# 3D Modeling Based on Disparity Estimation for Image-Based Virtual Environments

Xiaodong Huang

A thesis submitted to the University of Ottawa in partial fulfillment

of the requirements for the degree of Doctor of Philosophy.

May 2008

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario, Canada

Copyright ©Xiaodong Huang, 2008

# Abstract

This thesis addresses the challenging problem of obtaining 3D models for real environments from stereo images and translational video sequences. The problem is partitioned into two main parts: matching and disparity estimations to obtain depth maps and separate 3D models for different image locations, and the combining of these separate 3D models into one 3D model for the whole environment. Solutions are proposed dealing with these two main issues respectively, and the results from implementing these solutions are also presented. The novelty for the solution of the first part – which deals with the matching problem – lies in the fact that it combines the pixel-based approach and region-based approach. A hybrid algorithm is developed for disparity estimation between stereo images and translational video sequences, so that some intrinsic problems in the pixel-based and region-based approaches (like the detection of sky) can be solved in a combined way while keeping the object boundaries sharp and crisp, and hence give disparity and depth maps which are qualitatively better than traditional methods. The novelty for the solution of the second part – which deals with the integration of separate 3D models from different image locations – comes from the usage of region information obtained from the disparity estimation process, the estimation of ego-motion parameters, as well as to the integration of the object surfaces from different 3D models. For the ego-motion estimation, instead of using the bundle adjustment or iterative closest points (ICP) which perform the estimation in 3D space, our algorithm uses large regions with correspondence information in each image to determine the homogeneous transformations in image space. For the integration of separate 3D models, these large regions from different models are also used to adjust and expand the shape of regions which belong to the same surfaces, so that even the occluded surfaces in one image location can be filled by integration of surfaces from other images. The results are shown with the image rendering at novel viewpoints as well as with PSNR values measured between the rendered images and existing images at real image locations.

# Acknowledgements

I would like to express my heartfelt gratitude to my supervisor Professor Eric Dubois, for his willingness to support my research work while giving me key help whenever necessary, as well as his patience and invaluable guidance that helped me finish all the research phases and this thesis.

My thanks go to the Natural Sciences and Engineering Research Council of Canada for their financial assistance. My thanks also go to the staff of the School of Information Technology and Engineering of the University of Ottawa for their administrative help.

Last but not least, my thanks to my parents and my wife who gave me precious moral support with great patience.

Xiaodong Huang

Ottawa, Canada, May 2008.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| LDI | Layered Depth Image |
| IBR | Image-Based Rendering |
| SSD | Sum of Squared Difference |
| SAD | Sum of Absolute Difference |
| GC | Graph Cut |
| PSNR | Peak Signal to Noise Ratio |
| ICP | Iterative Closest Points |
| SVD | Singular Value Decomposition |

# Chapter 1

# Introduction

The field of image-based virtual environments is a promising research area. This technology allows one to navigate through a photo-realistic environment, in which all the novel views are rendered based on a set of precaptured images or video sequences of that environment. The related applications include virtual museums, virtual sightseeing, flight simulation, augmented reality, computer games, etc.

The objective of this thesis is to develop an application in which, based on a set of precaptured images and videos in and around a real environment, the 3D model of the whole environment can be obtained through disparity estimation and 3D model integration. Then a user can virtually and freely move around in this environment and see the scene from novel viewpoints along his/her virtual trajectory, just like moving a virtual camera through the environment and generating the image sequence that camera could have acquired. As shown in Fig. 1.1, where the solid line represents the path of the real camera capturing the scene, and the dashed line represents the path of a virtual camera generating novel views, the path of the virtual camera and view directions can be controlled by the user and be completely different from those of the actual camera.

This application has some aspects in common with 3D model construction and rendering in computer graphics, in which one can arbitrarily synthesize a 3D model of an environment and then move around this synthetic environment through graphic view rendering. However, the difference between the proposed application and the computer graphics approach lies in

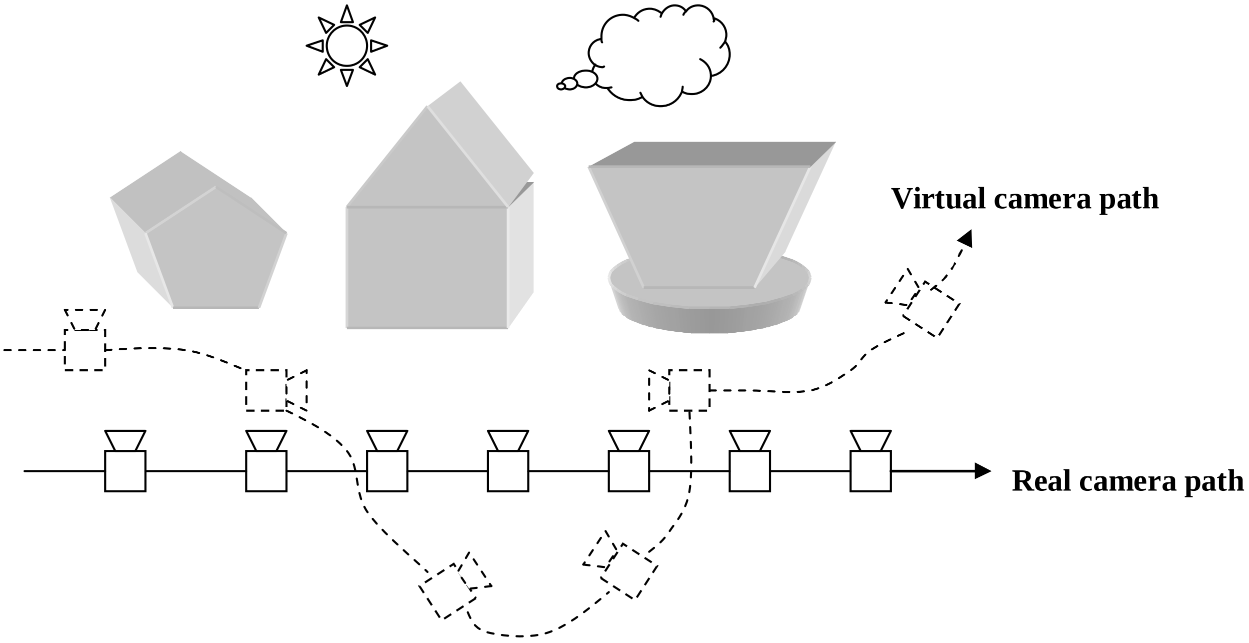


Figure 1.1: Image/video capture and virtual rendering

the fact that the representation of the environment is obtained from the acquired images and video sequences rather than synthesized by computer graphics. Therefore, our objective can be summarized as follows: acquire a set of images and videos of a real environment, and then generate an arbitrary sequence of novel views within this environment.

## 1.1 Problem Area

The objective of this thesis is closely related to the techniques of *image-based rendering* (IBR). As shown in Fig. 1.2 [1], the existing techniques for image-based rendering can be arranged according to their dependence on geometric information of the scene, from the most geometry-dependent texture mapping, to light-field rendering which requires no geometric information. The number of images and the constraints on the positions where pre-captured images are taken increase as the geometric dependence decreases. We summarize these techniques in the following in the order of increasing dependence on geometries:

(1) *Light field*, *lumigraph*, *concentric mosaic* and *panorama* – these techniques require

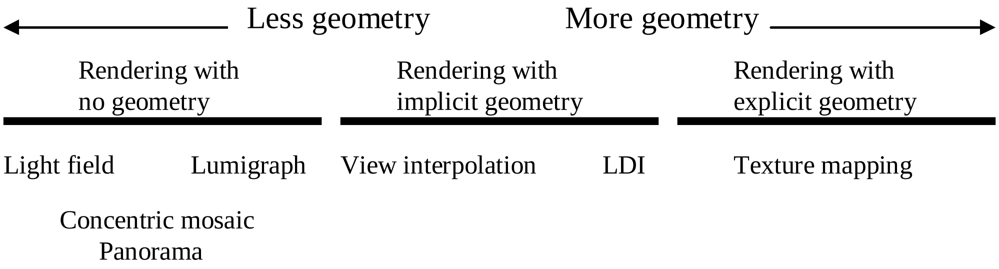


Figure 1.2: Classification of image-based rendering (adapted from [1])

no geometric information or only a little geometric information (lumigraph). However, they require a large number of images to be acquired. Especially for the light field and lumigraph, those images have to be captured at equally-spaced grid positions surrounding the scene. This makes the image acquisition stage very demanding, and the effective storage and retrieval of these images are very challenging work. For the concentric mosaic and the well-known panorama technique, circular movements are used for image acquisition. The requirements for the number of images and acquisition positions are not as demanding as for the light field and lumigraph, but the rendered viewpoints have to be constrained to the central point of the circle (panorama), or within a very small area around the central point (concentric mosaic).

(2) *View interpolation* and *layered depth image (LDI)* – these techniques require implicit geometric information. The "*implicit geometric information*" means the matching or correspondence relation among the acquired images, which is usually related to the scene depth. For *view interpolation*, instead of setting up the 3D models of the scene, the matching or depth information is exploited to render novel views by interpolations based on multiple images, e.g., [5]– [10]. These algorithms usually render the intermediate views between a stereo pair or any two consecutive images in a video sequence [5]–[9], or a novel view which is a little bit off the line that connects the stereo pair [10]. Although these algorithms do not deal with 3D models directly, they need dense or sparse correspondence for the interpolation processes, and the quality of the correspondence is directly linked to the performance of interpolation results. The layered depth image (LDI) was developed so that there is no need to construct the actual objects from their depth information, but just to combine the depth

information from different images to one reference location to form a layered depth – a kind of 3D image where each pixel coordinates has not only one pixel value, but also the values of pixels behind it for the same coordinates [11][12]. Those pixels are detected from other image locations and are arranged by their depth values. The rendering can thus be achieved based on such 3D LDI to any novel viewpoints through transformation and interpolation as long as the LDI is dense enough to supply sufficient information to these viewpoints. Thus, we can see that the main difference between the view interpolation and LDI is that the rendering of LDI is less constrained, rather than being limited to be near the intermediate viewpoints as for the view interpolation. However, this comes at the cost of higher demand for the accuracy of the matching process since the construction of LDI needs accurate information for the relative locations of different images in order to transform the scene depth at different locations to the reference location. A method of obtaining LDI for a real scene, which is actually a small object located in the center of a rotating round table was developed in [13], where the rotation angles are known and only the LDI of the central object rather than the whole environment is constructed.

(3) *Texture mapping* – using explicit geometry. *Texture mapping* is usually used in computer graphics where the scene is composed of synthetic objects and thus the geometric information of the whole scene is known. However, until now, using texture mapping for a real environment is very difficult due to the difficulty in obtaining the depth information of the scene from images, and the difficulty in combining the depth information from different images to get the geometric information of all the objects in order to construct the whole scene. Although 3D scanners can be used to obtain the model of the real scene, there are still matching problems – the matching of scanned models and the texture on the surface of these models – to be solved, along with the problem of 3D registration in order to merge the 3D models from different viewpoints. In addition, those 3D scanners are usually very expensive and hence makes the application very costly.

Therefore, for IBR, there is always a trade-off between the difficulty of obtaining the depth and geometric information of the real scene and the difficulty of capturing and storing a large number of images at required positions. Due to these difficulties, until now, only the algorithms related to *concentric mosaic* and *panaroma*, which require no geometric information nor large amounts of image data, are comparatively mature, at the cost that the rendering is only limited to the center point or a central area.

## 1.2 Unsolved Problems for 3D Model Construction of Real-World Scenes

Comparing our objective and the summary of the existing IBR techniques, we can see that in order to achieve this objective we should use the method of either LDI or texture mapping, since only these two methods allow us to render novel views comparatively far away from the actual camera paths, rather than near to the locations of acquired images. The reason that we do not choose the light field or lumigraph is that they are too demanding and expensive (they need a large number of cameras) for their image acquisition processes to apply them to practical applications, especially for large environments. Choosing between the LDI and texture mapping methods, we propose to use the method of texture mapping. The reason is that, after transforming the depth informations from all images to one reference location, the texture mapping approach tries to form models of the scene by converting these 3D points to surfaces while the LDI approach only organizes these points as 3D point clouds according to their depth values without constructing surfaces and geometric models for different objects. This makes the rendering process for the LDI approach inconvenient and vulnerable to artifacts since there is a complex formula to determine a block size for each pixel in LDI to interpolate on the novel image [11], and if the novel viewpoint is far from the reference location then those interpolated blocks will affect each other. For the texture mapping approach, no matter whether the novel viewpoint is far from, or near to, the reference location, its rendering process will not bring artifacts such as blocking and ringing, since the rendering is based on the constructed model with certain geometries. Furthermore, there are software tools like OpenGL which supply integrated functions for the interpolation process of texture mapping, and this will make the rendering procedure in our objective efficient and convenient.

To apply texture mapping on the real environment, a matching process, which is actually the estimation of pixel displacements among different images for the same 3D points, has to be performed on the captured images and videos in order to obtain the depth information at different image locations. Some recent examples are [7], [14] and [15], among others. Such a matching process can be performed either by extracting and matching feature points [16] or edges [17], or by using disparity estimation from stereo images [18].

Thus we can see that, for the purpose of setting up the application of image-based virtual environments using explicit geometry, the basic starting point is the estimation of displacement and depth maps; the following issue is how to combine those depth maps together or how to make use of them. *Displacement estimation* includes motion estimation for video sequences, disparity estimation for stereo images or multiview images, or joint disparity/motion estimation for stereo video sequences. Fig. 1.3 shows a general block architecture for the construction of 3D models for the real scene. Once the matching process (disparity and/or motion estimation) is performed for one image location, then the depth map for that location can be obtained from disparity or from motion values through the algorithm of *structure from motion* (SfM). Then, to obtain the 3D models for other locations, a stereo video sequence is usually adopted for disparity and stereo approach (upper branch of Fig. 1.3) and joint disparity/motion estimation should be applied to obtain disparity/depth maps of different locations as well as to obtain the homogeneous transform parameters among these locations. If a monocular video sequence is adopted (lower branch of Fig. 1.3) then motion estimation and SfM can be applied to the following images so that the depth maps for other locations can be obtained together with the estimation of homogeneous transform parameters from motion values. Finally, with the estimated depth maps for different locations and the associated homogeneous transform parameters, the structure or 3D model for the whole real environment can be constructed by combining the separate 3D models transformed to the reference location, and novel views can be rendered based on such structure or 3D model.

The basic reason that the image-based virtual environment using an explicit-geometry approach is not becoming a popular approach so far lies in the fact that there are no robust

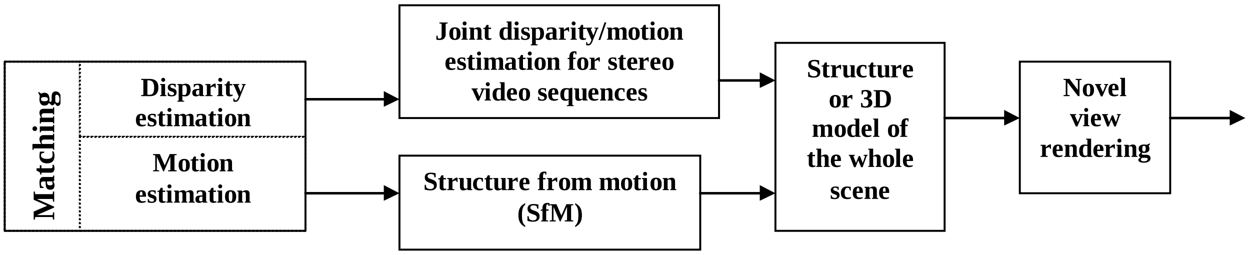

Matching
Disparity estimation
Motion estimation
Joint disparity/motion estimation for stereo video sequences
Structure from motion (SfM)
Structure or 3D model of the whole scene
Novel view rendering

Figure 1.3: Block architecture for setting up 3D model for real environment

and general solutions for both disparity and motion estimation, i.e., there are no algorithms which can accurately estimate scene depth either from stereo images or from motion analysis. In addition, even with perfect depth maps at different locations, we still have the problem of how to integrate them. In detail, the difficulties come from the following facts:

(1) Both algorithms – disparity and motion estimation – are matching processes, and matching is an ill-posed inverse problem. The ambiguities contained in these matching processes can bring many errors in the estimated depth images, such as noisy outliers and incorrect depth values for an entire large region. Such errors can greatly affect the following steps to set up one 3D model for the whole environment. Some typical difficulties in depth estimation are the matching for the pixels in untextured and slanted surfaces, the matching for sky areas (especially when the sky is segmented by some trees), etc.

(2) In order to set up a 3D model within which free navigation can be achieved, each surface of different objects has to be represented as a whole entity, e.g., by triangular meshes or NURB splines. Otherwise, if the 3D model is constructed simply by putting all the 3D points together, then in the rendering process the scene will split. This requires that the 3D surfaces be identified based on depth maps from disparity estimation, or SfM must be performed, so that the surfaces that belong to different objects can be distinguished and prevented from mixing with other objects. The difficulties in this issue are mainly related to the first problem – if the quality of depth maps is poor (like outliers or one surface with several discontinuous depth values) then no accurate 3D segmentation can be achieved.

(3) In most cases, the depth and the related 3D models estimated at different locations cover only part of the whole scene, i.e., the 3D models for each location are only separate 3D

models. In order to combine these separate 3D models to form *one* 3D model for the whole scene, the relations among the locations of these separate 3D models need to be found. After obtaining such relations, the separate 3D models can be transformed to *one* reference. At this reference location, each separate 3D model should have part of the 3D surfaces overlapping with other 3D models, and part of the surfaces not covered in other 3D models. Thus, in the combining stage, the overlapping as well as non-overlapping surfaces for each separate 3D model need to be identified, and the integration strategy on how to combine such overlapping surfaces along with non-overlapping surfaces needs to be set up. In image-based modelling, the separate 3D models estimated from images are usually noisy, and this will result in many ambuiguities for the integration stage and affect the quality of the final model.

## 1.3 Proposed Solutions

Until now, most of the matching algorithms (both disparity and motion estimation) are pixel-based, which means the matching process is performed pixel-by-pixel. Compared to the large amount of literature on pixel-based matching algorithms, there are only a few papers that make use of image segmentation and region matching techniques. Using region-based matching techniques can largely alleviate the ambiguities associated with the pixel-based matching, and the disparity and depth values with good quality can further be used to obtain 3D surfaces by evaluating if any adjacent regions belong to one 3D surface according to their disparity and depth values. Therefore, we will combine segmentation and region-based approaches in order to solve the first two of the three above-mentioned problems in a joint way. For the third problem, 3D integration, we will estimate the homogeneous transformation parameters (three rotations plus three translations) between each location with respect to the reference location. Then, the integration process will be performed by adjusting and expanding the shape of regions at the reference image location before the Delaunay triangulation is applied to each region. In detail, our solutions can be divided into the following steps relative to the three main problems mentioned above:

(1) Making use of region information which comes from the segmentation algorithm, and

performing region matching for disparity estimation. From the various possible segmentation algorithms (color-based, pattern-based, etc.), we will choose color-based segmentation for the first step. Because the segmentation in this first step will be applied to only *one* reference image on which the disparity or motion estimation is based, it is more objective to judge if a group of connected pixels belong to one surface by their colors. Since the color-based segmentation algorithm is based on color or intensity variations, one surface full of textures might be segmented into different regions. Although using a region matching scheme can dramatically reduce the ambiguities associated with the matching process, there are situations that region matching cannot effectively handle, e.g., the background sky segmented by foreground objects like trees or shrubs. In such situations, the region matching will bring results in which the segmented sky regions are considered as part of the foreground objects and thus have the same disparity/motion values. We handle this problem by developing a hybrid disparity estimation algorithm in which the pixel-based and region-based approaches are combined to solve all these problems and to obtain disparity/motion maps with high quality.

We start the algorithm by pixel-based approach through Gabor filtering or sum-of-squared-difference (SSD) for an initial and coarse disparity map. Then we implement the color-based segmentation. Based on the obtained regions and the coarse disparity map, a region manipulation and merging scheme is carried out so that different but adjacent regions which actually belong to one object surface can be merged into one region if their disparity values show some smooth and continuous properties along the borders of these adjacent regions. In addition, the regions with zero disparity value can be identified by a variational regularization approach. Finally, a region matching algorithm is applied to the regions with non-zero disparity values.

(2) Once we obtain regions with disparity values for each image, we apply a Delaunay triangulation to each region so that each region represents a 3D surface with triangular meshes. Then, the parameters of homogeneous transforms which describe the relations among different image locations in 3D space will be estimated in the image domain. This is unlike the usual alignment method used for laser scanned models in which the parameters of homoge-

neous transform are estimated in 3D space by searching the nearest 3D points, since we have not only the depth and 3D information as from laser scanned models, but also the texture information on all the 3D surfaces from the images. In order to fully exploit such texture information, we will use a cost function expressed by image data directly combined with parameters of homogeneous transforms. After these homogeneous transforms are estimated, the 3D surfaces from each of the separate 3D models will be transformed to the reference location. Then the integration process is applied to the regions at the reference image location by adjusting and expanding the regions according to the shapes of their corresponding regions from other images. After the regions in the reference image are integrated with their corresponding regions from other regions using such procedure, we apply the Delaunay triangulation to these regions, and the textures can be mapped to these regions from different images.

One distinct property of our algorithm is that it makes use of region information for the disparity estimation. Among the current disparity estimation algorithms, most of them are pixel-based, while some others use region matching. Due to the ill-posed nature of disparity estimation, there are intrinsic problems which both pixel-based matching and region matching could not solve, e.g., the matching for untextured areas in pixel-based matching, and the matching for sky areas with foreground objects in region matching. In addition, for region matching, there is the problem on how to obtain the correct regions through segmentation and how to efficiently manipulate them (e.g. merging). Our disparity estimation algorithm uses a set of procedures incorporating both pixel-based matching and region matching in order to solve such intrinsic problems in a joint way.

Currently, our disparity estimation algorithm only works for parallel stereo cases in which disparities only happen along horizontal direction. For non-parallel stereo cases in which there are also vertical disparities, rectification is needed to adjust the images so that there are only horizontal disparities between the rectified images, before applying our algorithm to the stereo image pair.

One feature of our algorithm is that there are many parameters to be chosen for different stages of the whole process. For example, the quantization parameter for the segmentation stage, the threshold of the number of pixels for the definition of large regions and small regions, the number of iteration steps for the variational regularization, etc. Currently, most of them are determined empirically, and are used for all the three image sets in this thesis. Although the robustness of these parameters to other image sets needs to be verified, some of them can compensate each other and therefore there exists some internal robustness (e.g., if the quantization parameter for segmentation is small, resulting in more small regions, such more regions might still be merged with surrounding regions in the later small region merging stage to eliminate outliers in the disparity maps).

## 1.4 Summary of Contributions

This thesis contains contributions in two main areas related to image-based 3D reconstructions: image matching and 3D model integration.

Our matching algorithm is mainly focused on dense disparity estimation for stereo images, which starts from pixel-based approaches [19][20], as well as from the traditional block-based SSD (*sum-of-squared-difference*). We extend these algorithms to a hybrid approach which depends on both pixel-based and region-based matching techniques. In this way, we can obtain disparity maps with very high subjective impression which are coherent to actual object surfaces. The main value of our algorithm, as well as a property which makes it distinct from other disparity estimation algorithms, lies in the fact that it combines the pixel-based and region-based approaches so that some intrinsic problems (keeping the object boundaries sharp and crisp, linear variation for slanted surfaces, identification of zero disparity for sky regions, etc.) of each approach can be solved in a combined way.

Our 3D model integration scheme is performed in image space by using image intensities and disparities, rather than in 3D space by using 3D point sets as for some popular algorithms. Thus, both of our ego-motion estimation and final integration procedure are performed in image space with implicit 3D information and with Delaunay triangulation applied only at the very final stage, rather than in 3D space with explicit 3D information and triangular meshes. This is a main difference of our integration procedure with most of the existing algorithms, which is also a novelty for our approach. Compared with the integration in 3D space for laser scanned models, such procedure will increase the efficiency while reduce the ambiguities cause by the usually noisy 3D models from image matching results.

## 1.5 Thesis Organization

In this thesis, we first give in Chapter 2 an overview of the disparity and motion estimation algorithms currently used, as well as the algorithms for 3D alignment. Then in Chapter 3, the new hybrid matching algorithms that we developed are shown, in which a robust pixel-based disparity estimation scheme based on the Gabor transform is first given, followed by the description on how we combine a color-based segmentation algorithm (*Mean Shift*) to the region matching based on the pixel-based matching results. Also in Chapter 3, we present our modified hybrid matching algorithm in which the problem on how to identify the regions with zero motion can be better handled (this is particularly useful for the detection of sky regions which usually give zero disparity values in stereo images), and apply this modified matching algorithm to a translational video sequence with outdoor scenes in order to show the effectiveness of this modified matching algorithm. In Chapter 4, we continue to improve our hybrid matching algorithm based on our own color-based segmentation scheme and region merging techniques along with some point-based 3D reconstructions. Then in Chapter 5, the methods that we developed for 3D model alignment and integration, which are different from the frequently used methods for laser scanned 3D models are presented, together with mesh-based 3D reconstructions and some quantitative performance measurements. Finally, conclusions and future directions are given in Chapter 6.

# Chapter 2

# Background on Disparity/Motion Estimation and 3D Model Integration

In this chapter, we give a review of the current literature and algorithms for disparity and motion estimation, as well as for the integration of separate 3D models.

## 2.1 Disparity Estimation

In the area of computer vision, disparity estimation using stereo image pairs has been a longstanding problem. As shown in Fig. 2.1, assume we have a pair of stereo images $I_l$ (left image) and $I_r$ (right image). The disparity value for a pixel with coordinates $(x, y)$ in $I_l$ is defined as the displacement between the coordinates of $(x, y)$ and its corresponding pixel $(x_r, y_r)$ in $I_r$. In the special case of parallel stereo, which we are dealing with in this thesis, this displacement can be approximately regarded as only occurring in horizontal direction

$$d(x, y) = x - x_r \tag{2.1}$$

with $y = y_r$. The depth value for $(x, y)$, which is the distance between its real 3D world coordinates $(X, Y, Z)$ and the baseline (the line that connects the two projection centres of $I_l$ and $I_r$), can be obtained as

$$Z(x, y) = \frac{fB}{d(x, y)} \tag{2.2}$$

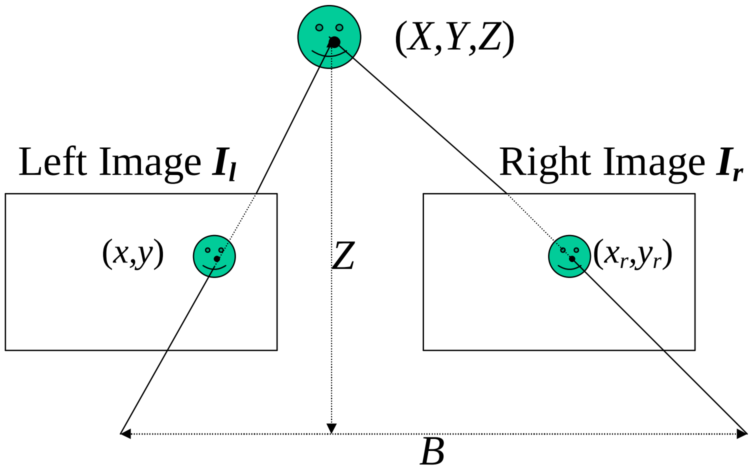


Figure 2.1: A standard stereo pair

where $f$ is the focal length of the camera, and $B$ is the baseline distance.

Many algorithms have been proposed to handle the issue of disparity estimation, and they can be classified into the following main categories:

- *Feature-based methods*: These algorithms find the correspondence between some feature points of the stereo image pair, and usually give sparse disparity maps.

- *Block-based methods*: The most often used methods, which find the correspondence for a block area in one image by comparing it with some blocks in shifted regions in another image.

- *Energy-based methods*: These approaches estimate the correspondence in a minimization and regularization formulation, which usually consists of an iterative solution of the associated discretized Euler-Lagrange equations [21]–[25], or use of some optimization algorithms like *graph cuts* [2][3] for the minimization of the associated functionals.

- *Phase-based methods*: Based on the Fourier phase information, the correspondence is estimated from the phase difference between the left and right Fourier-phase images. Another important algorithm that belongs to this category is to apply the Gabor transform to the images and find the disparity from the coefficients of the transform [26][27][10].

- *Parametric model methods*: This kind of method uses a function which depends on several parameters to express the shape of a disparity map, converting the problem of disparity estimation to the problem of parameter optimization for the disparity functions [28][29].

### 2.1.1 Feature-Based Methods

There is a great deal of literature dealing with the matching of feature points and edges, under various conditions like small-baseline or large-baseline, two-view or three-view, etc. Since such methods only give sparse disparity maps, while what we need are dense ones, they did not fall into our interests at first. However, as our research results so far have demonstrated, we found that it is very helpful to combine the matching results from feature-based methods with the matching process of dense disparity. This can give significant improvement in keeping the contours and shapes of objects clear in dense disparity matching.

There are different selection criteria for features and the resulting features selected vary, e.g. Harris corners [30] and SIFT features [31], and this gives only feature points. The matching for these feature points can be performed in two-view or three-view contexts [32] exploiting epipolar constraints. Other criteria include contour-based [33][34][35] as well as the one based on line segments [36]. These contour- or line-based algorithms usually detect edge pixels first, and then form a certain number of contours by linking connected edge pixels and try to find the mapping of those contours in another image by some optimization methods like least squares.

### 2.1.2 Block-Based Methods

Block-based methods are the most traditional methods for disparity estimation and are still used extensively. The basic idea is to compare the pixel values in a small block of one image in the stereo image pair with several candidate blocks in another image. There are three main matching cost functions for block-based methods: *sum-of-squared-difference* (SSD), *sum-of-absolute-difference* (SAD), and *normalized cross-correlations*. A very good survey of these methods as well as some extension methods based on block-based techniques can be found in [37].

The general matching cost function of SSD is given as (assuming horizontal disparity only)

$$E_{SSD}(d) = \sum \sum_{x,y \in \mathcal{B}} [I_l(x,y) - I_r(x-d,y)]^2 \tag{2.3}$$

where $I_l$ and $I_r$ are left and right images respectively, and $\mathcal{B}$ is the block containing a certain number of pixels. Usually there is a searching range for $d \in \{0, d_{max}\}$ $d \in \mathbf{Z}$ is an integer, and the disparity $\hat{d}$ for the pixels in $\mathcal{B}$ is determined to be the value which gives the minimum value of $E_{SSD}(d)$ in the searching range:

$$\hat{d} = \arg\min_{d} E_{SSD}(d). \tag{2.4}$$

Similarly, for the case of SAD, the matching cost function is defined as

$$E_{SAD}(d) = \sum \sum^{y}_{x,y \in \mathcal{B}} |I_l(x,y) - I_r(x-d,y)| \tag{2.5}$$

and the disparity value $\hat{d}$ for a block is determined in the same way as Eq. (2.4) for the SSD.

For the case of normalized cross correlation, its matching cost function is

$$E_{corr}(d) = \frac{\sum^{x} \sum^{y}_{x,y \in \mathcal{B}} [I_l(x,y) I_r(x-d,y)]}{\sqrt{\sum^{x} \sum^{y}_{x,y \in \mathcal{B}} I_l^2(x,y)} \sqrt{\sum^{x} \sum^{y}_{x,y \in \mathcal{B}} I_r^2(x-d,y)}} \tag{2.6}$$

and the disparity $\hat{d}$ for the pixels in $\mathcal{B}$ is determined to be the value which gives the maximum value of $E_{corr}(d)$ in the searching range:

$$\hat{d} = \arg\max_{d} E_{corr}(d). \tag{2.7}$$

We show some disparities estimated by SSD and SAD. The software can be downloaded from *www.middlebury.edu*, and the related document is [37]. First, we need to introduce some standard stereo image pairs that are used frequently in the literature on disparity estimation. The first one is called *Tsukuba*, its left and right images are shown in Fig.2.2. This pair also has a manually labeled disparity map – the so called *ground truth* disparity – in order that the estimated disparity for *Tsukuba* from any algorithm can be compared with a "true"

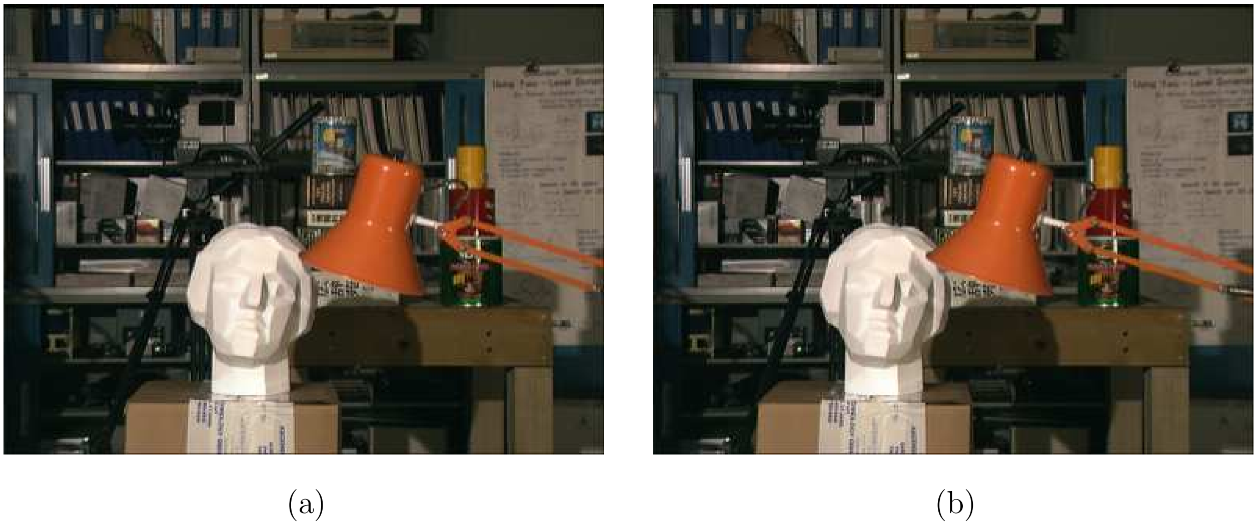


Figure 2.2: Original *Tsukuba* pair (288×384): (a) left image; (b) right image.

disparity map to obtain a quantitative performance measure for that algorithm. The *ground truth* disparity is shown in Fig. 2.3, in which the intensity values are proportional to disparity values – the brighter the intensity the larger the disparity values. Although Fig. 2.3 is called "*ground truth*", it doesn't mean that the disparities for all the pixels are correct, because it does not take into account occluded pixels, and does not reflect the continuous variation of disparity values (all the disparity values in Fig. 2.3 are integers). The disparity maps estimated by SSD and SAD (using a block with size $9 \times 9$) are shown in Fig. 2.4.

The second stereo pair is from the sequence *Flower Garden*, which was taken by translational movement of the camera. Here we use the $21st$ and $23rd$ images as left and right images, as shown in Fig. 2.5. The disparity maps for this stereo pair estimated by SSD and SAD are shown in Fig. 2.6 ($9 \times 9$ for block size).

Both *Tsukuba* and *Flower Garden* are images of real environments. There are more synthetic stereo pairs in [37], but since we are dealing with the disparity estimation for IBR of real environments, we did not use any synthetic image pairs.

The third image pair was taken by ourselves in VIVA Lab, and is shown in Fig. 2.7. The disparity maps for this stereo pair estimated by SSD and SAD are shown in Fig. 2.8 ($9 \times 9$ for block size).

