An Iterative Adaptive Multi-modal Stereo-vision Method using Mutual Information

Mustafa Yaman*, Sinan Kalkan

Dept. of Computer Engineering
Middle East Technical University
Ankara, Turkey
Email: {mustafa.yaman, skalkan}@ceng.metu.edu.tr

Abstract

We propose a method for computing disparity maps from a multi-modal stereo-vision system composed of an infrared-visible camera pair. The method uses mutual information (MI) as the basic similarity measure where a segment-based adaptive windowing mechanism is proposed along with a novel MI computation surface. The computed cost confidences are aggregated using s novel adaptive cost aggregation method, and the resultant minimum cost disparities in segments are plane-fitted in their respective segments. Finally, the estimated disparities are iteratively refined by merging and splitting segments according to the confident disparities, and in order to reduce the dependence of the disparity computation upon the initial segmentation, all these steps (*i.e.*, MI computation, cost aggregation, plane fitting, segment splitting and merging) are repeated. On an artificially-modified version of the Middlebury dataset and a Kinect dataset that we created in this study, we show that (i) our proposal improves the quality of existing MI formulation, and (ii) our

*Corresponding author

method can provide depth comparable to the quality of Kinect depth data.

Keywords: multi-modal stereo-vision, mutual information, adaptive windowing, adaptive cost aggregation, iterative stereo, RGB-D

1. Introduction

(a) (b)

Figure 1: An example illustrating the difficulty of finding correspondences in an IR-RGB image pair. **(a)** The RGB image. **(b)** The IR image. [Best viewed in color].

Using multi-modal cameras for surveillance systems has been popular since the year 2000 [1, 2, 3, 4] since using cameras of different modalities, such as a pair of infrared and visible cameras, has advantages over using unimodal cameras in surveillance systems. These advantages include being able to work under low visibility or lighting conditions, better segregation of a target from the background, allowing a richer set of information like thermal signatures in the scene or the different reflectance properties of objects in different bands of the electromagnetic spectrum etc. When considering to

enhance the performance and usefulness of such multi-modal systems, the question of whether stereo-vision from multi-modal cameras can yield an accurate depth information or not has attracted well-deserved attention. One reason for this attention is that, for such systems, the distance of an intruder or the depth map of the scene under surveillance is very valuable.

A powerful method for computing depth from multiple cameras is stereo-vision. Stereo-vision [5, 6] deals with computing depth by finding the corresponding pixels in different views. The correspondences, which are generally determined by comparing intensities of pixels, are used for computing the 3D positions using simple triangulation. It is one of the most studied problems of Computer Vision - for reviews, see [5, 7, 8, 9, 10, 11]. Stereo-vision methods are mainly clustered around two main axes: *Sparse or feature-based* methods (*e.g.*, [8, 18]) vs. *dense* methods (*e.g.*, [10, 19]); and *local* methods (*e.g.*, [20, 22]) vs. *global* methods (*e.g.*, [29, 31]) . The former grouping describes whether correspondences (and therefore the pixel disparities) are computed for all the pixels in the images (*i.e.*, the dense methods), or only for some reliable features (such as salient points, edges, corners, curves etc.) extracted from the images. Regarding the latter grouping, local methods use only the local neighborhood and intensity information for finding stereo correspondences. Global methods, on the other hand, use global constraints to correct false correspondences that would be otherwise impossible to correct locally.

Although classical stereo-vision techniques have had tremendous success in terms of both accuracy and running time, they are not directly applicable in a multi-modal setting. The reason is that computing similarities between intensities of pixels or windows will not work using unimodal matching meth-

ods simply because the intensities of the corresponding pixels will be different. For example, an RGB-thermal image pair would have totally different intensities for corresponding pixels (see, *e.g.*, Figure 1). This study aims to investigate how to compute reliable stereo correspondences for such an image pair and compute its depth information.

1.1. Related Studies

Stereo-vision from multi-modal cameras was not studied much until the 2000's. The earliest of such studies, per the authors' knowledge, is from Egnal [22], who, influenced by Viola's studies of multi-modal registration [33], applied mutual information (MI) as the basic similarity measure for stereo correspondence. Egnal tested his method on images that were made multi-modal by red-blue filtering or altering the illumination of the different views. The results were promising and revealed the power of MI compared to standard correlation-based methods, especially on images with different spectral characteristics. However, using MI still not produced depth information of sufficient quality.

Fookes *et al.* extended the MI-based approach with adaptive windowing [34] and integrated prior probabilities using a 2D matching surface [35]. However, their methods were only tested on synthetically-altered unimodal images, which do not actually include different segmentation or the edge characteristics that genuine multi-modal images have. Nonetheless, Fookes's contributions are important for showing that stereo-vision using mutual information could be significantly enhanced when combined with other state-of-the-art stereo-vision techniques.

Later, Krotosky and Trivedi [1, 2, 3] used mutual information for an

infrared-visible camera pair in order to detect and track pedestrians. They applied mutual information for stereo correspondence within regions of interests (ROI) including human bodies, and proposed a disparity voting method for computing the final depth information of the corresponding regions as a significant restriction. Finally, this depth information was used to accurately register the multi-modal images for the ROIs.

In a very recent work on multi-modal stereo-vision, Campo *et al.* [36] proposed an MI-based method where the similarity measures were extended using the gradient information. They developed a multi-modal stereo rig (with thermal and visible cameras) and a database. The 3D depth results presented in their work were quite sparse for the scenes tested; however, their results are promising for showing that stereo-vision is possible from images with very distinct spectral characteristics.

Recently, a measure, called local self similarity (LSS), originally proposed for image template matching [37], has been applied as a thermal-visible stereo correspondence measure by Torabi and Bilodeau [38]. They implemented a ROI-based image matching system by tracking people in the scene according to their silhouettes, and compared it against MI-based similarity descriptors. In their first publication [39], they showed that the LSS measure outperforms MI and HoG (Histogram of Oriented Gradients). Later, they used the LSS measure in an energy minimization framework, enhancing the results when compared to their previous work [40]. In a recent study [41], with more data, they compared LSS and MI with (i) "traditional" descriptors such as SIFT, SURF, HOG, (ii) binary descriptors such as Census, Fast REtina Keypoint (FREAK) or Binary Robust Independent Elemantary Feature (BRIEF) and

(iii) direct comparisons of windows based on SSD, NCC. In their study, MI and LSS were shown to be the leading measures for ROI-based image matching of human silhouettes. MI outperformed LSS showing that it is still the best choice for multi-modal image windows matching; however, for smaller window sizes where the objects of interest were small or segmented into small fragments or there were many occlusions between objects, LSS performed better. On the other hand, LSS measure has not yet been tested for a dense disparity map estimation and still requires larger windows than is used in our study. Moreover, it is computationally more expensive, and performs poorly on uniform regions or small regions at salient points that are dissimilar to their neighboring regions [38]. Such regions constitute non-informative descriptors and for this reason, they are eliminated in the beginning of their method, which makes their method sparse, *i.e.*, not suitable for dense disparity map calculation.

1.2. The Current Study

In this article, we propose a new multi-modal stereo-vision method based on mutual information which can accurately generate *dense* disparity maps of images taken from cameras of different modalities. The method is compared to previous MI-based methods in the literature quantitatively and visually, and it is shown to outperform them. The contributions of the article are summarized as follows:

- Contribution of two datasets for evaluating multi-modal stereo-vision methods. One is based on cosine-transformed versions of the widely-used Middlebury Stereo Evaluation Dataset [32], and the other is collected from the RGB and IR cameras of a Kinect device.

- Adaptive computation of the window used in computing the cost matrix. The adaptively sized and shaped windows for matching the pixels are determined by the segments in the images, and in turn, these windows help generate a robust correlation surface when computing joint probabilities to be used in calculating joint entropy and the mutual information similarity metric.

- Adaptive aggregation of raw costs for all pixels, enabling us to determine stable disparities, which are used for fitting planes in a segment.

- An iterative method which uses the estimated disparities for re-calculating the prior probabilities in MI calculation, and repeating the subsequent steps (*i.e.*, cost estimation, cost aggregation, plane fitting, and segment splitting and merging).

This study extends an earlier version of our article [45] where only the preliminary results on for the adaptive windowing mechanism were presented. The current article differs mainly in the following aspects - see also Figure 2: (i) The method is now iterative, (ii) an adaptive cost aggregation method is proposed using confidences of disparities (iii) the segments are split and merged using confidence information of disparities and the disparity planes fitted, thus reducing the dependence of the method to the initial segmentation.

2. Methodology

The overview of our method is depicted in Figure 2. Our method takes as input a pair of rectified multi-modal images, satisfying the epipolar line

constraint so that correspondences can be found on horizontal scanlines. The initial step is to segment the left(IR) image. Next, the cost matrix for all candidate matching pixels in each scanline of the rectified image pair according to designated maximum disparity is computed by the MI computation algorithm using the adaptive windowing method proposed uniquely in this study. Later, the raw costs are adaptively aggregated using confidence metrics and segmentation information. Next, the disparity planes corresponding to segments are computed from the stable pixels where the outliers of each disparity plane are inspected for segment splitting. Finally, the segments are inspected for merging with a neighboring segment by comparing the similarities between the associated disparity planes. The new iteration uses refined segmentation and the current disparity map for the new disparity plane computation. In the following subsections, each of these steps are explained in more detail. Table 1 provides definitions of the symbols used throughout the article.

2.1. Segmentation of the IR Image

We segment the IR image since the rest of the processing will rely on this segmentation. The reason for segmenting only the IR image is that the surfaces in IR images are also common in the RGB images, but the reverse is not true (see Figure 1) since RGB images contain more detailed and textured surfaces which do not exist in the IR images of our datasets. With this step, we get non-overlapping segments representing homogeneous regions in the IR image (see Figure 3). We assume that each segment corresponds to a planar surface in the scene, which is a common assumption in segmentation-based stereo-vision techniques [46, 47, 48, 49, 50, 51, 52].

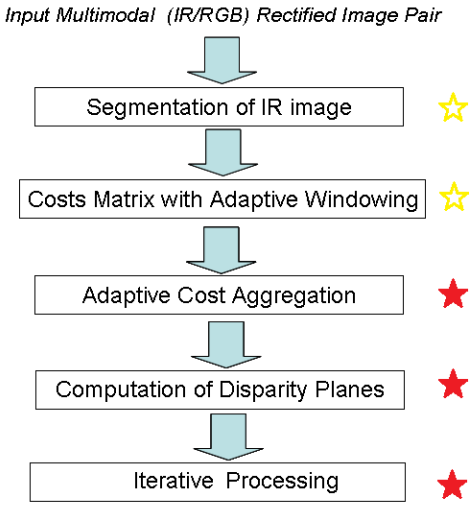


Figure 2: Overview of our method. The red (filled) stars are the extensions over the preliminary version of our work [45], and the yellow (empty) stars are the steps that are modified compared to our previous work. [Best viewed in color]

We use the Synergistic Image Segmentation algorithm [53] for segmenting the IR image. This method incorporates an edge magnitude/confidence map into the mean-shift segmentation algorithm [54] enhancing the results especially on weak edges, hence, separating the objects better. The algorithm makes use of the parameters of the mean shift segmentation algorithm; the spatial bandwidth h_s, the feature (range) bandwidth h_r, and the minimum segment size M as well as the size of the gradient window n used, the mixture parameter for blending of gradient magnitude a_{ij}, and the threshold for the discontinuities t_e - see [53] for the details.

