, as in Fig. 7:

$$\frac{C_p}{f/N} = \frac{z'_p - z'_U}{z'_p},$$
$$C_p = \frac{(z'_p - z'_U) f/N}{z'_p}. \tag{15}$$

In Eq. (15), z'_p stands for behind lens distance of point P. The behind lens distance can be directly converted from object's distance to the camera sensor using the thin lens model as shown in Fig. 12. Tables I and II summarize the symbols used for *d.o.f.* calculation and circle of confusion size calculation, respectively.

B. Depth-of-Field Rendering Using Gaussian Point Spread Function

To render a convincing and realistic refocused image I^R based on the proposed physical model of *d.o.f.*, we propose two primary techniques for the rendering process. Firstly, each pixel that is outside of the *d.o.f.* must be diffused throughout the region of its circle of confusion, which we accomplish through convolution with a blur kernel with spread based on the diameter of the circle of confusion. Secondly, in-focus objects must have sharp edges that do not leak into the blurred background and foreground areas, so we adapt the support of the blur kernel based on the depth of neighboring pixels.

Since our depth map refinement process, which uses the color image as guidance, provides sharp, well-aligned depth boundaries, we can safely rely on the produced depth map D_{out} to not cause edge leaking problems in the refocused image. A hard threshold on the depth is set based on the previously calculated *d.o.f.*, so that, if the depth value of the pixel is within the *d.o.f.*, then its color intensity is stored directly in the refocused image I^R, and any blur kernel that is applied to the image excludes this point from its support. For other points which are outside of the *d.o.f.*, an adaptive Gaussian PSF is applied. The diffusion of light rays from an out-of-focus object in the camera's image plane is similar to the distribution of the energy of a Gaussian kernel of appropriate spread. Therefore, the *d.o.f.*-rendered image I^R is generated according to the following relationship:

$$I^R(p) = \begin{cases} I(p) & D_{out}(p) \in [Z_N, Z_F], \\ (I * G)(p) & D_{out}(p) \notin [Z_N, Z_F], \end{cases} \tag{16}$$

where the filter coefficient for the Gaussian PSF $G_p(u, v)$ is defined for each point p adaptively with relative image coordinate (u, v) as

$$G_p(u, v) = \frac{1}{2{\sigma_p}^2} \exp\left(\frac{-(u^2+v^2)}{2{\sigma_p}^2}\right) \cdot \delta\left(D_{out}(p + (u, v))\right), \tag{17}$$

In Eq. (16), $*$ denotes the convolution operator. The binary function $\delta(D_{out}(p + (u, v)))$ in Eq. (17) evaluates whether $D_{out}(p+(u, v)) \notin [Z_N, Z_F]$. If the condition is true, the filter kernel mask at point $p + (u, v)$ is activated; otherwise, it is set to 0 to avoid color bleeding between adjacent in-focus and out-of-focus regions. To model the out-of-focus effect, points in the scene with a larger circle of confusion should be more heavily blurred and points with a smaller circle of confusion should incur less blur. Therefore, we relate the spread of the kernel, parametrized by σ_p, linearly to the circle of confusion for each pixel p as follows:

$$\sigma_p = K * \frac{C_p}{p_s}, \tag{18}$$

where K controls the linear relation and p_s is the sensor's pixel size. Although we have assumed a circular shape of the aperture in this paper, other real-life aperture shapes could also be implemented to further improve this artistic effect.

Fig. 8 shows our *d.o.f.* rendering result as well as the value of sigma at each pixel, which is displayed in the lower graph, according to various user strokes. The graph shows how the value of sigma changes according to the depth of scene and the user-selected *d.o.f.*

V. INTERACTIVE MOBILE PHOTO REFOCUSING SYSTEM

Though any camera pair can be used to capture stereoscopic images, motivated by the mobile computational photography trend, we specially pick NVIDIA's Tegra 3 Android tablet [27] as an image acquisition and manipulation device. It is a powerful tablet, which houses a quad-core CPU and a dedicated GPU. At the current stage of the development, we use the tablet's stereo cameras to acquire images. The stereo