From the results in Fig. 2.4, Fig. 2.6 and Fig. 2.8, we can see that the disparities estimated

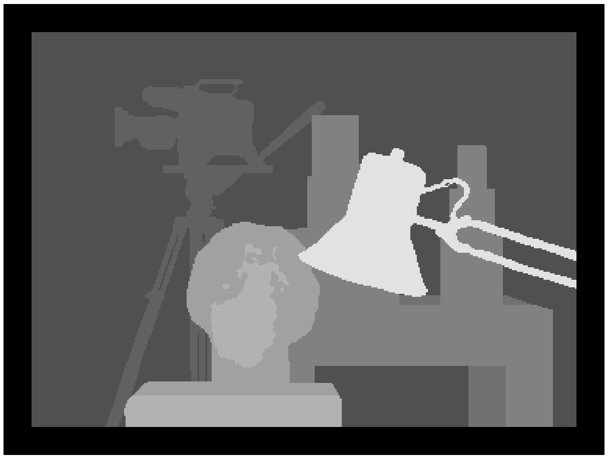


Figure 2.3: *Ground truth* disparity map for *Tsukuba*

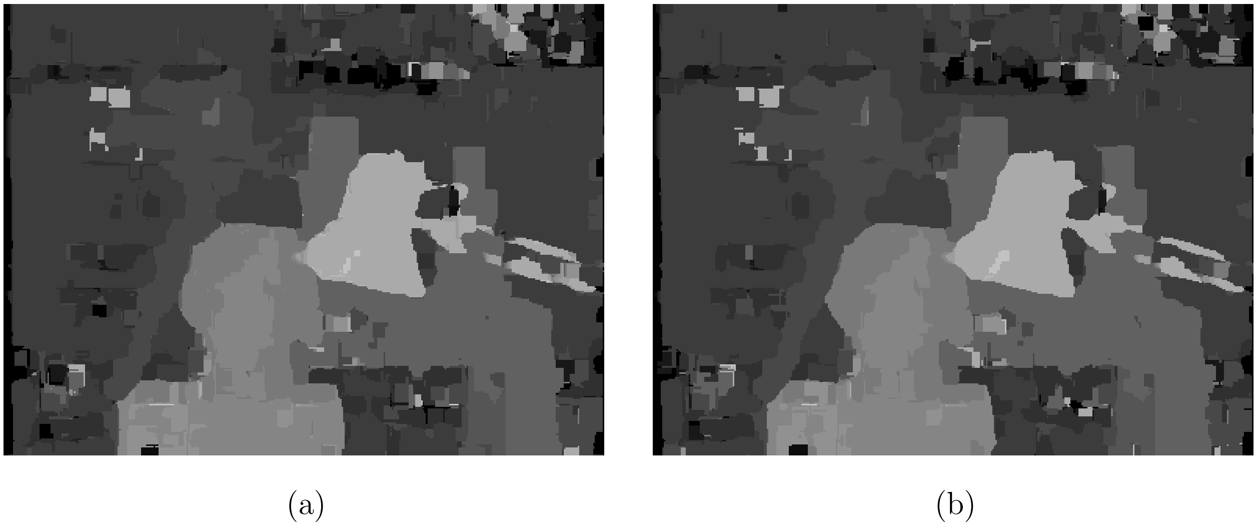
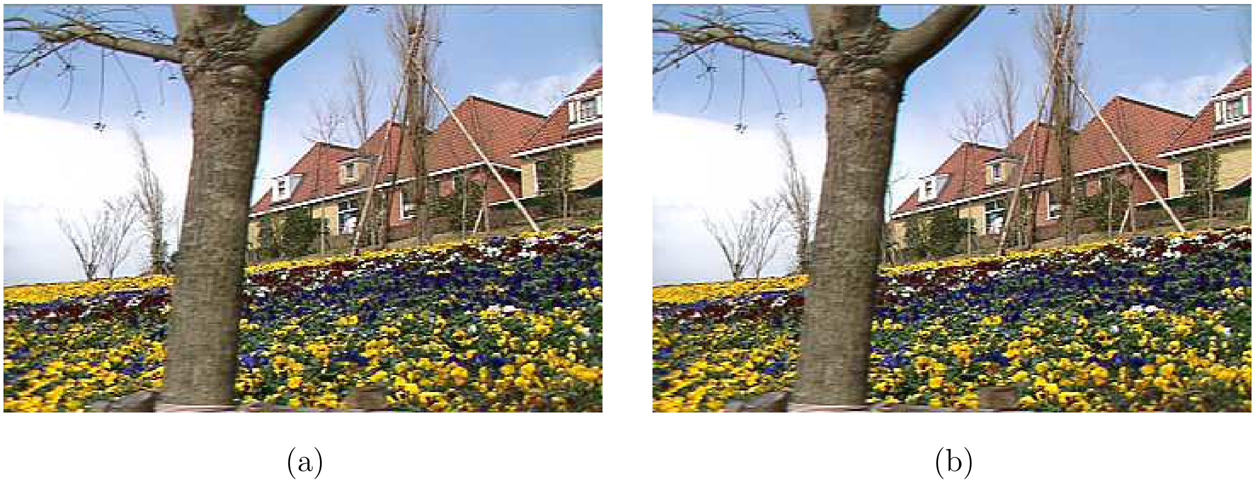
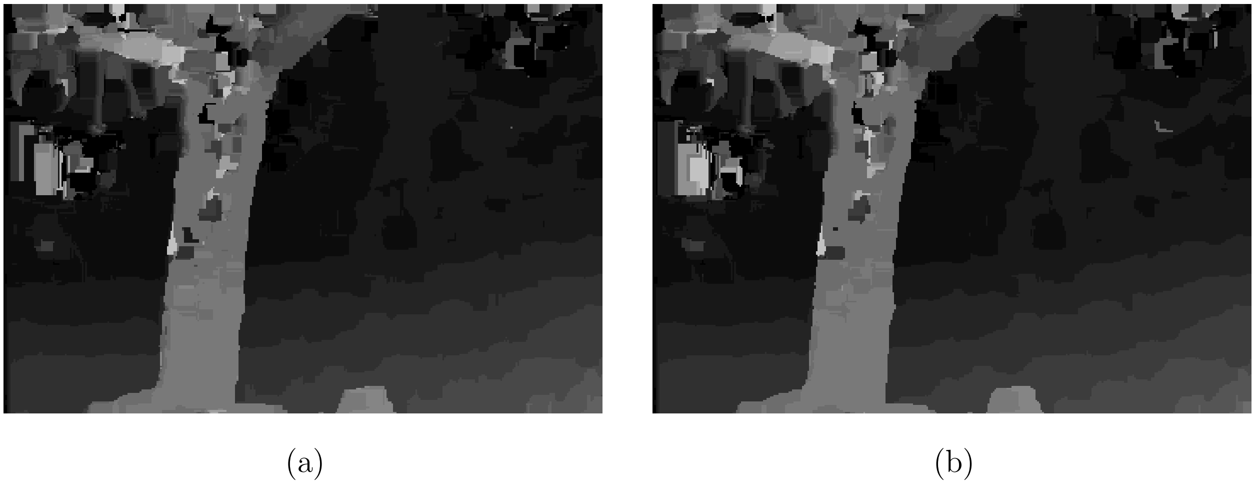


Figure 2.4: Estimated disparity for *Tsukuba* by block-based methods: (a) SSD; (b) SAD.



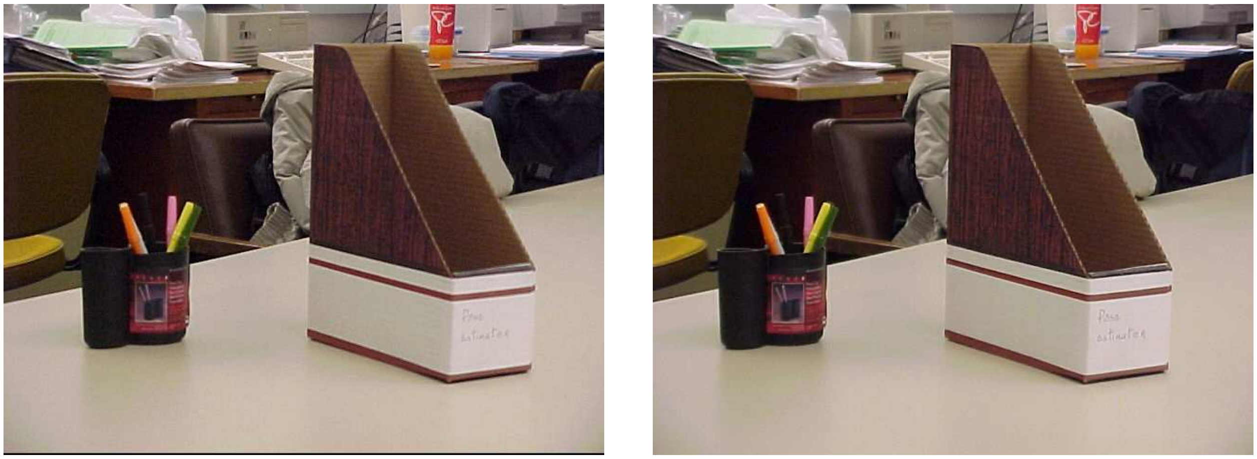
(a) (b)

Figure 2.5: Original images from *Flower Garden* (240×352): (a) left image; (b) right image.



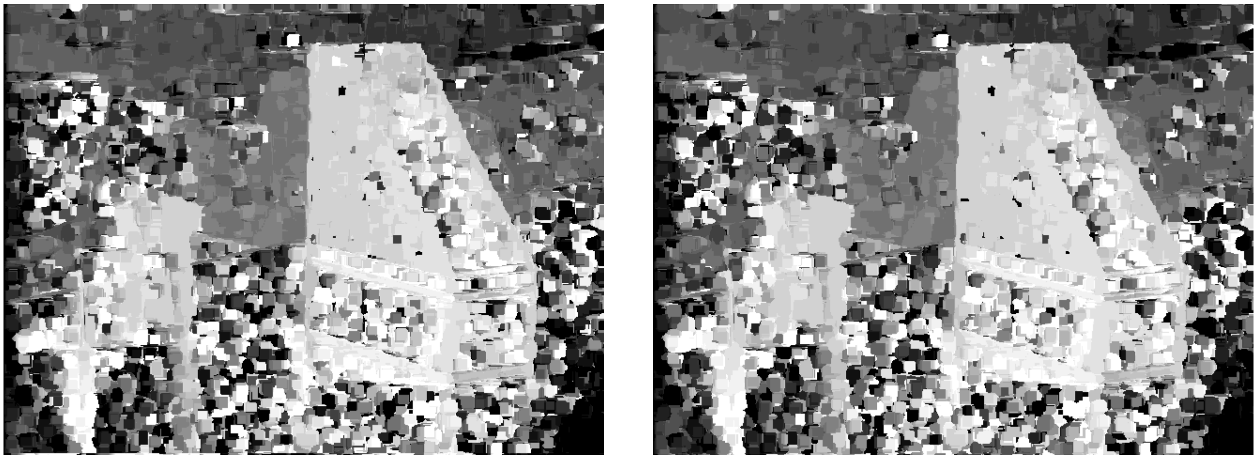
(a) (b)

Figure 2.6: Estimated disparity for *Flower Garden* by block-based methods: (a) SSD; (b) SAD.



(a) (b)

Figure 2.7: *VIVA Lab* (480×640): (a) left image; (b) right image.



(a) (b)

Figure 2.8: Estimated disparity for *VIVA Lab* by block-based methods: (a) SSD; (b) SAD.

by SSD and SAD usually distort the object boundaries (as for *Tsukuba*), and give blocking effect for untextured areas (as for *VIVA Lab*). We will use these three stereo pairs to test and compare the performance of different disparity estimation algorithms, including our own in Chapter 3.

### 2.1.3 Energy-Based Methods

The energy-based methods can be classified into two approaches: variational regularization with a partial differential equation (PDE) approach, and the approach using discrete optimization methods. They are all based on finding the minimization for the following energy functional:

$$E = \int\int [I_l(x,y) - I_r(x-d,y)]^2 \,\mathrm{d}x\,\mathrm{d}y \tag{2.8}$$

where $I_l$ and $I_r$ are left and right images respectively, and $d$ is the disparity value at location $(x,y)$ in the left image. Since disparity estimation is an ill-posed inverse problem in which one pixel in an image might have many matching pixels in another image, regularization is needed to control the smoothness of the disparity values. Therefore, there is usually a regularization term added to Eq. (2.8):

$$E = \int\int [I_l(x,y) - I_r(x-d,y)]^2 \,\mathrm{d}x\,\mathrm{d}y + \lambda E_R(d, I_l) \tag{2.9}$$

where $E_R(d, I_l)$ is a disparity- and image-related regularization term and $\lambda$ is the regularization coefficient. The variational regularization approach and the discrete optimization approach treat the minimization of Eq. (2.9) in a different way.

#### Variational Regularization

The variational regularization approach usually uses a regularization term involving the gradient values of the disparity field as well as the image values. This comes from the regularization functionals used in optical flow techniques in which the gradient values of the motion vectors are used. Some functionals also exploit image gradients to make the regularization more precise along edge pixels, and this feature is usually adopted for the regularization functionals in disparity estimation in order to preserve depth discontinuities.

In the following, we use $\nabla d = [\frac{\partial d}{\partial x}, \frac{\partial d}{\partial y}]^t$ and $\nabla I_l = [\frac{\partial I_l}{\partial x}, \frac{\partial I_l}{\partial y}]^t$ to represent the gradients of $d$ and $I_l$.

There are two main forms of the regularization functionals used in disparity estimation. The first one is [21][23]:

$$E_R(d, I_l) = \int\int \left[(\nabla d)^t D(\nabla I_l)\nabla d\right] \mathrm{d}x\, \mathrm{d}y \tag{2.10}$$

where $D(\nabla I_l)$ is a matrix defined by:

$$D(\nabla I_l) = \frac{1}{|\nabla I_l|^2 + 2\upsilon^2}\left\{\begin{bmatrix} \frac{\partial I_l}{\partial y} \\ -\frac{\partial I_l}{\partial x}\end{bmatrix}\begin{bmatrix} \frac{\partial I_l}{\partial y} \\ -\frac{\partial I_l}{\partial x}\end{bmatrix}^t + \upsilon^2 Id\right\} \tag{2.11}$$

where $Id$ is the identity matrix and $\upsilon$ is an arbitrary positive real number. The regularization term $E_R(d, I_l)$ is anisotropic: in homogeneous areas the disparities are smoothed in all directions since the values of $\nabla I_l$ are very small and thus the smoothing power of $E_R(d, I_l)$ is large, while in textured areas including edges the smoothing is mainly along the edge but not across it since the values of $\nabla I_l$ are large along the edges and thus the smoothing power of $E_R(d, I_l)$ is small. This functional was developed from the Nagel and Enkelmann functional [38], which was shown to be the best quadratic smoothness constraint for optical flow estimation [39].

The second regularization functional frequently used is [24][25]:

$$E_R(d, I_l) = \int\int \frac{1}{(1 + |\nabla I_l|^2)^2}|\nabla d|^2 \,\mathrm{d}x\, \mathrm{d}y. \tag{2.12}$$

Thus, in the variational regularization approach, the final goal is to minimize the overall energy functional of (2.9) with respect to the disparity $d$. The minimization process is carried out by first obtaining the associated Euler-Lagrange equation of (2.9); then assuming a pseudo-time variable $t$, we apply the gradient descent method to the Euler-Lagrange equation to obtain a converged disparity $d$. The corresponding Euler-Lagrange equation for (2.9) with regularization functional as (2.10) is:

$$\frac{\partial d}{\partial t} = [I_l(x, y) - I_r(x - d, y)] \times I_{r,x}(x - d, y) + \lambda \mathrm{div}(D(\nabla I_l)\nabla d) \tag{2.13}$$

and the Euler-Lagrange equation for (2.9) with regularization functional as (2.12) is:

$$\frac{\partial d}{\partial t} = [I_l(x,y) - I_r(x-d,y)] \times I_{r,x}(x-d,y) - \lambda \left\{ \frac{\partial}{\partial x}\left[\frac{d_x}{(1+I_{l,x}^2)^2}\right] + \frac{\partial}{\partial y}\left[\frac{d_y}{(1+I_{l,y}^2)^2}\right] \right\} \quad (2.14)$$

in which we modified (2.12) since from our experiments we found that using the regularization term reflected in (2.14) gives better results. We will leave the topic of the discretization scheme for the numerical solutions of (2.13) and (2.14) to next chapter in which we actually use the variational regularization approach for the disparity estimation.

In [21]–[24], the disparity estimated from block-based methods (correlation or SSD) is used as a coarse estimation for the initial values of $d$ in (2.13) and (2.14), and the iteration procedures for solving (2.13) and (2.14) act as refinement processes for the disparity field. Thus, the disadvantage of the block-based methods also affect the performance of the variational regularization approach, because the final solutions or converged values of (2.13) and (2.14) can easily fall into local minima if the initial values are not accurate enough.

### Discrete Optimization Approach

Several years ago, a new stereo algorithm called *graph cuts* was developed for disparity estimation [2][3]. This algorithm is based on discrete combinatorial optimization techniques. The idea of this approach is to construct a graph consisting the pixels of the image as well as the labels (disparity values) for the energy function to be minimized, and using an efficient combinatorial optimization algorithm like the max-flow algorithm so that the minimum cut applied on the graph also minimizes the energy. It also treats Eq. (2.9) with two terms

$$E(f) = E_{data}(f) + E_{smooth}(f) \quad (2.15)$$

where $f$ is a labeling which assigns a label $f_p$ to a pixel $p$. $E_{data}$ is a data penalty term similar to the first term in Eq. (2.9) which measures how well $f_p$ fits pixel $p$

$$E_{data}(f) = \sum_p D_p(f_p) \quad (2.16)$$

where $D_p$ could be a squared difference function as in the first term of Eq. (2.9). $E_{smooth}$ performs as a regularization term corresponding to the second term in Eq. (2.9), which measures the smoothness of the labels $f$. In [2], $E_{smooth}$ is defined as:

$$E_{smooth}(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) \tag{2.17}$$

where $\mathcal{N}$ is the set of neighboring pixels, and $V_{p,q}(f_p, f_q)$ can be defined as

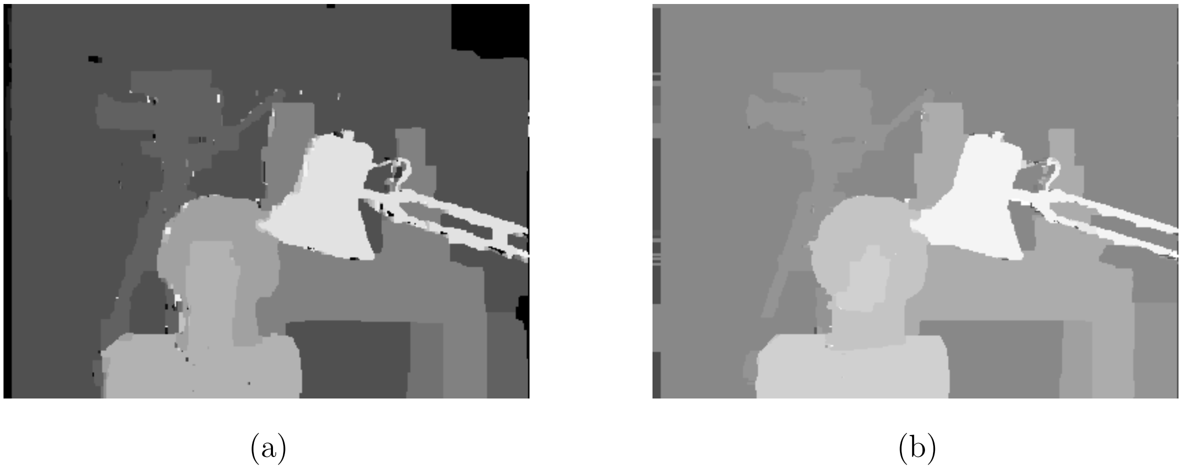
$$V_{p,q}(f_p, f_q) = \min(K, |f_p - f_q|^2) \tag{2.18}$$

or

$$V_{p,q}(f_p, f_q) = \min(K, |f_p - f_q|) \tag{2.19}$$

with $K$ a constant.

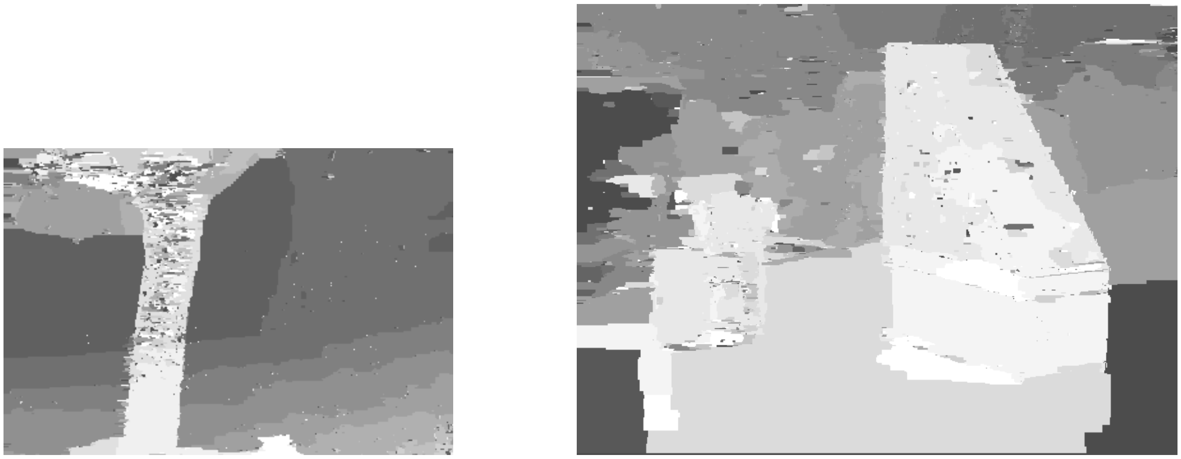
The approach of graph cuts can give, up to the present, the best disparity maps for some images in which the surfaces of most of the objects are fronto-parallel (surface that is parallel to the lens of the camera). The typical example is *Tsukuba*, as shown in Fig. 2.9. Although graph cuts can not give a global optimization, it can bring the solution to a strong local minimum.


(a)

(b)

Figure 2.9: Disparity for *Tsukuba* by graph cuts: (a) result from [2]; (b) result from [3].

However, one main disadvantage of graph cuts is the fact that it does not handle textured/untextured slanted surfaces well. This is due to the fact that the labels assigned to all the disparity values are discrete, and hence the disparity values are all integers, and also because the minimization processes of graph cuts does not take into account the derivatives of image densities as well as disparity values. As shown in Fig. 2.10, we can see that graph cuts gives poor performance especially for those slanted surfaces, in which it could not identify linear variation for untextured surfaces, or gives highly quantized disparity values for textured slanted surfaces. In addition, some fine features like the twigs in *Flower Garden* could not be clearly distinguished by graph cuts.



(a)

(b)

Figure 2.10: Disparity estimated by graph cuts [3]: (a) *Flower Garden*; (b) *VIVA Lab*.

### 2.1.4 Phase-Based Methods

Disparity can also be estimated by comparing the phase difference between the two images. Since disparity values vary over the whole image, the localized phase information is needed. Therefore, the windowed Fourier transform, or preferably, the Gabor transform, is usually employed.

In order to show how the disparity can be obtained from the Fourier phase information of the stereo images, first assume that the right image is a pure horizontal translation of the left image:

$$I_r(x, y) = I_l(x - d, y) \tag{2.20}$$

where $d$ is constant over the whole image. From the properties of the Fourier transform:

$$\hat{I}_r(\omega_1, \omega_2) = \hat{I}_l(\omega_1, \omega_2)e^{-i\omega_1 d}. \tag{2.21}$$

Therefore, we have:

$$\frac{\hat{I}_l(\omega_1,\omega_2)\hat{I}_r^*(\omega_1,\omega_2)}{|\hat{I}_l(\omega_1,\omega_2)||\hat{I}_r(\omega_1,\omega_2)|} = e^{i\omega_1 d} \tag{2.22}$$

Hence from the above normalized phase-correlation we can obtain the phase difference between $\hat{I}_l(\omega_1,\omega_2)$ and $\hat{I}_r(\omega_1,\omega_2)$, and the disparity can be obtained by taking the inverse Fourier transform of the correlation product, resulting in an impulse at the location $d$.

However, in practice, the disparity values vary over the whole image. Thus it is desirable to measure the phase difference locally rather than globally. In order to do this we need to use the windowed Fourier transform. The best choice is to use the Gabor function because the Gaussian window performs the localization in both the spatial and the frequency domains simultaneously. The Gabor functions are Gaussian functions modulated by complex sinusoidals. For the 2-D case, they are defined as follows:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\left(\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2}\right)\right] \times \exp\left[i\left(\omega_{10}x+\omega_{20}y\right)\right] \tag{2.23}$$

where $\omega_{10}$ and $\omega_{20}$ define the spatial frequencies in $x$ and $y$ directions respectively. Its Fourier transform has the form:

$$G(\omega_1,\omega_2) = e^{-\pi[\sigma_x^2(\omega_1-\omega_{10})^2+\sigma_y^2(\omega_2-\omega_{20})^2]}. \tag{2.24}$$

The Gabor functions are practically implemented by Gabor filters, which are actually discretized versions of (2.23). In the method of [26], assume the Gabor filter is tuned to a single frequency, and the outputs for the left and the right images are $G_l(x,y)$ and $G_r(x,y)$, with $\phi_l(x,y)$ and $\phi_r(x,y)$ representing their phase components. Let $\phi_{l,x}(x,y)$ and $\phi_{r,x}(x,y)$ represent the x-derivatives of $\phi_l(x,y)$ and $\phi_r(x,y)$, which is calculated as

$$\phi_{l,x}(x,y) = \frac{Im[G_l^*(x,y)G_{l,x}(x,y)]}{|G_l(x,y)|^2} \tag{2.25}$$

and similarly for $\phi_{r,x}(x,y)$. Then the estimated disparity $d(x,y)$ for location $(x,y)$ is determined by [26]:

$$d(x,y) = \frac{2[\phi_l(x,y)-\phi_r(x,y)]_{2\pi}}{\phi_{l,x}(x,y)+\phi_{r,x}(x,y)} \tag{2.26}$$

where $[\theta]_{2\pi}$ denotes phase-wrapping, i.e., $[\theta]_{2\pi} \in (-\pi,\pi]$.

In [27], a set of quadrature-pair Gabor filters is used. Each quadrature-pair Gabor filter is a set of discretized samples of a Gabor function tuned with different $\omega_{10}$ and $\omega_{20}$ to different directions. Assume that the outputs of the $k^{th}$ filter pair are $G_l^k(x,y)$ and $G_r^k(x,y)$ for the left and right images respectively. Instead of doing the phase-wrapping, a local weighted phase-correlation between the two images is calculated as

$$C_k(x,y,\tau) = \frac{W(x,y) \otimes [G_l^k(x,y) G_r^{k*}(x+\tau,y)]}{\sqrt{W(x,y) \otimes |G_l^k(x,y)|^2} \sqrt{W(x,y) \otimes |G_r^k(x+\tau,y)|^2}} \tag{2.27}$$

where $W(x,y)$ is a small and localized window, $\otimes$ represents correlation, and $\tau$ is a preshift of the right filter output. Then a summation is obtained over all the filters:

$$S(x,y,\tau) = \sum_k C_k(x,y,\tau) \tag{2.28}$$

and the disparity for location $(x,y)$ can be estimated by finding a peak in the real part of $S(x,y,\tau)$ and verified by a zero near $(x,y)$ in its imaginary part.

### 2.1.5 Parametric Model Methods

Parametric model-based disparity estimation differs from the above mentioned algorithms in that it tries to express the disparity map as a function, and the function itself is dependent on several parameters. Therefore, the problem of disparity estimation for each pixel is transformed to the estimation of parameters for the function that express the shape of the disparity map for a certain region [28][29].

In [29], a hierarchical scheme is used and the disparity map is expressed as

$$d(x,y) = \sum_{k=1}^{N} w_k \phi_k + w_0 \tag{2.29}$$

where $\phi_k$ is the $k$th Gaussian function:

$$\phi_k = e^{-\left[\frac{(x-t_{x,k})^2}{\sigma_{x,k}^2} + \frac{(y-t_{y,k})^2}{\sigma_{y,k}^2}\right]} \tag{2.30}$$

and $N$ is the number of Gaussians which is dependent on the hierarchical scheme; $w_k$ is the combination weight; $w_0$ is a constant shift. To save the computational cost, in [29], only the $w_k$ are treated as free parameters, and are estimated iteratively through a variational function containing the derivatives of an error function with the $w_k$.

In [28], a disparity surface is defined as

$$S(x,y) = w_0 + w_y \cdot y + w_x \cdot x + \sum_{i=1}^{N} w_i \cdot e^{-\frac{(x-\mu_i^x)^2+(y-\mu_i^y)^2}{\sigma_i^2}} \tag{2.31}$$

Then the parameters of Gaussian functions, i.e., the centers $(\mu_i^x, \mu_i^y)$, the spreads $\sigma_i$ are estimated along with the weighting parameters, by a constrained nonlinear optimization scheme – sequential quadratic programming (SQP) [40].

The main advantage of parametric model methods is that such algorithms can avoid the noise and outliers in the estimated disparity maps that usually happen in pixelwise-based methods.

## 2.2 Motion Estimation with Variational Regularization

Among the large amount of literature and algorithms for optical flow and motion estimation, differential techniques [41][42] with variational regularization form a major class. These techniques involve a functional including the displaced frame difference and a smoothing term, and usually descent-based methods are used to minimize the functional by solving its associated Euler-Lagrange equations. Since this class is closely related to the variational regularization approach for the disparity estimation, and is also an approach that we adopted in later chapters, so we give a short introduction to this class of motion estimation.

The algorithms of this class are all based on the assumption that the intensities of image objects in subsequent frames remain constant:

$$I(x+u, y+v, t+1) = I(x,y,t) \tag{2.32}$$

where the displacement field $(u,v)^T$ is called *optical flow* and is in pixel unit, and the frame interval is assumed to be 1. Under the condition of small displacements, a first order Taylor expansion can be applied to (2.32) yielding the well known *optical flow constraint*

$$I_x u + I_y v + I_t = 0 \tag{2.33}$$

where subscripts represent partial derivatives. To alleviate the noise and outliers in the estimated optical flow field, Horn and Schunck embedded a quadrature regularization functional into a global energy function

$$E_{HS}(u,v) = \iint [(I_x u + I_y v + I_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)] \, \mathrm{d}x \, \mathrm{d}y \tag{2.34}$$

and $(u,v)^T$ can be calculated in a recursive descent approach by solving the associated Euler-Lagrange equations of (2.34) with respect to $u$ and $v$.

(2.34) is an early functional with regularization term used for motion estimation, and there are extensions based on it which, except for the gradients of optical flow field, also make use of the gardient values of image intensities in the regularization term, similar to (2.10) and (2.12).

## 2.3 3D Model Integration

The algorithms involved in the problem of 3D model alignment, or 3D registration, can be divided into two major steps: ego-motion estimation, and integration of separate 3D models.

### 2.3.1 Ego-Motion Estimation

Ego-motion estimation, or global motion estimation, is the estimation of the camera motion in 3D space represented by six parameters – three rotational parameters and three translational parameters. The transformation that transfers the 3D coordinates of one camera location to another one is called *homogeneous transformation.* In order to accurately align 3D models estimated at different locations, we need to transfer these 3D models to one reference location, which can be achieved once we have accurate ego-motion parameters.

Currently, depending on the application area, there are two main approaches for ego-motion estimation: the *bundle adjustment* (usually for real image sequences) and the *iterative closest point* (ICP, usually for laser scanned 3D models).

### Bundle Adjustment

Bundle adjustment originated in the field of photogrammetry [43], and is widely used in the computer vision community for most of the feature-based multiview structure (3D positions of feature points) and ego-motion estimation (camera poses) algorithms. An overview of its applications in computer vision can be found in [44], and an implementation with C++ by exploiting its sparse structure can be found in [45].

According to [45], suppose there are $n$ 3D points in $m$ views, we represent the projection parameters of each camera $j$ by a vector $\mathbf{a}_j$, and the 3D coordinates of each 3D point $i$ by a vector $\mathbf{b}_i$. Assume the projection of the $i$-th point on image $j$ be $\mathbf{x}_{ij}$. Then the bundle adjustment tries to minimize the reprojection error with respect to all 3D points and camera parameters by

$$\min_{\mathbf{a}_j,\mathbf{b}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} ||\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i) - \mathbf{x}_{ij}||^2. \tag{2.35}$$

$\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i)$ is the predicted projection of point $i$ on image $j$, which implicitly includes camera pose parameters ($\mathbf{Q}(\mathbf{a}_j, \mathbf{b}_i) = A_j(\mathbf{R}|\mathbf{T})\mathbf{b}_i$ where $\mathbf{R}$ and $\mathbf{T}$ represent rotation matrix and translation vector, and $A_j$ represents the projection matrix of camera $j$), and $||\mathbf{x} - \mathbf{y}||$ represents the Euclidean distance between the image points $\mathbf{x}$ and $\mathbf{y}$. Then bundle adjustment minimizes (2.35) using a non-linear Levenberg-Marquardt optimization method to jointly estimate $\mathbf{a}_j$ which implicitly includes camera pose parameters, and $\mathbf{b}_i$.

### Iterative Closest Point (ICP)

Starting from the foundation paper by Besl and McKay [46], ICP has become the major algorithm for 3D model registration, especially for laser-scanned 3D models. Although there are many variants of ICP [47], it has two basic steps which should be carried out iteratively: (1) finding the closest point pairs by searching nearest neighbor in 3D space using kd-tree algorithm [48]; (2) calculating the best homogeneous transformation between all matched point pairs. During each iteration step, the parameters of the homogeneous transformation can be estimated by minimizing the following cost function

$$E = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{A}_i - \mathbf{R}\mathbf{B}_i - \mathbf{T}|^2 \tag{2.36}$$

where $\mathbf{A}_i$ and $\mathbf{B}_i$ are 3D coordinates of the two sets of given point pairs, and $N$ is the total number of points $\mathbf{A}_i$ as well as $\mathbf{B}_i$. The details on how to solve (2.36) to estimate the rotational matrix $\mathbf{R}$ and the translational vector $\mathbf{T}$ can be found in [49] and [46].

### Ego-Motion Estimation in Image Space

Both bundle adjustment and ICP algorithms perform 3D alignment in 3D space. On the other hand, there exist image intensity-based ego-motion estimation methods, such as [50]. In this approach, it is the image intensity, rather than the coordinates of 3D points like in (2.36), that is involved in the cost function which includes the parameters of a homogeneous transformation. In [50], it is assumed that the main motion of the camera is translational, with slight rotations around $x$ and $z$ axis. Then the two consecutive frames $I_{k-1}$ and $I_k$ have the following motion model:

$$\mathbf{P} + \mathbf{T_x} = \mathbf{R}^{-1}\mathbf{P}' \tag{2.37}$$

where $\mathbf{P}$ and $\mathbf{P}'$ are corresponding 3D points in the coordinate systems of $I_{k-1}$ and $I_k$, and $\mathbf{T_x} = [T_x, 0, 0]^t$ represents the horizontal translation. The matrix $\mathbf{R}$ represents the rotation from $I_{k-1}$ to $I_k$. Based on the assumption of small rotations, as well as the constant depth for each pixel due to the translational motion, the image coordinates of $I_{k-1}$ and $I_k$ have the following relations:

$$\begin{aligned} &x_{k-1} - T_x D_{k-1}(x_{k-1}, y_{k-1}) = \cos(\alpha)x_k - \sin(\alpha)y_k \\ &y_{k-1} - b = \sin(\alpha)x_k + \cos(\alpha)y_k \end{aligned} \tag{2.38}$$

where $D_{k-1}(x_{k-1}, y_{k-1})$ is the normalized disparity for $(x_{k-1}, y_{k-1})$ (disparity of $(x_{k-1}, y_{k-1})$ over $T_x$), $\alpha$ denotes the rotation about $z$ axis, and the rotation around the $x$ axis is approximated by a uniform vertical translation $b$. Also, with small rotation about the $z$ axis, we have $\cos(\alpha) \approx 1$ and $\sin(\alpha) \approx \alpha$, and these give the following cost function to be minimized:

$$E(T_x, b, \alpha) = \sum_{x_{k-1}, y_{k-1}} [I_{k-1}(x_{k-1} + T_x D_{k-1}(x_{k-1}, y_{k-1}), y_{k-1}) - I_k(x_k - \alpha y_k, y_k + b + \alpha x_k)]^2. \tag{2.39}$$

Using the first order Taylor expansion, we have following approximations:

$$\begin{aligned} I_{k-1}(x_{k-1} + T_x D_{k-1}(x_{k-1}, y_{k-1}), y_{k-1}) &\approx I_{k-1}(x_{k-1}, y_{k-1}) + \frac{\partial I_{k-1}}{\partial x_{k-1}} T_x D_{k-1}(x_{k-1}, y_{k-1}) \\ I_k(x_k - \alpha y_k, y_k + b + \alpha x_k) &\approx I_k(x_k, y_k) - \frac{\partial I_k}{\partial x_k}(\alpha y_k) + \frac{\partial I_k}{\partial y_k}(b + \alpha x_k) \end{aligned} \tag{2.40}$$

and we obtain the final cost function by substituting Eq. (2.40) into Eq. (2.39):

$$\begin{aligned} E(T_x, b, \alpha) = \sum_{x_{k-1}, y_{k-1}} & [I_{k-1}(x_{k-1}, y_{k-1}) - I_k(x_k, y_k) + \frac{\partial I_{k-1}}{\partial x_{k-1}} T_x D_{k-1}(x_{k-1}, y_{k-1}) \\ & + \frac{\partial I_k}{\partial x_k}(\alpha y_k) - \frac{\partial I_k}{\partial y_k}(b + \alpha x_k)]^2. \end{aligned} \tag{2.41}$$

The minimization of Eq. (2.41) can be achieved by taking the derivatives of the cost function with respect to the three ego-motion parameters and equalize them to zero, giving a set of three linear equations with three variables. Therefore, the translation as well as the rotation matrix of homogeneous transformation can be estimated by such direct method based on image intensities.

### 2.3.2 Integration of Separate 3D Models

Once the ego-motion parameters are obtained, we have the relative poses of cameras at different locations. Thus the separate 3D models at different locations can be transfered to the reference location by homogeneous transform. Then, as a final stage, the remaining problem is how to integrate them into one whole 3D model.

The existing algorithms for 3D model integration can be classified into two categories [51]: the volumetric approach, and the surface approach.

The volumetric approach, e.g. [52] and [53], partitions the 3D space into voxel grids, and puts 3D points from all models into such grids. Then, the surface of the model can be generated by triangulation using an Iso-Surface algorithm – marching cubes [54].

The surface approach, such as [55] and [56], generates initial triangular meshes in separate 3D models, and then these triangular meshes are stitched together. The stitching processes usually starts with the detection of the overlapping and non-overlapping mesh parts from different 3D models, and then determines which models' contribution to the overlapping region need to be kept, while the others to be discarded. Finally the gaps between the meshes of the overlapping and non-overlapping parts are filled by applying triangulation on the vertices of existing triangles along the gaps.

According to [51], surface approach is limited to processing 3D data in image format, and is superior to the volumetric approach if the purpose is to generate the most accurate triangular meshes possible relative to the original data. This is because the surface approach triangulates the data at the original resolution. However, the surface approach usually needs more memory space than that of the volumetric approach since it needs to keep all the original data points in memory, while for the volumetric approach once a point has been processed it can be discarded.

## 2.4 Summary

For all the disparity estimation algorithms listed in this chapter, one limitation is that most of them are not robust. There are two aspects to this issue:

- For block-based methods, the block size needs to be determined according to the complexity of the scene. Since most of the variational regularization methods use the disparity results from block-based methods as initial values, this also makes the algorithms involving variational regularization not robust. Although there are adaptive algorithms for the selection of block size, they are very time-consuming.

- The algorithms based on graph cuts method, which are popular approaches now, only favor objects with fronto-parallel surfaces. For scenes with slanted surfaces, the performance of such algorithms decreases dramatically.

In order to address the robustness issue for disparity estimation, we use the following two approaches:

- Use Gabor transform to estimate a coarse disparity map as initial disparity values for variational regularization, instead of block-based ones, since the methods that we use based on Gabor transform do not need to determine any block size in advance.
- Exploit image segmentation and perform region-matching for the further improvement of disparity maps.

For 3D model integration, we use the image intensity-based method for the ego-motion estimation, as well as the surface approach for the final 3D model integration, since such a procedure will give us a more direct and simple, less ambiguous method which is suitable for image-based 3D modeling with results close to the original image data, while the procedure with ICP and volumetric approach might be more suitable for laser scanned 3D models.

We will give the detailed discription and results for our disparity and motion estimation algorithms in Chapter 3 and Chapter 4. In Chapter 5, the detailed procedure that we used for integrating 3D models estimated at different camera locations will be presented with results measured both visually and quantitively, followed by a conclusion in Chapter 6.

# Chapter 3

# Region-Based Disparity Estimation

In this chapter, we present our new developments on the issue of disparity estimation exploiting segmentation techniques and region matching. First we would like to unify the use of the terms “disparity” and “motion”. Both disparity and motion estimation concern finding the pixel displacements between different images. The main difference is that the scenes in images for disparity estimation are usually static (only the camera is moving) and thus the displacements involved in these images can be seen as rigid motion, while for motion estimation both objects in the scenes and the camera could be moving. Therefore, disparity estimation can be seen as a special case of motion estimation. Usually disparity estimation is applied to a pair of stereo images and the pixel displacements involved can be as high as dozens of pixels, while motion estimation can be applied to any two or more consecutive image frames with (in many cases) comparatively smaller pixel displacements (several pixels). In this thesis, each sequence we use has more than two images and the scenes inside these images are all static. Therefore we will use the term “disparity” when we apply our algorithms to two images, and use “multiview disparity” when we apply our algorithms to more than two images in the sequences.