2.2. Computation of the Cost Matrix

The computation of the cost matrix is the key step in this study, representing a significant part of our contributions (see Algorithm 1). The inputs

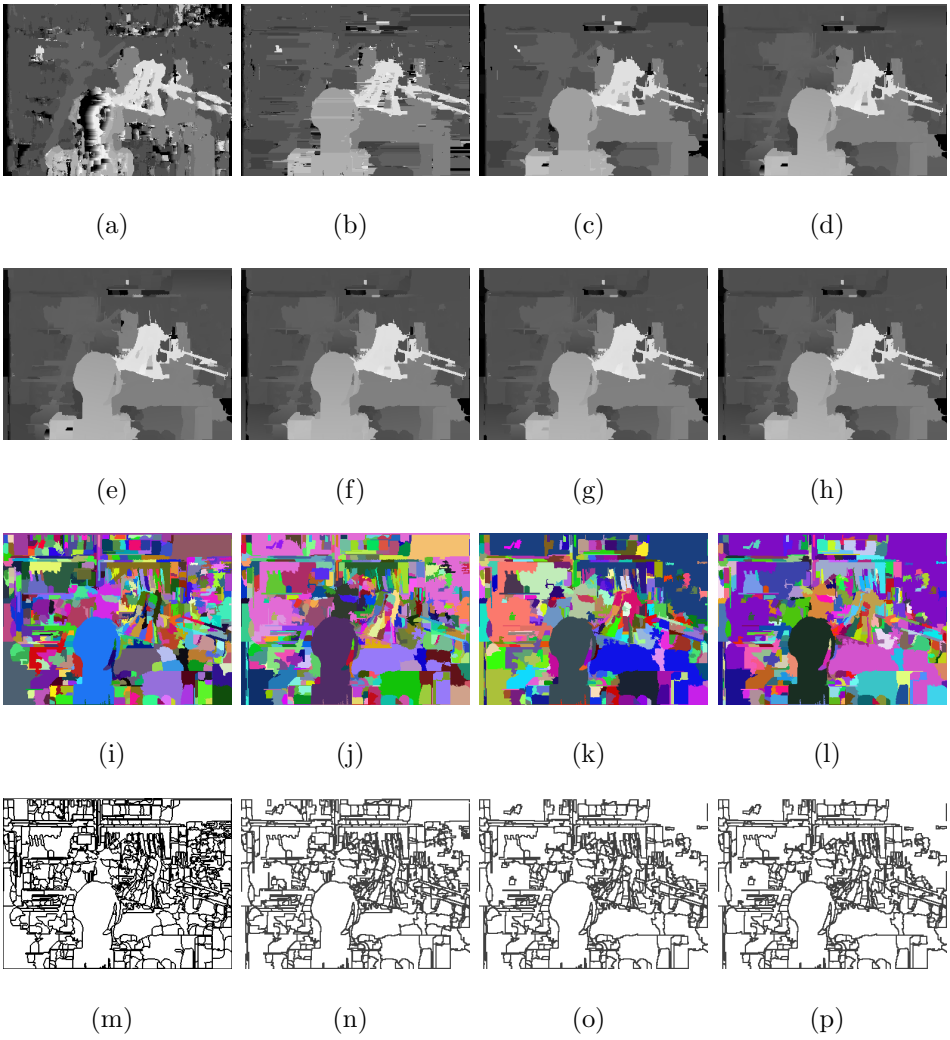


Figure 3: The intermediate steps. **(a)** WTA disparities of raw costs with No-Adaptive Windowing - 1st iteration. **(b)** WTA disparities of raw costs with Adaptive Windowing - 1st iteration. **(c)** WTA disparities after adaptive cost aggregation - 1st iteration. **(d)** Plane fitted disparities - 1st iteration. **(e-h)** Resultant plane-fitted disparities for iterations 1-4. **(i)** The initial segmentation of the left image. **(j-l)** The input segmentations for iterations 2-4 (after the segment splitting and merging steps are applied in the previous iteration). **(m-p)** Edge map of the corresponding input segmentation at each iteration. [Best viewed in color]

Table 1: List of notations and acronyms.

Symbol	Definition	Symbol	Definition
L	Left (IR) image	l_c	current center pixel in left image
R	Right (RGB) image	$Conf$	Confidence map regarding calculated costs
$^{(i)}$	Iteration number ($i \in [0, N]$)	c_1	Min. cost of the candidate disparities
S	Segmentation	c_2	Second min. cost of the candidate disparities
C	Cost matrix	ρ	Ceiling value for the maximum confidence
D	Disparity map	w	Weights for performing cost aggregation
MI	Mutual Information	b	Half-size of the window for cost aggregation
WTA	Winner Takes All	SD	Spatial Distance
x	Column number of a pixel	DD	Disparity Distance
y	Row number of a pixel	λ_{SD}	Designated scaling constant for spatial distance
d	Disparity in range $[0, d_{max}]$	λ_{DD}	Scaling constant for disparity distance
p	Current pixel	f	Function for subpixel disparity computation
q	Neighbor pixel	τ_{ic}	Confident inlier disparity threshold
s	Segment in ($s \in S$)	τ_{ir}	Stable segment ratio threshold
I_l	Intensities of left image pixels	τ_{od}	Outlier disparity distance threshold
I_r	Intensities of right image pixels	τ_{os}	Outlier disparities size threshold
W	Local window of computation for a center pixel	τ_{oc}	Confident outlier disparity threshold
ω	Assumed thickness of discontinuities in images	$Plane$	Set of disparity planes
P	Joint Probability	α	Angle between two disparity planes
P_{prior}	Prior Joint Prob. of Left & Right Images	τ_α	Angle threshold for parallel planes
P_{window}	Joint Prob. of Left & Right Images	τ_{pd}	Plane to plane distance threshold
λ	Ratio of incorporating prior prob. to joint prob.	h_s	Spatial bandwidth in mean-shift segm.
h_w	Histogram computed for the adaptive window	h_r	Feature (range) bandwidth in mean-shift segm.
$T()$	Counter function for hist. computation	M	Minimum segment size in mean-shift segm.
$L1$	L1 distance	n	Size of the grad. window in syn. image segm.
k	Increment of counts for histograms	a_{ij}	Mixture parameter in syn. image segm.
t_e	Threshold value for the edge computation		

to the algorithm are the left (IR) image L, the right (RGB) image R, the segmentation $S^{(i)}$ (computed from L for the initial iteration and modified for the following iterations) and the disparity map $D^{(i)}$ ($D^{(0)} = 0$, and otherwise, $D^{(i)}$ is the disparity map generated disparity map in the previous iteration).

Algorithm 1 first computes joint prior probabilities for all corresponding pixels in left and right images using the current disparity map (for the sake of simplicity, in the rest of the section, the current iteration superscript ($^{(i)}$)

Algorithm 1 Cost matrix computation.

Inputs: L : Left (IR) Image

R : Right (RGB) Image

$S^{(i)}$: Input segmentation ($i \in [0, N] : iteration$)

$D^{(i)}$: Input disparity map ($D^{(0)}$ is zero)

Outputs: $C^{(i)}$: The cost matrix

Compute $P^{(i)}_{prior}(L, R, D^{(i)})$ //See Eqn. 1
for $y = 0$ to $height$ **do**
  for $x = 0$ to $width$ **do**
    for $d = 0$ to d_{max} **do**
      $C^{(i)}(x, y, d) \leftarrow -M\left(W_L(x, y), W_R(x - d, y), S^{(i)}, P^{(i)}_{prior}\right)$
      // see Eq. 8 for $M()$
    end for
  end for
end for
return $C^{(i)}$

is omitted since all the variables are for the current iteration):

$$P_{prior}(I_l, I_r) = \frac{h(I_l, I_r)}{\sum_{l,r} h(I_l, I_r)}, \tag{1}$$

where I_l, I_r are respectively the intensities of the pixels $l(i, j) \in L$ and the corresponding pixel $r(i, j - D(l)) \in R$. Prior probabilities are computed using $h()$, the 2D histogram of all the corresponding pixel intensities.

Next, we compute the cost matrix for all pixels by computing MI (the negative of the MI measure is used as the cost) using the proposed adaptive

windowing scheme as:

$$
\begin{aligned}
W_L(x,y) &= L(x_{min} : x_{max}, y_{min} : y_{max}), && (2)\\
x_{min} &= x - \delta x_l - \omega, && (3)\\
x_{max} &= x + \delta x_r + \omega, && (4)\\
y_{min} &= y - \delta y, && (5)\\
y_{max} &= y + \delta y, && (6)
\end{aligned}
$$

where δx_l and δx_r are distances to the border of the segment to which the current pixel (x, y) belongs, and the window is enlarged by ω, the assumed thickness of discontinuity at segment borders (Figure 4). δy similarly provides the window size in the vertical direction and it is currently a user-configured parameter ($\delta y \leq 4$ pixels for the Middlebury database) determined experimentally. We did not consider the segment borders in the vertical direction because the segment plane may not be a fronto-planar surface and may confuse the cost calculation. We applied the same window to the right image by moving the window for each candidate disparity d.

$$W_R(x,y) = R(x_{min} - d : x_{max} - d, y_{min} : y_{max}). \quad (7)$$

After we determine the adaptive windows to be matched, we compute MI between the two windows using the segment information and the prior probabilities as:

$$M(W_L, W_R, S, P_{prior}) = \sum_W P(I_l, I_r) \ln \frac{P(I_l, I_r)}{P(I_l)P(I_r)}, \quad (8)$$

where joint probabilities are computed using the adaptive correlation surface that we developed and the prior probabilities incorporated just as Fookes did

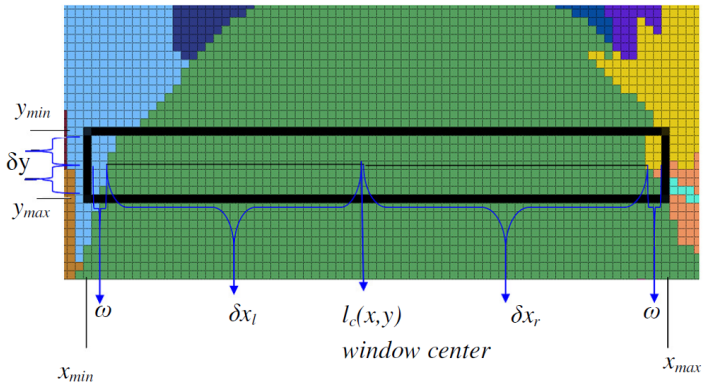


Figure 4: Adaptive window calculation.

[35] as follows:

$$P(I_l, I_r) = \lambda P_{window}(I_l, I_r) + (1 - \lambda) P_{prior}(I_l, I_r). \tag{9}$$

The correlation surface enabling joint probability calculation is another key contribution of our study to the MI cost calculation, where the joint histogram is calculated by considering the pixels within the current segment in the window and the pixels nearby the edge of the segment as:

$$P_{window}(I_l, I_r, S) = \frac{h_w(I_l, I_r, S)}{\sum_w h_w(I_l, I_r, S)}, \tag{10}$$

$$h_w(I_l, I_r, S) = \sum_w T(I_l, I_r, S), \tag{11}$$

$$\tag{12}$$

where the $T()$ function is defined as follows:

$$T(I_l, I_r, S) = \begin{cases} k & \text{if } S(l) = S(l_c) \ \& \ L1 > \omega \\ k - \frac{k}{\exp(L1)} & \text{elif } S(l) = S(l_c) \ \& \ L1 \leq \omega \\ \frac{k}{\exp(L1)} & \text{elif } L1 \leq \omega \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

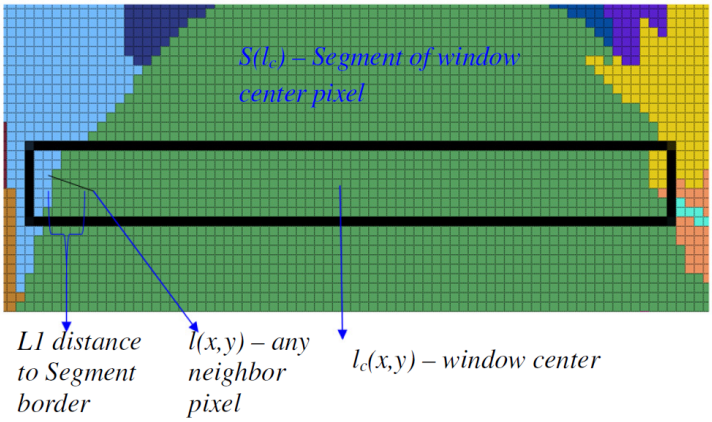


Figure 5: Adaptive MI computation surface using segmentation.

where $L1 = \|l - S(l_c)\|$ is the $L1$ distance between the neighbor pixel l within the generated window and the border of the segment to which the current pixel l_c belongs; and ω is the assumed thickness of the segment border as was defined in Equation 2.