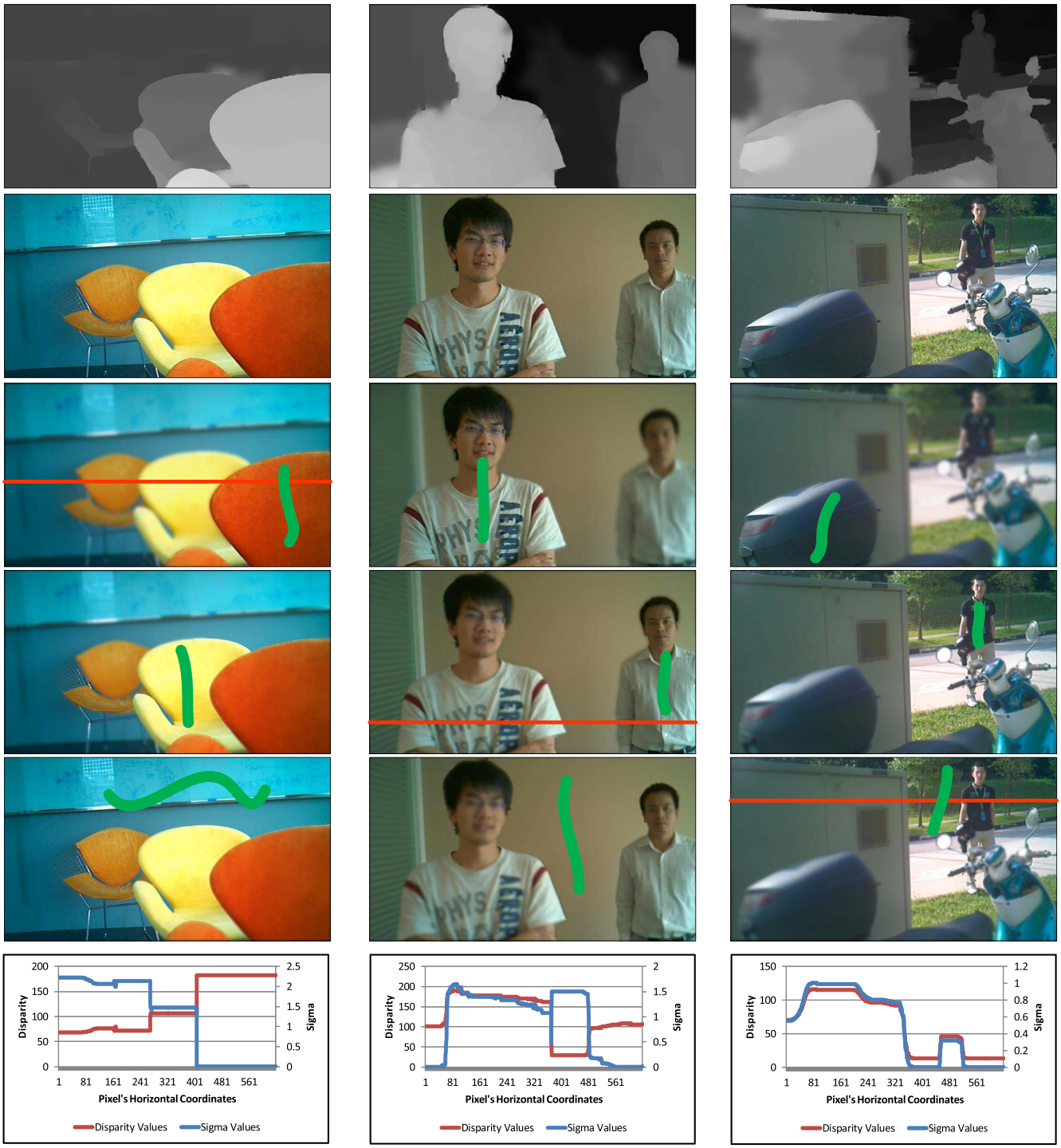


Fig. 8. Depth-of-field rendering result with different focus regions. Green strokes are users' strokes to select in-focus areas. Row 1: Estimated and refined depth map. Row 2: Original all in-focus color images. Row 3-5: Rendering result. Row 6: Gaussian PSF sigma value (blue) and estimated disparity value (red) of each pixel on the red horizontal line. This figure is best viewed on screen.

image pairs are then transferred to the PC for testing of our stereo matching correspondence and refocus-rendering algorithm. In addition to robustness, all of the components of our algorithm are designed with close attention also to efficiency, so that the system may easily be ported as an Android application to run on the tablet, which is our immediate future work.

A. System Overview

Fig. 9 presents our system design, which is separated into interactive and pre-processing sections. The first step is the calibration of the stereo cameras, which involves calculating the intrinsic and extrinsic parameters of the cameras. This is accomplished by the standard checker-board calibration method. On our mobile system, the stereo images

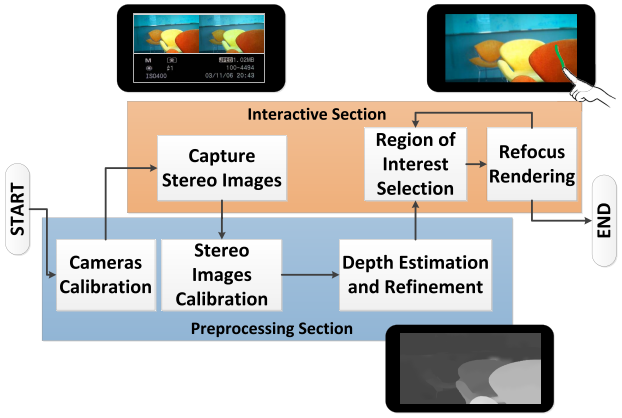


Fig. 9. Interactive mobile refocusing system overview.

are captured by an Android application that is built upon the FCam API [28]. After users capture the images, the system runs image processing in the background. The stereo images are rectified and tone calibrated. The corresponding depth map from this image pair is estimated and refined. Users can choose any color image to start the interactive refocus-rendering process. Strokes are drawn on the image in the GUI to indicate the region of interest in the image. On a mobile device, this could be done easily on its touch screen. The application then renders a depth-of-field effect on the image, which keeps the region of interest in focus while synthetically defocusing other regions according to their distance from the camera.