Our algorithm can be seen as a combined approach using both pixel-based and region-based matching techniques. For the pixel-based method, we use the Gabor transform and variational energy method, in which the disparity results from the Gabor transform are used as coarse disparities to serve as initial values for the partial differential equations (PDE) from the variational energy functional, so that these coarse disparities can be further refined by variational regularization in an iterative process. For the region-based method, a color-based segmentation algorithm is used and a region-matching process is applied to each region. The final disparity estimation values are joint results by analyzing the results from both pixel-based matching and region matching to each region in order to to obtain a better solution.

## 3.1 Disparity Estimation

As shown in section 2.1.2, the frequently used disparity estimation methods SSD and SAD have some disadvantages like the distortion of object boundaries and losing tiny features. Although we can alleviate these problems to some extent by adjusting the block size of SSD and SAD. This makes SSD and SAD not robust. In order to have a robust disparity estimation method independent of scene complexities, we developed a disparity estimation algorithm based on the Gabor transform for which there is no need to select a block size in advance.

### 3.1.1 Disparity Estimation Using the Gabor Transform

As stated in section 2.1.4, due to the fact that the disparity values vary over the whole image, it is desirable to perform disparity estimation by making use of the localization properties of the Gabor transform in both the spatial and the frequency domain. The localization in the spatial domain limits the regions taken into account by the Gabor transform to a small neighborhood around that particular pixel location, while in the frequency domain it can bring band-pass filtered information at different frequencies for detailed analysis. We show these ideas in Fig. 3.1 in which the real and imaginary parts of a Gabor function with $\omega_{10} = \omega_{20} = 0.5\pi$ are shown as well as its Fourier transform. We can observe the localizations in both spatial and frequency domains, and the function is tuned to 45° with respect to the $x$-axis since $\omega_1 = \omega_2$.

We use a set of discretized version of (2.23), which is shown in Chapter 2, as Gabor filters

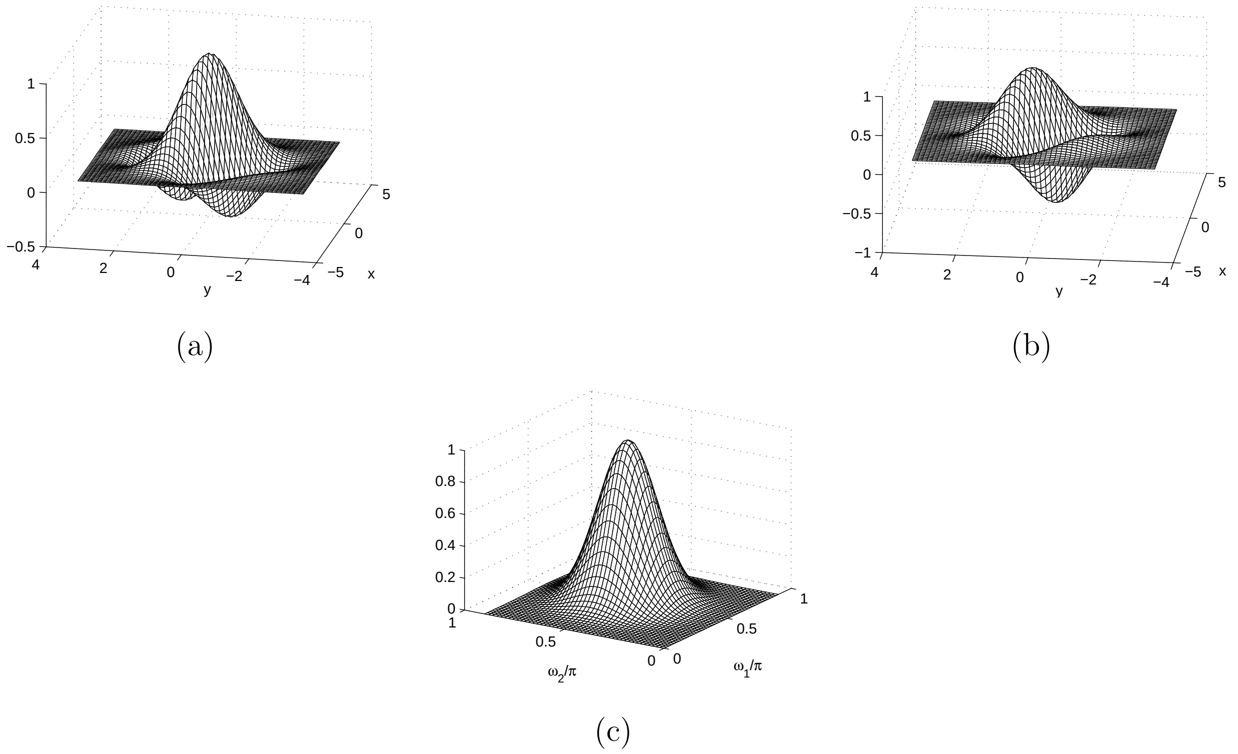


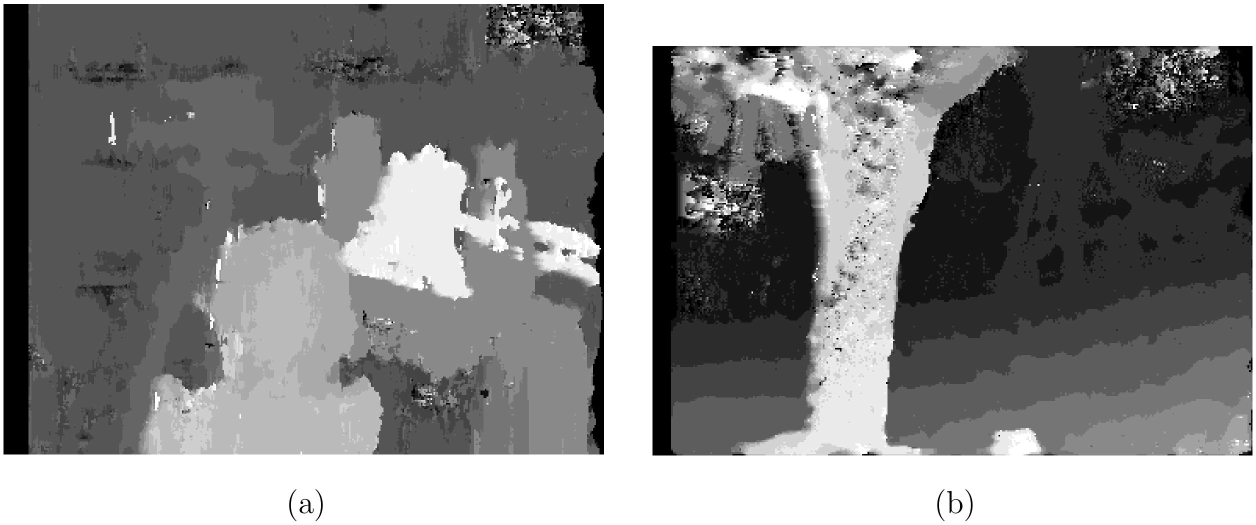
Figure 3.1: Gabor function with $\omega_{10} = \omega_{20} = 0.5\pi$: (a) real part; (b) imaginary part; (c) Fourier transform.

with different frequencies tuned to different directions, to implement the Gabor transform on the stereo images. The method that we use to estimate the disparity differs from the methods of Fleet et al. in that we avoid the phase-wrapping, and the uncertainty from phase correlation (2.28) in which a peak in its real part and a zero in its imaginary part need to be identified to jointly determine a disparity value. We propose a new and simple method to process those Gabor coefficients [20] in which the disparity $\hat{d} \in [0, d_{max}]$ for a position $(x, y)$ in the left image is determined as:

$$\begin{aligned}\hat{d}(x,y) = \arg\min_{d} \sum_{k} \big[ & |Re\{G_l^k(x,y)\} - Re\{G_r^k(x-d,y)\}|^2 \\ & + |Im\{G_l^k(x,y)\} - Im\{G_r^k(x-d,y)\}|^2 \big]\end{aligned} \tag{3.1}$$

where $Re\{G_l^k(x,y)\}$ and $Im\{G_l^k(x,y)\}$ are real and imaginary parts of $G_l^k(x,y)$, and similarly for $G_r^k(x,y)$. (3.1) is like performing SSD using Gabor coefficients, since each Gabor filter has a certain length (like a window with a certain width) and therefore each coefficient from Gabor filtering at a specific location $(x, y)$ is the result of convolving this Gabor window with the pixels at $(x, y)$ and its neighboring pixels under this Gabor window. Thus we can do this pixel-by-pixel rather than using a block with pre-determined size.

We use three values $\{\pi/16, \pi/8, \pi/4\}$ for the central frequency $\omega_{\mathbf{0}} = \sqrt{\omega_{01}^2 + \omega_{02}^2}$ of these Gabor filters. For each frequency, there are four filter pairs tuned to orientations 0°, 45°, 90° and 135° respectively. To test this algorithm under different scene conditions, we applied these filters on the three sets of stereo image pairs used in Chapter 2: *Tsukuba*, *Flower Garden* and *VIVA Lab*. The results are shown in Fig. 3.2 and Fig. 3.3, from which we can see that from a coarse point of view the disparities estimated by (3.1) are good except for *VIVA Lab*. In this case, the estimated disparity is noisy for the untextured and slanted surfaces. Also, the method that we used here is robust in the sense that (3.1) is a pixel-based approach for images with different kinds of scenes and texture levels, i.e., the disparities are estimated pixel-by-pixel independently of the characteristics of images, rather than predetermining a block size like in the SSD and SAD. However, there are still some obvious errors in the disparity maps, like some noisy values or outliers in homogeneous areas, or some distorted edges, as can be seen from Fig. 3.2. To alleviate such errors we need to ensure that


(a)

(b)

Figure 3.2: Coarse disparities estimated by (3.1): (a) *Tsukuba*; (b)*Flower Garden.*

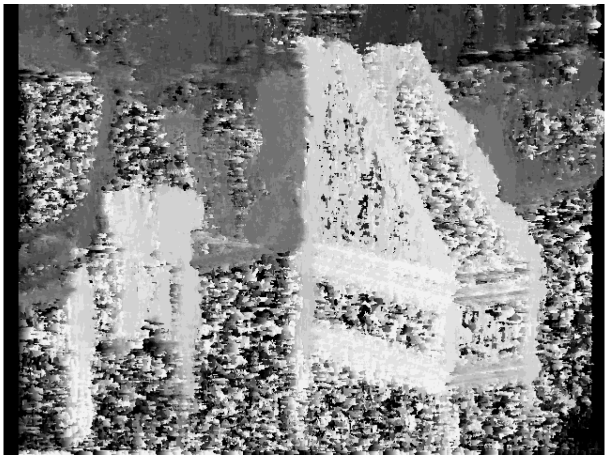


Figure 3.3: Coarse disparities estimated by (3.1) for *VIVA Lab*

the disparity values for continuous surfaces are changing smoothly, while maintaining the disparity discontinuities at the object boundaries. To achieve such properties, we chose to use an energy-based variational regularization approach.

Before we go further to the next section, we show some performance results for the disparities estimated by SSD, SAD, Gabor transform, and graph cut (GC) methods. The method that we used to obtain the performance is to interpolate the right image $I_r$ using the left image $I_l$ and the disparity map, and then measure the peak-signal-to-noise-ratio (PSNR) value between the interpolated $I_r$ and the original $I_r$. The interpolated $I_r$ for the three image sets based on SSD, SAD, Gabor transform, and graph cut (GC) are shown in Fig. 3.4 – 3.6.

The PSNR values for these interpolated right images $I_r$ are shown in Table 3.1. From

Table 3.1: PSNR values for interpolated $I_r$

| | *Tsukuba* | *Flower Garden* | *VIVA Lab* |
|---|---|---|---|
| SSD | 23.96 | 15.29 | 17.94 |
| SAD | 24.19 | 15.29 | 17.90 |
| Gabor transform | 20.87 | 15.25 | 18.02 |
| graph cut (GC) | 20.84 | 14.58 | 18.65 |

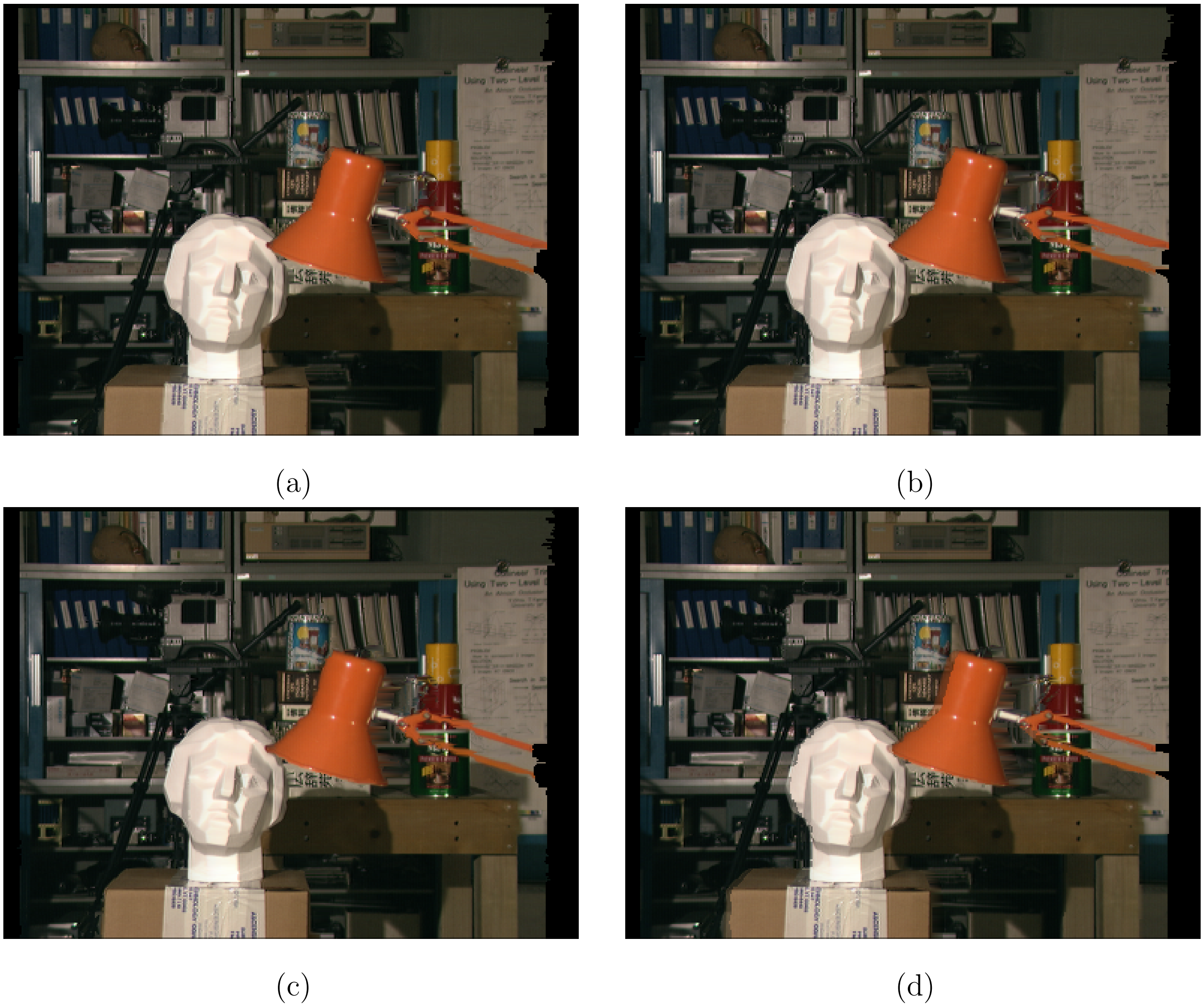
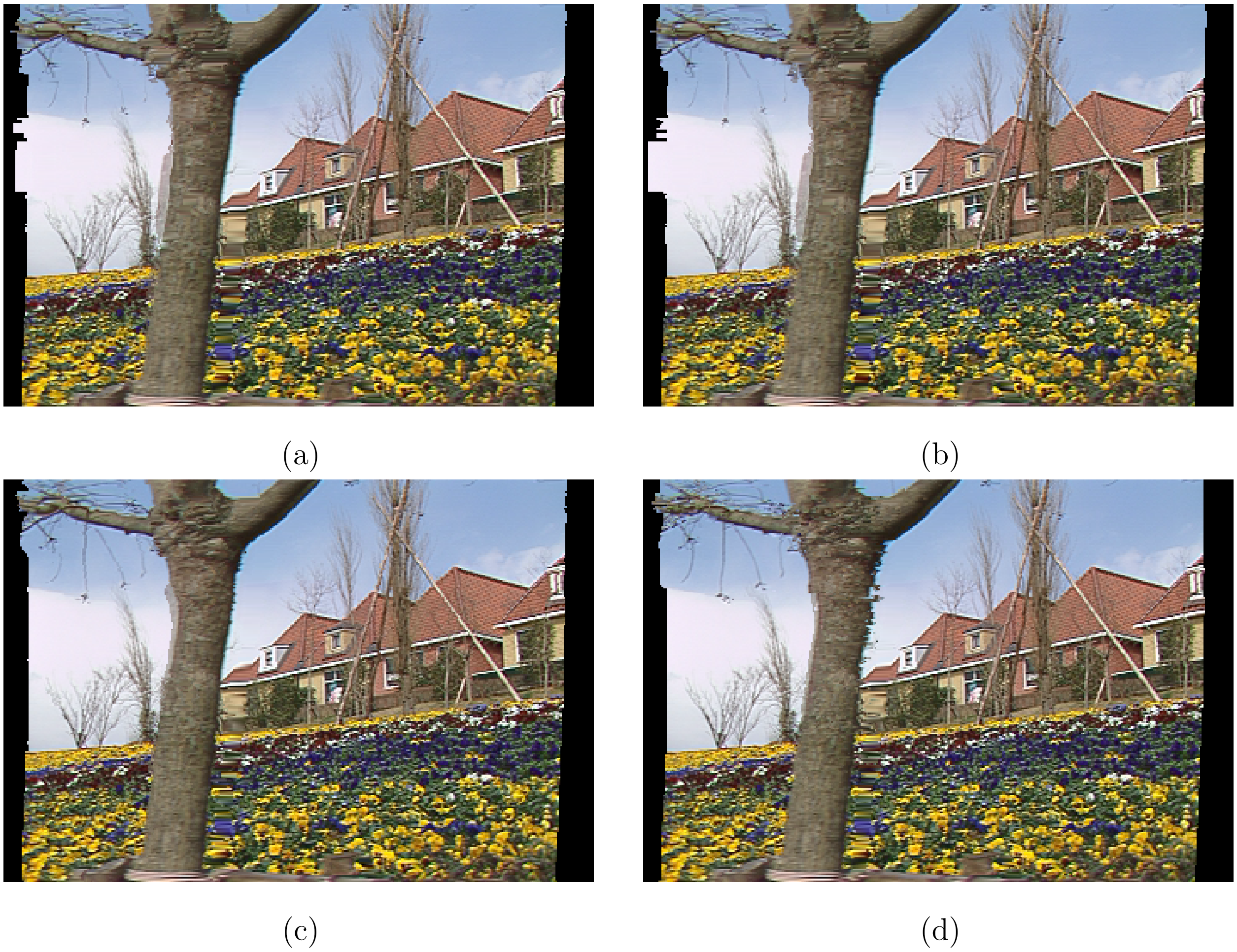


Figure 3.4: Interpolated $I_r$ for *Tsukuba*: (a) SSD; (b) SAD; (c) Gabor transform; (d) GC.

thesis PSNR values, we can see that PSNR obtained by interpolating $I_r$ does not really reflect the performance of a disparity estimation algorithm. Although graph cut gives the best visual quality for *Tsukuba*, its PSNR value is the lowest. We believe the performance of a disparity estimation algorithm should be measured by the visual quality of novel views rendered once the final 3D model is set up, and the views rendered at locations of existing images can be used to calculated PSNR values as quantitative performance.

With regard to implementation complexity, the running time of SSD, SAD and graph cut for those three image sets are shown in Table 3.2. Since we use MATLAB to implement our



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

Figure 3.5: Interpolated $I_r$ for *Flower Garden*: (a) SSD; (b) SAD; (c) Gabor transform; (d) GC.

algorithm for disparity estimation based on Gabor transform, the running time ranges from

Table 3.2: Running time (in seconds)

| | *Tsukuba* | *Flower Garden* | *VIVA Lab* |
|---|---|---|---|
| SSD | 2.3 | 1.8 | 6 |
| SAD | 2.2 | 1.7 | 6 |
| graph cut (GC) | 21.4 | 21.3 | 186 |

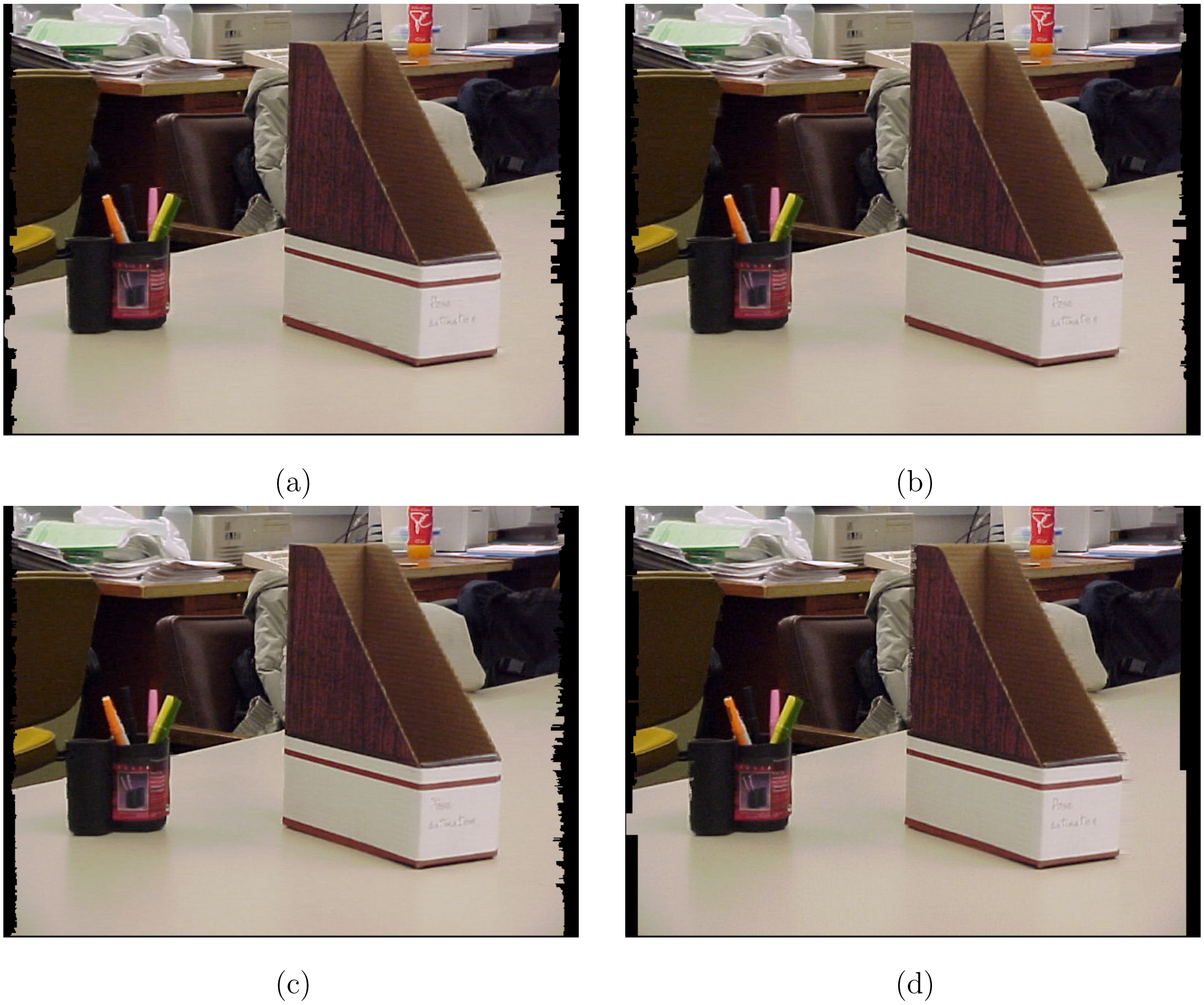

(a)
(b)
(c)
(d)

Figure 3.6: Interpolated $I_r$ for *VIVA Lab*: (a) SSD; (b) SAD; (c) Gabor transform; (d) GC.

1 to 3 minutes. However, from the nature of the Matlab codes, they should be completed in seconds if implemented in C, just a little more than SSD and SAD.

Therefore, based on the PSNR values and complexity, we select SSD and Gabor transform as starting point for our disparity estimation algorithm. In the remainder of this chapter as well as the next chapter, we show the evolution process of how we reached our final disparity estimation algorithm. In the remainder of this chapter, we only use the image sets *Tsukuba* and *Flower Garden*, which summarize our work in [20][57][58]. Once we reached our final algorithm for disparity estimation in the next chapter, we then apply it to all three image sets.

### 3.1.2 Refinement Using Variational Regularization

Variational regularization has had extensive application in optical flow estimation, as well as for disparity estimation [22][21]. The idea in [21] is to control the smoothing of disparity variations using the values of image gradients. When the value of image gradient is low the disparity would get smoothed, and the smoothing process is stopped when the value of image gradient is high, which represents a possible object boundary. As shown in Chapter 2, which we repeat here again for clarity, the Euler-Lagrange equations for the overall refinement functionals are

$$\frac{\partial d}{\partial t} = [I_l(x,y) - I_r(x-d,y)] \times I_{r,x}(x-d,y) + \lambda \mathrm{div}(D(\nabla I_l)\nabla d) \tag{3.2}$$

and

$$\frac{\partial d}{\partial t} = [I_l(x,y) - I_r(x-d,y)] \times I_{r,x}(x-d,y) - \lambda \left\{ \frac{\partial}{\partial x}\left[\frac{d_x}{(1+I_{l,x}^2)^2}\right] + \frac{\partial}{\partial y}\left[\frac{d_y}{(1+I_{l,y}^2)^2}\right] \right\} \tag{3.3}$$

for the two regularization functionals (2.10) and (2.12) respectively.

The numerical implementation of (3.2) and (3.3) is given by the forward Euler method, and the spatial derivatives are calculated by the central difference scheme. Let us further represent (2.11) by

$$D(\nabla I_l) = \frac{1}{|\nabla I_l|^2 + 2\upsilon^2} \left\{ \begin{bmatrix} \frac{\partial I_l}{\partial y} \\ -\frac{\partial I_l}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial I_l}{\partial y} \\ -\frac{\partial I_l}{\partial x} \end{bmatrix}^t + \upsilon^2 Id \right\} = \begin{bmatrix} g & f \\ f & e \end{bmatrix} \tag{3.4}$$

and use the subindex $(i,j)$ to represent the discretized coordinates of $(x,y)$:

$$\mathbf{x}_{i,j} = (x_i, y_j) \tag{3.5}$$

with, e.g.,

$$g_{i,j} = g(x_i, y_j). \tag{3.6}$$

Then, based on [21], the discretization of (3.2) can be given by

$$\begin{aligned}\frac{d_{i,j}^{k+1}-d_{i,j}^{k}}{\triangle t} =&[I_l(x_i,y_j)-I_r(x_i-d_{i,j}^k,y_j)]\times I_{r,x}(x_i-d_{i,j}^k,y_j)\\ &-\lambda\Bigg[\frac{g_{i+1,j}+g_{i,j}}{2}\times\frac{d_{i+1,j}^k-d_{i,j}^k}{h_1^2}+\frac{g_{i-1,j}+g_{i,j}}{2}\times\frac{d_{i-1,j}^k-d_{i,j}^k}{h_1^2}\\ &+\frac{f_{i,j+1}+f_{i,j}}{2}\times\frac{d_{i,j+1}^k-d_{i,j}^k}{h_2^2}+\frac{f_{i,j-1}+f_{i,j}}{2}\times\frac{d_{i,j-1}^k-d_{i,j}^k}{h_2^2}\\ &+\frac{e_{i+1,j+1}+e_{i,j}}{2}\times\frac{d_{i+1,j+1}^k-d_{i,j}^k}{2h_1h_2}+\frac{e_{i-1,j-1}+e_{i,j}}{2}\times\frac{d_{i-1,j-1}^k-d_{i,j}^k}{2h_1h_2}\\ &-\frac{e_{i+1,j-1}+e_{i,j}}{2}\times\frac{d_{i+1,j-1}^k-d_{i,j}^k}{2h_1h_2}-\frac{e_{i-1,j+1}+e_{i,j}}{2}\times\frac{d_{i-1,j+1}^k-d_{i,j}^k}{2h_1h_2}\Bigg],\end{aligned}\tag{3.7}$$

where $k$ is the iteration number, $\triangle t$ is the pseudo-time step, and $h_1$ and $h_2$ are pixel sizes for horizontal and vertical directions. After each iteration, the new $d_{i,j}$ for the discretized position $(x_i,y_j)$ is updated as:

$$\begin{aligned}d_{i,j}^{k+1} =&d_{i,j}^k+\triangle t\Bigg\{[I_l(x_i,y_j)-I_r(x_i-d_{i,j}^k,y_j)]\times I_{r,x}(x_i-d_{i,j}^k,y_j)\\ &-\lambda\Bigg[\frac{g_{i+1,j}+g_{i,j}}{2}\times\frac{d_{i+1,j}^k-d_{i,j}^k}{h_1^2}+\frac{g_{i-1,j}+g_{i,j}}{2}\times\frac{d_{i-1,j}^k-d_{i,j}^k}{h_1^2}\\ &+\frac{f_{i,j+1}+f_{i,j}}{2}\times\frac{d_{i,j+1}^k-d_{i,j}^k}{h_2^2}+\frac{f_{i,j-1}+f_{i,j}}{2}\times\frac{d_{i,j-1}^k-d_{i,j}^k}{h_2^2}\\ &+\frac{e_{i+1,j+1}+e_{i,j}}{2}\times\frac{d_{i+1,j+1}^k-d_{i,j}^k}{2h_1h_2}+\frac{e_{i-1,j-1}+e_{i,j}}{2}\times\frac{d_{i-1,j-1}^k-d_{i,j}^k}{2h_1h_2}\\ &-\frac{e_{i+1,j-1}+e_{i,j}}{2}\times\frac{d_{i+1,j-1}^k-d_{i,j}^k}{2h_1h_2}-\frac{e_{i-1,j+1}+e_{i,j}}{2}\times\frac{d_{i-1,j+1}^k-d_{i,j}^k}{2h_1h_2}\Bigg]\Bigg\}.\end{aligned}\tag{3.8}$$

For the implementation of (3.3), we use $\rho(x,y)$ and $\theta(x,y)$ for the following representations:

$$\begin{aligned}\rho(x,y)&=\left[\frac{1}{(1+I_{l,x}^2)^2}\right]\\ \theta(x,y)&=\left[\frac{1}{(1+I_{l,y}^2)^2}\right]\end{aligned}\tag{3.9}$$

Then, the discretized version of (3.3) is

$$\begin{aligned}\frac{d_{i,j}^{k+1}-d_{i,j}^{k}}{\triangle t} =& [I_l(x_i,y_j)-I_r(x_i-d_{i,j}^k,y_j)]\times I_{r,x}(x_i-d_{i,j}^k,y_j)\\ &-\lambda\Bigg[\frac{\rho_{i+1,j}+\rho_{i,j}}{2}\times\frac{d_{i+1,j}^k-d_{i,j}^k}{h_1^2}+\frac{\rho_{i-1,j}+\rho_{i,j}}{2}\times\frac{d_{i-1,j}^k-d_{i,j}^k}{h_1^2}\\ &+\frac{\theta_{i,j+1}+\theta_{i,j}}{2}\times\frac{d_{i,j+1}^k-d_{i,j}^k}{h_2^2}+\frac{\theta_{i,j-1}+\theta_{i,j}}{2}\times\frac{d_{i,j-1}^k-d_{i,j}^k}{h_2^2}\Bigg].\end{aligned} \tag{3.10}$$

Thus, after each iteration, the new $d_{i,j}$ for the discretized position $(x_i, y_j)$ is updated as

$$\begin{aligned}d_{i,j}^{k+1} =& d_{i,j}^k+\triangle t\Bigg\{[I_l(x_i,y_j)-I_r(x_i-d_{i,j}^k,y_j)]\times I_{r,x}(x_i-d_{i,j}^k,y_j)\\ &-\lambda\Bigg[\frac{\rho_{i+1,j}+\rho_{i,j}}{2}\times\frac{d_{i+1,j}^k-d_{i,j}^k}{h_1^2}+\frac{\rho_{i-1,j}+\rho_{i,j}}{2}\times\frac{d_{i-1,j}^k-d_{i,j}^k}{h_1^2}\\ &+\frac{\theta_{i,j+1}+\theta_{i,j}}{2}\times\frac{d_{i,j+1}^k-d_{i,j}^k}{h_2^2}+\frac{\theta_{i,j-1}+\theta_{i,j}}{2}\times\frac{d_{i,j-1}^k-d_{i,j}^k}{h_2^2}\Bigg]\Bigg\}.\end{aligned} \tag{3.11}$$

In our simulation, we used $h_1 = h_2 = 1$, $\triangle t = 0.01$ and $k = 800$ which were determined empirically to ensure good convergence. The initial values of $d_{i,j}^0$ for all pixel positions $(i, j)$ are obtained from the coarse disparities estimated using the Gabor transform (3.1), which are shown in Fig. 3.2. The refinement results are shown in Fig. 3.7 and Fig. 3.8 for (3.8) and (3.11) respectively.

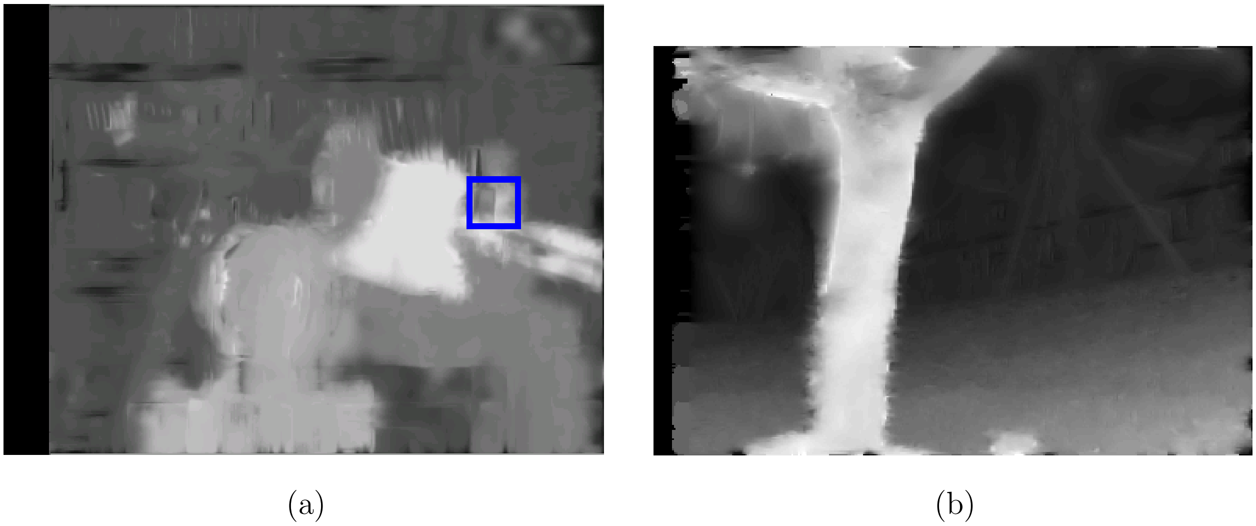
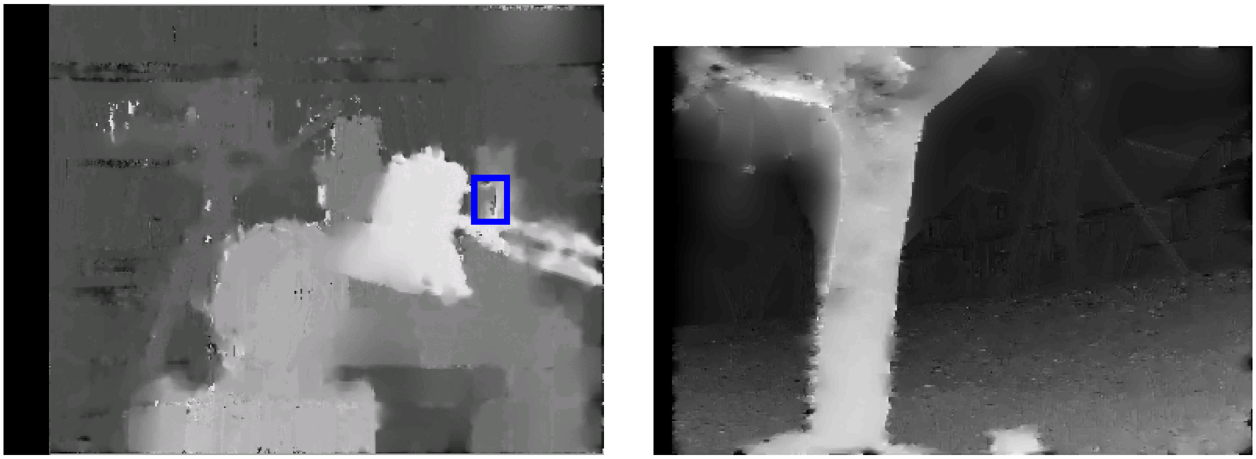


Figure 3.7: Refinement by (3.8): (a) *Tsukuba*; (b) *Flower Garden*.



(a) (b)

Figure 3.8: Refinement by (3.11): (a) *Tsukuba*; (b) *Flower Garden.*

From these results, we can see that most of the noisy outliers can be removed from the coarse disparities. However, as is the case for refinement using variational regularization, the contours and object boundaries get more or less blurred, especially for tiny features (like the handle of the lamp in *Tsukuba* as annotated in these two figures).

### 3.1.3 Variational Refinement Taking into Account Edge Information

As can be seen from the last section, variational regularization can smooth a coarse disparity and eliminate some outliers in untextured areas, but at the cost of blurring the object contours, even though some edge-preserving functionals are used. This is because, especially for real images, the difference of intensity variations between the edge areas and non-edge areas are not very big, since the intensity variations in some untextured areas are not zero (an ideal value which only can be reached for many synthetic images). In order to solve this problem, we propose to consider edge pixels separately. Specifically, in the variational refinement stage, we allow fewer iterations on the edge pixels while more iterations are used on non-edge pixels, to suppress the smoothing of edge pixels in the refinement stage and thus to keep the object contours crisp.