The use of $L1$ distance (Figure 5) in Eq. 13 incorporates the pixels near the segment borders into the MI calculation with some penalty due to possible occlusions around borders. This allowed us to consider both the segment and the edges excluding other segments within the rectangular window in our MI measure computation.

2.3. Adaptive Cost Aggregation

In this step, our major concern is to detect, revise, and reduce those *un-confident* cost estimates computed in the previous step and causing the majority of incorrect winning disparities in the WTA or in the subpixel disparity calculations. We use cost confidence measures to detect *un-confident* costs computed for the pair of corresponding pixels and for this purpose, a

modified version of the confidence measure that was used in [50] is proposed as:

$$Conf(x,y) = \min\left(\frac{|c_1 - c_2|}{|c_1|}, \rho\right), \tag{14}$$

where c_1 is the minimum cost within the disparity range $[0..d_{max}]$, and c_2 is the second minimum cost. The ratio of the minimum and the second minimum cost value margin to the minimum cost value is used as the confidence measure and the obtained values are truncated with respect to some predetermined value ρ. This way, higher confidence values are prevented from dominating the cost aggregation step. Figure 6 shows confidence values.

Figure 6: Cost confidences for Tsukuba (scaled and truncated to [0..255] range for the sake of visibility).

We perform cost aggregation by visiting all the pixels p in the initially computed cost matrix $C^{(i)}(p,d)$ (see Algorithm 1) and aggregating the costs

according to the following weights within a local neighborhood for all the disparities d in range $[0..d_{max}]$:

$$C^{(i)}_{agg}(p,d) = \sum_{q \in w_p} w(p,q) C^{(i)}(q,d), \tag{15}$$

where w_p is a square support window of size $(2b+1) \times (2b+1)$ (b: the half window size) where the costs of the same disparity in the neighborhood are aggregated by a weighting mechanism which incorporates the current segmentation effectively as:

$$w(p,q) = \begin{cases} Conf^{(i)}(q) & \text{if } S^{(i)}(p) = S^{(i)}(q) \\ Conf^{(i)}(q) \exp(-(\frac{SD(p,q)}{\lambda_{SD}} + \frac{DD(p,q)}{\lambda_{DD}})) & \text{if } S^{(i)}(p) \neq S^{(i)}(q) \end{cases} \tag{16}$$

where $Conf^{(i)}(q)$ is the confidence of aggregating pixel q (See Eqn. 14); $SD(p,q)$ is the spatial distance between pixels p and q; $DD(p,q)$ is the WTA (winner takes all) disparity distance of initial costs $C^{(i)}$ between pixels p and q; λ_{SD} and λ_{DD} are the designated scaling constants for the spatial distance and the disparity distance.

Therefore, the proposed scheme aggregates costs of neighboring pixels within the same segment according to the confidence of aggregating pixels while penalizing aggregating weights according to their spatial distance and their WTA disparity distances of initial costs for the pixels in the neighboring segments.

2.4. Computation of Disparity Planes

In this step, the main idea is to fit planes to the segments using disparities of confident pixels in the aggregated disparity map yielding disparity planes as the final disparity map. Algorithm 2 shows the major steps of the method proposed in this step.

Algorithm 2 Computation of disparity planes.

Inputs: $S^{(i)}$: Segments of current iteration, ($i \in [0, N]$: iteration)
$C^{(i)}_{agg}$: Aggregated cost matrix, (refer to Section 2.3)

Outputs: $S^{(i)}_{final}$: Revised segmentation,
$D^{(i)}_{final}$: Disparity map computed from the fitted planes

1: $D^{(i)}_{agg} \leftarrow$ WTA disparity map corresponding to $C^{(i)}_{agg}$ aggregated cost matrix
2: $D^{(i)}_{aggsub} \leftarrow D^{(i)}_{agg} + f(C^{(i)}_{agg})$ //estimate subpixel disparities - Eq. 18
3: $D^{(i)}_{aggsub,m} \leftarrow med_{W_m}(D^{(i)}_{aggsub})$ //3x3 median filter to subpixel disparities
4: $Conf^{(i)}_{agg} \leftarrow$ Compute confidences for aggregated cost matrix $C^{(i)}_{agg}$
// Eq. 14
5: $(S^{(i)}_{split}, P^{(i)}_{split}) \leftarrow$ Perform iterative segment splitting step
using ($D^{(i)}_{aggsub,m}$, $S^{(i)}$, $Conf^{(i)}_{agg}$) // Alg. 3
6: $(S^{(i)}_{final}, P^{(i)}_{final}, D^{(i)}_{final}) \leftarrow$ Perform segment merging & finalization step
using ($D^{(i)}_{aggsub,m}$, $S^{(i)}_{split}$, $P^{(i)}_{split}$, $Conf^{(i)}_{agg}$) // Alg. 4
7: **return** $(S^{(i)}_{final}, D^{(i)}_{final})$

The inputs for the algorithm are the segmentation for current iteration $S^{(i)}$ and the aggregated cost matrix $C^{(i)}_{agg}$ (computed as described in the previous step - Section 2.3). Below, each step of the algorithm is described in detail:

1. **WTA of aggregated costs:** The Winner Takes All (WTA) disparities $D^{(i)}_{agg}$ corresponding to the aggregated cost matrix $C^{(i)}_{agg}$ that was computed in the previous step (See Equation 15) are computed in the first step.
2. **Subpixel disparity computation:** For the next step, the subpixel disparity estimates ($D^{(i)}_{aggsub}$) are computed using the aggregated cost matrix $C^{(i)}_{agg}$ by fitting a parabola to the minimum cost disparity in $D^{(i)}_{agg}$

Algorithm 3 Iterative plane fitting & segment splitting.

Inputs: D : Input disparity map
S : Initial segmentation map
$Conf$: Confidences of the disparities
Outputs: S : Revised segmentation,
$Plane_S$: Fitted disparity plane equations for each segment

repeat
  for all segment $s \in S$ **do**
    repeat
      $Cloud \leftarrow \{(p,d) \mid \forall p \in s, d = D(p), Conf(p) > \tau_{ic}\}$ //extract the point cloud of confident pixels p in s
      if $size(Cloud) < 4$ or $size(Cloud)/size(s) < \tau_{ir}$ **then**
        $stable(s) \leftarrow$FALSE
      else
        $stable(s) \leftarrow$TRUE
        Fit plane $Plane_S(s)$ to $Cloud$ using RANSAC
        $OutCloud \leftarrow \{(p,d) \mid \forall p \in s : d = D(p), |d - Plane_S(s,p)| > \tau_{od}\}$ //extract outlier point cloud of disparities according to fitted plane
        if $(size(OutCloud) > \tau_{os})$ **then**
          $OutCloud2 \leftarrow \{(p,d) \mid \forall (p,d) \in OutCloud, Conf(p) > \tau_{oc}\}$
          Split segment s for all the connected subsets of $OutCloud2$
          Append splitted segments to segments list S
        end if
      end if
    until segment s is not splitted
  end for
  Re-compute segments map S //since new segments can break bigger segments to two or more disconnected sub-segments
until no segment splitting performed
return (S,$Plane_S$)

and the two neighboring cost values and then analytically solving for the minimum.

Therefore, defining d as the integer disparity of minimum cost (the WTA disparity) in the cost matrix C within the disparity range d_0 to

Algorithm 4 Segment Merging and Finalization

Inputs: D : Input disparity map
S : Input segmentation map
$Conf$: Confidences of the disparities
$Plane_S$: Fitted disparity planes for segments
Outputs: S : Revised segmentation by merged segments
$Plane_S$: Fitted disparity plane equations for each segment
D_{Plane} : Disparity map computed from fitted disparity plane equations

// **Phase 1**: merge stable segments & retry for unstable segments
for all segment $s \in S$ **do**
  if $stable(s)$ =TRUE /* See Algorithm 3 */ **then**
    for all $s' \in \Omega(s)$ {$\Omega(s)$: neighboring segments of s} **do**
      if $cop(s, s')$ =TRUE /* Coplanar planes - See Equation 20 */ **then**
        $s \leftarrow Merge(s, s')$ //Segments are merged
        $merged(s) \leftarrow$ TRUE
        $S \leftarrow S - s'$ //remove s' from set S since it is merged with s
      end if
    end for
  else if $stable(s)$ =FALSE **then**
    repeat
      $\tau_{ic2} \leftarrow \tau_{ic} * \lambda_{\tau_{ic}}$ //Decrement confidence threshold by $\lambda_{\tau_{ic}} \in (0, 1)$
      Re-compute plane fitting $Plane_S(s)$ for $Cloud$ where: $Cloud \leftarrow \{(p, d) \mid p \in s, d = D(p), Conf(p) > \tau_{ic2}\}$
      if $size(Cloud) < 4$ or $size(Cloud)/size(s) < \tau_{ir}$ **then**
        $stable(s) \leftarrow$FALSE
      else
        $stable(s) \leftarrow$TRUE
      end if
      $decrement(\lambda_{\tau_{ic}}, \gamma)$ //Decrement $\lambda_{\tau_{ic}}$ by $\gamma \in (0, 1)$
    until $(\lambda_{\tau_{ic2}} = 0)$ or $(stable(s)$ =TRUE)
  end if
end for

// **Phase 2**: recompute plane fits for merged segments
for all (segment $s \in S$) and ($stable(s)$ =TRUE) and ($merged(s)$ =TRUE) **do**
  Re-compute plane fitting $Plane_S(s)$ for $Cloud$ where $Cloud \leftarrow \{(p, d) \mid p \in s, d = D(p), Conf(p) > \tau_c\}$
end for

//**Phase 3**: compute final disparity map
for all segment $s \in S$ **do**
  if $stable(s)$ =TRUE **then**
    Compute disparities $D_{Plane(s)}$ for pixels in segment s using fitted plane equation $Plane(s)$
  else if $stable(s)$ =FALSE **then**
    Set disparities $D_{Plane(s)}$ for pixels in segment s to original disparities in input $D(s)$ // for segments which are still unstable
  end if
end for
return $(S, Plane_S, D_{Plane})$

d_{max};

$$d = \underset{d_i \in \{d_0,\ldots,d_{max}\}}{\arg\min} C(d_i). \tag{17}$$

The subpixel disparity estimate is defined as:

$$d_{sub} = d + f(C(d-1), C(d), C(d+1)), \tag{18}$$

where f is the function for parabola interpolation defined as:

$$f = \frac{C(d-1) - C(d+1)}{2(C(d-1) - 2C(d) + C(d+1))}. \tag{19}$$

This way we have floating point disparity values with more smooth transitions within the segment in which plane-fitting is performed.

3. **Median filtering:** Next, a median filter is applied to the subpixel disparities so that noisy disparities are eliminated (yielding $D^{(i)}_{aggsub,m}$), if there are any. This step improves the quality of the planes fitted to disparities.