B. Camera Calibration

The standard checkerboard method is employed to find the cameras' intrinsic and extrinsic parameters, followed by estimation of the relative position and orientation between the two cameras. We compute the rectification transformation to enforce that the cameras' corresponding epipolar lines on the left and right images have the same y-coordinate, so that the left and right images are shifted only by horizontal disparities. The intrinsic parameters of the cameras are also required for the later computation of the depth-of-field effect. Additionally, we balance the color between the left and right images by employing a Grey World algorithm [29]. This algorithm is especially good at the removal of the color cast problem of digital images. The algorithm is based upon the assumption that images which are captured from different camera sensors or attributes of the same scene would converge to a similar mean color. Therefore, we compute the mean color of each image and then transform each image according to the average of the mean colors. After this calibration step, the output stereo image pairs are appropriately color-balanced and rectified.

VI. Experiment and Evaluation

In this section, we evaluate our depth estimation and refinement algorithm and discuss a depth-based post-processing application, Scribble2focus–an application for interactive refocusing. We use the Middlebury 2003 dataset to compare the ranking of our proposed stereo method with other state-of-the-art algorithms in stereo matching. The Middlebury 2006 dataset is also used to further challenge our algorithm with larger images and additional difficult test cases. In addition, our algorithm is evaluated on images captured by the NVIDIA Tegra 3 tablet. The tablet's camera sensors have the following specifications:

- Baseline: $b = 65mm$,
- Focal length: $f = 10.11mm$,
- Sensor size: $4592\mu m \times 3423\mu m$,
- Pixel size: $1.75\mu m \times 1.75\mu m$,
- f-Number: $N = 2.8$.

The depth estimation and refinement process is applied using a PC with an Intel Quad Core 2.8Ghz and 4GB RAM. For an image pair of 640×360 resolution, our algorithm takes an average of 1.55s to produce the final refined disparity map D_{out}, with 60 pixels as the maximum disparity, which includes 245ms to generate the superpixel segmentation with a default superpixel size of 150 pixels, 725ms to construct the multi-MST and aggregate the cost, and 585ms to refine the disparity map. In the case of images of a smaller resolution of 384×288 and a smaller maximum disparity of 19 pixels, such as the Tsukuba test image, our method only takes 0.47s to estimate and refine the disparity map.

A. Middlebury Stereo Matching Evaluation

We evaluate our depth estimation algorithm using the quality assessment method proposed by [30] and the Middlebury stereo database [31]. The parameters are set to constant values across all the test datasets: $\beta = 0.11$, $T_i = 8$, $T_g = 2, \sigma = 0.1$. Evaluation is based on the rate of wrong disparity values over the entire image and over three different regions: non-occluded regions, discontinuous regions, and all regions. Four standard datasets–Tsukuba, Venus, Teddy, and Cones–were used to obtain evaluation results and temporary rankings at the time of submission. Fig. 10 shows our depth estimation results, both with and without our CLMF depth refinement method in comparison with the result of the non-local filter [1] and the ground truth. Visually, our algorithm produces better estimates of the depth of non-occluded regions. For example, our algorithm successfully infers the depth of the top-right portion of the wall in the Tsukuba case. It also provides a better result in the Teddy case in the area to the left of the teddy bear. The repetitive background pattern misleads the non-local filter algorithm into computing an unreliable pixel matching cost. Cost aggregation using only the pixel-level MST further propagates this error, leading to patches of incorrectly estimated disparity around the teddy bear. In our algorithm, the integration of a region-level MST into cost aggregation helps to suppress the propagation of this error. Fig. 10d shows that the application of depth refinement using the corresponding color image further preserves the edges of objects, such as the table in the Tsukuba case, the cones in the Cones case, and the papers in the Venus case.

Table III presents our method's quantitative performance on four different test cases from the Middlebury dataset in comparison to state-of-the-art methods either focusing on the cost aggregation step or using the tree structures. Based solely upon quantitative comparison, we found that our algorithm performs slightly better without the CLMF-based depth refine-