To do this, we multiply the left image $I_l$ by Sobel masks which are the flipped version of the impulse response of Sobel filters to obtain the Sobel coefficients for each pixel. The Sobel masks are defined as:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.12}$$

for $x$ and $y$ directions respectively. Assume that the Sobel filter outputs are $S_{l\_x}(x, y)$ in $x$ direction and $S_{l\_y}(x, y)$ in $y$ direction for $I_l$. Then the Sobel coefficients for $I_l$ are obtained by taking the sum of the absolute values of $S_{l\_x}(x, y)$ and $S_{l\_y}(x, y)$:

$$S_l(x, y) = |S_{l\_x}(x, y)| + |S_{l\_y}(x, y)| \tag{3.13}$$

and a pixel at $(x, y)$ is determined to be an edge pixel if $S_l(x, y)$ is larger than a threshold value (we use 0.8 which is determined empirically). The detected edge pixels are shown in Fig. 3.9.

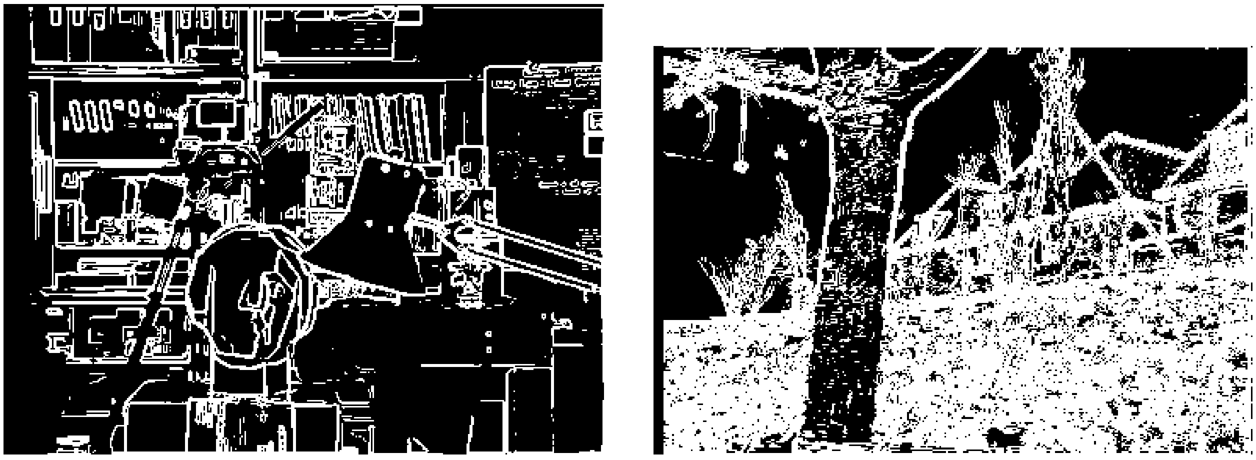
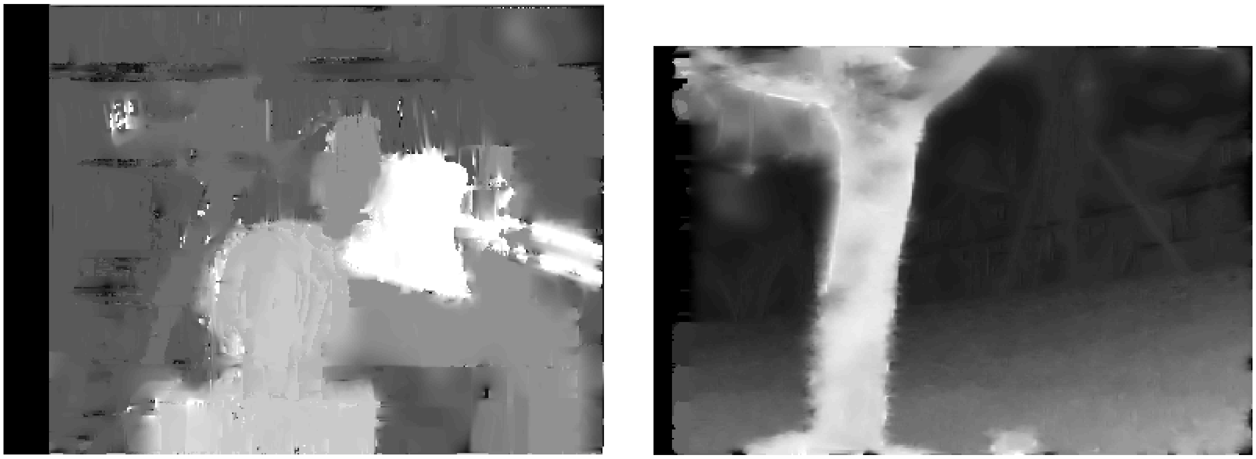

(a) (b)

Figure 3.9: Edge pixels detected using Sobel masks: (a) *Tsukuba*; (b)*Flower Garden.*

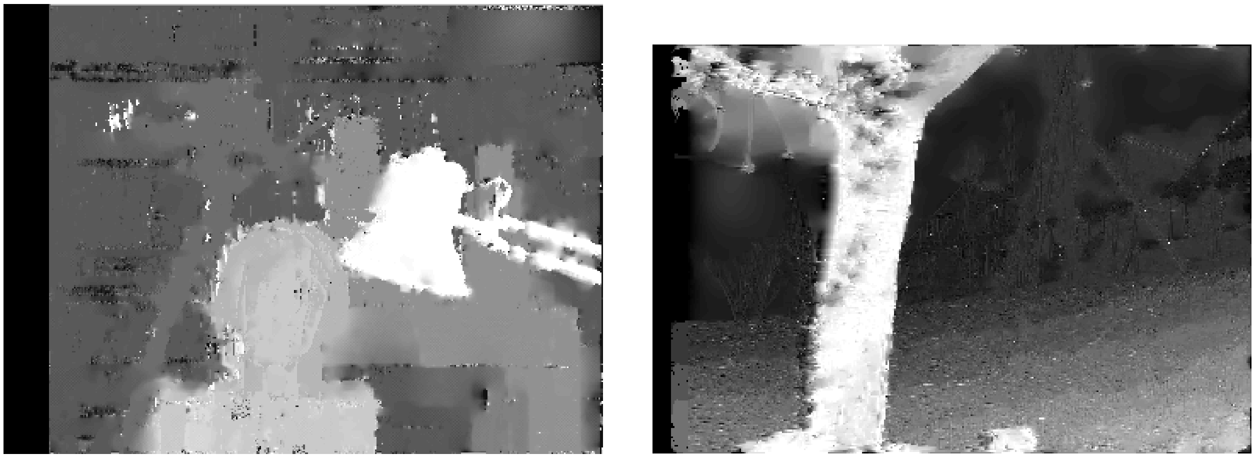
Then we apply our new edge-based refinement scheme to the same coarse disparity as in Fig. 3.2. A certain amount of refinement iterations still need to be applied to these edge pixels. In our experiments, we applied 400 iterations on edge pixels, and 800 iterations on non-edge pixels, which are determined empirically.

The final refinement results using the above scheme which distinguishes between edge and non-edge pixels are shown in Fig. 3.10 and Fig. 3.11 using (3.8) and (3.11) respectively.



(a) (b)

Figure 3.10: Refinement by (3.8): (a) *Tsukuba*; (b)*Flower Garden*.



(a) (b)

Figure 3.11: Refinement by (3.11): (a) *Tsukuba*; (b)*Flower Garden*.

Compared with the results shown in Fig. 3.7 and Fig. 3.8, we can see that for the refinement process distinguishing edge pixels, the results in Fig. 3.10 and Fig. 3.11 are better from the point of view of keeping object boundaries clearer, especially for those tiny features like the contours of the lamp in *Tsukuba* and the twigs in *Flower Garden*. Here we are facing a longstanding problem in the area of disparity estimation – how to evaluate the performance of a disparity estimation algorithm. Although there are benchmark stereo images like *Tsukuba* with so called ground-truth disparities, as we stated in Chapter 2, that is not a completely satisfactory way for the performance evaluation of disparity estimation algorithms since those ground-truth disparities are manually labeled integers. If 3D models are set up based on those ground-truth disparities, then we can see the quantization effects if viewing from the side. Another drawback for those benchmark image sets is that they do not contain images with complex scenes like in the *Flower Garden*. Although we used PSNR values for the coarse disparities in section 3.1.1, using variational refinement may reduce those PSNR values even though the visual quality is improved. This is because the functionals like (3.1) or SSD are actually square values of displaced frame difference, and the minimization of them is just the same thing as obtaining a better PSNR. While variational regularization is like adding a term involving gradient values of the image to the original square values of displaced frame difference, so the overall functional is not optimized only for the minimization of the square values of displaced frame difference. Therefore we can only judge the quality of these disparity maps by visual impression here. We will propose an objective way for the performance evaluation of disparity maps in Chapter 5 using the final 3D models.

### 3.1.4 Disparity from Region Matching

To further increase the quality of our disparity estimation algorithm, we now incorporate region matching techniques.

Until now, similarly to most of the existing disparity-estimation algorithms, our disparity estimation approach – from Gabor-based coarse estimation to variational refinement – is pixel-based; this is one common feature for various otherwise quite different existing disparity estimation algorithms. Compared to the large number of papers on pixel-based disparity estimation algorithms, there are only a few dealing with region-based disparity estimation [59, 60]. In [59], the *mean shift* segmentation algorithm developed in [4] was used to segment the images into different regions. However, in the next steps, like the methods used in [60], oversegmentation was applied to each region in order to handle the linear variation problems for untextured and slanted surfaces. Such oversegmentation is a step towards a pixel-based approach reducing the advantage of a region-based approach. We make the assumption that each region in one image of the stereo pair can be approximately considered as an affine transform from the same region in another image, and the region-based disparity estimation is thus converted to the estimation of the affine parameters for each region.

To better explain our idea on how to combine region matching techinques with our disparity-estimation algorithm, we show in Fig. 3.12 the block diagram for our overall disparity-estimation scheme. Our approach starts by filtering the left image $I_l$ and the right image $I_r$ with a set of Gabor filters. The left image $I_l$ is also put through a segmentation process using the mean shift algorithm, in which each region is formed by grouping pixels with similar color values and represented by *one* color value for this region. The Gabor-filtered outputs of $I_l$ and $I_r$ are compared and a coarse disparity map is estimated. Then a variational regularization using an edge-preserving functional is applied on this coarse disparity map as a refinement process. After variational refinement, the disparity values in each region of $I_l$ (obtained from segmentation) are used to estimate a set of affine transform parameters by least squares, so that the matching relation for the pixels in this region with their corresponding pixels in $I_r$ can be represented by an affine transform. The affine parameters for each region are further adjusted using a descent-based region matching technique, and these adjusted affine parameters can be used in turn to calculate a more refined disparity map.

### Segmentation of $I_l$

We applied the mean shift segmentation algorithm [4] to the image $I_l$, and the segmentation results for *Tsukuba* and *Flower Garden* are shown in Fig. 3.13. The source and binary codes of mean shift can be downloaded from [61]. The mean shift algorithm applies an averaging operation on the image and then groups some adjacent pixels with similar color values

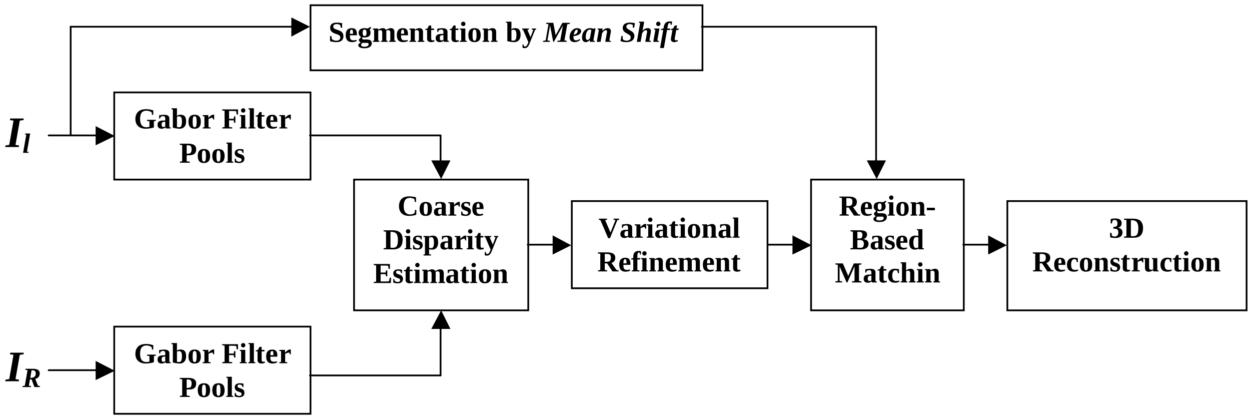


Figure 3.12: Block diagram of our approach

together by assigning these pixels with their mean color value, i.e., each region is indicated by one color value. Comparing Fig. 3.13(b) with its original in Fig. 2.5(a), we can find that the mean shift algorithm could not identify some tiny features, which are missing after segmentation (e.g., some twigs on the tree, and part of the shrubs). To alleviate such a problem, we have performed an edge-detection using Canny detector and edge linking on $I_l$ and on Fig. 3.13(b), and then compare the detected edges between the two images to pick out the missing tiny contours that are not detected in Fig. 3.13(b). The new segmentation result for *Flower Garden* is shown in Fig. 3.14. Although we get most of the missing tiny contours back, this method also introduces some extra contours on some existing regions.

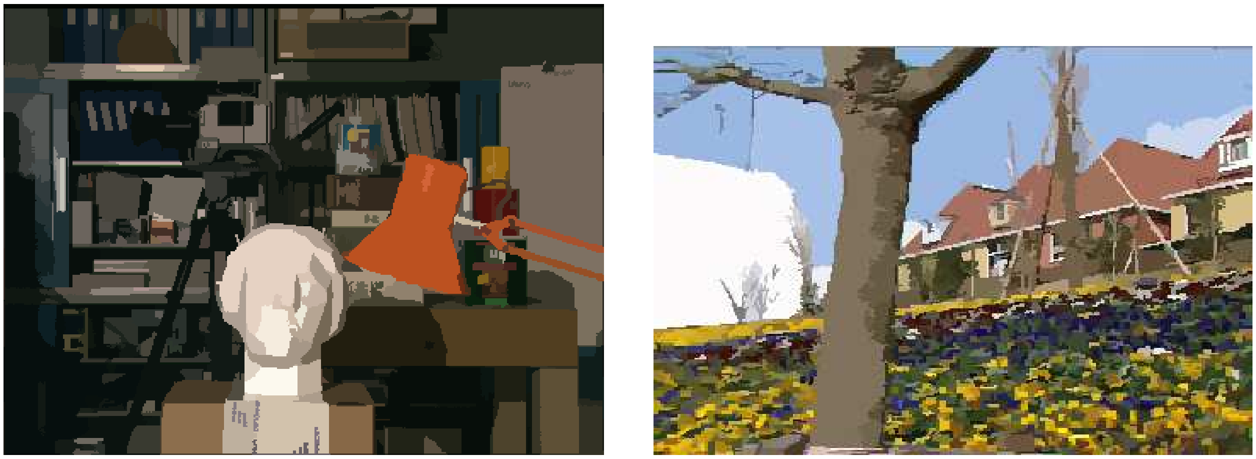
### Representing Disparity by an Affine Transform

We assume that the coordinates $(x, y)^T$ of each pixel in a region in $I_l$ are related to those of corresponding pixels $(x_r, y_r)^T$ in $I_r$ by an affine transform. In the case of parallel stereo without vertical displacement ($y = y_r$), we have:

$$x_r = a_{11}x + a_{12}y + a_{13}. \tag{3.14}$$

Therefore, the disparity $d(x, y)$ is related to these affine parameters by

$$d(x, y) = x - a_{11}x - a_{12}y - a_{13}. \tag{3.15}$$



(a) (b)

Figure 3.13: Segmentation by meah shift: (a) *Tsukuba*. (b) *Flower Garden*.

Thus, the estimated $d(x, y)$ for each pixel in one region from the previous variational refinement can be grouped and used as known variables so that the affine parameters can be estimated from (3.14). Since each pixel in the region gives one equation as in (3.14), and for most of the cases, the number of pixels in a region is larger than the number of affine parameters (three for 1-D affine transform), the estimation of the three parameters $(a_{11}, a_{12}, a_{13})$ can be done by least squares, implemented using singular value decomposition (SVD). Assume there are $N$ pixels in a region, then we have $N$ equations of (3.15) for this region, which can be expressed in a form using matrix and vectors

$$\mathbf{Qa} = \mathbf{b}, \tag{3.16}$$

where $\mathbf{Q}$ is a $N \times 3$ matrix defined as

$$\mathbf{Q} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix}, \tag{3.17}$$

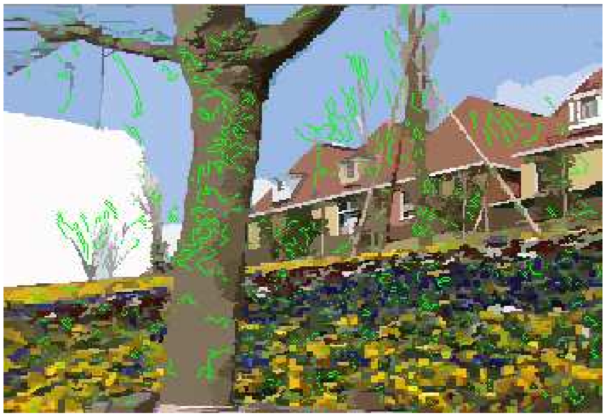


Figure 3.14: New segmentation with tiny contours recovered (in green color).

and **a** and **b** are two vectors with sizes of 3 × 1 and $N \times 1$ respectively defined as

$$\mathbf{a} = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} x_1 - d(x_1, y_1) \\ x_2 - d(x_2, y_2) \\ \vdots \\ x_N - d(x_N, y_N) \end{bmatrix}. \tag{3.18}$$

The SVD can decompose matrix **Q** into a product form:

$$\mathbf{Q} = \mathbf{U} \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \mathbf{V} \tag{3.19}$$

where **U** is a $N \times 3$ matrix and **V** a 3 × 3 matrix. Both **U** and **V** are orthogonal matrices, which means:

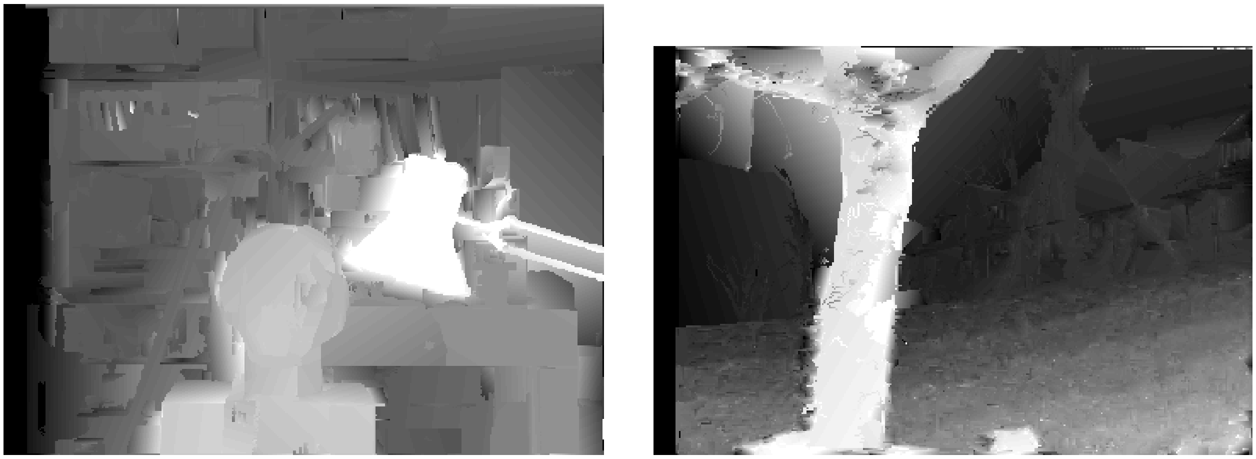
$$\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

where $\mathbf{V}^T$ and $\mathbf{U}^T$ is the transpose of matrices **V** and **U**. Then the three affine parameters in **a** can be calculated as:

$$\mathbf{a} = \mathbf{V}^T \begin{bmatrix} 1/w_1 & 0 & 0 \\ 0 & 1/w_2 & 0 \\ 0 & 0 & 1/w_3 \end{bmatrix} \mathbf{U}^T. \tag{3.21}$$

Once the affine parameters are estimated, a new disparity $d(x, y)$ for each pixel in the region can be in turn calculated by (3.14).

The new results for *Tsukuba* and *Flower Garden* from the above procedure are shown in Fig. 3.15 based on the previous variational refinement results shown in Fig. 3.11. We can see that this kind of parameterized estimation process can give more reasonable results in which the noise in each region is somewhat removed, but non-smoothness exists among some adjacent regions. To solve this problem, we use region matching to improve the affine parameters.



(a) (b)

Figure 3.15: New results by applying the affine parameters to the calculation of the disparities for each region: (a) *Tsukuba*; (b) *Flower Garden*.

### Further Refinement by Region Matching

The error function that we need to minimize for each region is:

$$E = \sum_{(x,y) \in W_i} [I_r(a_{11}x + a_{12}y + a_{13}, y) - I_l(x, y)]^2 \tag{3.22}$$

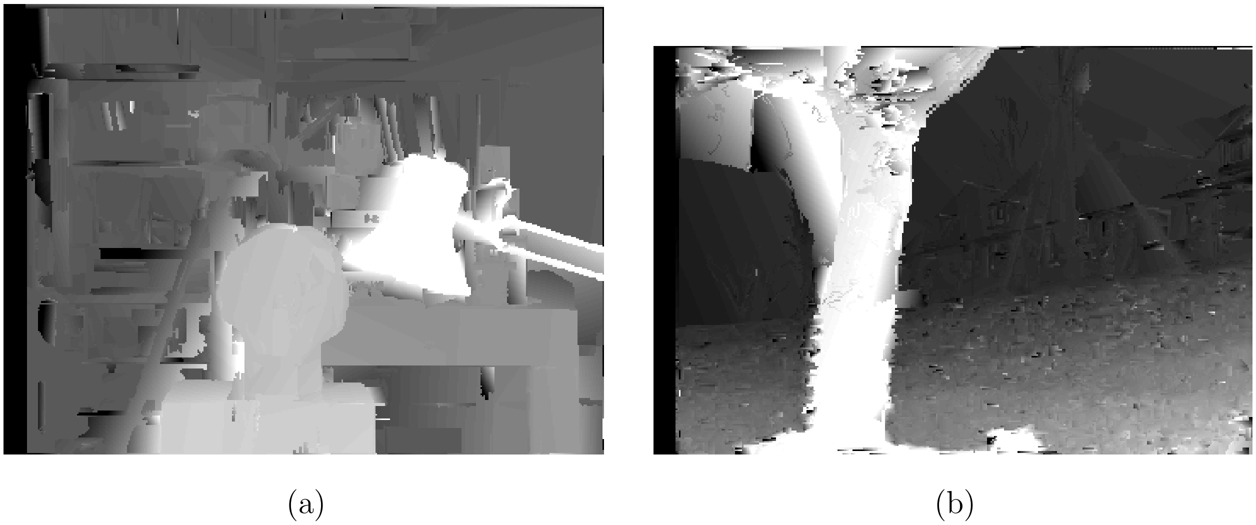
where $W_i$ represents a region. We need to minimize (3.22) by updating affine parameters $\mathbf{a} = [a_{11}, a_{12}, a_{13}]^T$ iteratively using least squares with Taylor expansion. Assume $\mathbf{X} = [x, y, 1]^T$. Let $\hat{\mathbf{a}}$ be the current estimate of affine parameters, and $\mathbf{a} = \hat{\mathbf{a}} + \Delta\hat{\mathbf{a}}$. Then expand $I_r$ around the current estimate $I_r(\mathbf{a}^T\mathbf{X}, y) \approx I_r(\hat{\mathbf{a}}^T\mathbf{X}, y) + \Delta\hat{\mathbf{a}}^T\mathbf{X}I_{r,x}(\hat{\mathbf{a}}^T\mathbf{X}, y)$; this first order expansion is valid only when $\hat{\mathbf{a}}$ is close to $\mathbf{a}$. This is the reason that we start the region matching with the result from pixel-based approach, rather than doing it from the very beginning without pixel-based results. Substituting the above first order expansion into (3.22), the error function becomes:

$$E(\Delta\hat{\mathbf{a}}) = \sum_{(x,y)\in W_i} [\psi^T\Delta\hat{\mathbf{a}} - D]^2 \tag{3.23}$$

where $\psi = I_{r,x}(\hat{\mathbf{a}}^T\mathbf{X}, y)\mathbf{X}$ and $D = I_l(x, y) - I_r(\hat{\mathbf{a}}^T\mathbf{X}, y)$. The solution of (3.23) by least squares is [62]:

$$\Delta\hat{\mathbf{a}} = [\sum_{(x,y)\in W_i} \psi\psi^T]^{-1} \sum_{(x,y)\in W_i} D\psi. \tag{3.24}$$

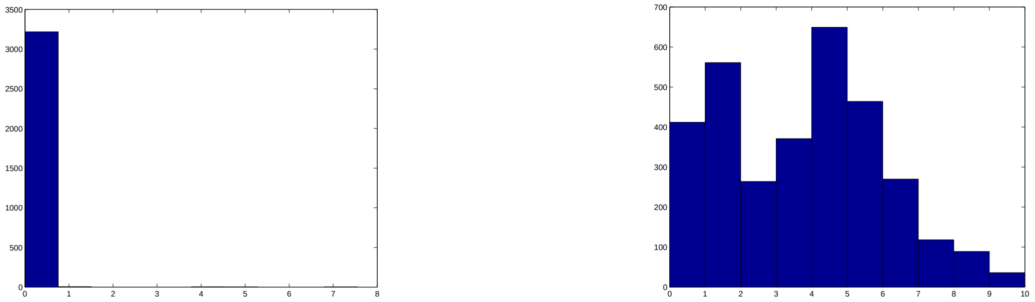
The resulting disparities obtained from the new affine parameters updated using (3.24) are shown in Fig. 3.16. Compared with Fig. 3.15, the new disparities have some improvements for regions which belong to the same object surfaces, like the slanted slope surface in *Flower Garden*. However, some regions where there are occlusions give worse effects than the same areas in Fig. 3.15, e.g., the sky areas with twigs and shrubs as foreground objects. This is because the minimization of (3.23) through (3.24) is trying to minimize the squared intensity difference between *all* the pixels (including occluded pixels) in a region of $I_l$ with their


(a)

(b)

Figure 3.16: New disparities after region matching: (a) *Tsukuba*; (b) *Flower Garden*.

corresponding pixels in $I_r$. Thus the region matching technique can make such regions shift along with their foreground objects. This effect can be detected by comparing patterns in the disparity histogram for a region before and after region matching, and the correct disparities can thus be determined from these patterns. For example, the disparity histogram for a sky region (between a twig and the tree) has one peak near zero value before region matching, and after region matching there are two peaks. There are 3256 pixels in this region, and most of them have disparity values range from 0 to 0.8, as shown in Fig. 3.17(a). After region matching, for the same region, the disparity histogram spreads over a range from 0 to 10 with a second peak located around the values 4 to 5, as shown in Fig. 3.17(b). This second peak with a higher value comes from those occluded pixels near foreground objects. Once such big pattern changes have been detected, which means it is very possible that some problems happened from the second peak due to the occlusion, the real disparity values for such regions is then determined by the lower peak in Fig. 3.17(a) (adopting the lowest value for that peak), and verified if the *mean absolute difference* between $I_l$ and $I_r$ for that region (excluding those occluded pixels which are detected from the technique in [20]) is less or equal to its value by using Fig. 3.17(b). The mean absolute difference for that region is 0.15 in Fig. 3.16(b), and 0.11 in Fig. 3.18(b).

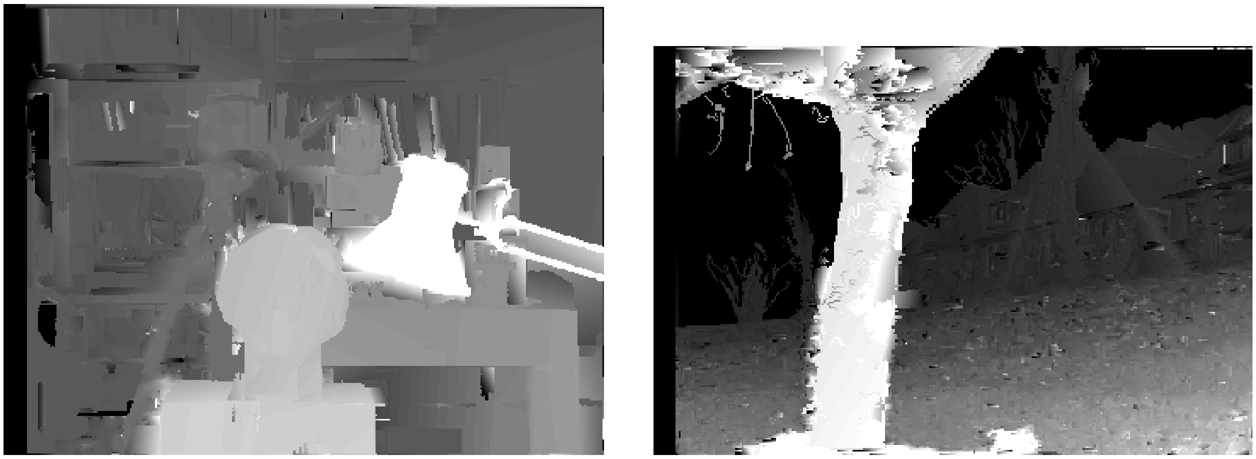


(a)

(b)

Figure 3.17: Histogram change for a sky region of *Flower Garden*: (a) before region matching; (b) after region matching.

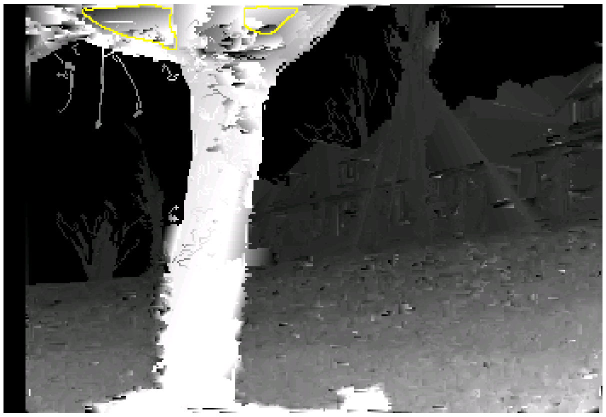
Fig. 3.18 shows the final results after making use of such pattern detections, where we can see some regions containing occlusions, like the regions left of the lamp and of the neck-of-head in *Tsukuba*, and the sky regions in *Flower Garden*, have been identified and the correct disparities have been assigned.



(a) (b)

Figure 3.18: Results after disparity histogram analysis: (a) *Tsukuba*; (b) *Flower Garden*.

### 3.1.5 Summary for Disparity Estimation



(b)

Figure 3.19: Two sky regions couldn't be detected

In this section, we showed our development process for disparity estimation which evolved from pixel-based techniques (Gabor filtering plus variational regularization) to region-based approaches. The results obtained until now (as shown in Fig. 3.18), from the visual quality point of view, show big improvements for some longstanding problems in disparity estimation, like the linear variation for slanted surfaces, the detection of zero displacements for sky regions, and keeping the object contours sharp and clear. Although we can not show these improvements through quantitative performance, these improvements are especially useful for the purpose of 3D reconstructions.

However, such pixel- and region-based approach still needs further improvements. For example, for the sky region detection using disparity histograms, there are sky regions that can not be detected from the method we used in section 3.1.4, as circled out with yellow lines in Fig. 3.19. This is because, after region matching, such regions do not have a second peak coming out in their disparity histogram. Therefore, we need some other methods in such region-based disparity histogram analysis, which we show in the next section.

## 3.2 Multiview Disparity with 3D Modeling Using Point Sets

In order to obtain the structure of a whole environment, we need to capture multiview images or monocular/binocular video sequences throughout it. Therefore, obtaining the depth information for one location using disparity estimation is not enough, since we still need to deal with multiview images. In this section, we will improve our pixel- and region-based disparity estimation approach and apply it to the translational sequence *Flower Garden*, so that a preliminary 3D model based on 3D point sets can be integrated by combining the separate depth maps of two image locations.

### 3.2.1 Region-Based Disparity Analysis for Translation Video Sequence

We now extend our disparity estimation algorithm to the case of translational video sequence, and the depth information for each image location can be obtained in a straightforward fashion, similarly as in the parallel stereo cases. The video sequence that we use for our simulation is also *Flower Garden*. However, unlike for the stereo case, we will apply our algorithm on more consecutive images.

As shown in Fig. 3.20, our system starts by filtering the two consecutive images $I_t$ and $I_{t+1}$ from a translational video sequence with a set of Gabor filters. Also $I_t$ is put through a segmentation process using the mean shift algorithm. The filtered versions of $I_t$ and $I_{t+1}$ are compared and a coarse disparity $d_G$ is estimated. Another disparity $d_R$ based on variational regularization using an edge-preserving functional is also estimated iteratively with disparity values for each pixel initialized with zero. Then the histograms of disparity values from $d_G$ and $d_R$ in each region of $I_t$ (obtained from the segmentation) are compared in order to identify those regions without movements (zero displacement). Once such regions with zero displacement are identified, the disparity values for the other regions of $I_t$ are used to estimate a set of affine transform parameters by least squares, so that the matching relation for the pixels in each region with their corresponding pixels in $I_{t+1}$ can be represented by the resulting affine transform. The affine parameters for each region are further adjusted using a descent-based region-matching technique, and these adjusted affine parameters can be used in turn to calculate a more refined disparity map. Once we get this final disparity for $I_t$, the depth image for the location of $I_t$ is obtained using the reciprocal values of the disparity values for each pixel. Because we are dealing with a translational video sequence, and similar to the relations between disparity and depth for parallel stereo, the disparity value is in a reciprocal relation with the depth value up to a scale factor.

Among the large amount of literature and algorithms for optical flow and motion estimation, differential techniques [41][42] with variational regularization form a major class. These techniques involve a functional including the displaced frame difference and a smoothing term, and usually descent-based methods are used to minimize the functional by solving its associated Euler-Lagrange equations. Recently, Brox et al. significantly improved this approach by embedding a multiresolution strategy and gradient constancy to a nonlinear objective functional, and obtained the best results until now for some standard test sequences like *Yosemite* [63]. Kim et al. used a similar functional with a modified regularization term

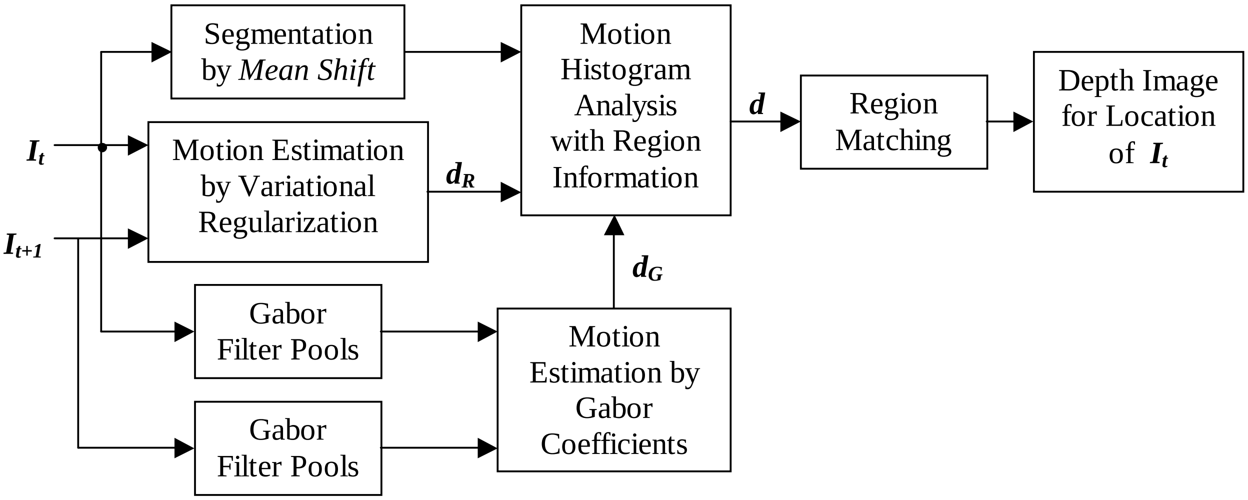
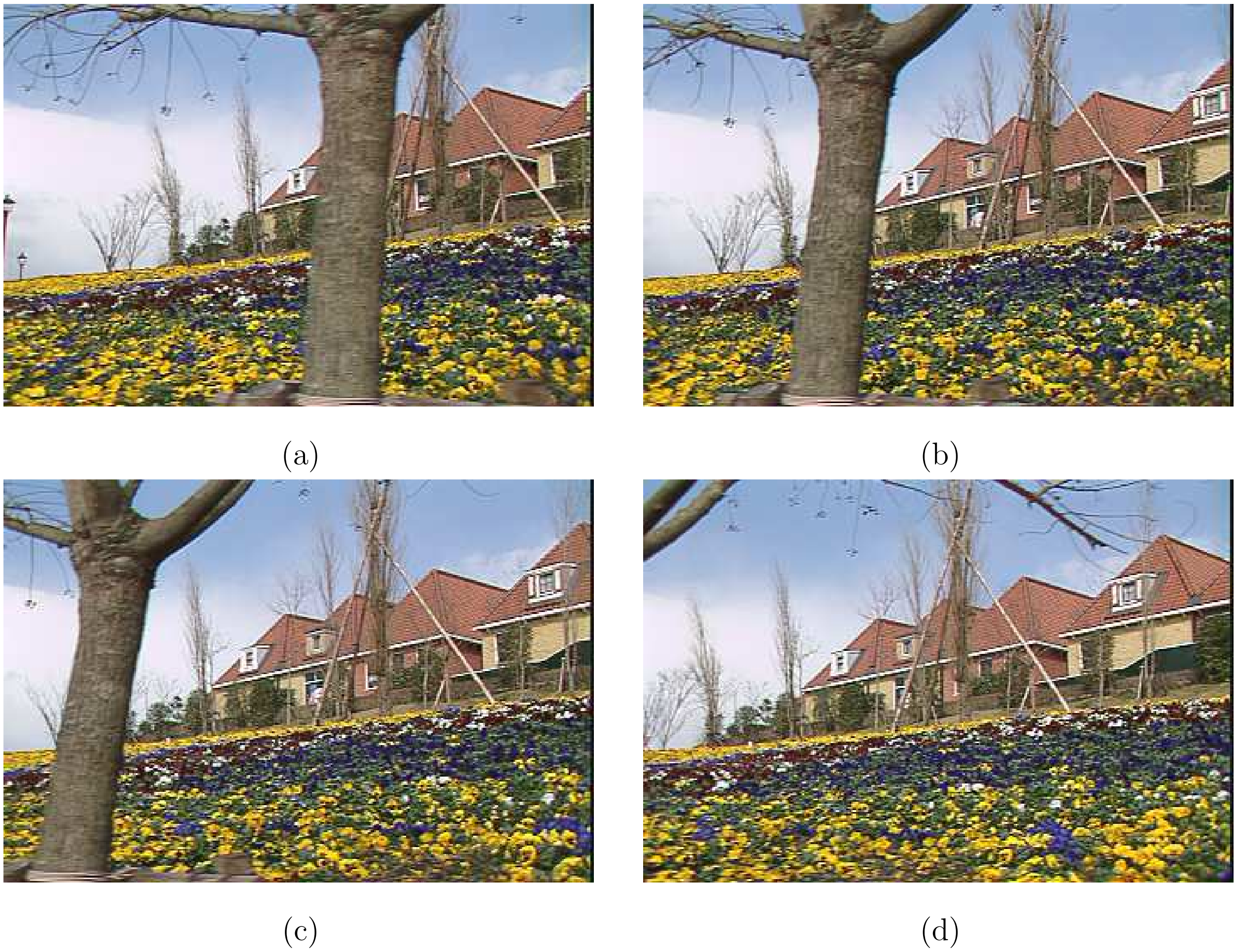

Segmentation by *Mean Shift*
Motion Histogram Analysis with Region Information
Region Matching
Depth Image for Location of $I_t$
$I_t$
$I_{t+1}$
Motion Estimation by Variational Regularization
$d_R$
$d$
$d_G$
Gabor Filter Pools
Gabor Filter Pools
Motion Estimation by Gabor Coefficients

Figure 3.20: Block architecture for motion analysis

and, to handle large motion fields, used a coarse-to-fine scheme, and solved the associated Euler-Lagrange equation using recursive iterations [64].