4. **Compute confidences of the aggregated cost matrix:** The confidence values ($Conf^{(i)}_{agg}$) corresponding to the aggregated cost matrix $C^{(i)}_{agg}$ is computed in order to take into consideration only the confident pixels in the subsequent steps.

5. **Iterative segment splitting:** In the next step, the confident disparities within the segments are plane fitted and the outlier disparities are re-evaluated by splitting the segments. The step is iteratively re-applied for the new segmentation map ($S^{(i)}_{split}$) until no further segment splitting occurs - see Section 2.4.1 for the details of this step.

6. **Segment merging & finalization:** Finally, the split segments ($S^{(i)}_{split}$) and corresponding disparity planes ($P^{(i)}_{split}$) are inspected for finalization

by (i) merging neighbor segments which are co-planar at the same disparity level and (ii) refining unstable segments that may be generated during the segment splitting operations or which may have an inadequate number of confident pixels to be able to compute a disparity plane (*i.e.*, when the number of pixels <4). The final disparity map is computed from the resultant segmentation and the corresponding segment plane equations - see Section 2.4.2 for the details of this step.

2.4.1. Iterative Plane Fitting and Segment Splitting Step

Algorithm 3 includes the details of this step where the aim is to revise the input segmentation according to the confident outlier disparities within the corresponding segments once plane fitting is performed. This way, we reduce the dependency of the performance of the algorithm upon the initial segmentation.

In the algorithm, the disparity plane for each segment in the current segmentation map S is computed from the confident disparities only. The plane fitting is performed using RANSAC (RANdom SAmple Consensus - [55]). Next, we analyze the outlier disparities in each plane fitted segment and check whether they constitute connected regions of a significant size; if so, the outlier region is split out. This operation is performed iteratively until the segment is no longer split. Finally, in the outer loop, the segmentation map is re-computed and the above described steps are re-applied since the segments may break more than once. This way, the segmentations and the plane fits are revised iteratively until no further segment splits can be performed. The algorithm returns the revised segmentation map along with the fitted disparity plane equations to the segments.

Algorithm 3 makes use of several thresholds. τ_{ic} is the disparity confidence value threshold used to construct the initial point cloud of disparities from the segment disparities. τ_{ir} is the stable segment ratio threshold which determines whether a segment is stable or not by checking the ratio of the size of the confident disparities point cloud and the segment size. If the size of the cloud is smaller than 4 pixels then it will not be possible to fit a plane and therefore such segments are marked as unstable and left for the next step for correction. τ_{od} threshold is used for determining the outlier points of the fitted plane which is designed to be greater than the RANSAC distance threshold parameter used for plane fitting. τ_{os} determines the minimum size of the outlier point cloud of disparities to continue the splitting operation and τ_{oc} is the confidence threshold for the outlier points which are to be selected for segment splitting. Therefore, to be able to create a new segment by splitting from the original segment, a connected region whose size is greater than a designated threshold should be available.

2.4.2. Segment Merging and Finalization of Disparity Planes Step

This step computes the final segmentation and disparity map of the scene where the details are presented in Algorithm 4. The step is composed of three phases: In the first phase, all the stable segments are inspected along with their neighbors and merged if they are coplanar. The coplanarity of two disparity planes are defined as:

$$cop(s, s') = \begin{cases} 1, & \text{if } (\alpha(s, s') < \tau_\alpha) \text{ and } (\|s - s'\| < \tau_{pd}) \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

which checks if the normals of the planes are parallel (the difference in their normals α is smaller than a threshold τ_α) and if they are at the same disparity

level (the distance between planes is smaller than a threshold τ_{pd}). Moreover, the segments that were marked as unstable are re-evaluated by decrementing the confidence threshold iteratively.

In the second phase, the equations for the disparity planes are recomputed for the merged segments and finally, in the third phase, the disparity of each pixel is computed from the disparity plane equations, except for the still-unstable segments where the input disparity map is accepted as-is for those pixels.

2.5. Iterative Refinement

Finally, we have an updated segmentation map along with its corresponding disparity map computed from the fitted disparity planes for each segment. We can go over the same steps again, as a new iteration. A new iteration can use the current disparity map for better estimation of the joint prior probabilities (see Equations 1 and 9) along with some adjustments that can be performed with such a priori data. Therefore, this step starts with the segmentation $S^{(i+1)}$ set to resultant segmentation of the current iteration $S^{(i)}_{final}$, and disparities $D^{(i+1)}$ set to resultant disparity map $D^{(i)}_{final}$ for the current iteration (see Algorithm 2).

Moreover, for iterations after the first one (*i.e.*, $i \geq 1$), we now have the opportunity to adjust the adaptive window calculation method in Equation 2, where the x_{max} value can be moved back in the direction of the center pixel if the right neighboring segment has a disparity level higher than the current segment. In such a case, how much x_{max} is shifted is determined by the difference in the disparity levels of the segments. This enables us to not use the pixels in the right segment when the same window is applied to the

right image for correspondence matching.

3. Experiments and Results

In this section, we test our method on the two datasets generated. The synthetically altered sets from the Middlebury Stereo Dataset [56] and the RGB-IR images captured from a Kinect device.

3.1. Dataset #1 - The Middlebury Dataset

This dataset contains the four *popular* image pairs (Tsukuba, Venus, Cones and Teddy) in the Middlebury Stereo-vision Dataset [56], where the left images are synthetically altered by using a cosine transform ($\cos(\pi I/255)$) of pixel intensities just as Fookes did [35]. See Figure 7 for the image pairs used in the experiments. Note that, in the left images, important details are lost due to the cosine transformation. In our experiments, we used the non-occluded regions and discontinuity regions "as is" as provided by the Middlebury page [56][1]. Figure 8 shows the "all" regions used for evaluating the results. For the experiments in the Middlebury set, we use the empirically-set parameters listed in Table 2. The parameter values were determined experimentally except for the histogram binning $Size(h_w)$ and λ for

[1]Regarding the "all" regions, we performed clipping on the left border for the regions that do not exist in the right image because we do not perform any extrapolation for those regions. Besides, in Teddy and Cones, image borders are also excluded for 20 pixels as they were for Tsukuba & Venus since this is a window-based method. This decision was explained in the description page of "Middlebury Stereo-Vision Evaluation Version 2" [57] *item-5* in the section describing differences to the "old table".

375 joint prior probability incorporation which was already analyzed by Fookes in [35]

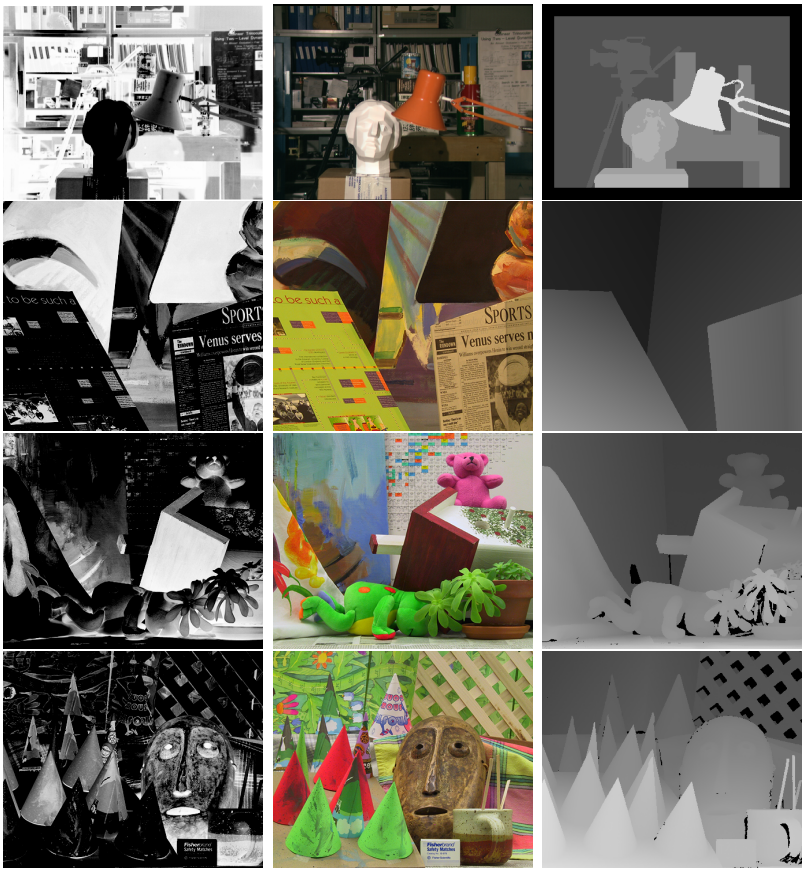
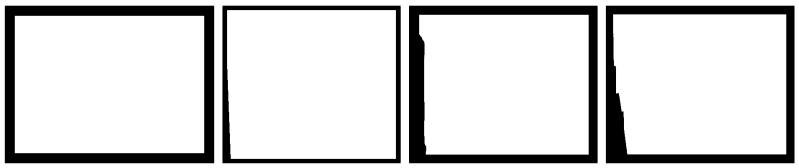


Figure 7: Tsukuba, Venus, Teddy and Cones stereo pairs from the Middlebury Stereo Vision Page - Evaluation Version 2. *Left column*: Synthetically altered left images. *Middle column*: The right images. *Right column*: The ground truth disparities. Note that, in the left image, important details are lost due to cosine transformation.



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

Figure 8: Regions where evaluations are performed, including both non-occluded and discontinuities: the "all" regions provided as "white pixels" for Tsukuba (a), Venus (b), Teddy (c), and Cones (d) pairs.

Table 2: Parameter Settings Used in Dataset #1 (Synt. Altered Middlebury) Experiments.

Segmentation	h_s	h_r	M	n	a_{ij}	t_e
	7	6	50	7	0.5	0.2
Adaptive Windowing	δ_y	λ	ω	$Size(h_w)$	k	
	4	0.3	1	40	5	
Adaptive Cost Aggregation	ρ	λ_{SD}	λ_{DD}	$Size(w(p,q))$		
	0.25	1	1	17x17		
Iterative Plane Fitting & Segment Splitting	τ_{ic}	τ_{ir}	τ_{od}	τ_{os}	τ_{oc}	
	0.007	0.25	1.0	20	0.014	
Segment Merging & Finalizing	τ_α (^{o})	τ_{pd}	γ			
	0.1	0.15	0.25			

Table 3 shows the performance of the proposed method for each step of (i) adaptive windowing (Section 2.2)(ii) cost aggregation (Section 2.3) (iii) plane fitting (Section 2.4) without any iterations yet. The resulting disparity maps of adaptive windowing and cost aggregation steps are computed as Winner Take All ("WTA") disparities where the disparity having the minimum cost is selected. The performance metrics are computed in terms of both *RMS* (root mean squared) error and the *Bad* pixels (*i.e.*, percentage of bad matching pixels). These metrics are computed between the estimated disparity map

$d_C(x, y)$ and the ground truth $d_T(x, y)$ as follows [10]:

$$RMS = \sqrt{\frac{1}{N} \sum_{(x,y)} |d_C(x, y) - d_T(x, y)|^2}, \tag{21}$$

$$Bad = \frac{1}{N} \sum_{(x,y)} \left(|d_C(x, y) - d_T(x, y)| > \delta_d \right). \tag{22}$$

The error threshold value δ_d for the Bad pixels metric is set to 1.5 disparity distance as was performed in the Middlebury Stereo Vision Evaluation Page (in the description page [57] for non-integer subpixel disparities unless they are rounded).