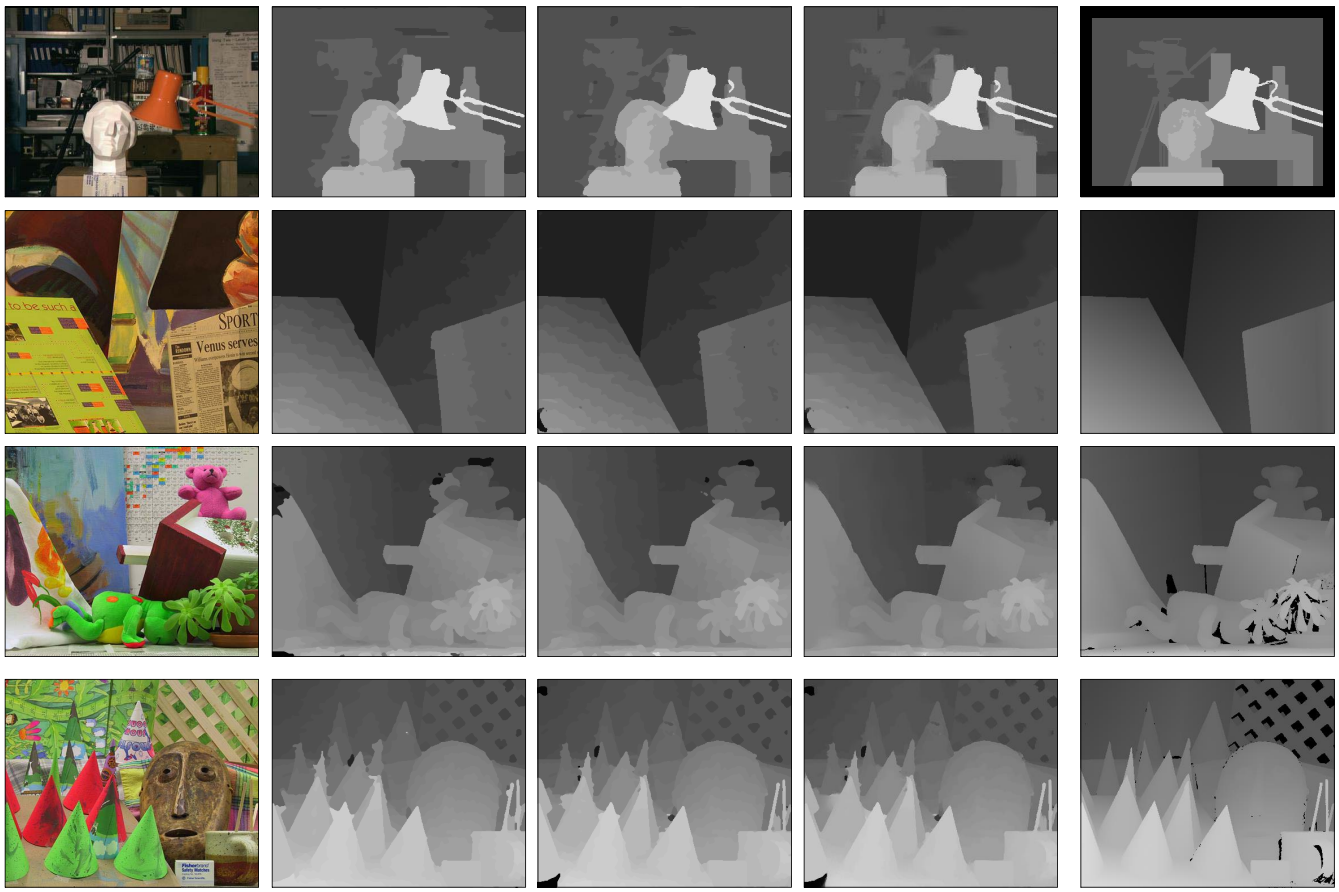


Fig. 10. Results on four Middlebury stereo vision datasets: Tsukuba, Venus, Teddy, and Cones. (a) Original left image. Disparity map of (b) non-local cost aggregation method [1], (c) our method without CLMF depth refinement, (d) our method with CLMF depth refinement, and (e) ground truth.

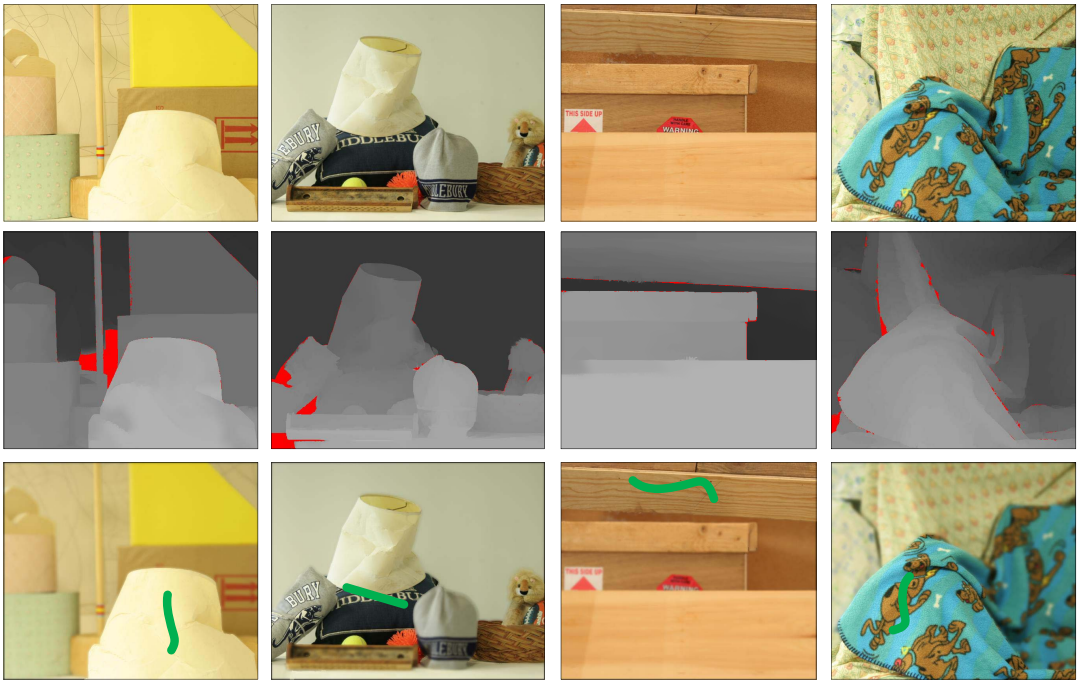
ment, so the results reported in Table III were generated without the use of this refinement. The better quantitative performance that is achieved by bypassing this additional refinement step is likely due to the disparity averaging that is performed during the color-guided refinement, in addition to discrepancies with the integer-valued ground truth. This additional refinement creates sharper edges, which is desirable for our refocusing application, but it may diffuse incorrect depth values into surrounding regions through convolution with the non-linear kernel. These errors do not impact the visual quality of our refocused images, but if quantitative performance is the primary consideration, it may be preferable to leave out this additional processing step. Generally, our algorithm achieved the temporal rank 20^{th} (out of nearly 170 methods) as of March 2014 while having competitive runtime efficiency. Our algorithm is ranked above some sophisticated local methods, such as Patch Match (rank 26^{rd}) [32] and also the original non-local filter (rank 40^{th}) [1].