However, the functionals used in the variational regularization approach usually do not take the occlusion effect into account, i.e., the objective functional that this approach tries to minimize is the displaced frame difference between *all* the pixels of $I_t$ and their corresponding pixels in $I_{t+1}$. Due to this reason, after iterative calculations to minimize such objective functionals, those background pixels (which should be occluded) along the foreground objects usually have motion values similar to the motion values of those foreground pixels, since the iteration process also tries to find a solution for such occluded background pixels. This will bring wrong motion values for such occluded background pixels. For example, in [64], the video sequence *Flower Garden* was used and from the result of its motion maps, most of the sky areas are merged with the middle objects and even with the twigs of the foreground tree. Therefore, although the displaced frame difference between $I_t$ and $I_{t+1}$ can be minimized to a small value which is good enough for some other purposes like compression and coding, the motion values estimated by variational regularization approach could not satisfy the purpose of 3D model constructions, since part of or most of the untextured background areas will be merged with the foreground objects, especially when those foreground objects have complex geometries.

We will try to solve this problem for translational video sequence by comparing the results from the variational regularization approach with the disparity estimation results from the Gabor transform and image segmentation. The *Flower Garden* sequence is taken along a straight line, and is approximately equi-distant for any two consecutive images. The maximum horizontal motion is about 6 pixels/frame. We show the *5th*, *22nd*, *35th* and *65th* images in Fig. 3.21; three of them contain the foreground tree and therefore the disparity estimation for these images is more difficult than for those without the tree. We will also show the disparity estimation results for these images.



(a)

(b)

(c)

(d)

Figure 3.21: Original images in *Flower Garden*: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

### Detecting Regions with Zero Displacement Using Variational Regularization Approach

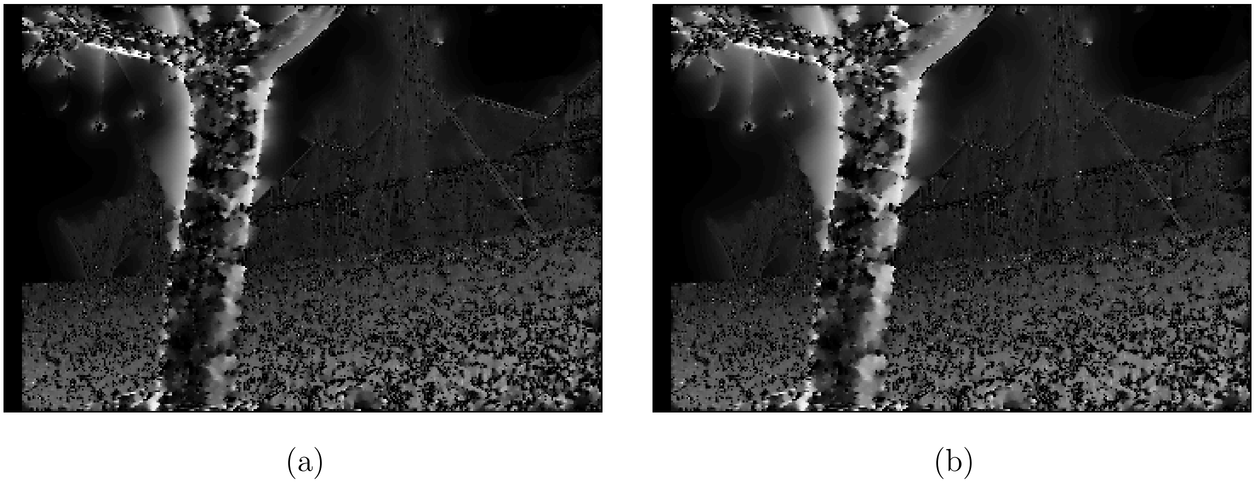
Similar to the general variational regularization approach, we use the same regularization functional (2.12) which we write down here together with the data fidelity term for convenience:

$$\begin{aligned} E(d_R) = & \iint [I_t(x,y) - I_{t+1}(x-d_R,y)]^2 \,\mathrm{d}x\,\mathrm{d}y \\ & + \lambda \iint \Big\{ \frac{1}{(1+I_{t,x}^2)^2} d_{R,x}^2 + \frac{1}{(1+I_{t,y}^2)^2} d_{R,y}^2 \Big\} \mathrm{d}x\,\mathrm{d}y \end{aligned} \tag{3.25}$$

where $\lambda$ is a regularization parameter, $d_{R,x}$ and $d_{R,y}$ are derivatives of $d_R(x,y)$ in $x$ and $y$ directions respectively, and similarly for $I_{t,x}$ and $I_{t,y}$. The minimization of (3.25) to estimate $d_R$ is carried out by applying a gradient descent method to solve its associated Euler-Lagrange equation with respect to $d_R$:

$$\begin{aligned} \frac{\partial d_R}{\partial t} = & [I_t(x,y) - I_{t+1}(x-d_R,y)] \times I_{t+1,x}(x-d_R,y) \\ & - \lambda \Big\{ \frac{\partial}{\partial x}\Big[ \frac{d_{R,x}}{(1+I_{t,x}^2)^2} \Big] + \frac{\partial}{\partial y}\Big[ \frac{d_{R,y}}{(1+I_{t,y}^2)^2} \Big] \Big\}. \end{aligned} \tag{3.26}$$

Unlike the coarse-to-fine scheme as in [63] and [64] to prevent the solution from falling into local minima, we just use the original images and $d_R$ are initialized with zero for all pixels. As shown in Fig. 3.22, with the increase of iteration numbers, $d_R$ could reach their true values for those pixels with small movements (like those houses and shrubs), and could not completely reach their true values for the pixels with large movements (like the foreground tree) since they fell into local minima. Most important for 3D reconstruction purpose is that the disparity values for those background pixels (sky) leave their true values (zero) and approach the disparity values of their foreground objects with the increase of iteration numbers. Therefore, as we stated in the beginning of this subsection, the objective functional can be further minimized with the increase of iteration numbers, but this does not fit our purpose of 3D model construction. Our solution for this dilemma is to use fewer iterations, such that most of the pixels with small movements can reach their true disparity values, and most of the background pixels with zero displacement stay where they are; the finding of



(a) (b)

Figure 3.22: $d_R$ for $22nd$ image after different numbers of iterations: (a) 2500; (b) 4000.

large disparity values for those foreground object can be left to some other techniques (as we show later).

For the images in Fig. 3.21, we used 800 iterations for their disparity estimation, and the results are shown in Fig. 3.23. We can see that the values of $d_R$ for most of the sky regions are black (zero value), and the majority of pixels which should have small displacements (like the houses and shrubs) also have small disparity values.

### Estimation of $d_G$ by Gabor Transform

The method that we used for the estimation of $d_G$ through the Gabor transform is similar to the one we used in section 3.1.1, in which a set of quadrature-pair Gabor filters are used. Assume that the outputs of $k^{th}$ filter pair are $G^k_{I_t}(x,y)$ and $G^k_{I_{t+1}}(x,y)$ for $I_t$ and $I_{t+1}$ respectively. Then the disparity $\hat{d}_G \in [0, d_{max}]$ for a position $(x,y)$ in $I_t$ is determined as:

$$\begin{aligned}\hat{d}_G = &\arg\min_{d_G} \sum_k \big[ |Re\{G^k_{I_t}(x,y)\} - Re\{G^k_{I_{t+1}}(x-d_G,y)\}|^2 \\ &+ |Im\{G^k_{I_t}(x,y)\} - Im\{G^k_{I_{t+1}}(x-d_G,y)\}|^2 \big] \end{aligned} \tag{3.27}$$

where $Re\{G^k_{I_t}(x,y)\}$ and $Im\{G^k_{I_t}(x,y)\}$ are the real and imaginary parts of $G^k_{I_t}(x,y)$, and similarly for $G^k_{I_{t+1}}(x,y)$.

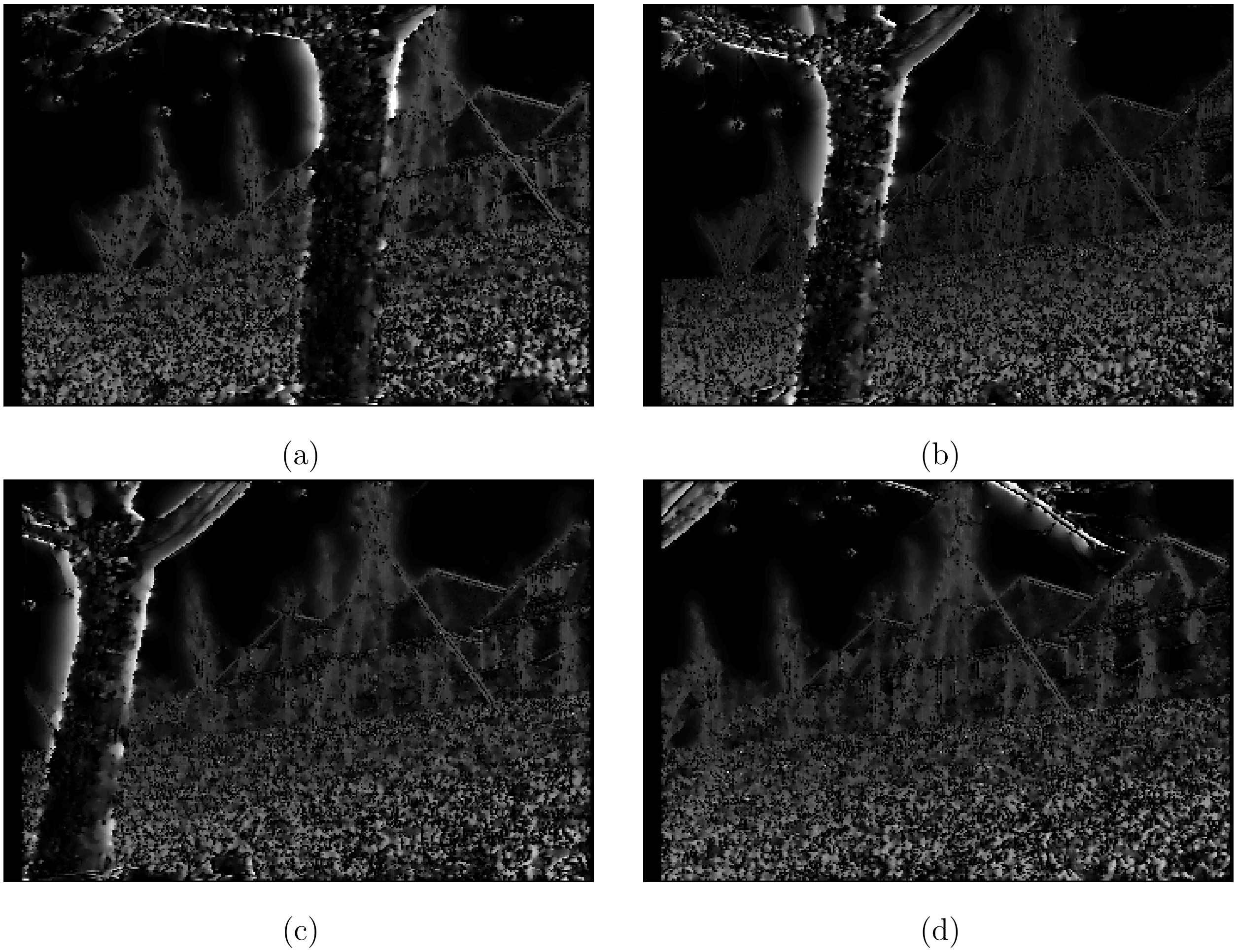


Figure 3.23: $d_R$ maps after 800 iterations: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

The disparity maps $d_G$ estimated by (3.27) for the four images in Fig. 3.21 are shown in Fig. 3.24. We can see that these results are good for pixels with large and medium displacements. However, for part of the pixels with zero displacement but near some middle and foreground objects, their disparity values tend to be confused with the disparity values for those objects. For example, for the sky areas above the houses in Fig. 3.21(b), most of their disparity values are the same as the houses with small displacements as shown in Fig. 4.5(b), rather than zero which should be shown as completely black.

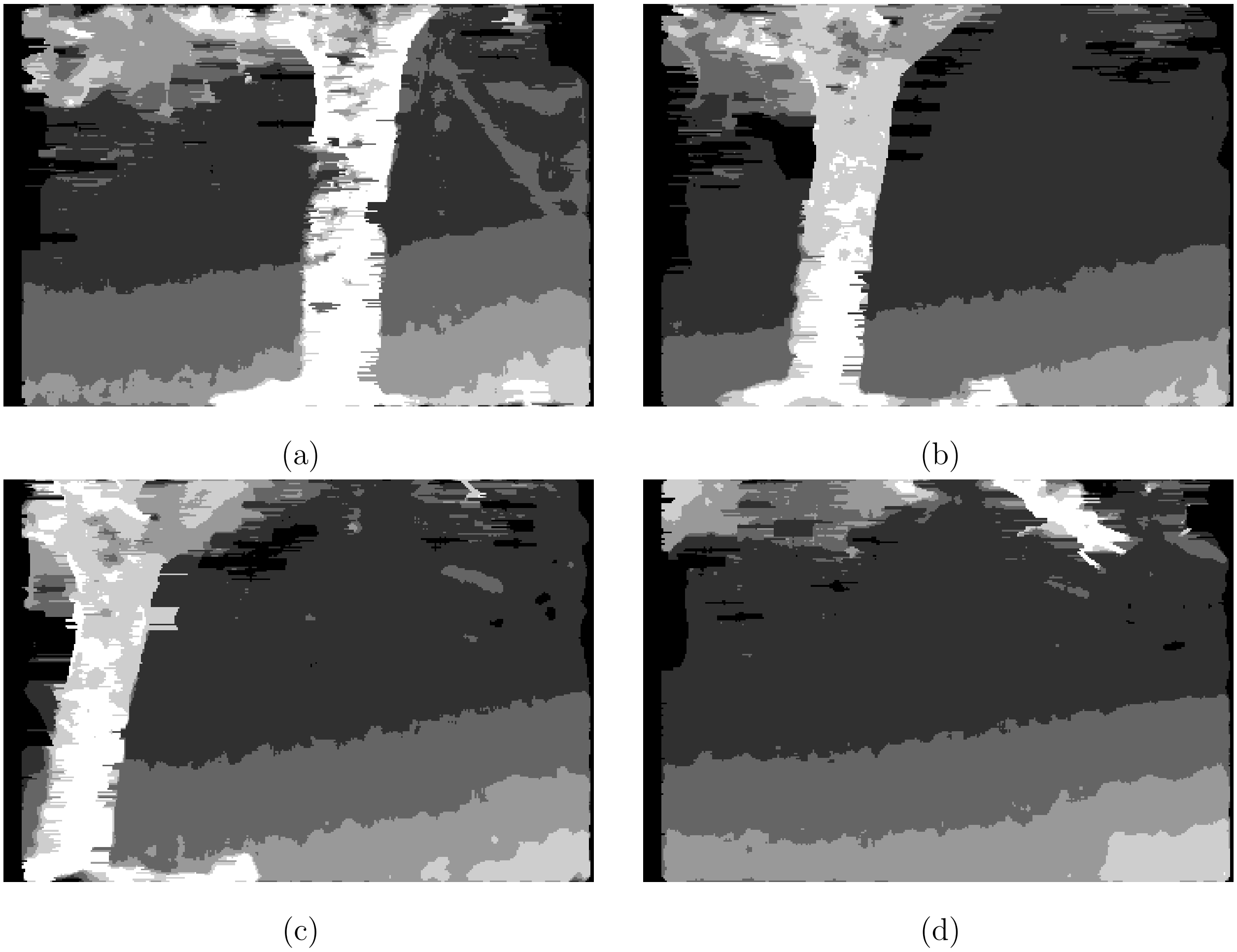
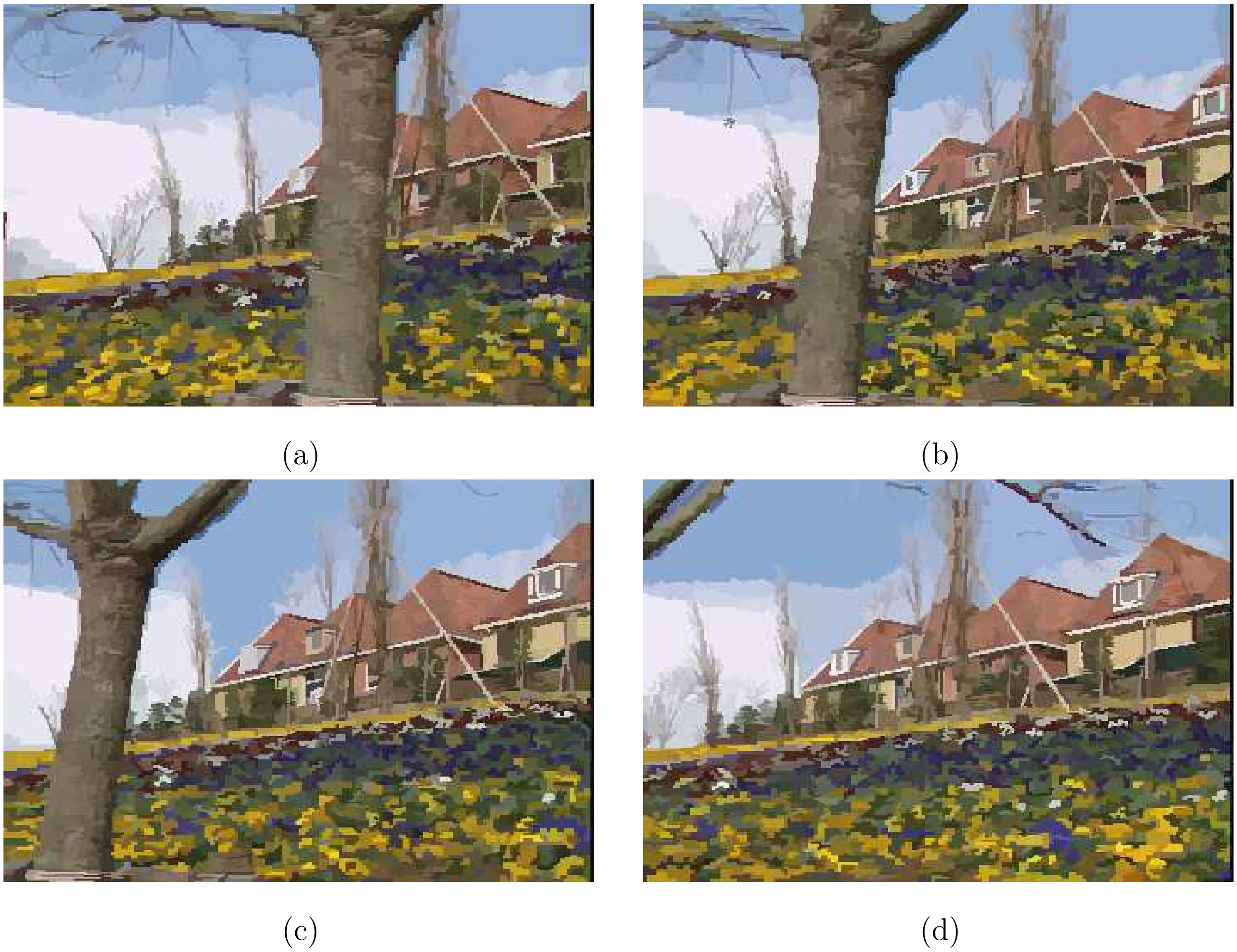


Figure 3.24: $d_G$ maps from Gabor transform: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

**Region-Based Analysis**

Comparing Fig. 3.23 and Fig. 3.24, we find that the results from the two methods are complementary to each other, in which the results from variational regularization approach are good for zero and small motions and the results from the Gabor transform are good for large as well as for small displacements. Therefore, we need to complement the two kinds of results from each other and obtain one good disparity map for the whole displacement range. In order to do that, we need to consider them in groups of connected pixels that fall in the same kind of regions that should have similar disparity values. Thus, we need to have region information from segmentation applied to images $I_t$.

The same as in section 3.1.4, we applied the mean shift segmentation algorithm [4] to the images $I_t$, and the segmentation results for the four images in Fig. 3.21 are shown in Fig. 3.25. Also, we have to run the contour detection program on these images in order to get the missing tiny contours back. This is because the mean shift algorithm groups some connected pixels by averaging their color values first, and once the color values are within a small range to the average value then they are considered to be in the same region. This kind of averaging operation might eliminate the color difference of some tiny features with their backgrounds, like the twigs in *Flower Garden*.


(a)

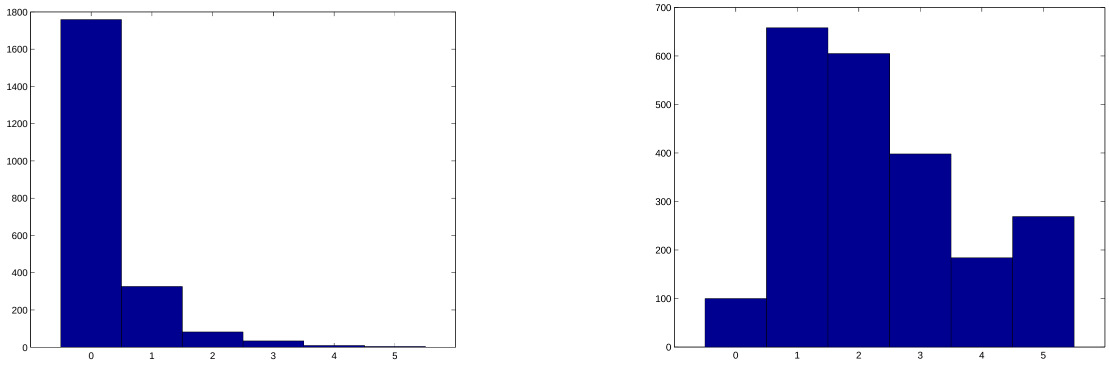
(b)

(c)

(d)

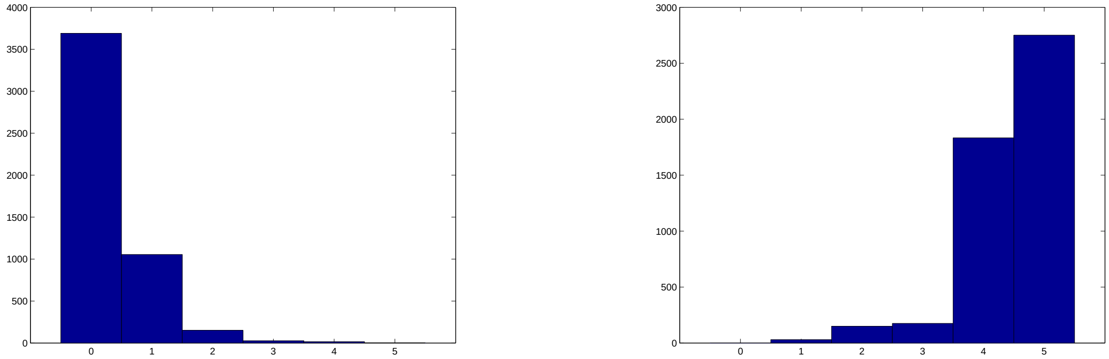
Figure 3.25: Segmentation by *Mean Shifts* [4]: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

Once we have the region information, we can compare and analyse the histograms of disparity values from $d_R$ and $d_G$ for each region. For example, as shown in Fig. 3.26 for a sky region in the $5th$ image which is between the upper twigs and the foreground tree, the histogram from $d_R$ is mainly located around zero which is the correct disparity value for this region, while the histogram from $d_G$ is spread across the whole range. As another example shown in Fig. 3.27 for a region of the foreground tree in the $22nd$ image, we can see that the histogram from $d_R$ is mainly located around zero while the histogram from $d_G$ is mainly located around the highest disparity values (which are correct). From our previous analysis, we already conclude that the disparity values from $d_R$ are good for zero and small displacements, while the disparity values from $d_G$ are good for large displacements and the



(a) (b)

Figure 3.26: Histograms for a sky region in $5th$ image : (a) from $d_R$; (b) from $d_G$.



(a) (b)

Figure 3.27: Histograms for a foreground tree region in $22nd$ image : (a) from $d_R$; (b) from $d_G$.

small displacements. Based on these observations, we can identify those regions with zero displacements by comparing the histograms from $d_R$ and $d_G$. For example, for the case in Fig. 3.26, since $d_R$ concentrates around zero while $d_G$ spreads across the whole range which means there is uncertainty in the process of obtaining the values of $d_G$ for this region, then $d_R$ is selected as the disparity values for this region; for the case in Fig. 3.27, since $d_R$ concentrates around zero while $d_G$ mainly concentrates on large disparity values which means there is more certainty in the process of determining the values of $d_G$, then $d_G$ is selected as the disparity values for this region.

The adjusted disparity maps $d$ after analysis of the histograms of disparity values for each region are shown in Fig. 3.28. Although we identified most of the sky regions now, most of the other regions with large disparity values are still in a coarse stage since the motion values from the Gabor transform are integers (e.g., those slope regions with quantization effects). We still need to further refine the disparity values for those regions by region matching techniques.

The refinement process using region matching is the same as in section 3.1.4 using Eqs. (3.14) – (3.24) except that $I_l$ and $I_r$ are changed to $I_t$ and $I_{t+1}$.

After region matching, the final results for the disparity maps of Fig. 3.21 are shown in Fig. 3.29.

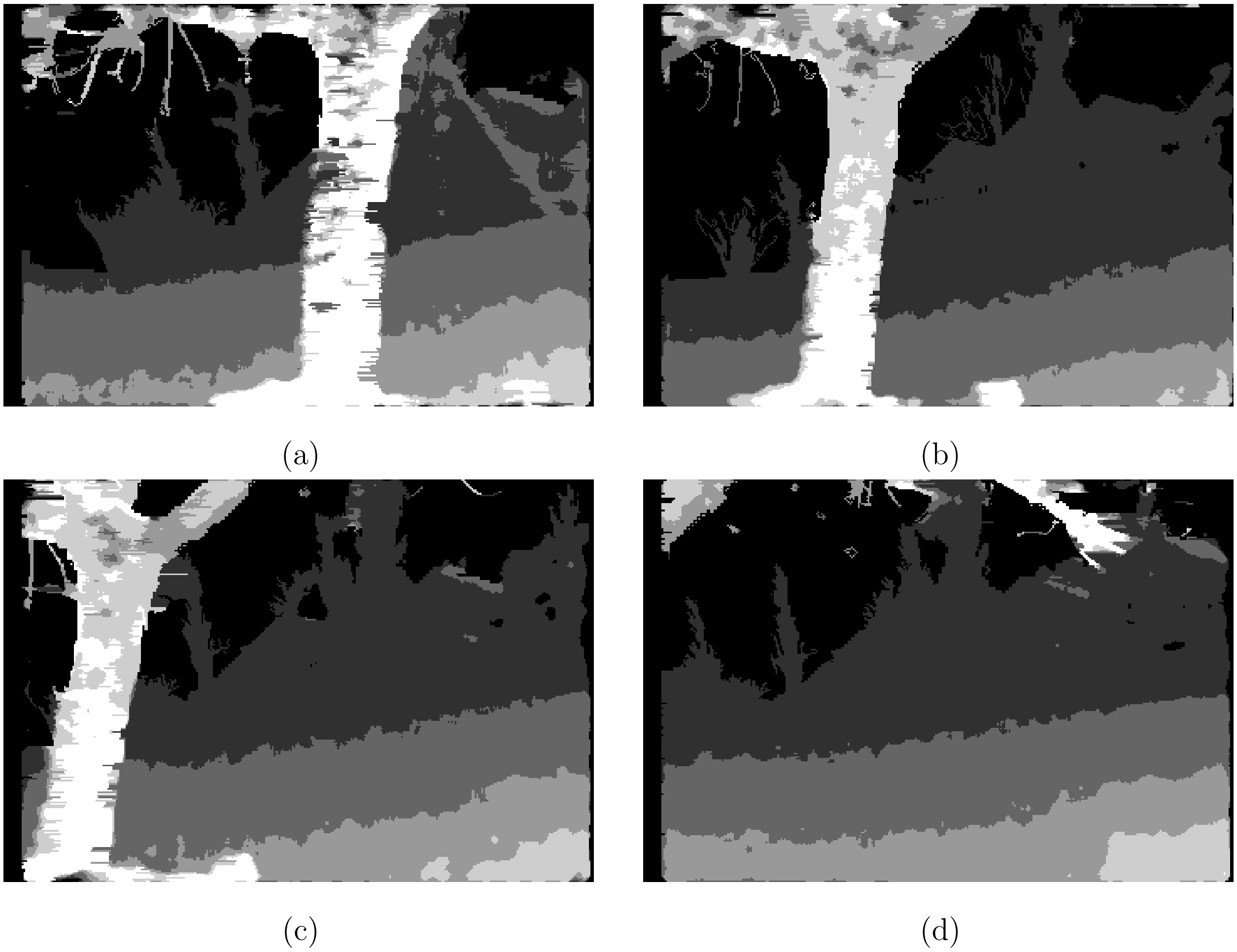
Then, for each location, the depth value $z(x, y)$ for a pixel at $(x, y)$ can be obtained as:

$$z(x,y) = \frac{Bf}{d(x,y)} \tag{3.28}$$

where $B$ is the baseline distance between $I_t$ and $I_{t+1}$, and $f$ is the focal length.

### 3.2.2 3D Reconstruction Using the Estimated Disparity Maps

We show in this section some 3D reconstructions based on the disparity or depth images we obtained. We set up the 3D models in OpenGL using 3D point arrays, and using *orthographic* projection mode for the rendering of novel views (only rotations allowed). The flow chart for the setting up of this 3D model and the rendering process is shown in Fig. 3.30.



(a)

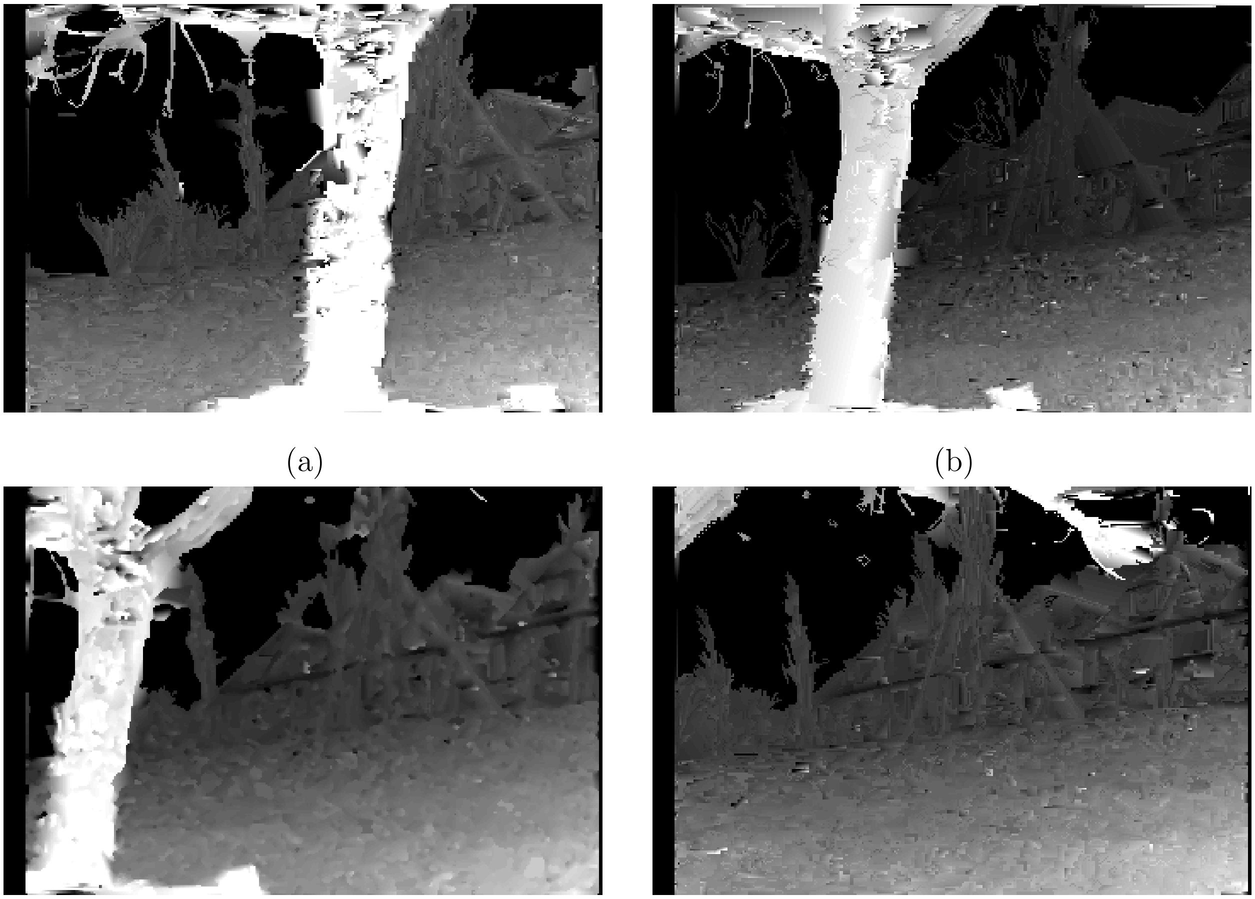
(b)

(c)

(d)

Figure 3.28: Disparity maps after histogram analysis: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

We first show in Fig. 3.31 some separate reconstructions based on each depth image on the four locations of Fig. 3.29 respectively. Fig. 3.31(a) is rendered by rotating about 10° around $y$-axis (vertical axis) from the original viewpoint to the right. Fig. 3.31(b) is rendered by rotating about 20° around $y$-axis to the left, then rotating up 15° around $x$-axis (horizontal axis). Fig. 3.31(c)(d) are rendered by rotating about 10° around $y$-axis to the right, then rotating up 10° and 5° around $x$-axis respectively. From these reconstructions, we can see that the sky has more shifting than the foreground scenes since it has the largest depth, and the occlusion from the foreground trees and shrubs on the sky can be clearly seen (black areas). Also, the linear variation of the depth values for the slanted slope surface can also be seen, especially from Fig. 3.31(d). All these facts indicate that the complex geometric



(a) (b)

(c) (d)

Figure 3.29: Final disparity maps after region matching: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.

structures detected by our algorithm are largely correct.

Then we attempted to combine the two depth images ($5th$ and $65th$) and their textures together with the $5th$ image as reference location. Since we assume that there is only translational shifting for the whole video sequence, the homogeneous transformation between the two locations is represented by only one parameter $K$ for the horizontal translation. Therefore, to shift the pixels $(x, y)$ of $65th$ image with depth value $z(x, y)$ to their corresponding image coordinates $(x_0, y_0)$ in the $5th$ image location, the $x_0$-components can be calculated as

$$x_0 = x + \frac{K}{z} \tag{3.29}$$

and $y_0 = y$, where $K$ is a constant determined by the baseline distance and the focal length

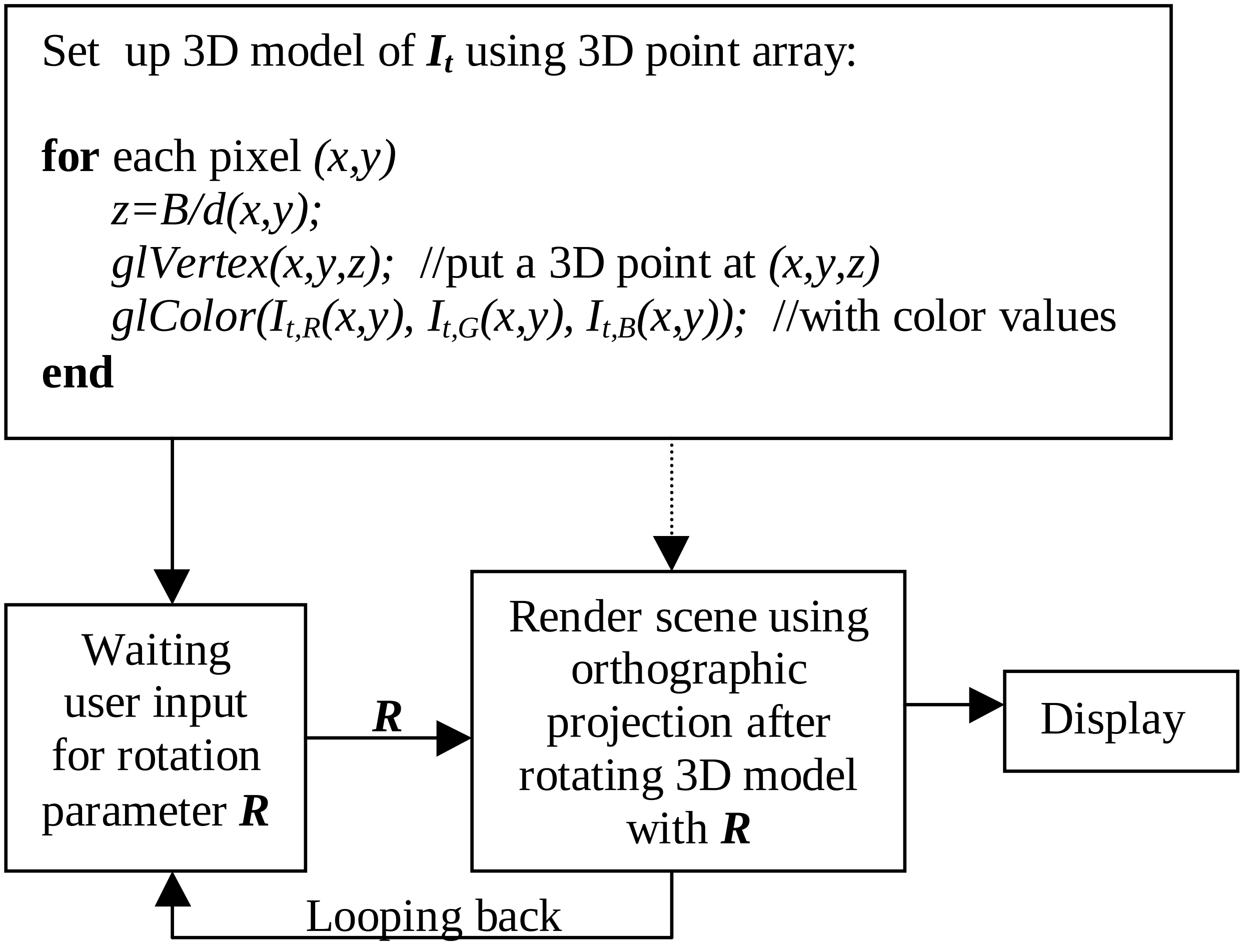

Set up 3D model of $I_t$ using 3D point array:
for each pixel (x,y)
z=B/d(x,y);
glVertex(x,y,z); //put a 3D point at (x,y,z)
glColor($I_{t,R}$(x,y), $I_{t,G}$(x,y), $I_{t,B}$(x,y)); //with color values
end
Waiting user input for rotation parameter R
R
Render scene using orthographic projection after rotating 3D model with R
Display
Looping back

Figure 3.30: Flow chart for point-based 3D model setting up and rendering.