As one can observe in the results, the WTA results after cost aggregation improve the disparity estimation of the adaptive windowing step. Plane-fitting, however, improves the RMS values of the estimated disparities in all image pairs. The bad matching percentage is almost the same as WTA of cost aggregation in Tsukuba image pair, composed of fronto-parallel surfaces, while it is worse in Teddy image pair, which includes mostly curved surfaces, whereas for Venus, which is only composed of planar surfaces, the bad matching percentage improves significantly. We also observe that, for discontinuities, the performances can get slightly worse due to the fact that, in the initial segmentation of the cosine transformed images, some of the edges may get lost, generating inaccurate disparities within those regions.

Table 3: Results of the Proposed Method on Synt. Altered Middlebury Images for WTA of Adaptive Windowing Costs, WTA of Adaptively Aggregated Costs and Plane Fitting.

Image*	Method	RMS (all)	RMS (nonocc.)	RMS (disc.)	Bad (all)	Bad (nonocc.)	Bad (disc.)
Tsukuba	WTA of Adap.W.	1.621	1.495	**2.419**	7.72%	6.64%	**16.88** %
	WTA of Agg.	1.425	1.315	2.433	**6.15**%	**5.43**%	17.09%
	Plane Fitting	**1.378**	**1.269**	2.484	6.20%	5.47%	18.32%
Venus	WTA of Adap.W.	1.729	1.689	2.464	8.77%	8.09%	25.45%
	WTA of Agg.	1.224	1.173	3.259	6.21%	5.54%	27.18%
	Plane Fitting	**1.003**	**0.939**	**2.754**	**3.42**%	**2.75**%	**19.18**%
Teddy	WTA of Adap.W.	8.092	5.420	6.570	24.27%	23.69%	**31.61** %
	WTA of Agg.	7.111	4.100	5.439	**19.42**%	**20.18**%	32.24 %
	Plane Fitting	**6.721**	**3.543**	**4.700**	24.28%	24.03%	32.72%
Cones	WTA of Adap.W.	10.102	8.384	10.285	27.32%	22.67%	35.24%
	WTA of Agg.	7.802	5.715	8.037	19.93%	14.80%	28.25%
	Plane Fitting	**7.240**	**5.150**	**7.359**	**19.48**%	**14.58**%	**27.24**%

* adaptive windowing vertical window size=9 and adaptive cost aggregation step window size =17x17, no iterative refinement yet

Next, we analyze the effect of iterative refinement on the estimated disparities. Figures 9 and 10 show the qualities of the estimated disparities in 10 iterations following Section 2.5. We see from the figure that the RMS and bad matching percentage decrease drastically in the second iteration and the values more or less stabilize. This suggests that iterating twice over the disparity estimation steps as suggested in Section 2.5 is sufficient.

Figure 3 shows the resultant WTA disparity maps of non-adaptive windowing costs, adaptive windowing costs, aggregated costs, and plane-fitted disparities as well as the resultant first 4 iterations of disparity maps, and the segmentation maps computed for the Tsukuba image. Figure 11 on the other

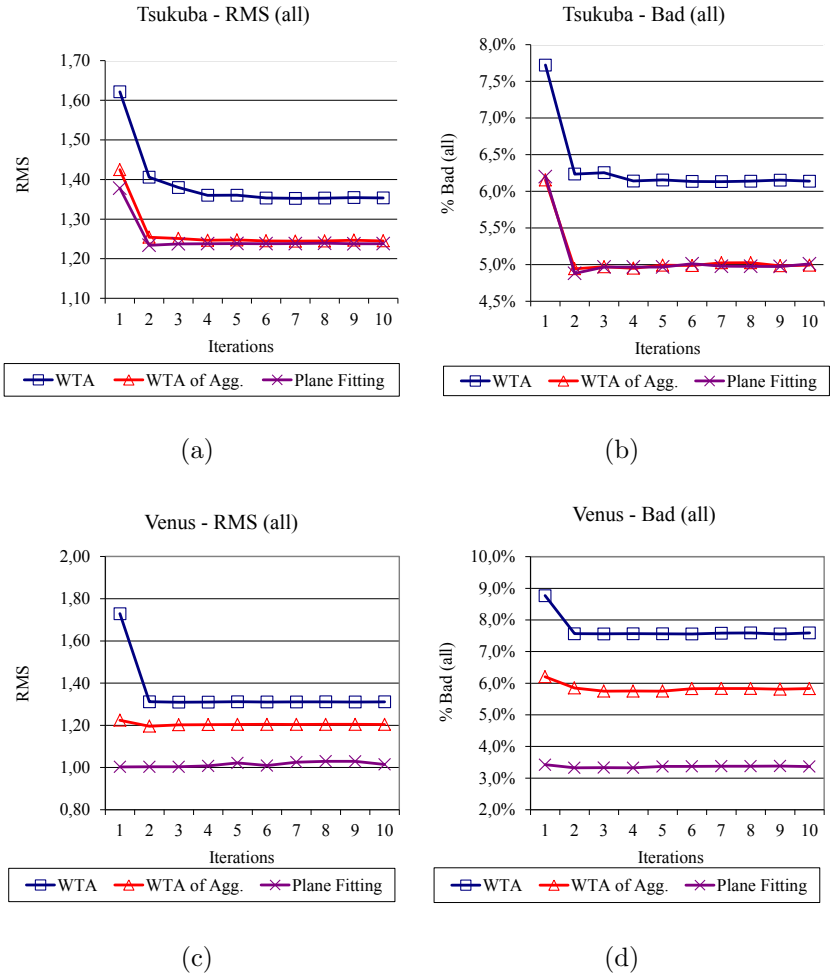


Figure 9: Effect of iterations on RMS and the percentage of bad pixels for "all" regions. **(a-b)** Tsukuba pair. **(c-d)** Venus pair.

hand presents the 2nd iteration (since it was concluded that two iterations are sufficient) results regarding WTA of aggregated costs and plane-fitted disparities for the Venus, Teddy, and Cones images.

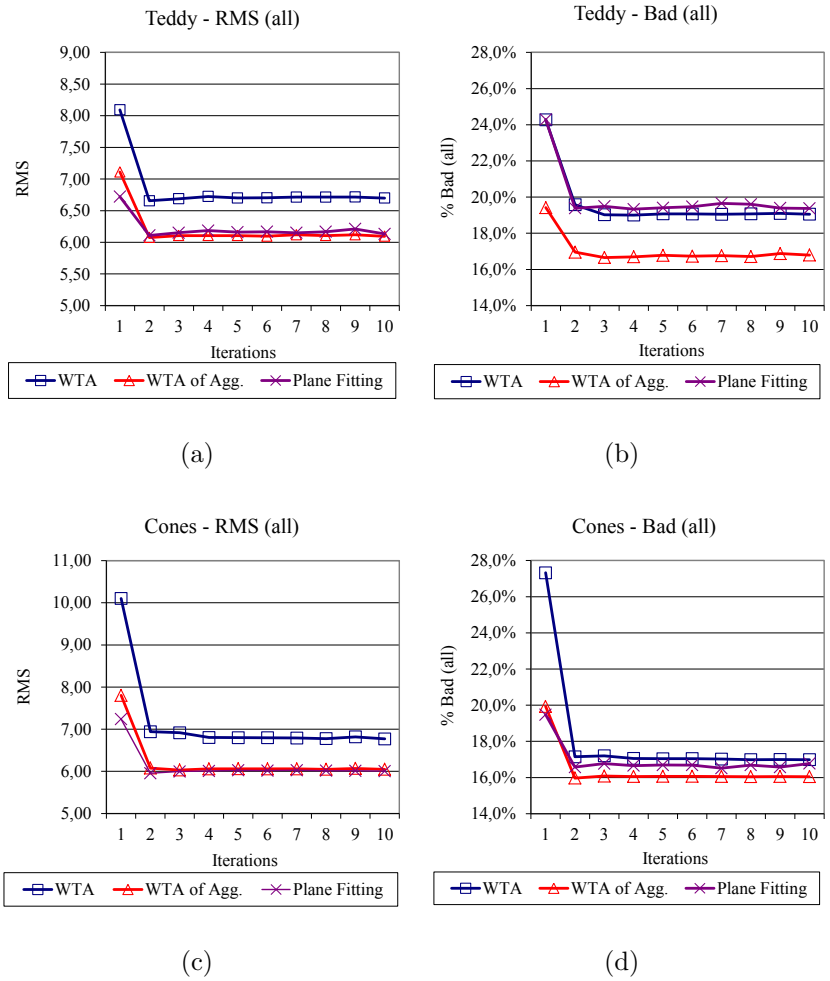
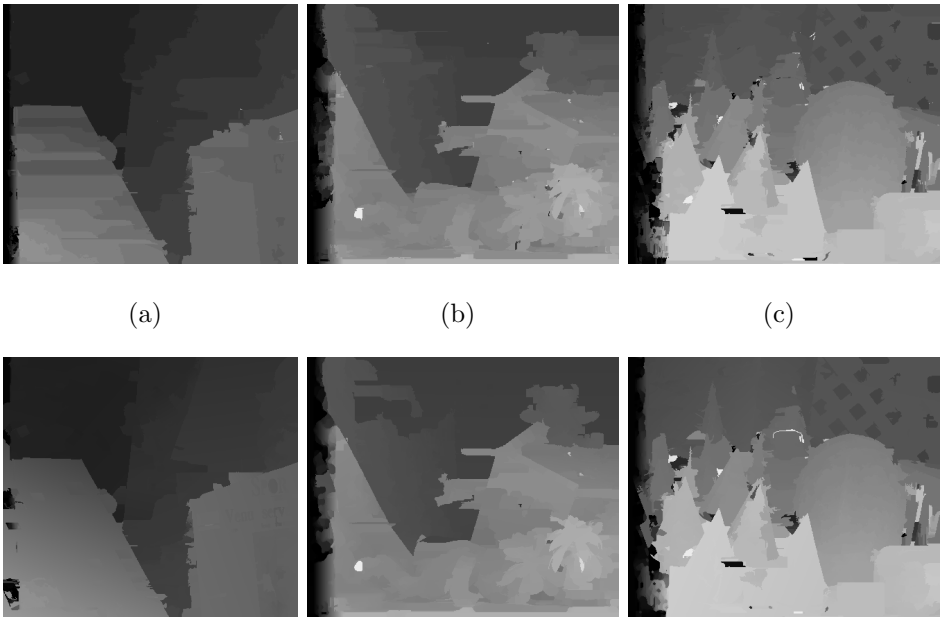


Figure 10: Effect of iterations on RMS and the percentage of bad pixels for "all" regions. **(a-b)** Teddy pair. **(c-d)** Cones pairs.

3.2. Dataset #2 - The Kinect Dataset

The Kinect dataset contains IR (left) and RGB (right) images captured from a Kinect device. The cameras are first calibrated (using RGBDemo software with OpenNI backend [58] with a set of 50 checkerboard images with different poses to find the extrinsic and intrinsic parameters for both



(a) (b) (c)

(d) (e) (f)

Figure 11: The Venus, Teddy and Cones stereo pair iteration-2 results. **(a-c)** WTA of aggregated costs. **(d-f)** Plane fitted disparities.

the IR and RGB cameras). The image pairs are stereo-rectified so that the Epipolar constraint is satisfied. Table 4 describes the images used in this part of the article, and Figure 12 shows the images along with the depth images generated by Kinect.

We compare our results against the depth that Kinect estimates. We look at two criteria for comparison: (i) "Percentage Good Depth" (PGD): Percentage of computed depth values z_c that are close to the Kinect's native depth z_k for different thresholds δ_z (namely, 10, 20 or 30 cm) where z_k is valid. Note that Kinect's depth is limited to (0., 5.0] meters which comprises the valid z_k values. (ii) "Percentage Total Coverage" (PTC): The percentage

Table 4: The Kinect dataset.

Dataset	Image No	Image Name	Resolution	Max. Disparity
Dataset #2	1	Kinect01	640×480	36
Dataset #2	2	Kinect02	640×480	21
Dataset #2	3	Kinect03	640×480	30

Table 5: Parameter Settings Used in Dataset #2 (Kinect) Experiments.