To further evaluate our method's performance, we use the challenging Middlebury 2006 dataset. The second row of Fig. 11 shows the result of our depth estimation and refinement method. The red-colored pixels mark errors in the estimated disparity map as compared to the ground truth of each test case. Generally, our stereo matching method excels in large, low texture regions and provides good edge preservation because of the guidance of the color image in the refinement process. The disparity of large, uniform-color objects, such as the yellow boxes in Lamp Shade and the wall in Middlebury, are well estimated with piece-wise smooth disparity values. The edges in all the cases are sharp and well-preserved with few errors, especially in the Wood case. Visual comparison with the non-local method [1] is provided in Fig. 1.

For the purpose of evaluating the processing speed, we have run the non-local algorithm [1] source code on our Quad Core CPU with single-core implementation. The non-local algorithm takes an average processing time of 0.7 seconds while our method takes an average processing time of 1.01 seconds on the Middlebury 2003 dataset. Such a tree-based aggregation structure has provided a speed advantage over other stereo matching algorithms: OverSegmBP [38] takes 50 seconds, GlobalGCP [36] takes 130 seconds, and FastBilateral [39] takes 32 seconds. Compared to the non-local method, our algorithm is slower by 0.31 seconds, which is due to the additional computation required to build the region-level MST and perform region-level cost aggregation. However, we believe that the increase in estimation accuracy, as shown by the difference between rankings on the Middlebury dataset, and the better handling of challenging cases, as shown in Fig. 1b, is worth the small trade-off in computation time for

TABLE III
MIDDLEBURY STEREO MATCHING EVALUATION

Method	Ranking	Tsukuba			Venus			Teddy			Cones			AP
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	BP
OUR METHOD	20	1.29	1.71	6.95	0.15	0.30	1.23	6.12	11.4	15.8	2.82	8.68	7.76	5.35
PMF [20]	22	1.74	2.04	8.07	0.33	0.49	4.16	2.52	5.87	8.30	2.13	6.80	6.32	4.06
SegmentTree [33]	23	1.25	1.68	6.69	0.20	0.30	1.77	6.00	11.9	15.0	2.77	8.82	7.71	5.35
Patch Match [32]	26	2.09	2.33	9.31	0.21	0.39	2.62	2.99	8.16	9.62	2.47	7.80	7.11	4.59
HistoAggr2 [34]	27	1.93	2.30	6.39	0.16	0.46	2.22	5.88	11.3	14.7	2.41	7.78	6.89	5.20
Impr NonLocal [35]	29	1.38	1.83	7.38	0.21	0.41	2.26	5.99	11.5	14.3	2.85	6.68	7.98	5.23
CrossLMF [23]	32	2.46	2.78	6.26	0.27	0.38	2.15	5.50	10.6	14.2	2.34	7.82	6.80	5.13
Cost Filter [14]	36	1.51	1.85	7.61	0.20	0.39	2.42	6.16	11.8	16.0	2.71	8.24	7.66	5.55
GlobalGCP [36]	39	0.87	2.54	4.69	0.16	0.53	2.22	6.44	11.5	16.2	3.59	9.49	8.95	5.60
Non-Local Aggr. [1]	40	1.47	1.85	7.88	0.25	0.42	2.60	6.01	11.6	14.3	2.87	8.45	8.10	5.48
RegionTreeDP [37]	71	1.39	1.64	6.85	0.22	0.57	1.93	7.42	11.9	16.8	6.31	11.9	16.8	6.56
OverSegmBP [38]	72	1.69	1.97	8.47	0.51	0.68	4.69	6.74	11.9	15.8	3.19	8.81	8.89	6.11
SegTreeDP [19]	80	2.21	2.76	10.3	0.46	0.60	2.44	9.58	15.2	18.4	3.23	7.86	8.83	6.82
FastBilateral [39]	89	2.95	4.75	8.69	1.29	2.87	7.62	10.71	19.8	20.82	5.23	15.3	11.34	7.31
ACTMF [40]	n.a.	n.a.	2.67	n.a.	n.a.	0.90	n.a.	n.a.	18.32	n.a.	n.a.	12.91	n.a.	n.a.



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

Fig. 11. Depth estimation result of our method on Middlebury 2006 dataset. First row presents the original left color images. Second row presents our depth estimation result together with error pixels (red color) in comparison with the ground truth. Third row shows the synthetic refocusing effect based on the depth estimation result.

most of the applications that we have considered, especially computational photography applications.