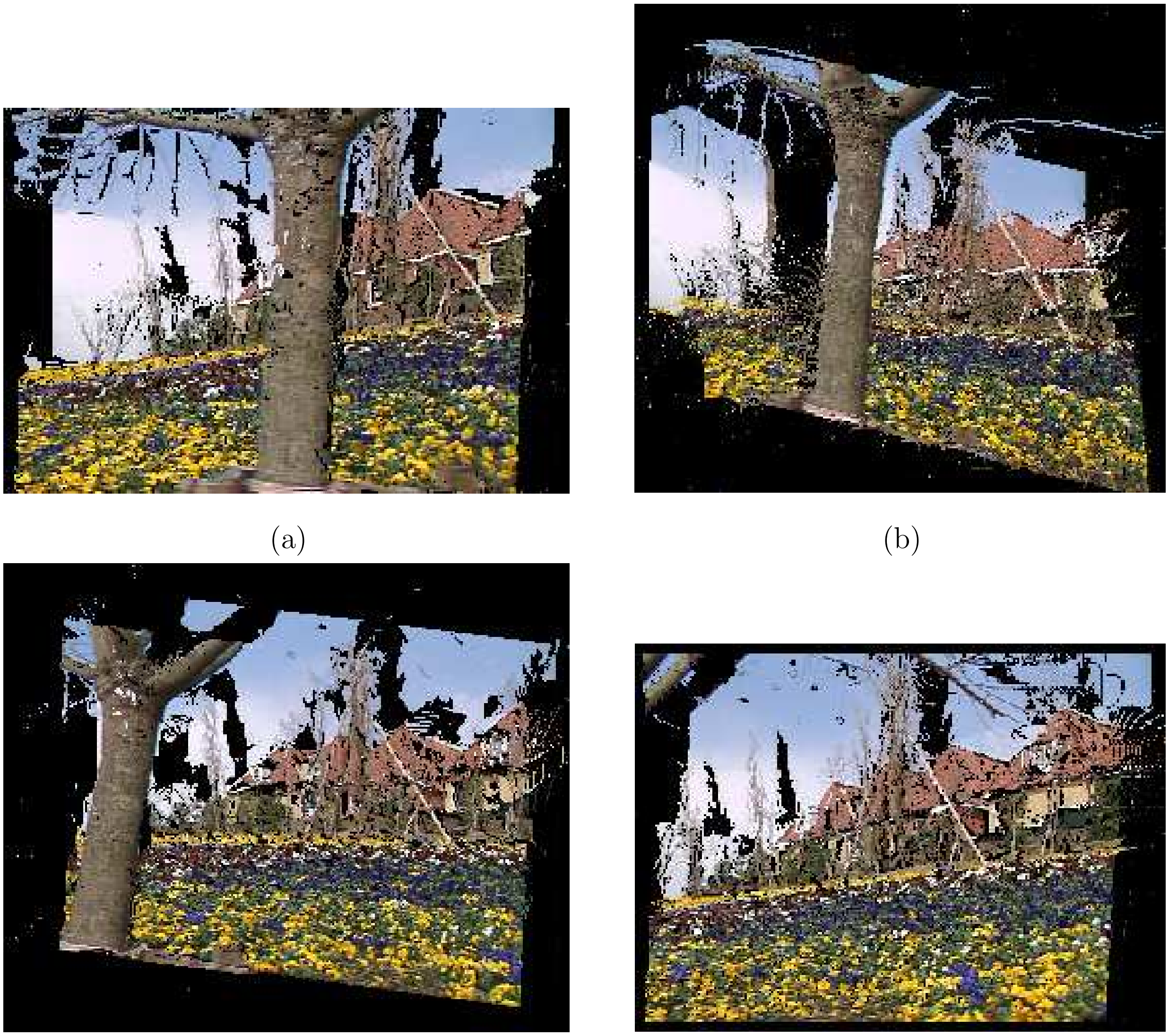
(we used $K = 60$ for the $5th$ and $65th$ images, which was obtained empirically by aligning the two views). We show in Fig. 3.32 the novel views after combining those two depth images without the foreground tree (disregard those pixels with lowest $z(x, y)$ values) in order to clearly show the fusion of the two images for those middle objects. From Fig. 3.32(a), we can see that the occluded areas in the $5th$ image (the areas behind the foreground tree) are recovered after combining with the $65th$ image, and the missing parts on the right side of the $5th$ image and on the left side of the $65th$ image are also filled into one image. A small amount of discrepancy on the right side of Fig. 3.32(a) can be seen for the right-most house. The reason for this discrepancy is due to the fact that the motion of the *Flower Garden* sequence is not strictly horizontal. There are small vertical displacements between any two consecutive images, and from the $5th$ to the $65th$ image, these vertical displacements accumulate to a significant amount. Therefore, to construct the 3D model for the whole scene, we need to extend our motion estimation algorithm to 2D if we want to accurately combine the depth images for the whole sequence, rather than assuming pure translational motion and using (3.29) only. Also, in Fig. 3.32(b) we can see occlusion areas on the background sky after a small rotation from the original viewpoint.

Finally we show in Fig. 3.33 the full fusion of the $5th$ and $65th$ images through their depth images with the $5th$ image as reference location. We can see that the tree branches on the middle top portion of the $65th$ image have moved to the very top-right part of the fused images without background, since these tree branches have lowest depth values in the $65th$ image (or largest motion values) and, while seeing from the location of the $5th$ image, the background for these tree branches should come from those images after the $65th$ image which we did not put into the fusion process. Also, from Fig. 3.33(b), we can see that there are no occlusion effects from the foreground tree after a small rotation from the original viewpoint, because those occluded areas in the $5th$ image are fused with the $65th$ image. In order to show these more clearly, We generated a video file "*garden.avi*" which contains a sequence of rendered images around the location of the $5th$ image. The file is put under the link *www.site.uottawa.ca/∼xhuang/demo*. The black holes in the video file are mainly caused by the occlusion effect of the house on the background sky, for which the information is not contained in the original $5th$ and $65th$ images.

## 3.3 Summary

In this chapter, we described a hybrid disparity estimation algorithm which combines pixel-based and region-based approaches. The novelty of our disparity estimation algorithm lies in the fact that it provides a robust method to solve some longstanding issues in disparity estimation, like the smoothness of surfaces, while keeping object boundaries sharp and clear, and the identification of occluding regions to recover their true disparities by analyzing the histograms from pixel-based and region-based approaches. These problems cannot be solved by either approach separately.

After showing the effectiveness of the combined pixel-based and region-based matching algorithm for disparity estimation, we further improved it for the detection of regions with zero displacement and applied our algorithm to a translational video sequence *Flower Garden*. The obtained disparity maps for the four image locations in this sequence have good visual quality (e.g. linear variation for slope areas, the preserving of tiny objects like those twigs), as shown in Fig. 3.29. However, after we set up the 3D models with 3D point sets based on these disparity maps in OpenGL, there are still many disturbing outliers in the rendered novel views. Although these point-based 3D models can reflect the approximate 3D locations for different objects (like the tree, the houses, etc.) in the scene, the disturbing effect caused by those outliers increases along with the increase of the rotation angles from the original viewpoints, as can be seen from the *avi* file mentioned at the end of last section. Those outliers are caused by the wrong matching results for many tiny regions which contain only several pixels. Therefore, to further improve our matching algorithm by eliminating those outliers, we will use region-merging techniques so that most of the tiny regions can be checked to see if they actually belong to the surrounding (big) regions, and if a tiny region is determined to belong to a neighboring region, then it will be merged with that neighboring region. We show our further improved disparity estimation algorithm in the next chapter.



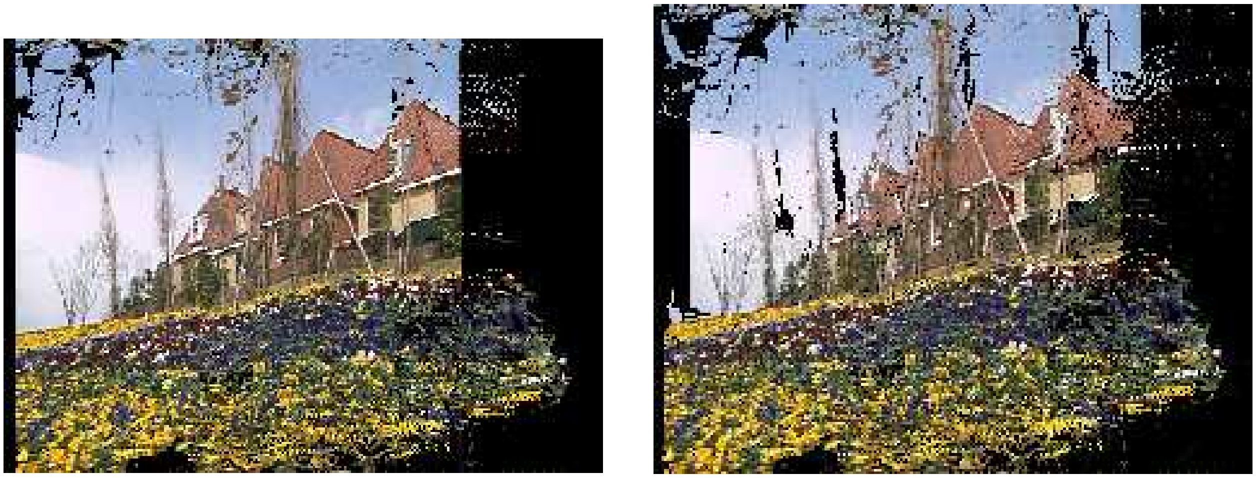
(a)

(b)

(c)

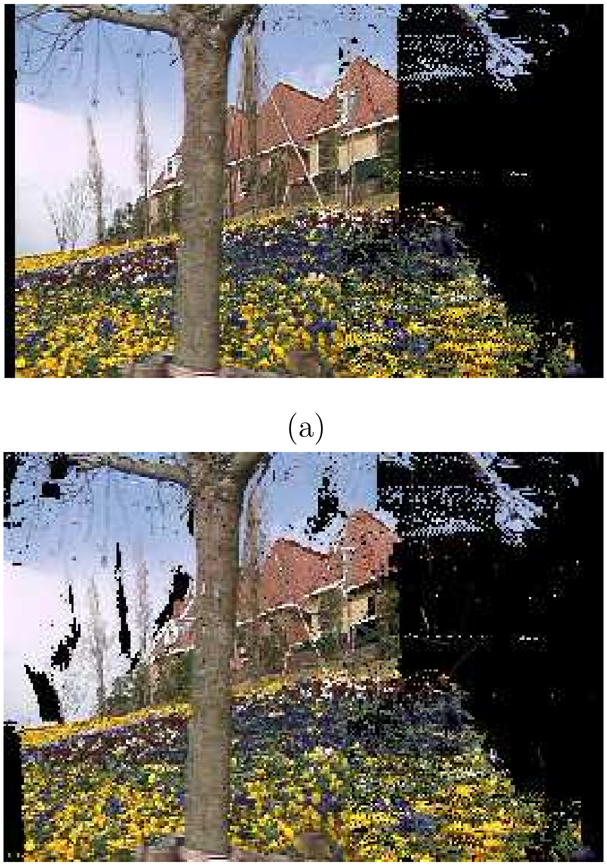
(d)

Figure 3.31: Separate reconstruction for different locations: (a) 5th; (b) 22nd; (c) 35th; (d) 65th.



(a) (b)

Figure 3.32: The fusion of $5th$ and $65th$ images without the foreground tree (with $5th$ image as reference location): (a) direct reconstruction; (b) with rotation.



(a)

(b)

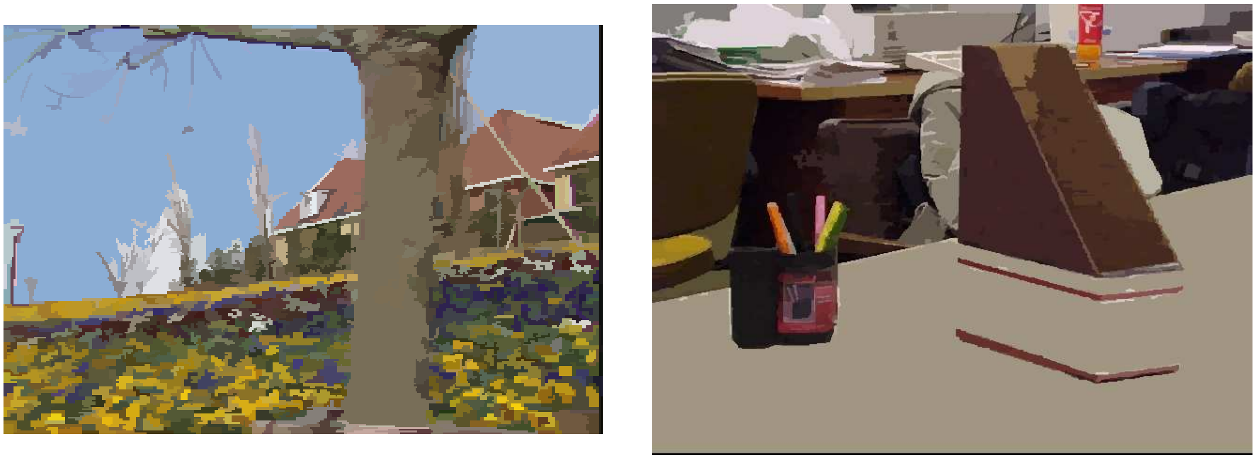
Figure 3.33: Full fusion of $5th$ and $65th$ images (with $5th$ image as reference location): (a) direct reconstruction; (b) with rotation.

# Chapter 4

# Disparity Estimation Based on a New Region Technique with 3D Modeling

In the last chapter, we showed the power of combining the region matching technique into the disparity estimation algorithm. However, the segmentation algorithm – mean shift – which we used to obtain region information in the last chapter has some defects, in that it could not identify some tiny details in the image (like the twigs in *Flower Garden*). Although we can use some edge detection and contour linking techniques to get some of the details back, the artifacts are obvious. To solve this problem, we propose to use a simple color-based region technique – quantization of image intensity values and grouping. In addition, we further have developed a disparity-based region merging scheme to improve the regions obtained in order that one surface full of texture can be identified as one region. This kind of region merging scheme can be very helpful in eliminating outliers in the final matching results.

In addition to the two image sets – *Flower Garden* and *Tsukuba* – that we used in the last chapter, we also apply our final disparity estimation algorithm described in this chapter to *VIVA Lab*.



(a)

(b)

Figure 4.1: (a)Mean shift segmentation of *Flower Garden* with a different spacial bandwidth; (b)mean shift segmentation of *VIVA Lab*.

## 4.1 Defects of Mean Shift

In the user interface of mean shift software, there are some parameters to adjust for some finer segmentation results (like “spatial bandwidth”). Fig. 4.1(a) shows the segmentation of the $0th$ image of *Flower Garden* using a smaller value for the option of “spacial bandwidth”. We can see some more twigs show out, but still not all of them. In addition to losing tiny objects, it is also easier for mean shift to merge two adjacent regions with similar colors which actually belong to two different objects. We show this point using *VIVA Lab* in Fig. 4.1(b), in which the region on the table and one region on the lower part of the box are segmented as one region.

Therefore, we can see that the mean shift segmentation algorithm tends to merge some different regions under similar colors into one region. This is not what we need. As shown in the section about region merging of this chapter, we would even like over-segmentation in the segmentation stage and then merge those over-segmented regions according to their disparity values, rather than losing some regions from the beginning. This leads to our own color-based quantization-grouping process.

## 4.2 A Color-Based Region Technique

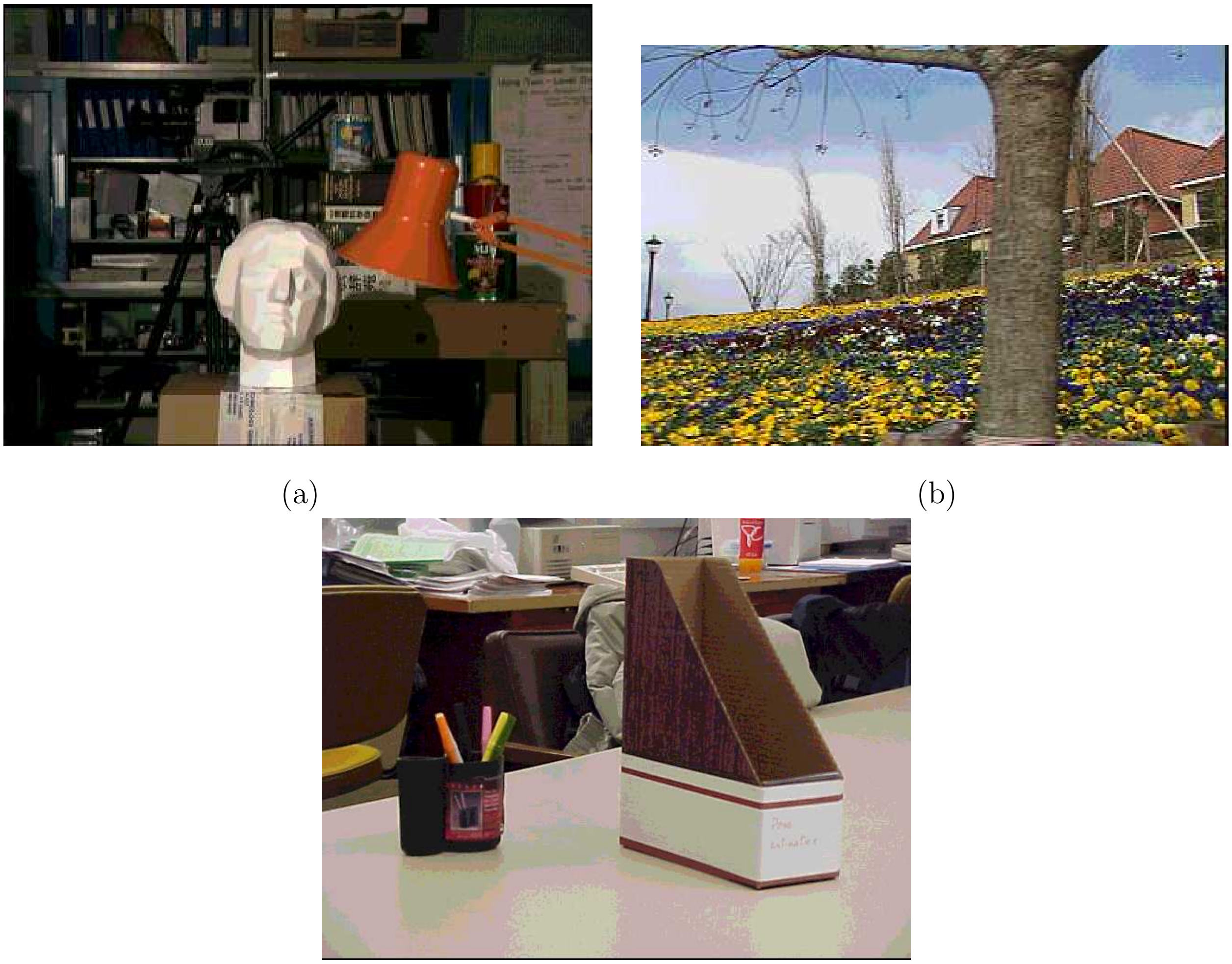
Our color-based region technique starts with the coarse quantization of the three color ($RGB$) values of the image. Assuming each component of the three color values $I(x, y)$ lies in the range of $[0, 255]$, then the quantized image $I_Q$ for each color component is obtained as

$$I_Q(x, y) = 25\text{round}[I(x, y)/25] \tag{4.1}$$

where round($X$) rounds $X$ to its nearest integer. The value 25 for the quantization step was determined empirically. The quantized images for the $0th$ images of *Tsukuba*, *Flower Garden* and *VIVA Lab* are shown in Fig. 4.2. Then the pixels with all three *R*, *G* and *B* quantized values the same and being adjacent to each other are grouped together to form a region. Although there might be more regions after such quantization-grouping process than that of the mean shift (especially for images like *Flower Garden*), as can be seen from the tree area in Fig. 4.2(b) comparing with Fig. 3.13(b), all tiny details like the twigs in *Flower Garden* are retained since their contrast with the sky background is larger than 25.

## 4.3 Region Manipulation and Matching Based on Our Region Technique

Similar to our method in Chapter 3, our new disparity estimation process based on our region technique is shown in Fig. 4.3 in which the quantization-grouping process is first applied to image $I_t$, along with the filtering of $I_t$ and $I_{t+1}$ using Gabor filters. Based on the Gabor filtering results, a coarse 1-D disparity map $d_G$ is obtained. Another disparity map $d_R$ based on variational regularization using an edge-preserving functional is also estimated iteratively with motion values for each pixel initialized with zero. Then the histograms of disparity values from $d_G$ and $d_R$ in each region of $I_t$ (obtained from the quantization-grouping process) are compared in order to identify those regions without movements (zero displacement). The regions detected as zero displacement for the $0th$ image of *Flower Garden* are shown in Fig. 4.4. We have no images of zero displacement to be shown for *Tsukuba* and *VIVA Lab*



(a) (b)

(c)

Figure 4.2: Image $I_Q$ after quantization: (a) *Tsukuba*; (b)*Flower Garden*; (c) *VIVA Lab.*

since our algorithm did not detect any zero displacement regions in these two images, which is a correct result because all pixels in the indoor environment of *Tsukuba* and *VIVA Lab* should have disparity values larger than zero.

Until now, these procedures in the matching process are similar to our disparity estimation algorithm applied to multiview image sequence in section 3.2 of Chapter 3. Then, before entering into the region matching stage, we further improved the shape of regions by developing a region merging scheme based on the Gabor matching result $d_G$. In such region merging, we merge the regions obtained from the quantization-grouping process that are

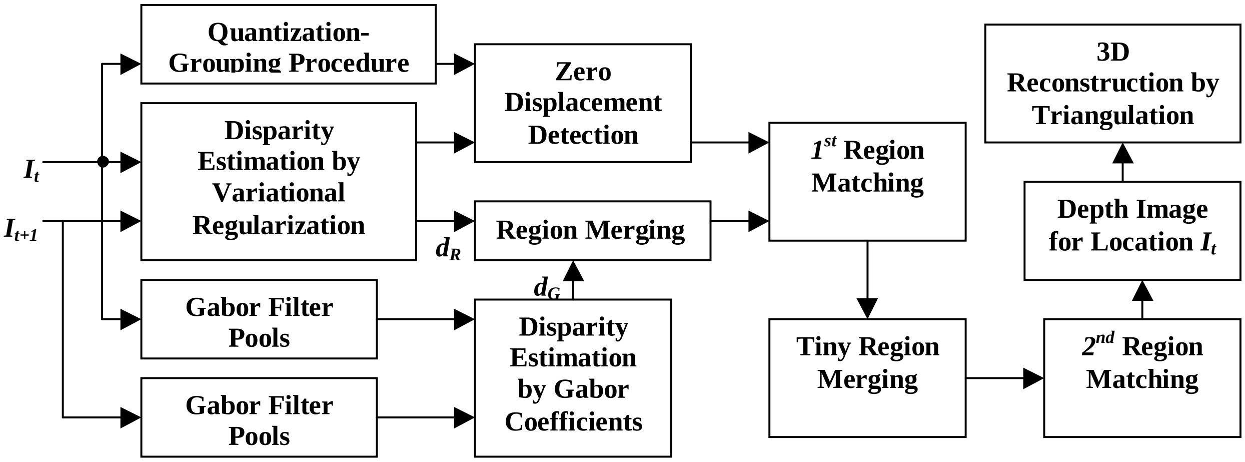


Figure 4.3: Block architecture for disparity estimation

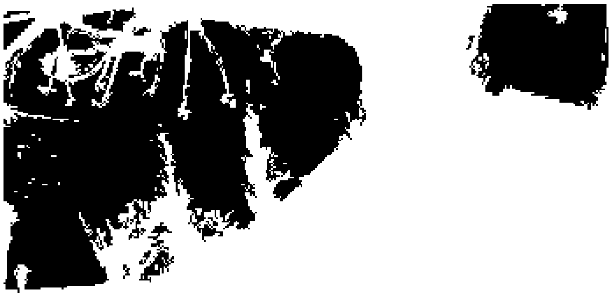


Figure 4.4: Region with zero displacement value (in black) detected for *Garden0*

adjacent to each other and have the same disparity values under $d_G$. The "same disparity values" is defined as follows: if we have adjacent region A and region B and more than 90% disparity values of region A (under $d_G$) is value A, and more than 90% disparity values of region B (under $d_G$) is value B, then region A and region B have "same disparity values" if value A = value B. This kind of merging technique is especially useful in identifying a surface full of texture on it, e.g., the slope surface full of flowers in *Flower Garden*. A direct benefit from such region merging is the reduction of outliers in the final matching results, since some tiny regions before region merging have less reliable matching results than that of the larger regions, and such region merging can just merge such tiny regions into larger regions.

In addition to checking the disparity values under $d_G$, we also check the difference of color values between adjacent regions. If the mean absolute difference of color values between two adjacent regions is under a threshold (we use 0.1, which was determined empirically), then these two adjacent regions will also be merged together.

Here we would like to state that although the block diagram in Fig. 4.3 was originally developed using the *Flower Garden* dataset, it is also compatible with the other image sets that we used in this thesis. As shown above, the same procedure for the detection of zero displacement regions did not find any zero displacement regions in *Tsukuba* and *VIVA Lab* (a correct result), while the region merging scheme can also reduce the outliers for the final disparity maps of *Tsukuba* and *VIVA Lab*.

The disparity maps $d_G$ for the $0th$ images of *Tsukuba*, *Flower Garden* and *VIVA Lab* are shown in Fig. 4.5 (it is the same for *VIVA Lab* as in Fig. 3.3), from which we can see that most of the slope areas with flowers in *Flower Garden* are under several large masks in $d_G$. Then, we merge the regions under the same masks in $d_G$, as well as merge the adjacent regions between which the mean absolute color difference is within a threshold. After these region merging, we apply the same region matching technique as in Chapter 3 to all the merged regions except for the regions determined as zero-displacement region, and the results are shown in Fig. 4.6, in which we got some basically desired purposes like the linear variation of depth values for the slope areas in *Flower Garden*, though there are still outliers from

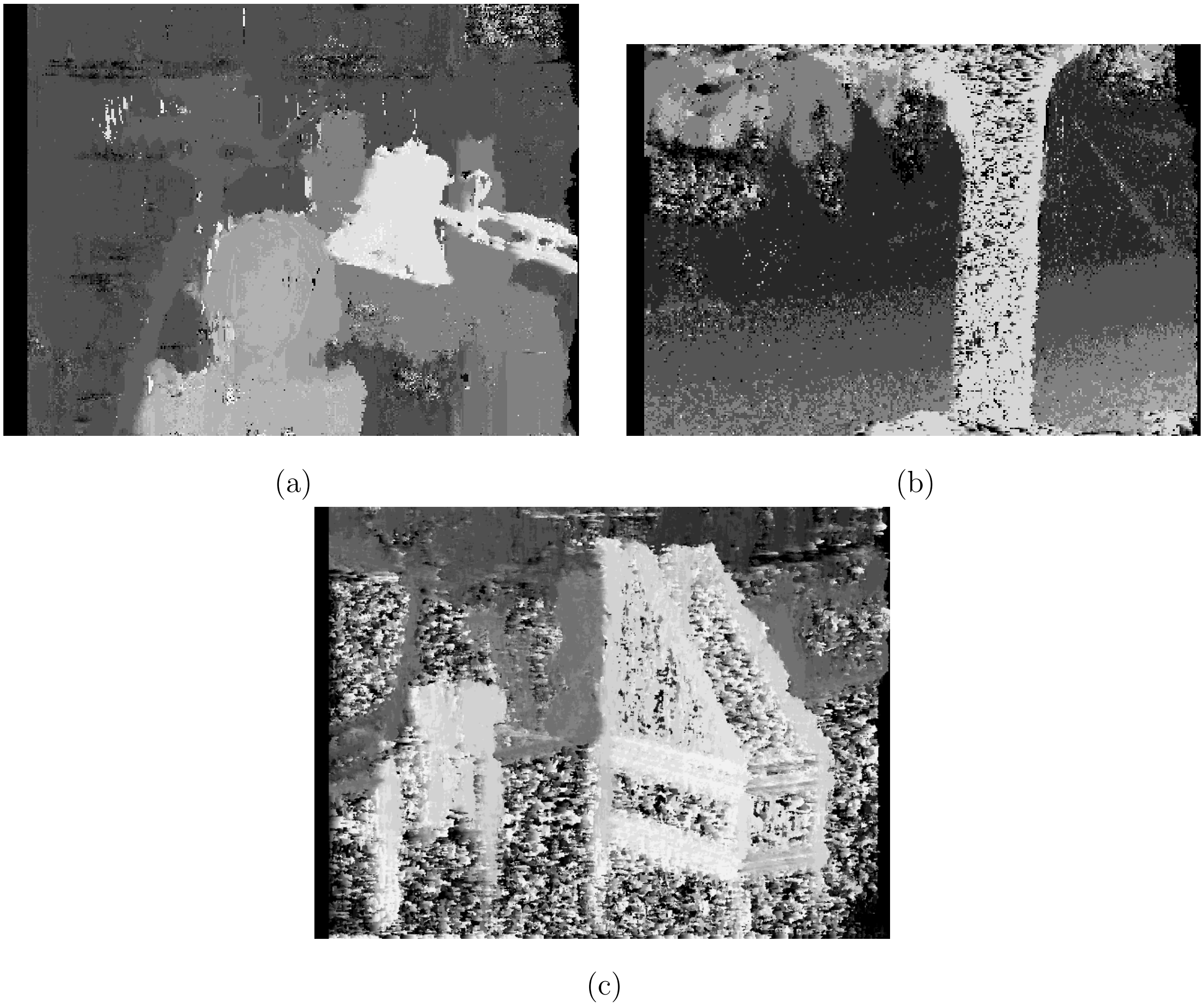


Figure 4.5: Gabor matching results $d_G$: (a) *Tsukuba0*; (b) *Garden0*; (c) *VIVA Lab.*

the false matching results for some tiny regions. To alleviate such outliers, we found that a large part of such tiny regions contain less than ten pixels and have disparity values out of the range after the region matching, and this allow us to detect them and merge them with surrounding regions. After this, a second round of region matching is applied to all the regions including the newly merged regions, and the results are shown in Fig. 4.7, from which we can see that many outliers are removed, and the visual quality for *Flower Garden* is similar to that of the method based the mean shift segmentation, while for *Flower Garden* it is better than mean shift-based method from keeping tiny features point-of-view. More importantly, we also have region information associated with the depth maps, and this will allow us to set up 3D model by triangulations.

## 4.4 Matching Results Based on SSD

In addition to matching results in Fig. 4.7 which is based on Gabor filtering, we also apply the same procedure to the initial matching results from SSD. The whole procedure, shown in Fig. 4.8, is similar to the one shown in Fig. 4.3 in which the Gabor filtering and $d_G$ are replaced with SSD and $d_{SSD}$.

The coarse disparity maps estimated by SSD for the $0th$ images of *Tsukuba*, *Flower Garden* and *VIVA Lab* are shown in Fig. 4.9 (it is the same for *VIVA Lab* as in Fig. 2.8), from which we can see that, as indicated in Chapter 2, the SSD method brings blocking effects and contour distortions. However, the disparity maps for *Tsukuba* and *Flower Garden* in Fig. 4.9 are less noisy than in Fig. 4.5, and this will give better results for our region merging technique as well as the final disparity maps and 3D models. After the region merging, we apply the first round of region matching to all the merged regions except for the regions determined as zero-displacement region, and the results are shown in Fig. 4.10. Similar to the results shown in Fig. 4.6, we have some outliers in each disparity image in Fig. 4.10 after the first round of region matching. Therefore we apply the same tiny region merging technique as in last section, then apply the second round of region matching, and the results are shown in Fig. 4.11, from which we can see that many outliers are removed, and the visual quality for *Flower Garden* is better than the Gabor-based result as shown in Fig. 4.7(b) in which there are less outliers left on the slope and tree areas; while for *Tsukuba* and *VIVA Lab* the visual impression in Fig. 4.10(a)(c) are similar to the Gabor-based results in Fig. 4.7(a)(c).

## 4.5 3D Modeling by Triangulation

The 3D modeling and rendering that we did at the end of Chapter 3 is only achieved by putting all 3D points together without any kind of connections among these points, and this will cause the scene surfaces to split into scattered dots when zooming into the scene. To overcome this problem, we need to use some connection methods such as triangular meshes to represent each scene surface, while keeping surfaces which belong to different objects disconnected. Our region information just meets this requirement.

We first detect the border pixels of each region. Then, these border pixels are put into the process of Delaunay triangulation, and the resulting triangular mesh will represent the surface covered by this region. For example, assume we have a region surrounded by border pixels which are represented using little circles in Fig. 4.12(a), and the depth of this region is zero. Then, after Delaunay triangulation, we have a triangular mesh covering this region as shown in Fig. 4.12(b). The code that we used for the Delaunay triangulation can be found at *http://cm.bell-labs.com/netlib/voronoi/triangle.zip*, and adapted into our code with interface on the data structure of border pixels. After triangulation, all the obtained triangles are used to set up the 3D model with the depth information associated with their vertices using OpenGL, and the related texture information for each triangle is put onto that triangle by texture mapping.

### 4.5.1 Gabor-Based Results

The rendered scenes based on the estimated disparity maps in Fig. 4.7 for the $0th$ images of *Flower Garden*, *Tsukuba* are shown in Fig. 4.13 and Fig. 4.14 respectively.

The rendered images have black holes and lines, even at the original positions, like the slopes in *Flower Garden* and the head face in *Tsukuba*. This is because some surfaces are still split even after region merging, and the matching results for these surfaces are different which make their positions in 3D space have gaps. Another thing to be noticed here is, for *Flower Garden*, we did not apply triangulation to the sky. We will do that at the end stage in the next chapter. From these rendered images, and comparing to the rendering results in the last chapter for *Flower Garden*, we can see that although many outliers are eliminated by region merging techniques, which makes the triangulation applicable to the scene surfaces, there are still some outliers left due to the incompleteness of region merging. These outliers could make the visual quality of rendered images at novel positions very disturbing, especially after triangulation is applied.

Two rendered images for *VIVA Lab* are shown in Fig. 4.15. From the table areas in these two images, we can find that they did not show the linear variation property for the slanted surface of the table. This is due to the reason that the image information contained in the original image set of *VIVA Lab* is not sufficient for our disparity estimation algorithm to determine the untextured table as a slanted surface. Because the table extends to the very bottom of the image, so our disparity estimation algorithm could not determine whether this untextured surface is a slanted one or, e.g., a vertical one.

### 4.5.2 SSD-Based Results

The rendered scenes based on the estimated disparity maps in Fig. 4.11 for the $0th$ images of *Flower Garden*, *Tsukuba* are shown in Fig. 4.16 and Fig. 4.17 respectively.