Segmentation	h_s	h_r	M	n	a_{ij}	t_e
	7	4	300	2	0.3	0.4
Adaptive Windowing	δ_y	λ	ω	$Size(h_w)$	k	
	18	0.4	2	40	5	
Adaptive Cost Aggregation	ρ	λ_{SD}	λ_{DD}	$Size(w(p,q))$		
	0.25	1	1	37x37		
Iterative Plane Fitting & Segment Splitting	τ_{ic}	τ_{ir}	τ_{od}	τ_{os}	τ_{oc}	
	0.0015	0.25	2.0	200	0.004	
Segment Merging & Finalizing	τ_α (^{o})	τ_{pd}	γ			
	0.1	0.15	0.25			

of pixels where Kinect does not provide an estimation ($z_k \notin (0.m, 5.0m]$), but our method provides an estimation in the valid range ($z_c \in (0.m, 5.0m]$). Note that, since there is no ground truth available, it is not possible to provide a quantitative evaluation of which estimation is better, except for these two criteria.

Table 5 provides the empirically determined parameter settings used in these experiments. Figure 13 shows the performance of our method for two iterations compared to Kinect and respective to each of the method steps. The graph and the performance figures are generated regarding the mean of the results achieved for each of the image pair in Dataset#2. The stacked

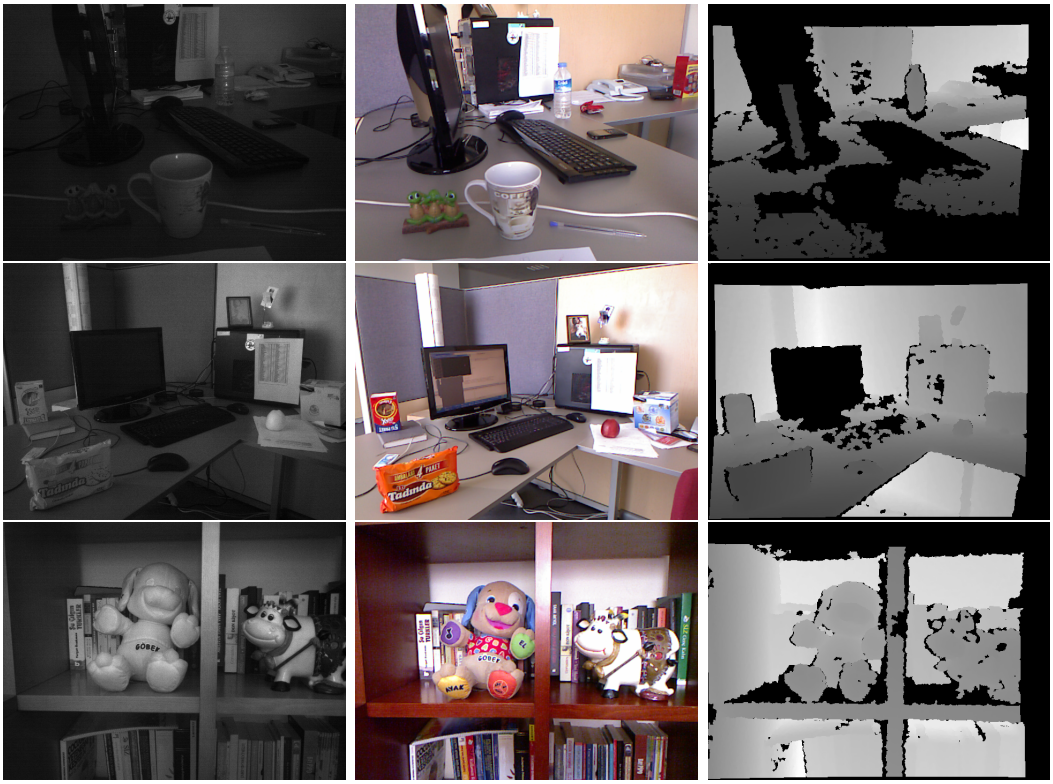


Figure 12: Kinect Dataset Images: *Left column*: Left (IR) camera images. *Middle column*: Right (RGB) camera images. *Right column*: Kinect's native depth computations

bar representation corresponds to additional pixels covered by the increased threshold from 10 cm to 30 cm.

For visual inspection, Figure 14 includes the results using 1st & 2nd iterations for the WTA of aggregated costs and plane fitting results and Figure 15 shows 3D views of the Kinect's native depth, and our computed depth for all Kinect image pairs.

From both the statistical and visual evaluation, we observe that the depth map generated by our method is comparable to Kinect native depth. Furthermore, our method can compute depth information on edges and non-fronto-

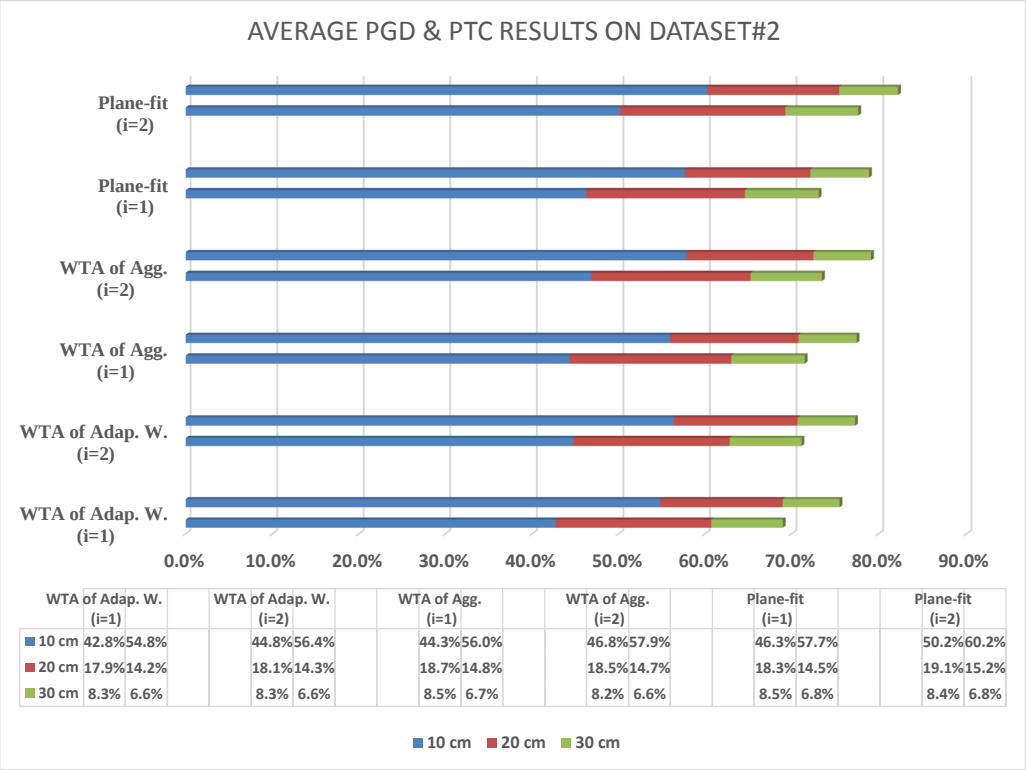
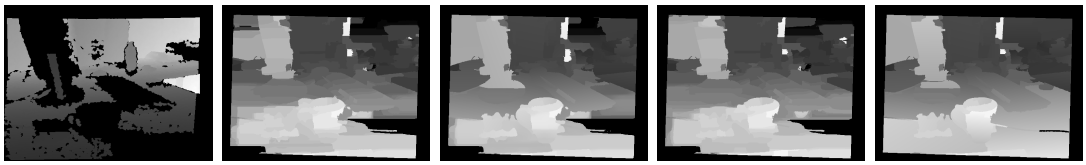


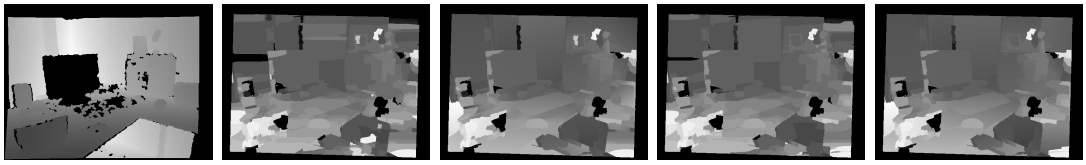
Figure 13: Average Percentage of Good Depth (PGD) and Percentage Total Coverage (PTC) results of the proposed method for the image pairs in Dataset#2 for WTA of Adaptive Windowing Costs (Adap.W.), WTA of Adaptively Aggregated Costs (Agg.), and Plane Fitting (Plane-fit) for 2 iterations (i). Note that on each line, the above bars represent PTC results and the bars beneath represent PGD results, and also the stacked bar representation shows additional depth data covered by the designated threshold [Best Viewed in Color]

planar surfaces where Kinect depth generation may fail. This suggests that our method can also be used in combination with Kinect to get better coverage of the scene. Figure 16 shows a merged 3D rendered view of the Kinect native depth map and the proposed method final depth map for the Kinect01 image in Dataset#2. As the figure suggests, the computed depth data from

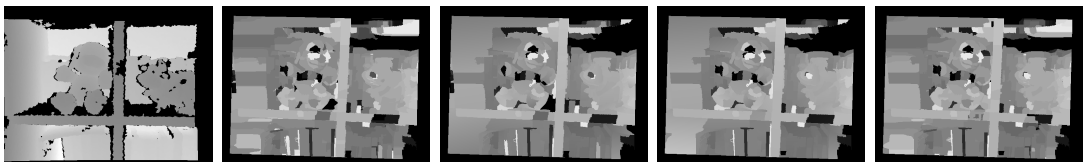
the proposed method can also be used to fill up empty depth information in the acquired scene.



(a) Kinect depth (b) WTA of Agg. ($i = 1$) (c) Plane fitting ($i = 1$) (d) WTA of Agg. ($i = 2$) (e) Plane fitting ($i = 2$)

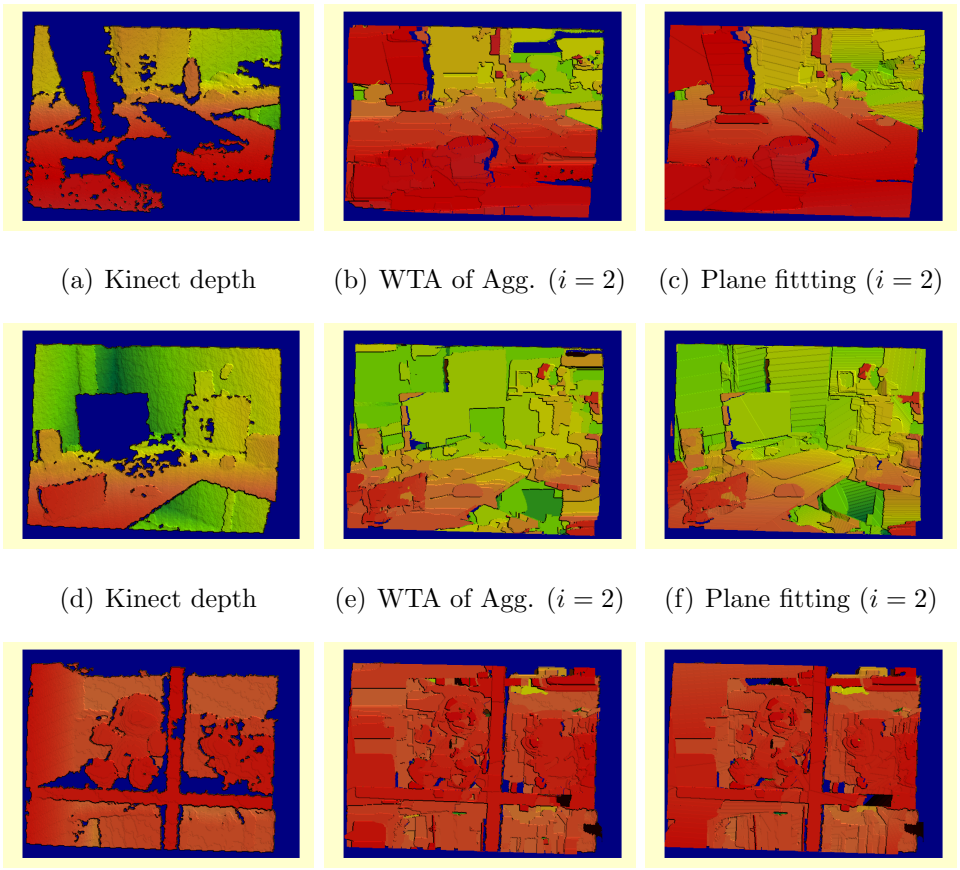


(f) Kinect depth (g) WTA of Agg. ($i = 1$) (h) Plane fitting ($i = 1$) (i) WTA of Agg. ($i = 2$) (j) Plane fitting ($i = 2$)