B. Scribble2focus – An Interactive Photo Refocusing System

We experiment with our interactive, post-capture refocusing application, using the NVIDIA Tegra 3 tablet to capture several test images under different indoor and outdoor conditions. We change the camera parameters manually to capture all-in-focus images. These images are passed to our Scribble2focus application, which performs image rectification, image calibration, depth estimation, and *d.o.f.* effect rendering. The calibrated images have a resolution of 640 × 360. Our PC required an average of 0.7 seconds to calculate the *d.o.f.* model from the color and depth images and render the *d.o.f.* effect according to the input scribble of the users. Fig. 8 shows the input and output result of our Scribble2focus application. The first row of the figure shows our depth estimation and refinement result on three real-world cases captured by the tablet. The result is not perfect, as our method still incurs error in low contrast areas or areas of highly-varying texture. However, the visual quality of the estimated depth is good; we can easily identify objects of different depth with well preserved depth discontinuities. Through the Scribble2focus interface, users simply mark a green scribble on captured color images to indicate their region of interest. Rows 3-5

TABLE IV
SSIM COMPARISON OF THE NON-LOCAL AGGREGATION METHOD [1] WITH AND WITHOUT OUR CLMF REFINEMENT AND OUR METHOD

Test Cases/SSIM Values	Non-Local Method [1]	Non-Local Method [1] with our CLMF refinement	Our Method
Middlebury	0.9527	0.9601	**0.9719**
Lamp Shade	0.9806	0.9844	**0.9927**

of Fig. 8 show the resulting real-to-life *d.o.f.* effect, which is rendered according to users' scribbles. The last row of Fig. 8, which shows the adaptive Gaussian PSF's sigma value for a scanline from each image, provides intuition of the blurring effect for a given selected region. The plots also justify the importance of our physical thin-lens based refocusing model, as the relationship between the Gaussian blurring level and the disparity value is *not* straightforward. In fact, the blur kernel size for a given point is jointly decided by the user-scribble placement, the stereo image depth range, and the camera-specific parameters such as N, f and p_s.

1) Chairs Test Case in Fig. 8(a): In this test case, each chair lies in a different depth layer and is quite far apart from the others. The color image has relatively low texture because each chair consists mostly of one color. Note that another challenge in this test case is the whiteboard, which causes a strong reflection and also creates a strong color border with the wall, which might result in different depth values on either side of the border. Our method successfully estimates the depth layer of each chair and also the wall, and the depth map inside each layer is very smooth. The algorithm also does not make any mistakes with the wall and whiteboard. The *d.o.f.* rendering according to each selected chair also shows that our algorithm preserves the edges well, as the color of the chair does not leak into the surrounding out-of-focus areas.

2) Two People Test Case in Fig. 8(b): This test case was also taken indoors and the color of the plain wall and two main people are similar. Our algorithm is able to estimate and separate depth layers robustly, though some errors still exist at the transition boundary between depth layers. The main reason for this is the low contrast of the image. However, when we apply the *d.o.f.* rendering, the result is still visually acceptable.

3) Outdoor Test Case in Fig. 8(c): This test case was taken outdoors under strong sunlight. This setup is also challenging, as it contains slanted surfaces, thin objects, and large textureless regions. Our estimated depth map is visually acceptable–it clearly shows objects with correct depth discontinuities. Note that our method can even detect the connection between the head of the motorbike and the mirror. More importantly, our refocused images look visually plausible, creating convincing, *d.o.f.* effects based upon the user's strokes.

4) Middlebury 2006 Dataset in Fig. 11: To further demonstrate the performance of our application, we also test our *d.o.f.* rendering algorithm on the depth estimation result of the Middlebury 2006 dataset. According to the information that we were able to gather from the Middlebury 2006 dataset's website [41] and Scharstein and Szeliski's paper discussing the dataset [42], we assume that theses images were captured with focal length $f = 13.11mm$ and baseline $b = 160mm$. The last row of Fig. 11 shows our final rendering result for different input strokes from the user, drawn in green.

Fig. 1(c), 1(d) shows a comparison between *d.o.f.* rendering on depth estimation results from the non-local method [1] and from our method. The cropped region clearly shows how incorrect depth estimation in textureless region would create undesirable visual defects on the *d.o.f.*-rendered images. Additionally, to evaluate the quality of our resulting refocused images, we compared our method to the non-local aggregation method [1] quantitatively using the popular SSIM metric [43] for visual quality. The reference refocusing image is generated by feeding the ground-truth disparity map into our thin-lens based computational refocusing model. As shown in Table IV, our method outperforms the non-local aggregation method both with and without our CLMF refinement, though our CLMF refinement improves the refocusing results of the original non-local method.

VII. CONCLUSION

We have proposed an efficient stereo matching algorithm for fast processing that is based on pixel-level and region-level MST representations of a stereo pair of images. Fusion of aggregation over these MSTs, one being of a finer resolution of the image and one being of a coarser resolution, allows for better depth estimation over large, textureless regions while still preserving depth discontinuities at object boundaries. The result of our depth estimation method is superior to state-of-the-art local methods on the Middlebury benchmark. Experiments show that our method performs exceptionally well in the notoriously difficult low texture regions and is able to preserve sharp depth discontinuities. Furthermore, our method provides depth inference of high visual quality on challenging, real-world cases captured under different indoor and outdoor conditions. The addition of color-guided filtering of the disparity map using CLMF refines edge boundaries, resulting in sharper disparity discontinuities at object borders, which we have shown is important for depth-based computational photography applications. We plan to improve our algorithm's performance in areas of low contrast by further research in robust matching measures and improved inference.