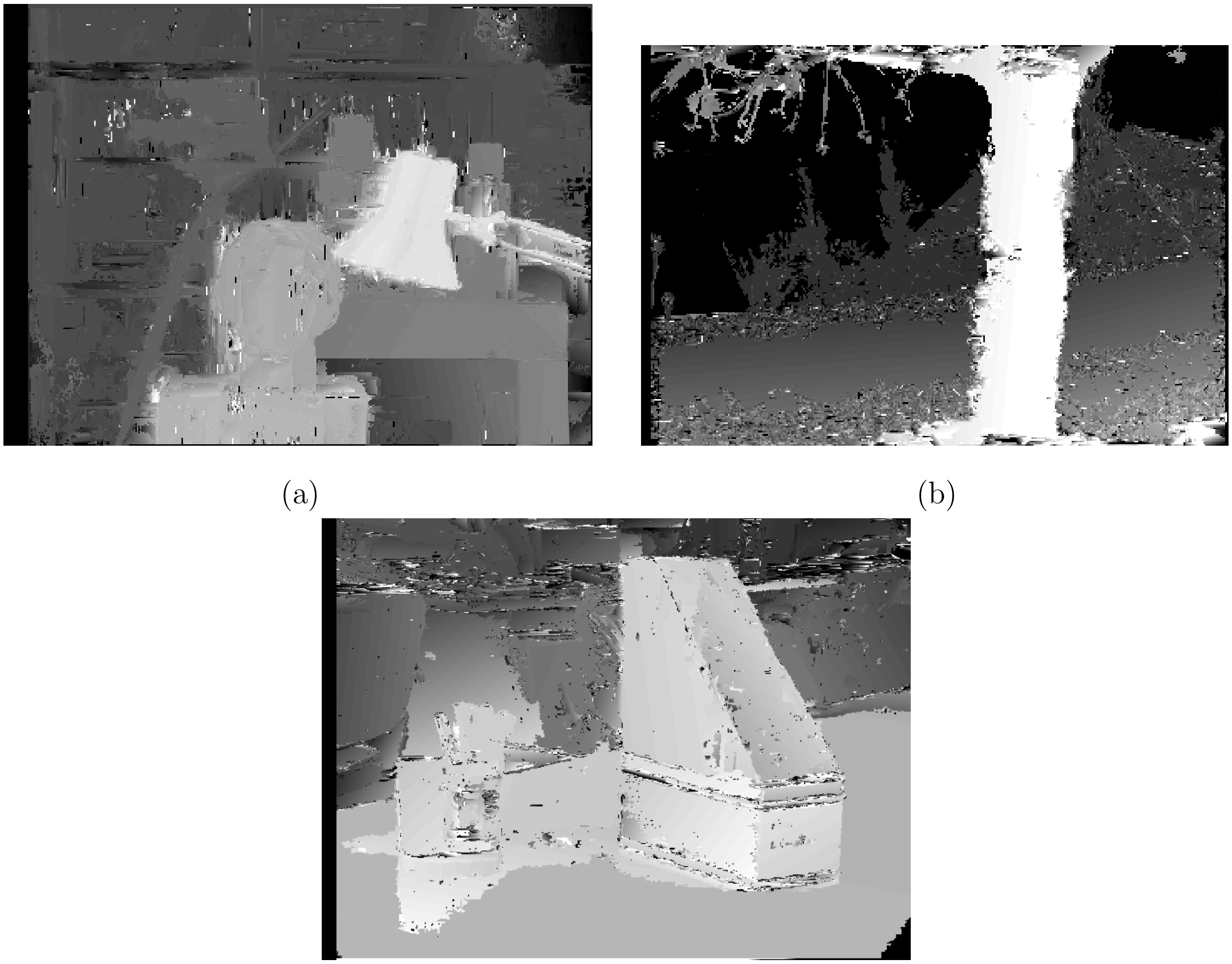
Two rendered images for *VIVA Lab* are shown in Fig. 4.18. Again, like the results in Gabor-based method for *VIVA Lab*, the table area did not show the necessary linear variation property.

## 4.6 Summary

This chapter shows a new set of region manipulation and matching algorithm based on our new color- and quantization-based region manipulation process. The final matching results show some improvements over the results in Chapter 3 which are based on the mean shift segmentation from the point of view of maintaining small features. In addition, the region information that we obtained can be used for the triangulation in the 3D model set up.

From the rendered images of *Tsukuba* and *Flower Garden*, we find that the SSD-based approach is better than Gabor-based approach since the original coarse disparity from SSD gives better information for region merging process. Although in the coarse disparity estimated by SSD we can see the distortion for some object contours, our color- and quantization-based region manipulation process can repair such distortions. However, for both Gabor- and SSD-based approaches, there are still outliers left in the final estimated disparity maps which will bring separate and spur-like triangles in the triangulation process for the 3D modeling. To improve the quality of the final 3D models, we need to further eliminate such outliers, which we will do in the final process of separate 3D model integration by considering the information from other locations. This will be presented in the next chapter.

For the image set of *VIVA Lab*, since our disparity estimation algorithm could not determine the linear variation property for the untextured and slanted table surface in that image set, we will not use this image set in the next chapter.

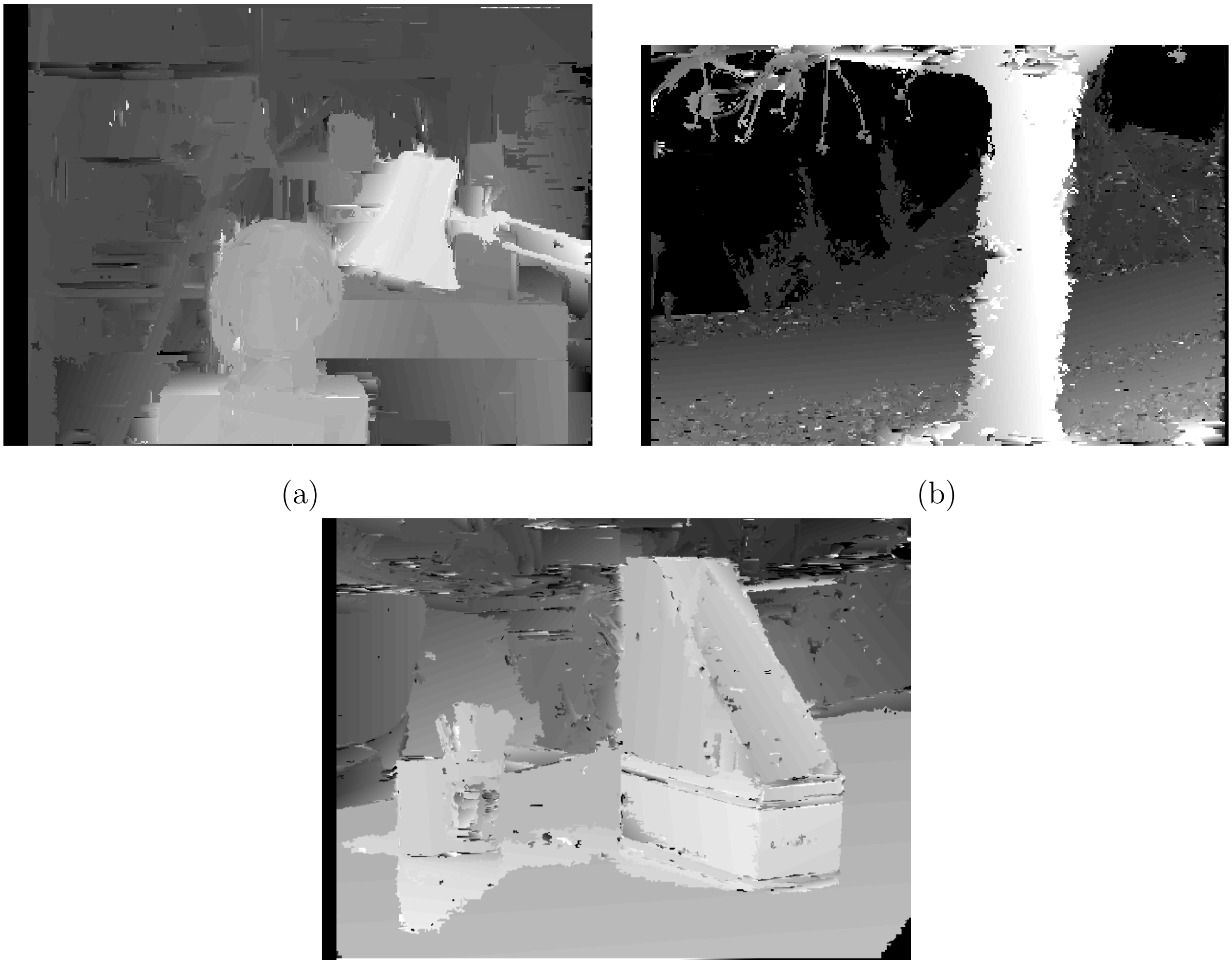


(a)

(b)

(c)

Figure 4.6: Results after first round of region matching: (a) *Tsukuba0*; (b)*Flower Garden0*; (c) *VIVA Lab*.



(a)

(b)

(c)

Figure 4.7: Final disparity estimation results: (a) *Tsukuba0*; (b) *Flower Garden0*; (c) *VIVA Lab*.

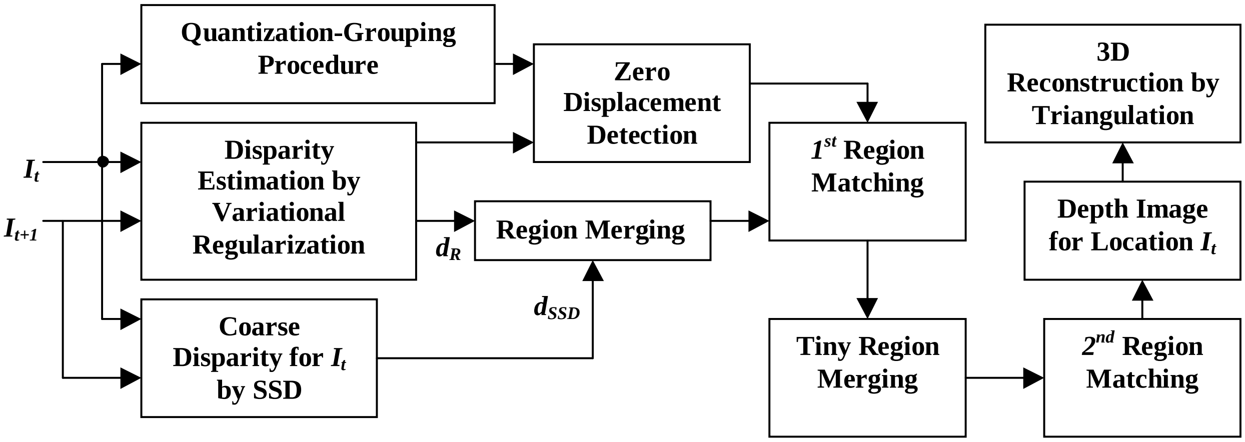
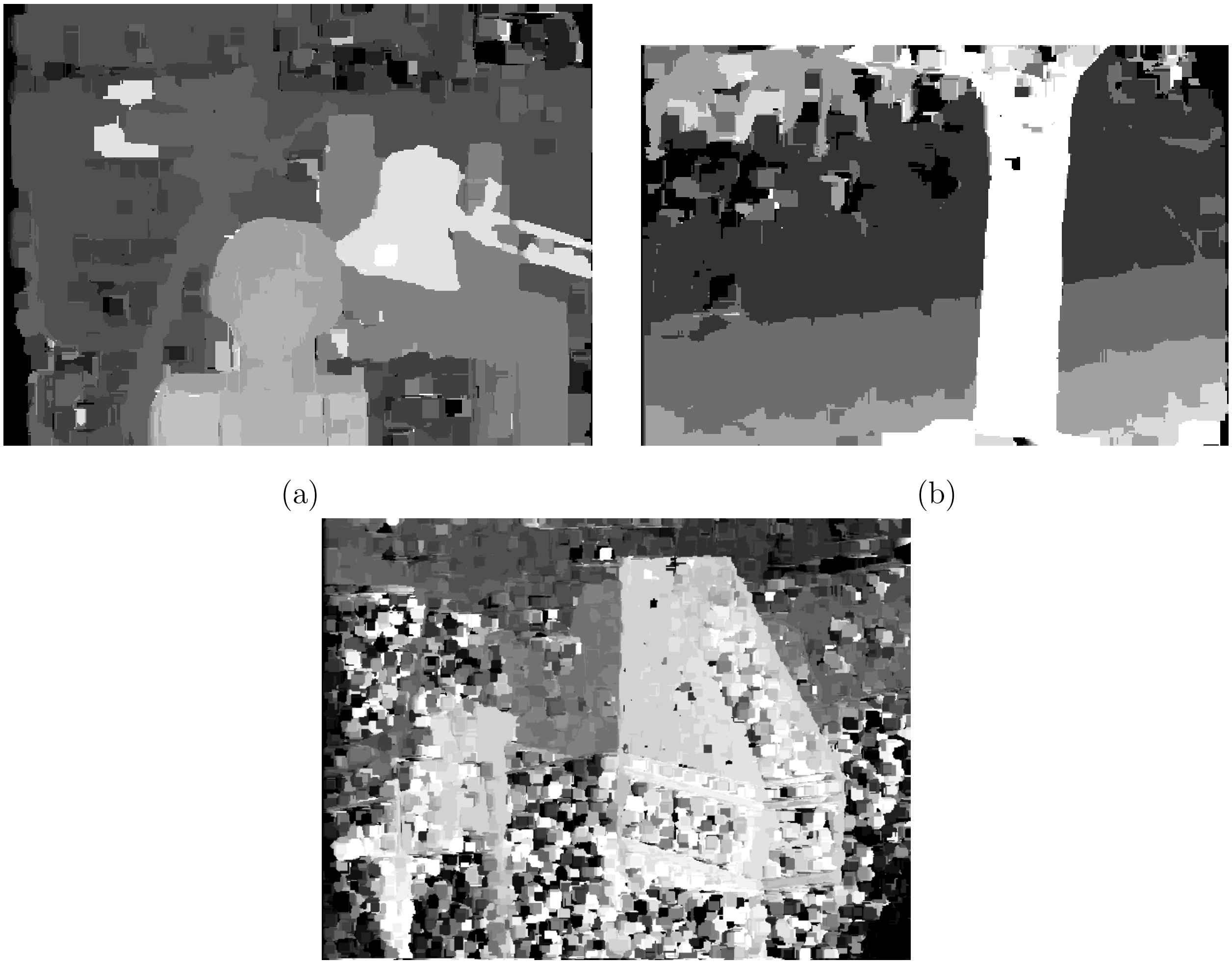

Quantization-Grouping Procedure
Zero Displacement Detection
3D Reconstruction by Triangulation
$I_t$
$I_{t+1}$
Disparity Estimation by Variational Regularization
1st Region Matching
Depth Image for Location $I_t$
$d_R$
Region Merging
$d_{SSD}$
Coarse Disparity for $I_t$ by SSD
Tiny Region Merging
2nd Region Matching

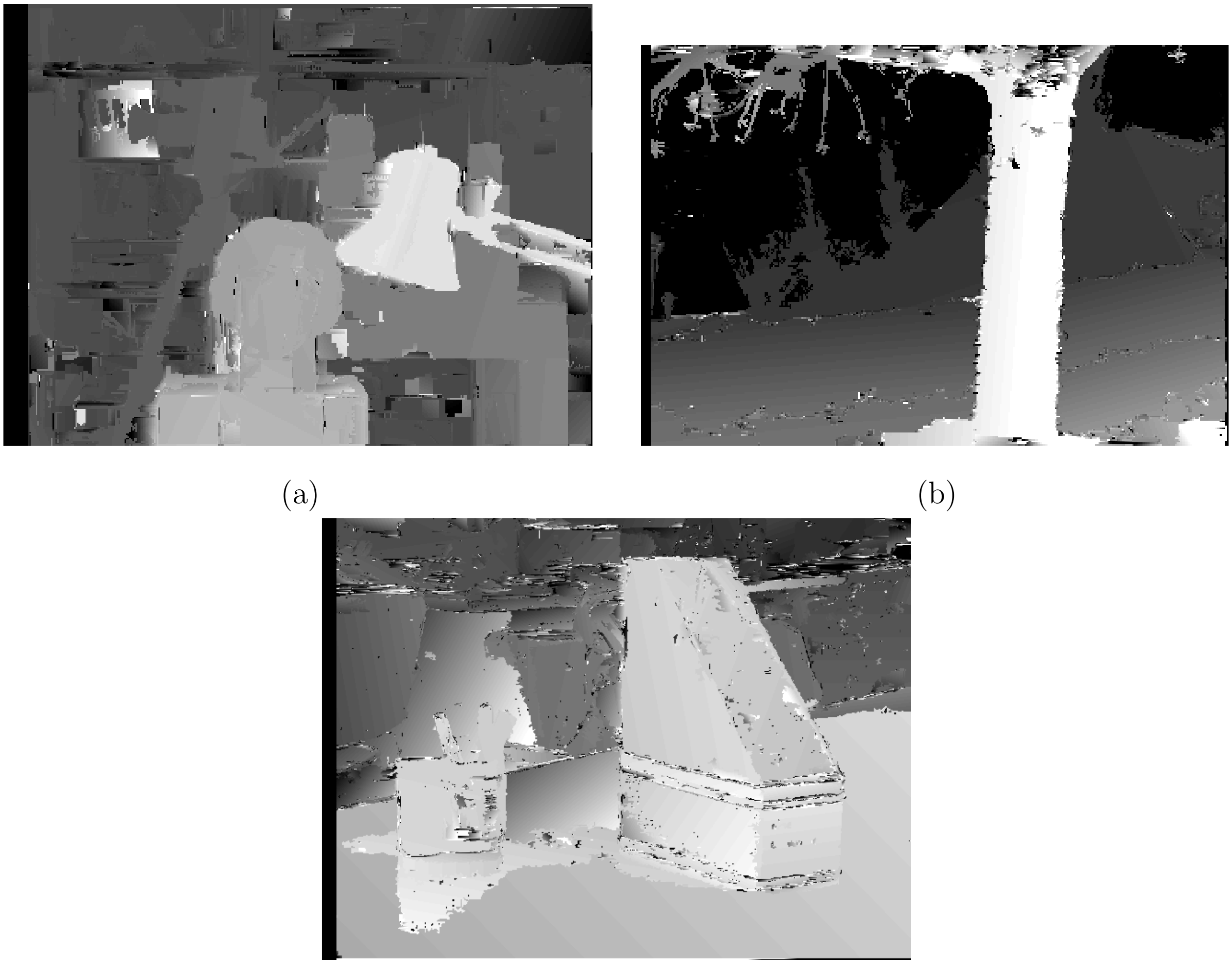
Figure 4.8: Block architecture for motion analysis



(a) (b)

(c)

Figure 4.9: SSD results $d_{SSD}$: (a) *Tsukuba0*; (b) *Garden0*; (c) *VIVA Lab*.

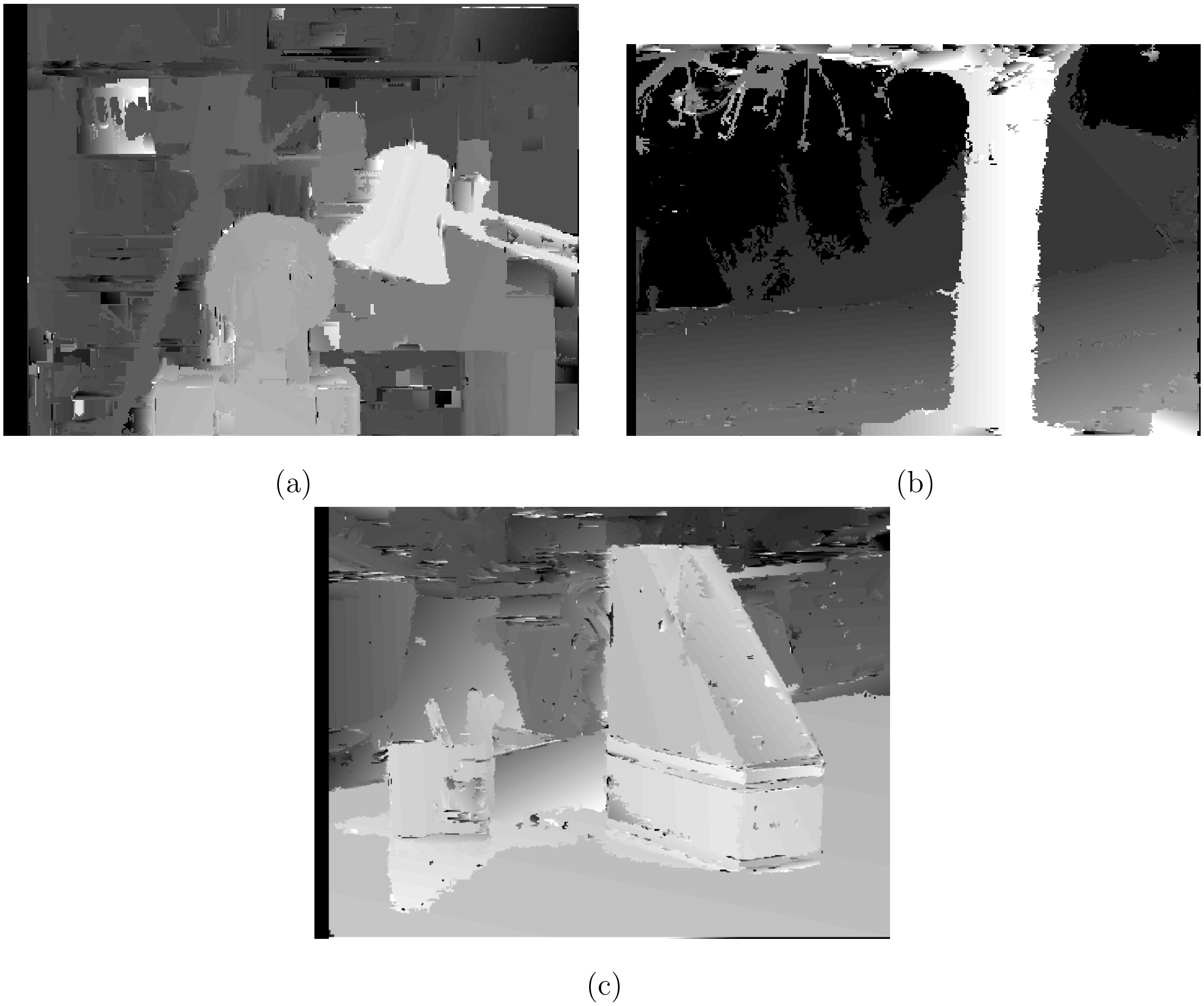


(a)

(b)

(c)

Figure 4.10: Results after first round of region matching based on $d_{SSD}$: (a) *Tsukuba0*; (b) *Flower Garden0*; (c) *VIVA Lab*.

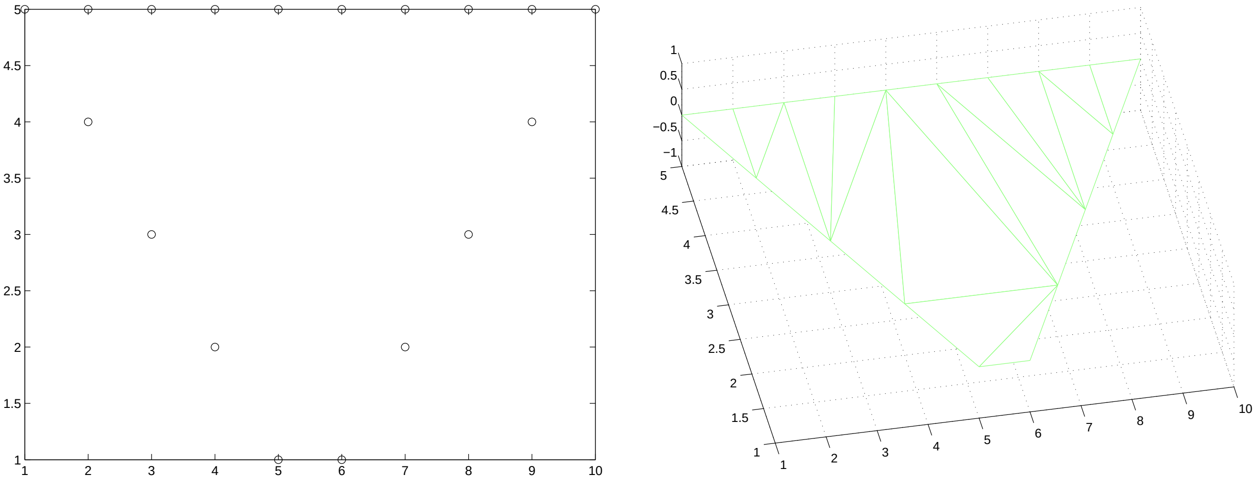


(a)

(b)

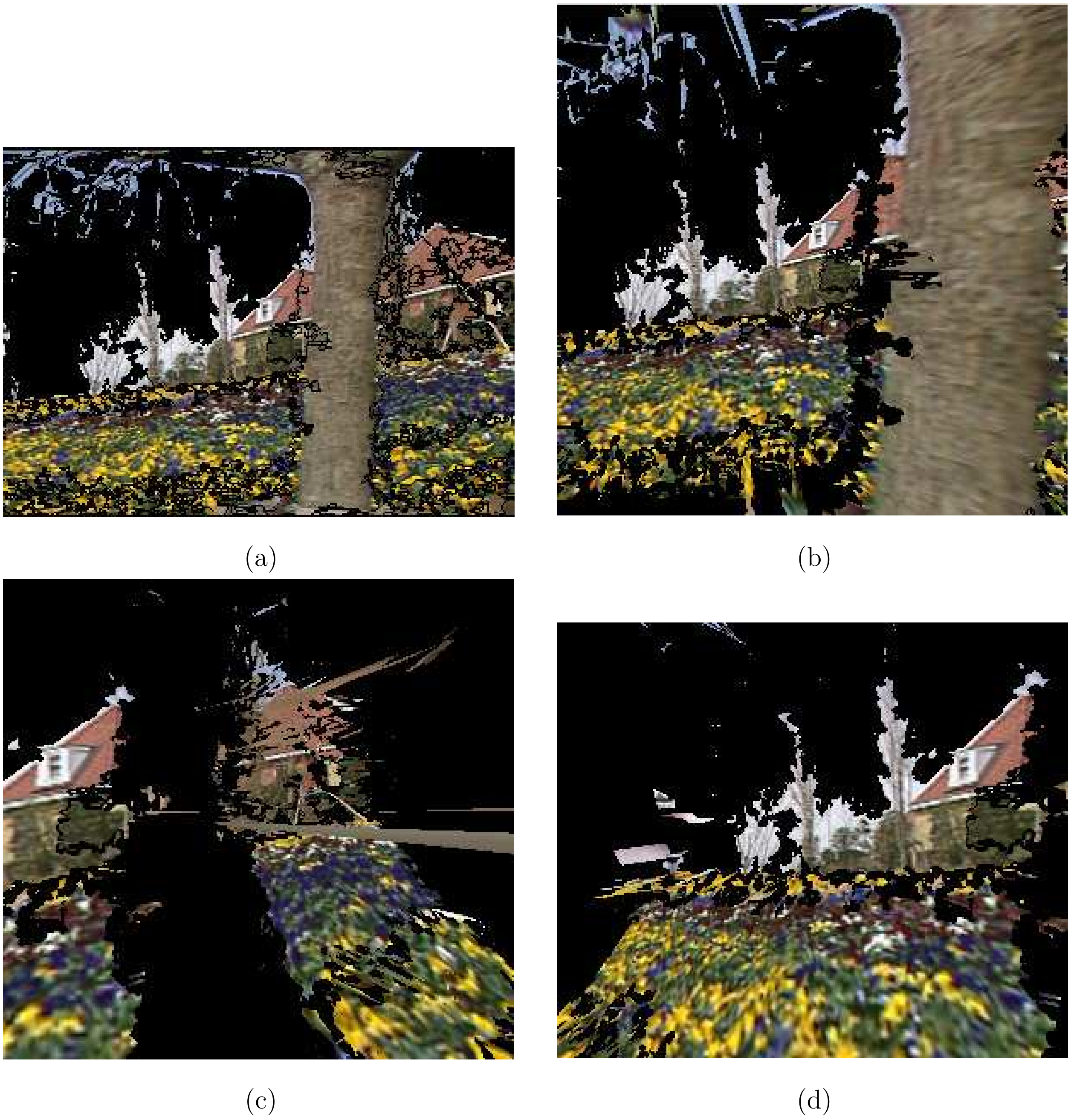
(c)

Figure 4.11: Final disparity estimation results based on SSD: (a) *Tsukuba0*; (b)*Flower Garden0*; (c) *VIVA Lab*.



(a) (b)

Figure 4.12: (a) A region surrounded by little circles (depth $z = 0$); (b) the region covered by triangular mesh after Delaunay triangulation.



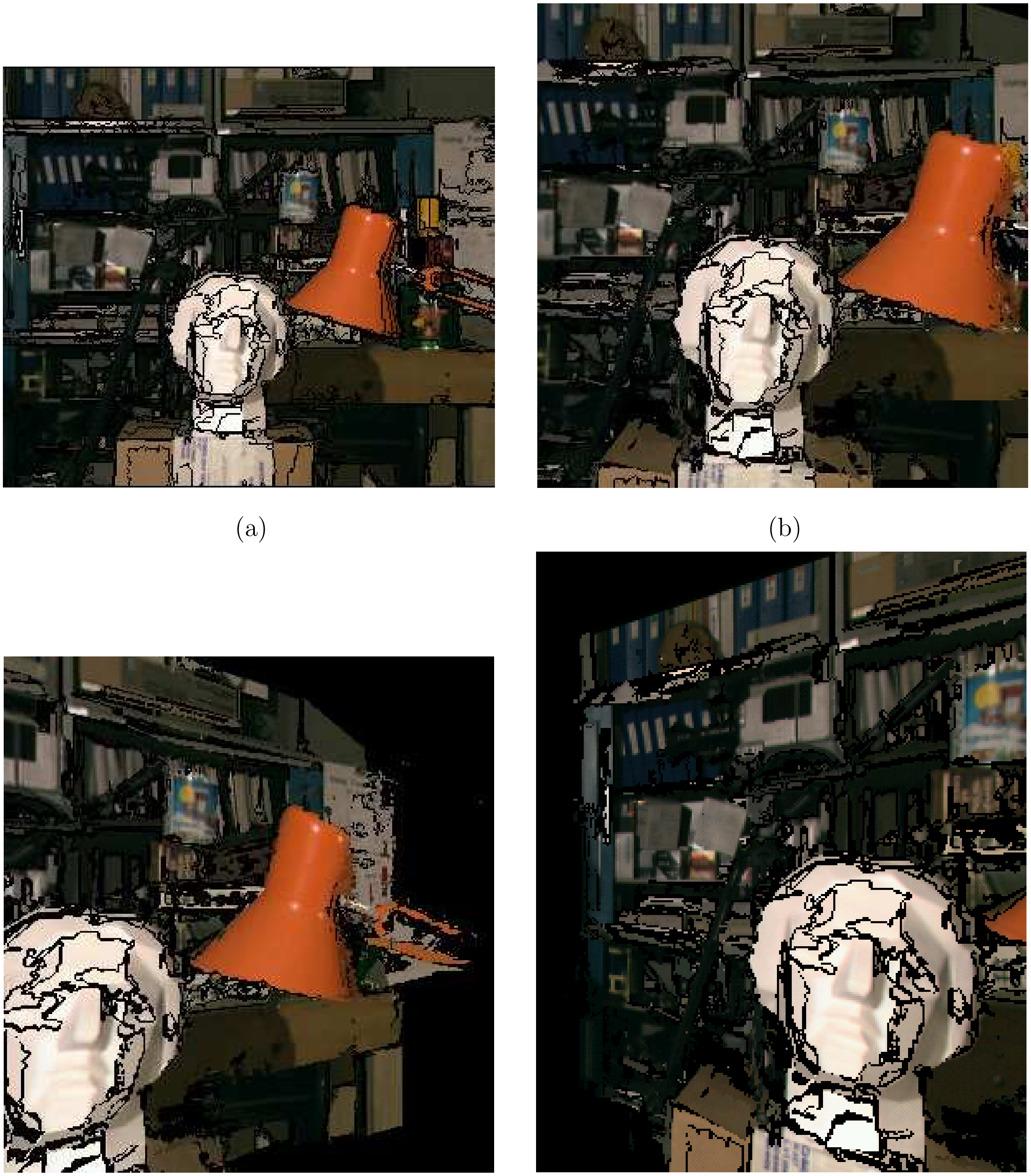
(a)

(b)

(c)

(d)

Figure 4.13: Gabor-based render results for *Garden0*: (a)original position; (b)zooming in; (c)zooming in and rotation to the right; (d)zooming in and rotation to the left.



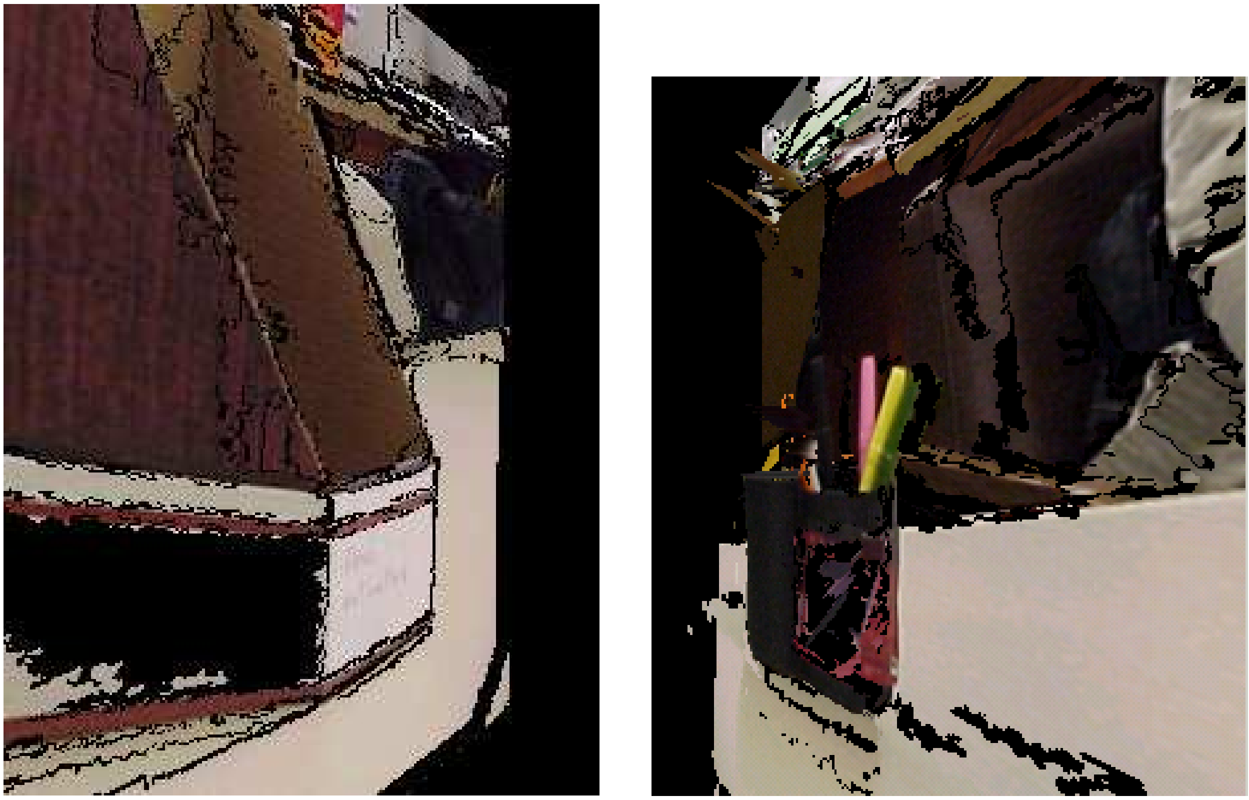
(a)

(b)

(c)

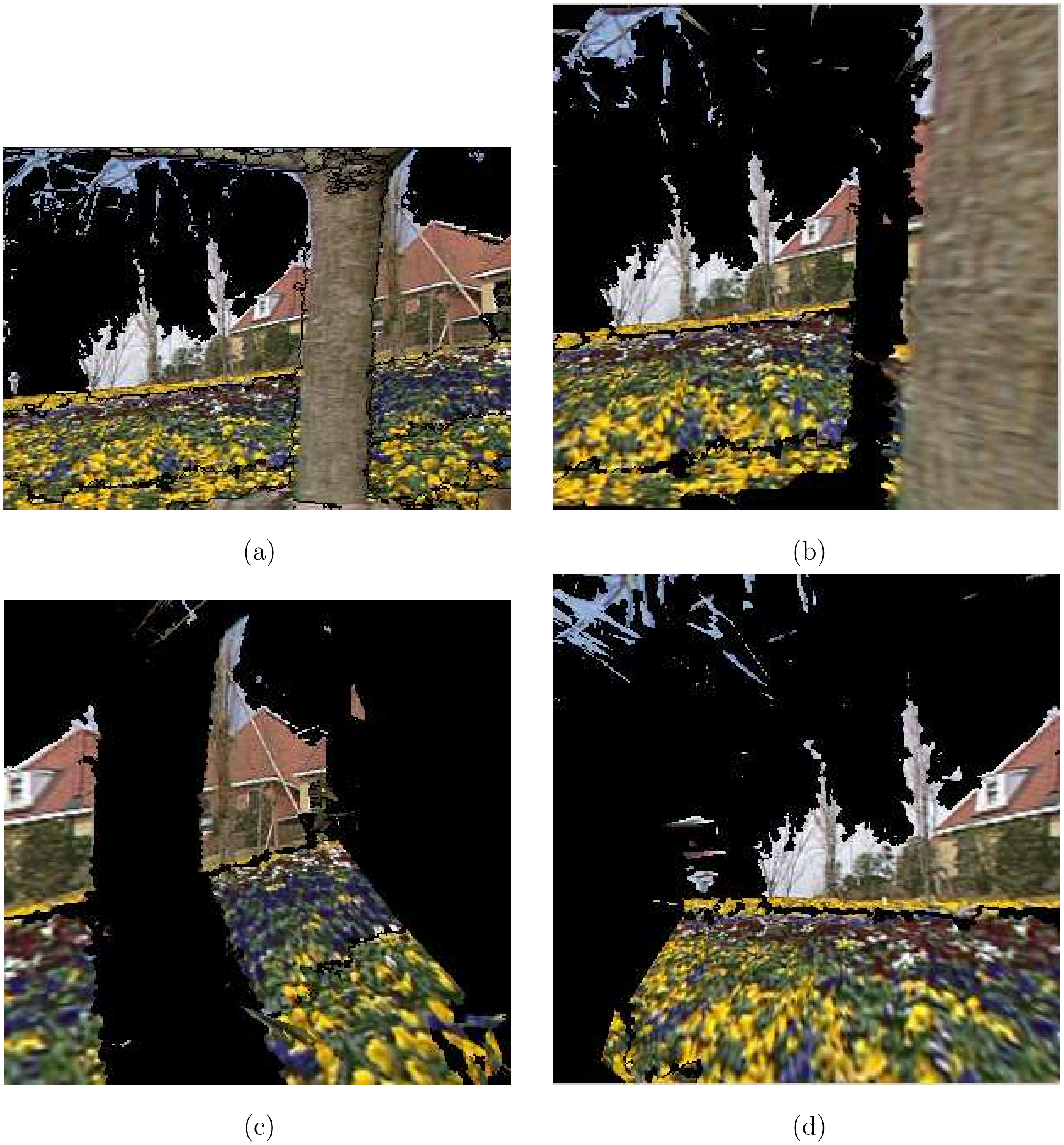
(d)

Figure 4.14: Gabor-based render results for *Tsukuba0*: (a)original position; (b)zooming in; (c)zooming in and rotation to the right; (d)zooming in and rotation to the left.



(a) (b)

Figure 4.15: Gabor-based render results for *VIVA Lab*: (a)zooming in and rotation to the right; (b)zooming in and rotation to the left.