(k) Kinect depth (l) WTA of Agg. ($i = 1$) (m) Plane fitting ($i = 1$) (n) WTA of Agg. ($i = 2$) (o) Plane fitting ($i = 2$)

Figure 14: Our disparity estimation on the Kinect Dataset compared to native depth of Kinect: *Left column*: Kinect's native depth image (*brighter pixels are farther*). *Second column*: WTA disparity of aggregation results- 1st iteration. *Third column*: Plane fitting disparity results - 1st iteration. *Fourth column*: WTA disparity of aggregation results - 2nd iteration. *Last column*: Plane fitting disparity results - 2nd iteration.

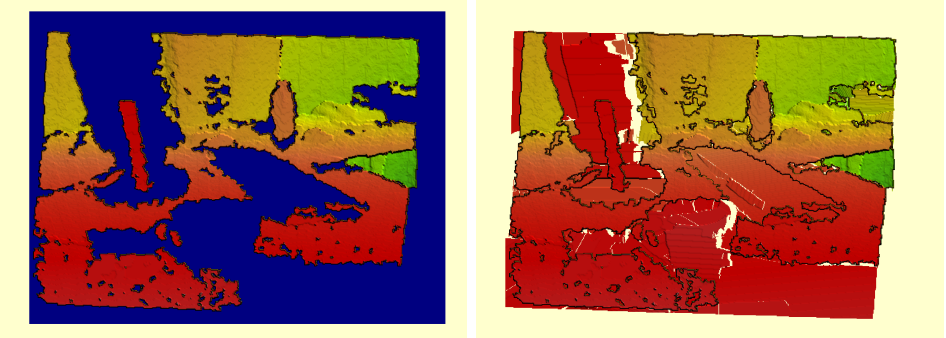


(a) Kinect depth (b) WTA of Agg. ($i = 2$) (c) Plane fittting ($i = 2$)

(d) Kinect depth (e) WTA of Agg. ($i = 2$) (f) Plane fitting ($i = 2$)

(g) Kinect depth (h) WTA of Agg. ($i = 2$) (i) Plane fitting ($i = 2$)

Figure 15: The Kinect results compared to native depth as 3D views: *Left column*: Kinect's native depths in 3D views. *Middle column*: Depth computed from WTA disparity of cost aggregation results - 2nd iteration. *Right column*: Depth computed from Plane fitting disparity results - 2nd iteration.



(a) Kinect depth (b) Merged depth

Figure 16: Merging Kinect's depth data with the results of the proposed method. **(a)** Kinect depth. **(b)** Merged depth. Note that invalid depth values in Kinect's data are filled with the estimated depth values. [Best Viewed in Color]

3.3. Comparison with Other Methods

In this section, we compare the performance of our method with existing MI-based methods in the literature. These methods are: the Egnal's method [22], which uses regular (non-adaptive) windows for the computation of MI measures (called "MI(woPR)" in the rest of the article) and the Fookes' method [35], which uses MI incorporating joint prior probabilities (called "MI(wPR)" in the rest of the article). Table 6 provides the comparison results for the WTA performances of MI(woPR) and MI(wPR) and the three main steps of the proposed method; *i.e.*, the adaptive windowing (WTA of Adap. Wind.), the cost aggregation (WTA of Agg.) and Plane Fitting for the 1st iteration and the 2nd iteration.

Table 6: Comparison with the state of the art methods on Synt. Altered Middlebury Images. MI(woPR) is MI without prior probabilities [22], and MI(wPR) is MI with prior probabilities [35].

Image*	Method	RMS (all)	RMS (nonocc.)	RMS (disc.)	Bad (all)	Bad (nonocc.)	Bad (disc.)
Tsukuba	MI(woPR)	3.701	3.660	3.473	31.15%	29.76%	35.02%
	MI(wPR)	2.640	2.552	2.812	20.51%	18.99%	24.94%
	WTA of Adap.W. ($i = 1$)	1.621	1.495	2.419	7.72%	6.64%	16.88%
	WTA of Agg. ($i = 1$)	1.425	1.315	2.433	6.15%	5.43%	17.09%
	Plane-fit ($i = 1$)	1.378	1.269	2.484	6.20%	5.47%	18.32%
	Plane-fit ($i = 2$)	1.233	1.130	2.309	4.90%	4.33%	17.02%
Venus	MI(woPR)	5.593	5.609	5.074	37.82%	37.33%	38.38%
	MI(wPR)	4.078	4.079	3.514	28.01%	27.31%	38.67%
	WTA of Adap.W. ($i = 1$)	1.729	1.689	2.464	8.77%	8.09%	25.45%
	WTA of Agg. ($i = 1$)	1.224	1.173	3.259	6.21%	5.54%	27.18%
	Plane-fit ($i = 1$)	1.003	0.939	2.754	3.42%	2.75%	19.18%
	Plane-fit ($i = 2$)	1.034	0.972	2.814	3.55%	2.88%	20.37%
Teddy	MI (woPR)	15.823	15.634	14.312	55.45%	55.01%	58.52%
	MI (wPR)	11.161	10.836	10.635	39.34%	40.57%	43.49%
	WTA of Adap.W.($i = 1$)	8.092	5.420	6.570	24.27%	23.69%	31.61%
	WTA of Agg. ($i = 1$)	7.111	4.100	5.439	19.42%	20.18%	32.24%
	Plane-fit ($i = 1$)	6.721	3.543	4.700	24.28%	24.03%	32.72%
	Plane-fit ($i = 2$)	6.133	3.175	4.817	20.33%	21.17%	32.13%
Cones	MI (woPR)	16.013	16.037	14.835	47.18%	43.75%	53.36%
	MI (wPR)	12.524	12.056	11.808	37.30%	34.85%	44.47%
	WTA of Adap.W. ($i = 1$)	10.102	8.384	10.285	27.32%	22.67%	35.24%
	WTA of Agg. ($i = 1$)	7.802	5.715	8.037	19.93%	14.80%	28.25%
	Plane-fit ($i = 1$)	7.240	5.150	7.359	19.48%	14.58%	27.24%
	Plane-fit ($i = 2$)	6.055	3.884	5.582	16.55%	11.18%	22.55%

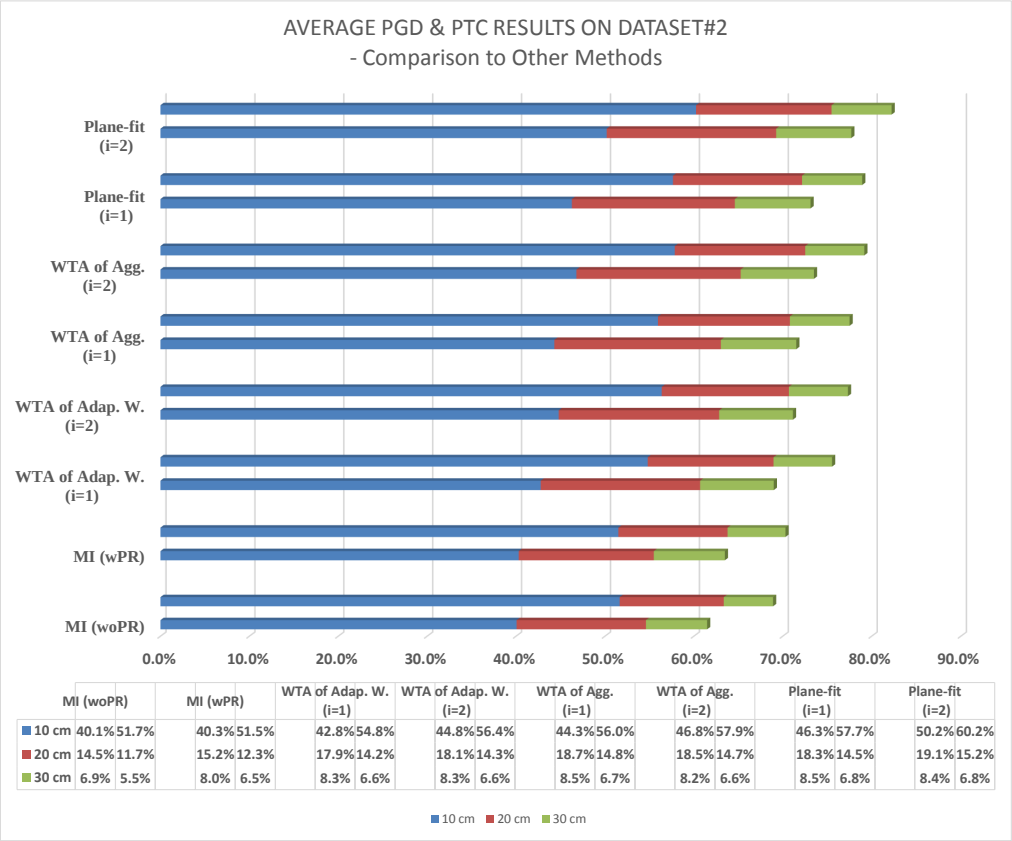
*MI(woPR), MI(wPR) methods used local window size=9x9, adaptive method vertical window size=9

We see from Table 6 and Figure 17 that the proposed method is already outperforming the other MI based methods in the literature at the initial phase of the computation, *i.e.*, computing the cost matrix using the developed adaptive windowing algorithm for MI computation. The performance figures are not only improved for the RMS error, but also the percentage of *good* pixels.

(a) (b) (c)

Figure 17: Visual results on synt. altered Middlebury images for WTA disparity selection of methods. **(a)** MI(woPR) [22]. **(b)** MI(wPR) [35] and **(c)** the proposed method Adaptive Windowing Step (WTA of Adap. W. iteration #1).

As for comparison on Dataset #2, Figure 18 shows the performance of the other MI based methods compared to the steps of the proposed method in the stacked bar graph representation. Again, in conformance with the Dataset #1 results, the proposed method outperforms the other methods compared already in the initial step of computation by the adaptive windowing step introduced.