We also evaluated this method in an application of refocus-rendering from computational photography. For this application, we derived a precise model of the lens to achieve a realistic *d.o.f.* effect. Motivated by this application, we presented an interactive depth-of-field rendering application named Scribble2focus that uses our proposed stereo matching algorithm to estimate depth. Our application enables the user to easily and interactively create a real-life, depth-of-field effect by simply drawing a stroke through the region to be emphasized. We carefully study the physical model of a real-life, depth-of-field effect and utilize the acquired depth map to guide the depth-based, pixel-adapted, Gaussian blurring in the rendering process. The Scribble2focus application provides

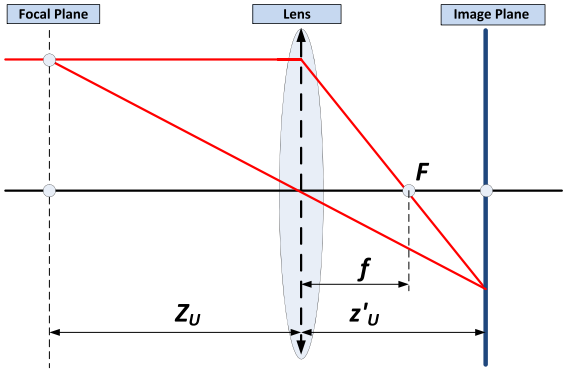


Fig. 12. Relationship between the focal length f, the object distance to camera Z_U and the object's image behind lens distance z'_U. According to the physical model of thin lens, z_U can be expressed in the form $z'_U = \frac{fZ_U}{Z_U - f}$.

users with quality, artistic, depth-of-field images with little effort. In future work, we plan to optimize the algorithm and port the entire framework to Android devices.

APPENDIX A
PROOF OF DEPTH OF FIELD RANGE

According to Fig. 6, we use the similar triangle formula to calculate z_F as

$$\frac{C_T}{f/N} = \frac{z_U - z_F}{z_F}$$
$$\Rightarrow z_F = \frac{fz_U}{C_T N + f}.$$

Fig. 12 shows how we use the thin lens model to calculate the behind lens distance. Application of this model gives us another expression of z_F and z_U in the form

$$z_F = \frac{fZ_F}{Z_F - f},$$
$$z_U = \frac{fZ_U}{Z_U - f}.$$

Therefore,

$$z_F = \frac{fZ_F}{Z_F - f} = \frac{fz_U}{C_T N + f}$$
$$\Rightarrow Z_F = \frac{fz_U}{C_T N + f - z_U}.$$

After substituting Z_U, the farthest distance value for object to be in focus is

$$Z_F = \frac{f^2 Z_U}{C_T N(Z_U - f) - f^2}.$$

Similarly, we calculate the nearest distance for objects to be in focus as

$$Z_N = \frac{f^2 Z_U}{C_T N(Z_U - f) + f^2}.$$

ACKNOWLEDGMENT

This work was supported by the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

REFERENCES

[1] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1402–1409.
[2] P. Green, W. Sun, W. Matusik, and F. Durand, "Multi-aperture photography," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 68:1–68:7, Jul. 2007.
[3] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 70:1–70:9, Jul. 2007.
[4] Y. Bando, B.-Y. Chen, and T. Nishita, "Extracting depth and matte using a color-filtered aperture," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 134:1–134:9, Dec. 2008.
[5] (2014, Jun. 13). *About the Camera*, Lytro, Mountain View, CA, USA [Online]. Available: http://www.lytro.com/
[6] K. Venkataraman *et al.*, "Picam: An ultra-thin high performance monolithic camera array," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 166:1–166:13, Nov. 2013.
[7] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *Int. J. Comput. Vis.*, vol. 33, no. 3, pp. 181–200, 1999.
[8] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 399–406.
[9] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. Int. Conf. Comput. Vis.*, Jul. 2001, pp. 508–515.
[10] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, Apr. 2006.
[11] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 556–561.
[12] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073–1079, Jul. 2009.
[13] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.
[14] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3017–3024.
[15] Y. Wei and L. Quan, "Region-based progressive stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004, pp. 106–113.
[16] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004, pp. 74–81.
[17] Y. Deng, Q. Yang, X. Lin, and X. Tang, "A symmetric patch-based correspondence model for occlusion handling," in *Proc. 10th Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1316–1322.
[18] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 384–390.
[19] Y. Deng and X. Lin, "A fast line segment based dense stereo algorithm using tree dynamic programming," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 201–212.
[20] J. Lu, H. Yang, D. Min, and M. N. Do, "PatchMatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1854–1861.
[21] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1297–1304.
[22] C. Zhou, A. Troccoli, and K. Pulli, "Robust stereo with flash and no-flash image pairs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 342–349.
[23] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do, "Cross-based local multipoint filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 430–437.
[24] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[25] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, Feb. 1956.
[26] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Jun. 1986.