	MI (woPR)		MI (wPR)		WTA of Adap. W. (i=1)		WTA of Adap. W. (i=2)		WTA of Agg. (i=1)		WTA of Agg. (i=2)		Plane-fit (i=1)		Plane-fit (i=2)	
10 cm	40.1%	51.7%	40.3%	51.5%	42.8%	54.8%	44.8%	56.4%	44.3%	56.0%	46.8%	57.9%	46.3%	57.7%	50.2%	60.2%
20 cm	14.5%	11.7%	15.2%	12.3%	17.9%	14.2%	18.1%	14.3%	18.7%	14.8%	18.5%	14.7%	18.3%	14.5%	19.1%	15.2%
30 cm	6.9%	5.5%	8.0%	6.5%	8.3%	6.6%	8.3%	6.6%	8.5%	6.7%	8.2%	6.6%	8.5%	6.8%	8.4%	6.8%

Figure 18: Average Percentage of Good Depth (PGD) and Percentage Total Coverage (PTC) results of the proposed method for the image pairs in Dataset#2 for WTA Disparity Selection of Methods MI(woPR) [22], MI(wPR) [35], and the proposed method's steps. Note that, on each line, the above bars represent PTC results and the bars beneath represent PGD results, and also the stacked bar representation shows additional depth data covered by the designated threshold. [Best Viewed in Color]

Finally, in Figure 19, the visual results are provided for the Kinect02 image pair in Dataset#2. A significant enhancement on the obtained disparity map can also be observed when compared to the other methods in Kinect image pairs.

(a) (b) (c)

Figure 19: Visual results on Dataset #2 - Kinect01 image for WTA disparity selection of methods. **(a)** MI(woPR) [22]. **(b)** MI(wPR) [35] and **(c)** the proposed method - Adaptive Windowing Step (WTA of Adap. W. iteration #1).

4. Conclusion

In this article, we have proposed a multi-modal stereo-vision method which (i) is iterative, (ii) uses adaptive windowing and (iii) adaptive cost aggregation (iv) along with iteratively refined disparity plane fitting. Our method uses mutual information as the basic similarity measure and we have tested it on a synthetically modified (multi-modal) version of the Middlebury Stereo Evaluation Dataset as well as IR-RGB images from a Kinect device.

Our results show that a significant increase is achieved by the adaptive windowing method when compared to other alternative MI-based methods for multi-modal stereo-vision. Moreover, the adaptively aggregated costs enhance the results while smoothing out the disparity maps, whereas plane fitting enables us to get more clean disparity maps, although it depends on the current segmentation. We balance this dependence by performing iterative segment splitting/merging over confident disparities and finally the whole method is re-applied in the next iteration where we now have an initial

disparity map to incorporate into the joint probability calculation as the prior probabilities. Our results show that two iterations are sufficient to converge with reasonable results.

With respect to the Kinect device experiments; from the quantitative and visual evaluations, we observe that the depth map generated by our method is comparable to Kinect native depth and our method can compute depth information on edges and non-fronto-planar surfaces where Kinect depth estimation fails due to insufficient reflectance of infrared beams on such surfaces. This suggests that our method can be used in combination with such RGB-D sensors especially on scenes including highly reflective surfaces.

Our method is limited only to planar surfaces like existing unimodal stereo-vision methods, though it provides reasonable estimations on curved surfaces as well. Moreover, our method does not run in real-time yet (the computational complexity is $O(Ndw)$, where N is the number of pixels in the image, d is maximum designated disparity and w is the maximum segment size in number of pixels in the image segmentation); our focus, rather, has been improving the accuracy of the existing methods. Last but not the least, our method can be integrated very well with other RGB-D sensors that use structured infra-red light for stereo, and it should also be compared against the newer versions of Kinect (the Kinect One - which uses time-of-flight principle rather than active stereo) which has not been possible during the course of this study.

References

[1] S. Krotosky, M. Trivedi, Multimodal stereo image registration for pedestrian detection, in: IEEE Intelligent Transportation Systems Conference, IEEE, 2006, pp. 109–114.

[2] S. Krotosky, M. Trivedi, Registration of multimodal stereo images using disparity voting from correspondence windows, in: IEEE International Conference on Video and Signal Based Surveillance, IEEE, 2006, pp. 91–91.

[3] S. J. Krotosky, M. M. Trivedi, Mutual information based registration of multimodal stereo videos for person tracking, Computer Vision and Image Understanding 106 (2) (2007) 270–287.

[4] Z. Zhu, T. Huang, Multimodal Surveillance: Sensors, Algorithms and Systems, Artech House Publisher, 2007, Ch. Multimodal surveillance: an introduction.

[5] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge Univ Press, 2000.

[6] R. Szeliski, Computer vision: algorithms and applications, Springer, 2011.

[7] M. Z. Brown, D. Burschka, G. D. Hager, Advances in computational stereo, IEEE Transactions On Pattern Analysis and Machine Intelligence 25 (8) (2003) 993–1008.

[8] U. R. Dhond, J. K. Aggarwal, Structure from stereo-a review, IEEE Transactions on Systems, Man and Cybernetics 19 (6) (1989) 1489–1510.

[9] N. Lazaros, G. C. Sirakoulis, A. Gasteratos, Review of stereo vision algorithms: from software to hardware, International Journal of Optomechatronics 2 (4) (2008) 435–462.

[10] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, International Journal of Computer Vision 47 (1-3) (2002) 7–42.

[11] B. Tippetts, D. J. Lee, K. Lillywhite, J. Archibald, Review of stereo vision algorithms and their suitability for resource-limited systems, Journal of Real-Time Image Processing (2013) 1–21.

[12] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: European Conference on Computer Vision–ECCV, Springer, 2006, pp. 404–417.

[13] C. Harris, M. Stephens, A combined corner and edge detector., in: Alvey vision conference, Vol. 15, Manchester, UK, 1988, p. 50.

[14] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[15] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: European Conference on Computer Vision, Springer, 2002, pp. 128–142.

[16] C. Schmid, R. Mohr, C. Bauckhage, Comparing and evaluating interest points, in: Sixth International Conference on Computer Vision, IEEE, 1998, pp. 230–235.

[17] B. Zitova, J. Flusser, Image registration methods: a survey, Image and Vision Computing 21 (11) (2003) 977–1000.

[18] V. Venkateswar, R. Chellappa, Hierarchical stereo and motion correspondence using feature groupings, International Journal of Computer Vision 15 (3) (1995) 245–269.

[19] S. Birchfield, C. Tomasi, Depth discontinuities by pixel-to-pixel stereo, International Journal of Computer Vision 35 (3) (1999) 269–293.

[20] K. Ambrosch, W. Kubinger, M. Humenberger, A. Steininger, Flexible hardware-based stereo matching, EURASIP Journal on Embedded Systems 2008 (2).

[21] P. Aschwanden, W. Guggenbuhl, Experimental results from a comparative study on correlation-type registration algorithms, Robust computer vision (1992) 268–289.

[22] G. Egnal, Mutual information as a stereo correspondence measure, Technical Report MS-CIS-00-20, University of Pennsylvania (2000) 113.

[23] M. J. Hannah, Computer matching of areas in stereo images., Ph.D. thesis, Stanford University (1974).

[24] T. Kanade, M. Okutomi, A stereo matching algorithm with an adap-

tive window: Theory and experiment, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (9) (1994) 920–932.

[25] N. Pugeault, N. Krger, Multi-modal matching applied to stereo, in: Proceedings of the British Machine Vision Conference, 2003, pp. 271–280.

[26] C. S. Park, H. W. Park, A robust stereo disparity estimation using adaptive window search and dynamic programming search, Pattern Recognition 34 (12) (2001) 2573–2576.

[27] O. Veksler, Stereo correspondence by dynamic programming on a tree, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, IEEE, 2005, pp. 384–390.

[28] C. Cassisa, Local vs global energy minimization methods: application to stereo matching, in: IEEE International Conference on Progress in Informatics and Computing (PIC), Vol. 2, IEEE, 2010, pp. 678–683.

[29] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: IEEE International Conference on Computer Vision, Vol. 2, IEEE, 2001, pp. 508–515.

[30] J. Marroquin, S. Mitter, T. Poggio, Probabilistic solution of ill-posed problems in computational vision, Journal of the American Statistical Association 82 (397) (1987) 76–89.

[31] J. Sun, N.-N. Zheng, H.-Y. Shum, Stereo matching using belief propagation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 787–800.

[32] The middlebury stereo vision page.
URL http://vision.middlebury.edu/stereo/

[33] P. Viola, W. M. Wells III, Alignment by maximization of mutual information, International Journal of Computer Vision 24 (2) (1997) 137–154.

[34] C. B. Fookes, A. Lamanna, M. Bennamoun, A new stereo image matching technique using mutual information, International Conference on Computer, Graphics and Imaging.

[35] C. Fookes, A. Maeder, S. Sridharan, J. Cook, Multi-spectral stereo image matching using mutual information, in: International Symposium on 3D Data Processing, Visualization and Transmission, IEEE, 2004, pp. 961–968.

[36] F. Barrera Campo, F. Lumbreras Ruiz, A. Sappa, Multimodal stereo vision system: 3d data extraction and algorithm evaluation, IEEE Journal of Selected Topics in Signal Processing 6 (5) (2012) 437–446.

[37] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[38] A. Torabi, G.-A. Bilodeau, Local self-similarity as a dense stereo correspondence measure for themal-visible video registration, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2011, pp. 61–67.

[39] A. Torabi, M. Najafianrazavi, G.-A. Bilodeau, A comparative evaluation

of multimodal dense stereo correspondence measures, in: IEEE International Symposium on Robotic and Sensors Environments (ROSE), IEEE, 2011, pp. 143–148.

[40] A. Torabi, G.-A. Bilodeau, A lss-based registration of stereo thermalvisible videos of multiple people using belief propagation, Computer Vision and Image Understanding 117 (12) (2013) 1736–1747.

[41] G.-A. Bilodeau, A. Torabi, P.-L. St-Charles, D. Riahi, Thermal-visible registration of human silhouettes: a similarity measure performance evaluation, Infrared Physics & Technology 64 (2014) 79–86.

[42] Microsoft's kinect for windows.
URL http://www.microsoft.com/en-us/kinectforwindows/

[43] The xbox 360 video game console.
URL http://www.xbox.com

[44] M. Beetz, D. Cremers, J. Gall, W. Li, Z. Liu, D. Pangercic, J. Sturm, Y.-W. Tai, Special issue on visual understanding and applications with rgb-d cameras, Journal of Visual Communication and Image Representation 25 (1) (2014) 1–238.

[45] M. Yaman, S. Kalkan, Multimodal stereo vision using mutual information with adaptive windowing, in: 13th IAPR International Conference on Machine Vision Applications, IAPR, 2013.

[46] A. Klaus, M. Sormann, K. Karner, Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure, in: Inter-

national Conference on Pattern Recognition, Vol. 3, IEEE, 2006, pp. 15–18.

[47] Y. Taguchi, B. Wilburn, C. L. Zitnick, Stereo reconstruction with mixed pixels using adaptive over-segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[48] H. Tao, H. S. Sawhney, R. Kumar, A global matching framework for stereo computation, in: IEEE International Conference on Computer Vision, Vol. 1, IEEE, 2001, pp. 532–539.

[49] Z.-F. Wang, Z.-G. Zheng, A region based stereo matching algorithm using cooperative optimization, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[50] Q. Yang, L. Wang, R. Yang, H. Stewénius, D. Nistér, Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (3) (2009) 492–504.

[51] C. L. Zitnick, S. B. Kang, Stereo for image-based rendering using image over-segmentation, International Journal of Computer Vision 75 (1) (2007) 49–65.

[52] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, ACM Transactions on Graphics (TOG) 23 (3) (2004) 600–608.

[53] C. M. Christoudias, B. Georgescu, P. Meer, Synergism in low-level vision, in: 16th International Conference on Pattern Recognition, 2002, pp. 150–155.

[54] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.

[55] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[56] Middlebury stereo evaluation - version 2.
URL http://vision.middlebury.edu/stereo/eval/

[57] Middlebury stereo evaluation - version 2 new features and main differences to version 1.
URL http://vision.middlebury.edu/stereo/eval/newFeatures.html

[58] Rgbdemo software - calibrating kinect with openni backend.
URL http://labs.manctl.com/rgbdemo/index.php/Documentation/Calibration