[27] (2014. Jun. 13). *Tegra Developer Zone*. NVIDIA, Santa Clara, CA, USA [Online]. Available: http://developer.nvidia.com/tegra-start
[28] A. Adams *et al.*, "The frankencamera: An experimental platform for computational photography," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 29:1–29:12, Jul. 2010.
[29] A. C. Shumate and H. Li. *Gray World Algorithm*. [Online]. Available: http://scien.stanford.edu/pages/labsite/2000/psych221/projects/00/trek
[30] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.
[31] (2014, Jun. 13). *Middlebury Stereo Vision*, Middlebury College, Middlebury, VT, USA [Online]. Available: http://vision.middlebury.edu/stereo/
[32] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo—Stereo matching with slanted support windows," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 14.1–14.11.
[33] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-tree based cost aggregation for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 313–320.
[34] D. Min, J. Lu, and M. N. Do, "Joint histogram-based cost aggregation for stereo matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2539–2545, Oct. 2013.
[35] D. Chen, M. Ardabilian, X. Wang, and L. Chen, "An improved non-local cost aggregation method for stereo matching based on color and boundary cue," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2013, pp. 1–6.
[36] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground control points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3033–3040.
[37] C. Lei, J. Selzer, and Y.-H. Yang, "Region-tree based stereo using dynamic programming optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2378–2385.
[38] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 49–65, Oct. 2007.
[39] S. Mattoccia, S. Giardino, and A. Gambini, "Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering," in *Proc. Asian Conf. Comput. Vis.*, 2009, pp. 23–27.
[40] M. Mueller, F. Zilly, and P. Kauff, "Adaptive cross-trilateral depth map filtering," in *Proc. 3DTV-Conf., True Vis. Capture, Transmiss. Display 3D Video, 3DTV-CON*, Jun. 2010, pp. 1–4.
[41] (2006). *Stereo Datasets with Ground Truth*, Middlebury College, Middlebury, VT, USA [Online]. Available: http://vision.middlebury.edu/stereo/data/scenes2006/
[42] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 195–202.
[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Dung T. Vu received the B.S. degree in electrical engineering from the National University of Singapore, Singapore, in 2010. He is currently with the Advanced Digital Sciences Center, which was jointly founded by the University of Illinois at Urbana-Champaign, Champaign, IL, USA, and the Agency for Science, Technology, and Research, a Singapore government agency. His research interests include 3D computer vision, 3D reconstruction, and hybrid sensor systems.

Benjamin Chidester received the B.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2009, and the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2013, where he is currently pursuing the Ph.D. degree. He was a recipient of one-Year Research Fellowship from MIT Lincoln Labs in 2013. His interests include computer vision, image processing, and machine learning.

Hongsheng Yang received the B.Eng. degree in electronic information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2011. He was with the Advanced Digital Sciences Center, Singapore, as a Software Engineer till 2013. He is currently pursuing the Ph.D. degree in computer science with the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. His current research interests include 3D computer vision, large-scale scene understanding, and optimization methods.

Minh N. Do (M'01–SM'07–F'14) was born in Vietnam in 1974. He received the B.Eng. degree in computer engineering from the University of Canberra, Canberra, ACT, Australia, in 1997, and the Dr. Sci. degree in communication systems from the Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland, in 2001.

He has been on the faculty with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, since 2002, where he is currently a Professor with the Department of Electrical and Computer Engineering, and hold joint appointments with the Coordinated Science Laboratory, the Beckman Institute for Advanced Science and Technology, and the Department of Bioengineering. His research interests include image and multidimensional signal processing, wavelets and multiscale geometric analysis, computational imaging, augmented reality, and visual information representation.

Prof. Do was a recipient of the Silver Medal from the 32nd International Mathematical Olympiad in 1991, the University Medal from the University of Canberra in 1997, the Doctorate Award from the EPFL in 2001, the CAREER Award from the National Science Foundation in 2003, and the Young Author Best Paper Award from the IEEE in 2008. He was named a Beckman Fellow at the Center for Advanced Study, UIUC, in 2006, and received the Xerox Award for Faculty Research from the College of Engineering, UIUC, in 2007. He was a member of the IEEE Signal Processing Theory and Methods Technical Committee, the Image, Video, and Multidimensional Signal Processing Technical Committee, and an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Jiangbo Lu (M'09) received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering, Katholieke Universiteit Leuven, Leuven, Belgium, in 2009.

He was with VIA-S3 Graphics, Shanghai, China, from 2003 to 2004, as a Graphics Processing Unit Architecture Design Engineer. In 2002 and 2005, he conducted visiting research at Microsoft Research Asia, Beijing, China. Since 2004, he has been with the Multimedia Group, Interuniversity Microelectronics Center, Leuven, Belgium, as a Ph.D. Researcher. Since 2009, he has been with the Advanced Digital Sciences Center, Singapore, which is a joint research center between the University of Illinois at Urbana-Champaign, Champaign, IL, USA, and the Agency for Science, Technology and Research, Singapore, where he is leading a few research projects. His research interests include computer vision, visual computing, image processing, video communication, interactive multimedia applications and systems, and efficient algorithms for various architectures.

Dr. Lu is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He was a recipient of the 2012 TCSVT Best Associate Editor Award.