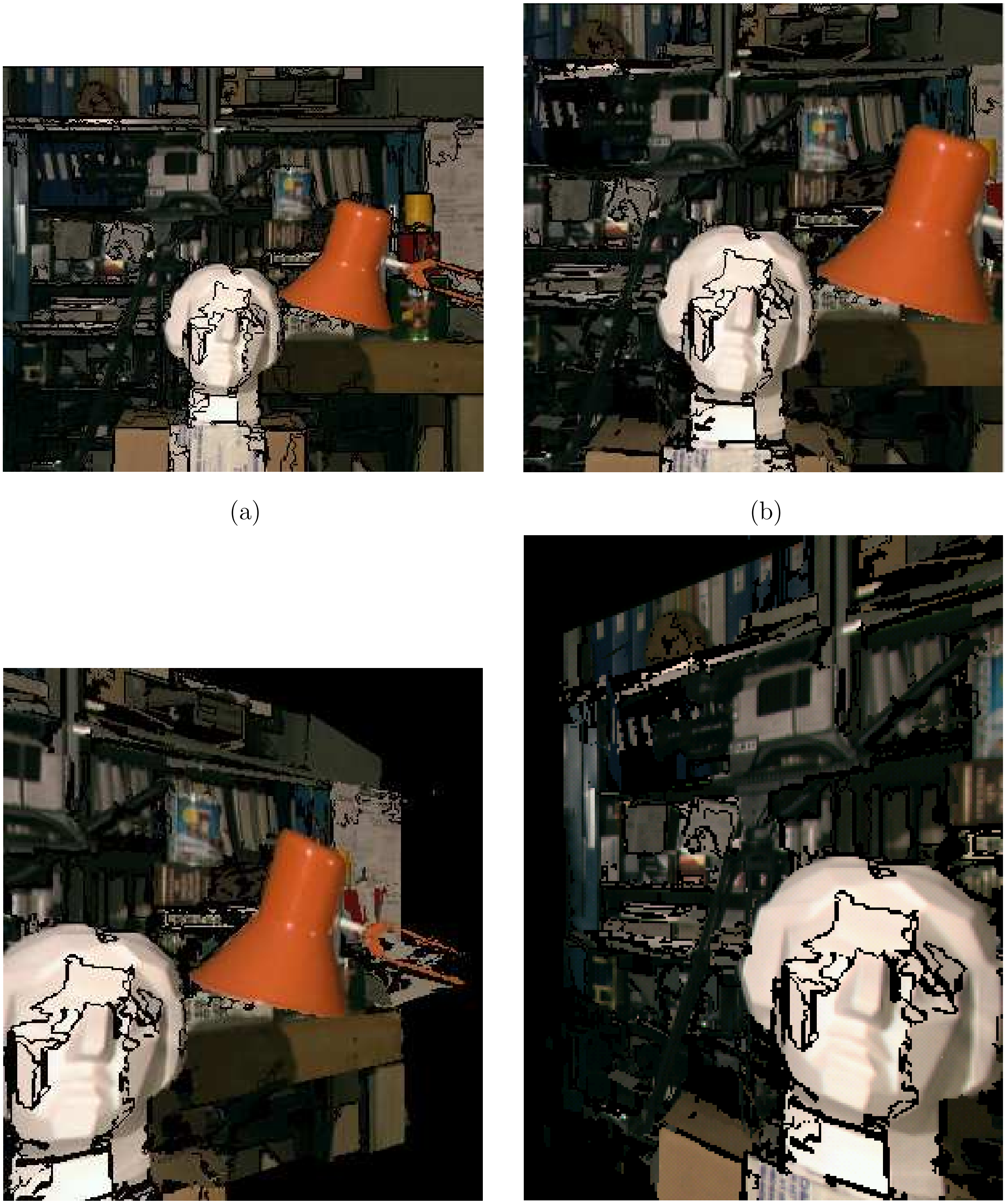
(a)

(b)

(c)

(d)

Figure 4.16: SSD-based render results for *Garden0*: (a)original position; (b)zooming in; (c)zooming in and rotation to the right; (d)zooming in and rotation to the left.



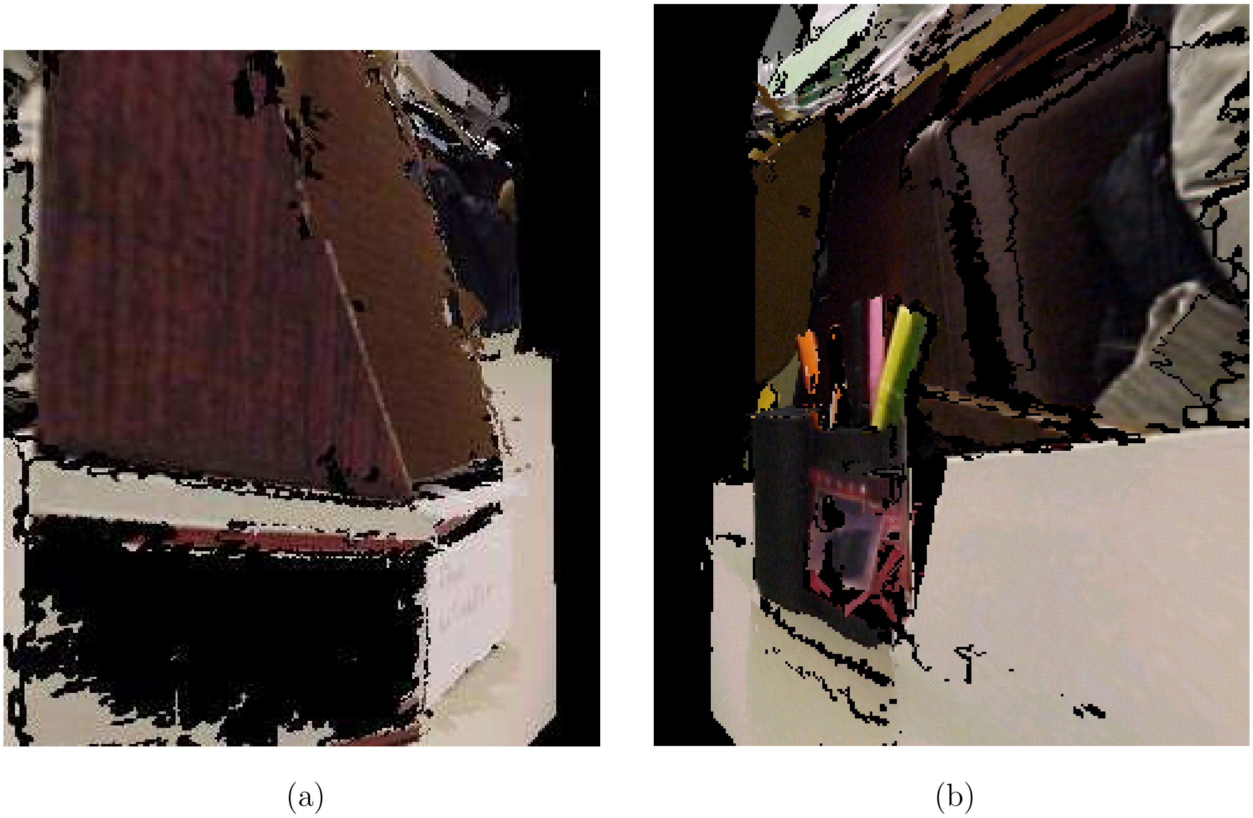
(a)

(b)

(c)

(d)

Figure 4.17: SSD-based render results for *Tsukuba0*: (a)original position; (b)zooming in; (c)zooming in and rotation to the right; (d)zooming in and rotation to the left.



(a) (b)

Figure 4.18: SSD-based render results for *VIVA Lab*: (a)zooming in and rotation to the right; (b)zooming in and rotation to the left.

# Chapter 5

# Integration of Separate 3D Models with Final Results

Although the matching results based on our new hybrid motion estimation algorithm in the last chapter give very good visual qualities, like the retention of tiny features as well as the linear variation of depth values for slanted surfaces in *Flower Garden*, there are still some outliers in the depth maps, and these outliers in depth values result in more disturbing outlier triangles in the separate 3D models based on the matching results. Therefore, it is desirable to further reduce those outliers before a final 3D model for the whole scene can be integrated. We will try to achieve this purpose by considering the matching results from separate camera positions, after the camera motion parameters are obtained by ego-motion estimation.

The block diagram for the final stage in our 3D model integration is shown in Fig. 5.1. The whole process starts from estimating disparity (depth) maps for each image location using the same procedure as shown in the last chapter. Then, to achieve the purpose of further eliminating outliers, we do the first round of ego-motion estimation and transform the large regions from other image locations to the reference location, so that the tiny regions causing those outliers in a large region can be removed by the information of its correspondent large regions from other images. This is done by selecting all the large regions in each image and

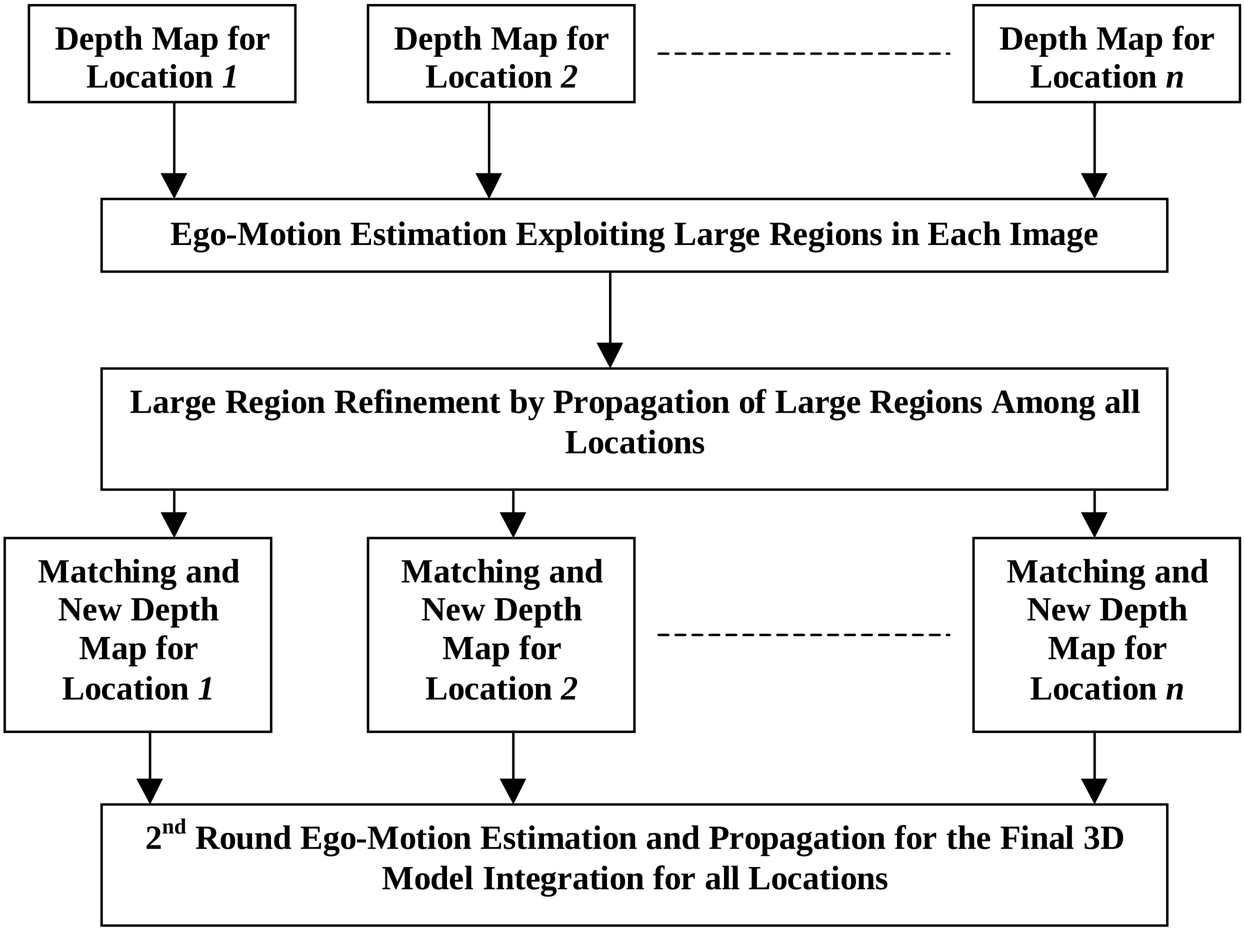

Depth Map for Location *1*
Depth Map for Location *2*
Depth Map for Location *n*
Ego-Motion Estimation Exploiting Large Regions in Each Image
Large Region Refinement by Propagation of Large Regions Among all Locations
Matching and New Depth Map for Location *1*
Matching and New Depth Map for Location *2*
Matching and New Depth Map for Location *n*
2nd Round Ego-Motion Estimation and Propagation for the Final 3D Model Integration for all Locations

Figure 5.1: Block architecture for ego-motion estimation and 3D integration

estimating the ego-motion parameters by minimizing the matching cost functions for these large regions between any pair of different image locations. Once we have all the ego-motion parameters among all the image locations, we propagate these large regions to all other locations in order to jointly refine the shape of all the large regions. After the large regions in each image location are further refined, a second matching process will be applied on *all* the pixels in that image, and this will give a further improved depth map with reduced outliers and thus a better separate 3D model for that location.

We present the whole process shown in Fig. 5.1 with results for each main step using two image sets – *Flower Garden* and *Tsukuba*. Also, for each image set, we show the results based on two different starting approaches for coarse disparity estimation – Gabor and SSD.

## 5.1 Data Sets

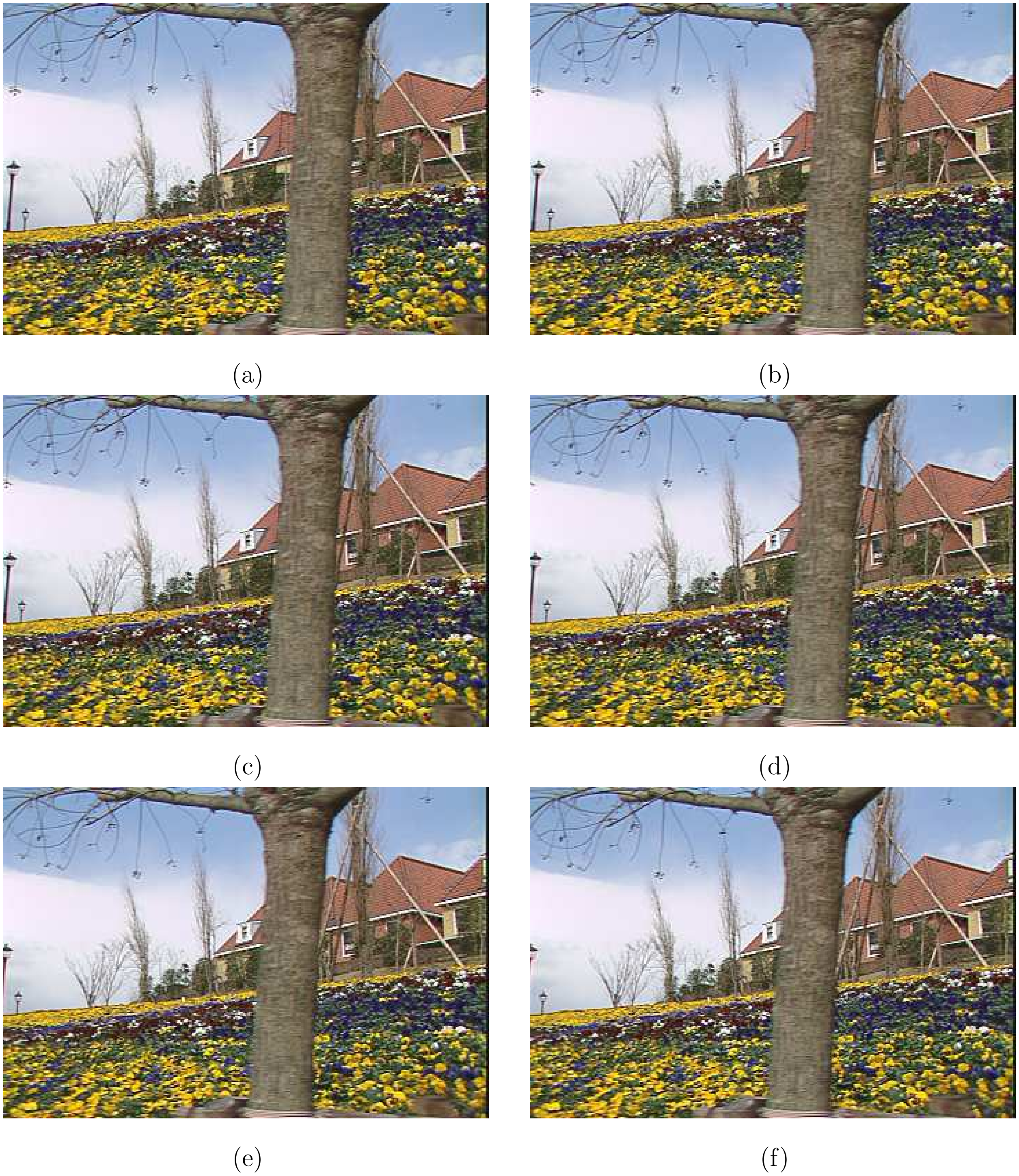
We will try to apply our motion estimation algorithm and 3D integration method on two sets of images: the first six images from the sequence *Flower Garden* (*garden0 – 5*) and the four images from *Tsukuba* (*tsukuba0 – 3*, there is also *tsukuba4* but that is the last one in the row so no 3D information can be estimated for that location, since we only estimate motion and disparity from the left image to the right image), as shown in Fig. 5.2 for *Flower Garden* and Fig. 5.3 for *Tsukuba*.

## 5.2 First Round of Disparity Estimation

As an initialization, the disparity maps for all image locations are estimated using the same disparity estimation algorithm described in Chapter 4.

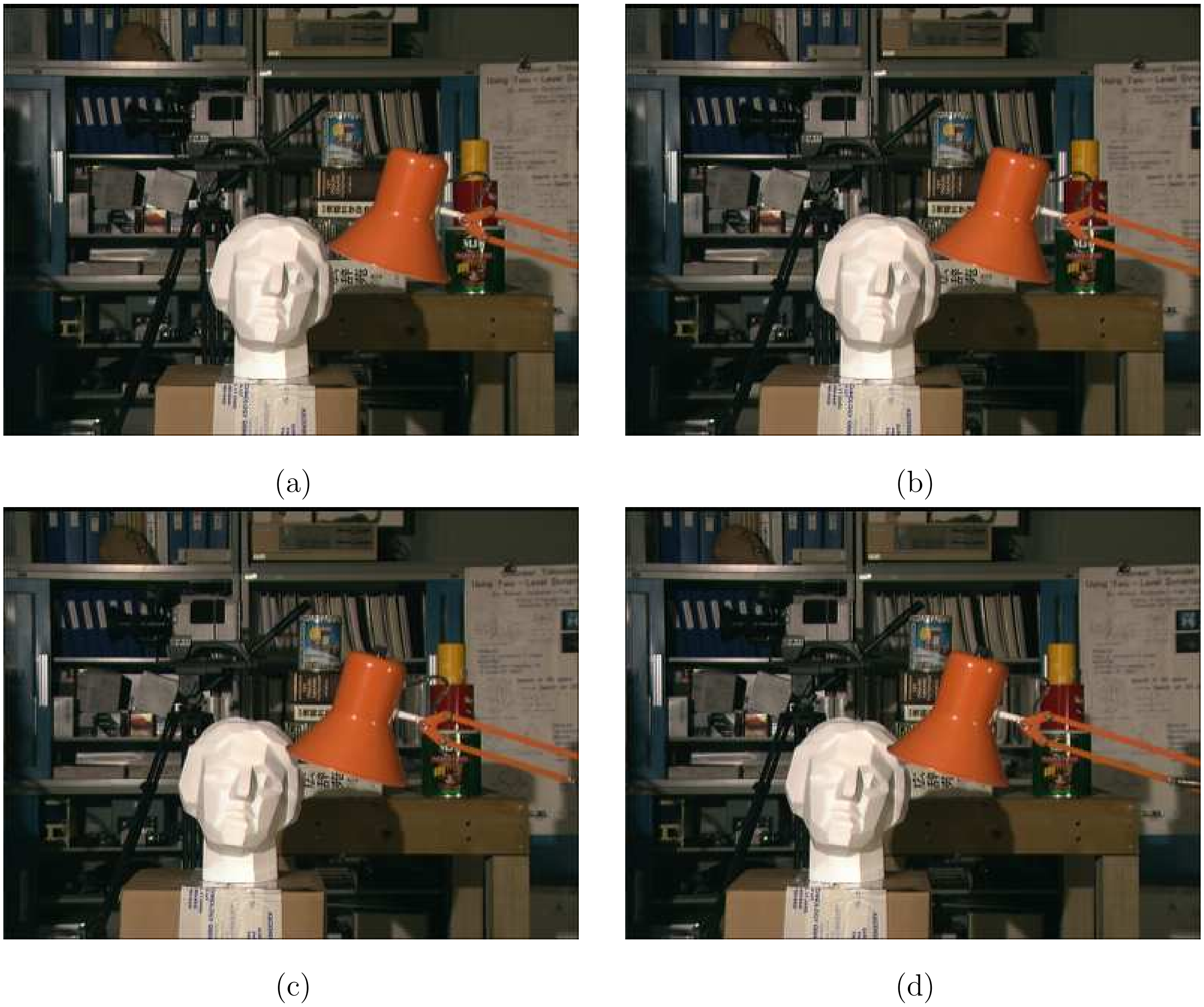
### 5.2.1 Disparity Maps from Gabor-Based and SSD-Based Approaches

The disparity maps using Gabor-based approach for all the image locations are estimated as we did to obtain the disparity maps for *tsukuba0* and *garden0* in Fig. 4.7(a)(b), and are shown in Fig. 5.4 for *Flower Garden*, and in Fig. 5.5 for *Tsukuba*.



(a) (b) (c) (d) (e) (f)

Figure 5.2: Original image set for *Flower Garden*: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



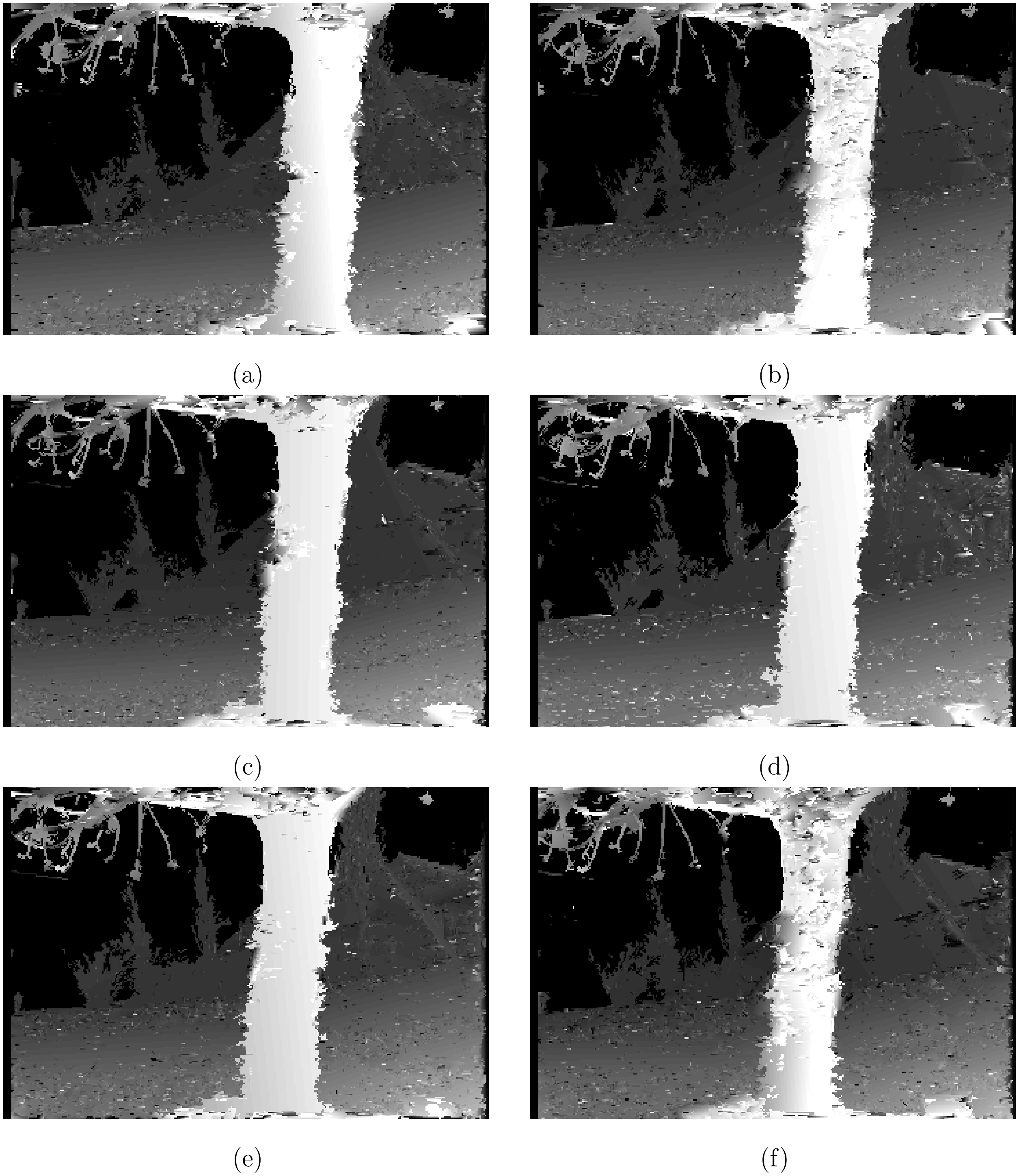
(a)

(b)

(c)

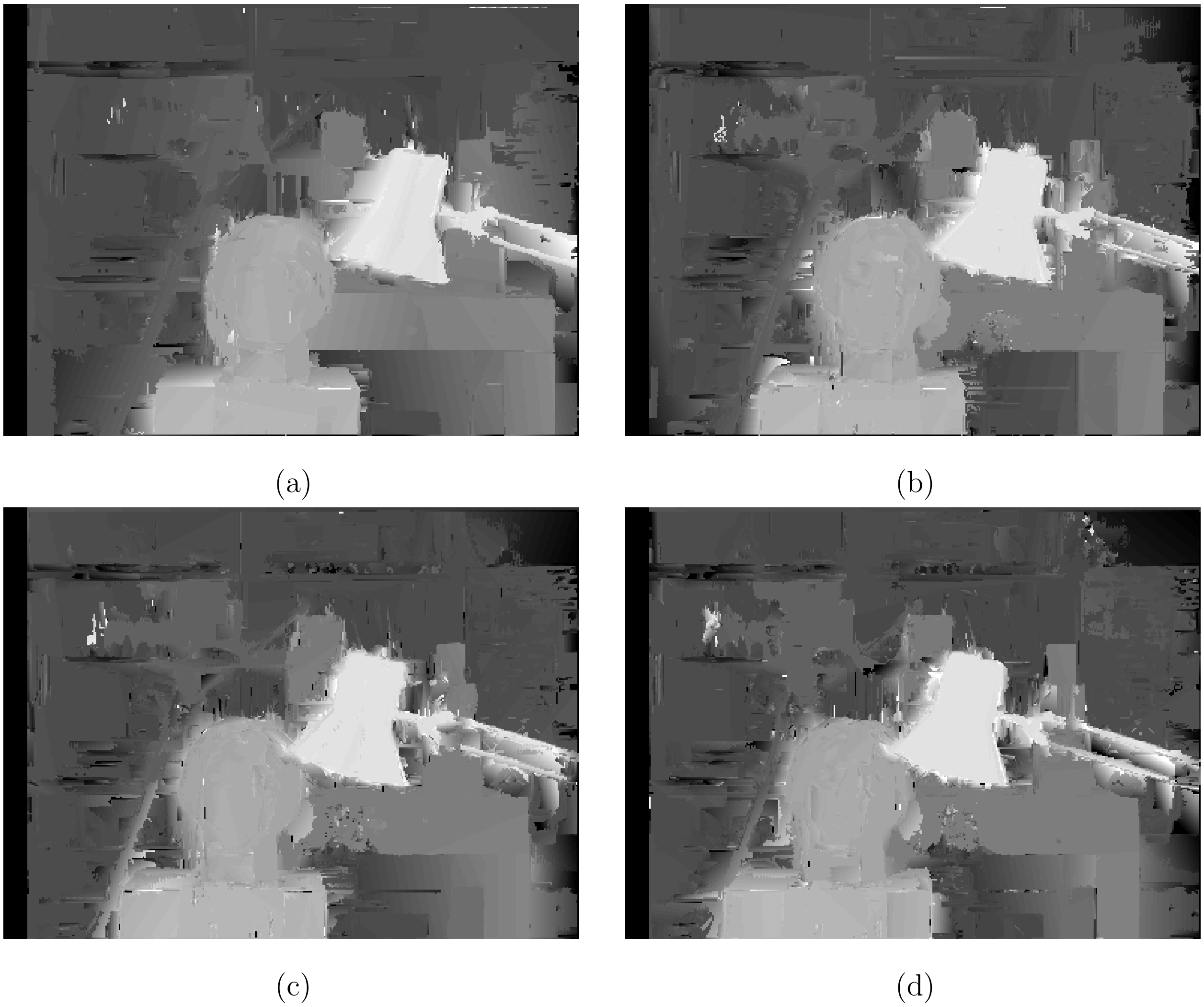
(d)

Figure 5.3: Original image set for *Tsukuba*: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.



(a) (b) (c) (d) (e) (f)

Figure 5.4: Disparity maps after 1st round of matching process using Gabor-based approach for *Flower Garden*: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



(a)

(b)

(c)

(d)

Figure 5.5: Disparity maps after 1st round of matching process using Gabor-based approach for *Tsukuba*: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.

The disparity maps using SSD-based approach for all the image locations are estimated as we did to obtain the disparity maps for *tsukuba0* and *garden0* in Fig. 4.7(a)(b), and are shown in Fig. 5.6 for *Flower Garden*, and in Fig. 5.7 for *Tsukuba*.

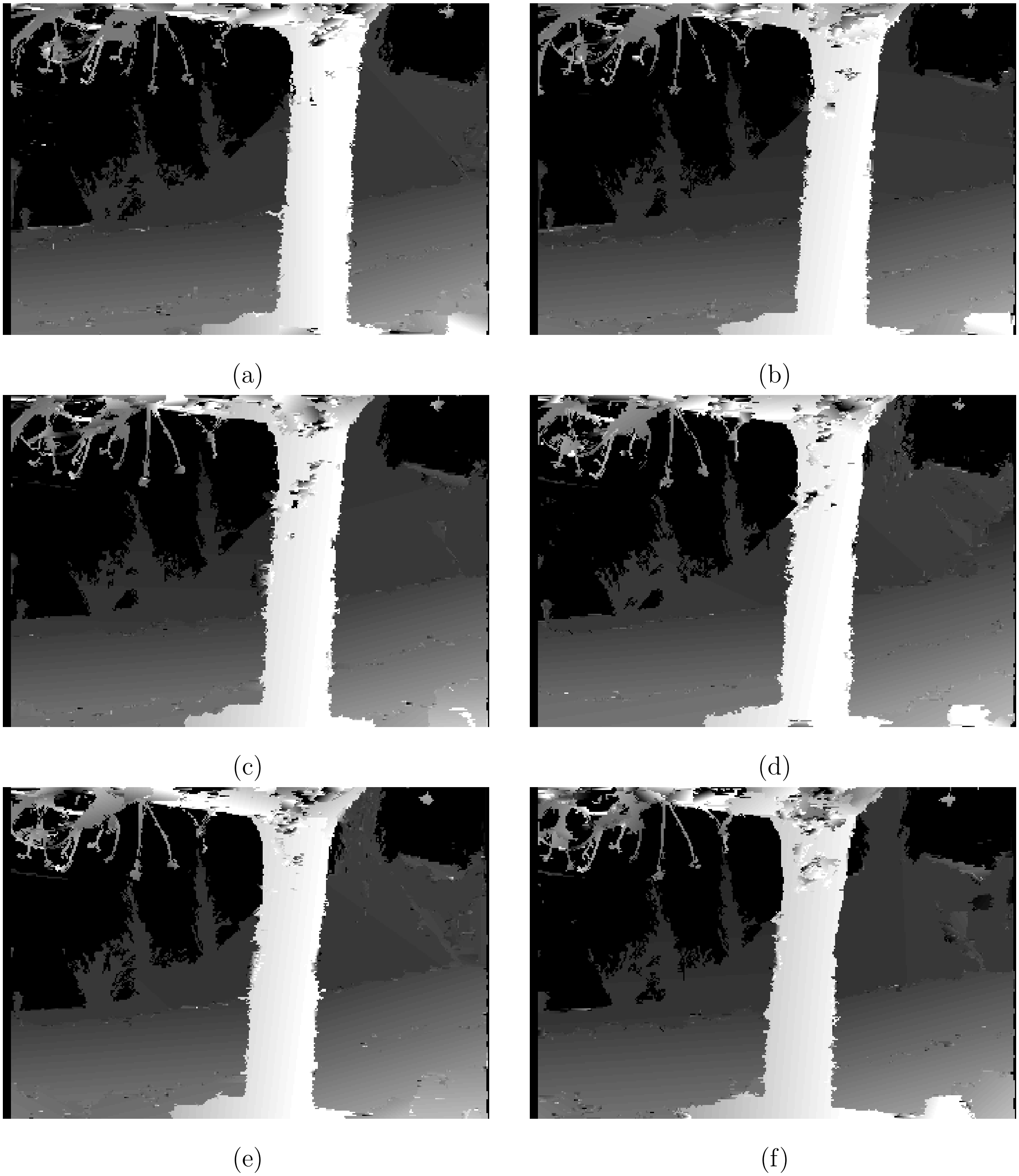
## 5.3 Ego-Motion Estimation Exploiting Large Regions

### 5.3.1 Gabor-Based Approach

As stated at the end of Chapter 2, we will use the image intensity-based method for the ego-motion estimation, similar to that of [50]. We first select the large regions in each image, since they have matching results more reliable than those of small regions, which might be outliers. The criterion for selection as a large region is that the region have more than five hundred pixels (determined empirically). The large regions selected for all the six images of *garden0 – 5* for the approach starting from the Gabor transform (Fig. 4.3) are shown in Fig. 5.8, in which each large region in each image has a unique label for display purpose. The large regions for *Tsukuba0 – 3* for the approach starting from the Gabor transform are shown in Fig. 5.9.

From these large regions, we can see that most of them do not have complete shapes as what they should be, and some large regions are even missing in some locations, like the region representing the tree in *garden1*. In addition, some large regions (especially the slope regions in *Flower Garden* which are full of textures) have very small holes inside, which are actually represented by other small regions not merged with the surrounding large regions; such very small regions will bring outliers in the 3D models. Therefore, we should further improve the shapes of these large regions before a final 3D model can be built up. However, in order to improve the shapes of large regions for one location, we need to propagate the large regions from other locations to that location, and this requires the ego-motion parameters to be estimated first. We will use the existing large regions that we have, as in Fig. 5.8 and Fig. 5.9, to do the ego-motion estimation.

Similarly to the approach in [50], based on the fact that we are using two sets of transla-

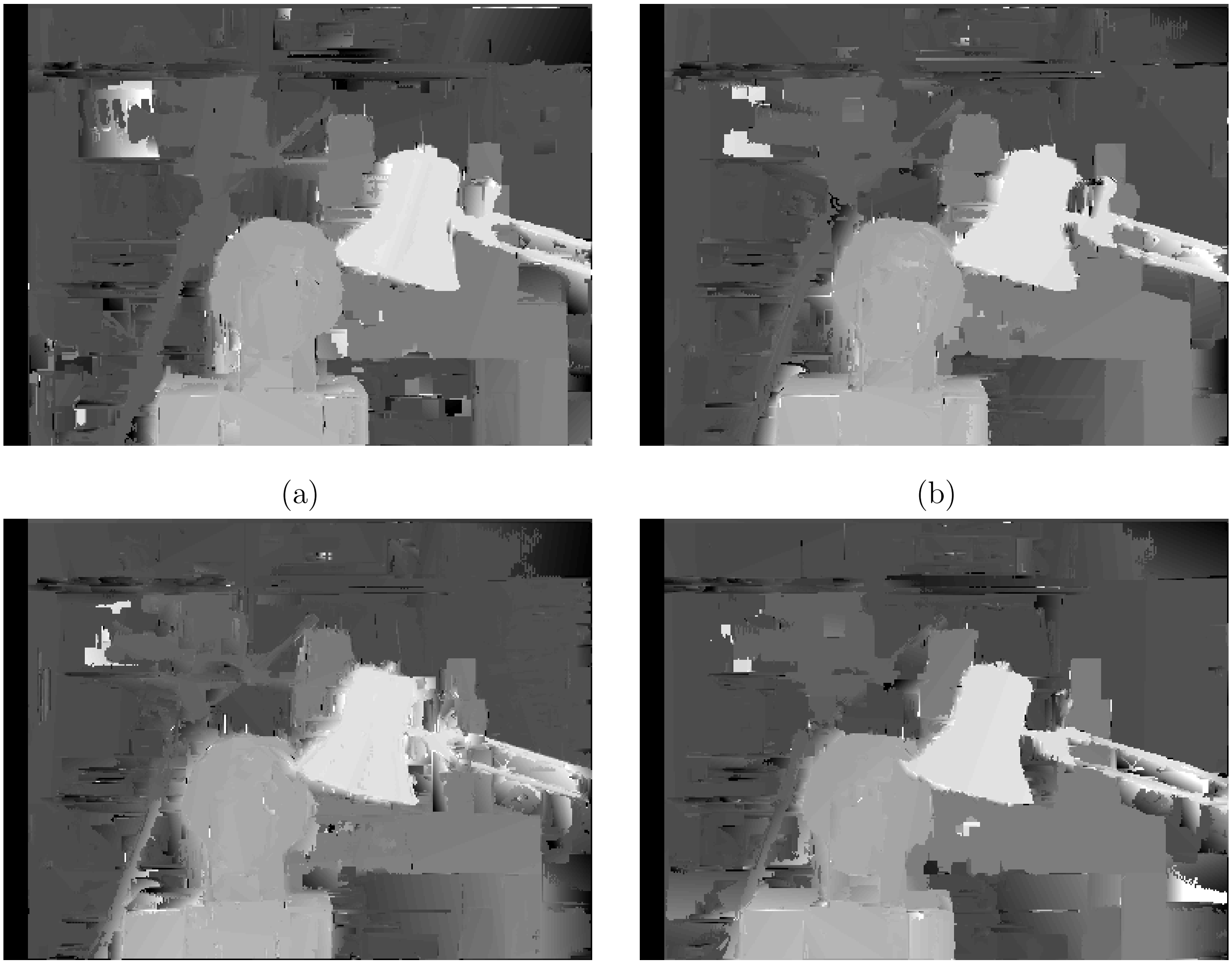


(a) (b)

(c) (d)

(e) (f)

Figure 5.6: Disparity maps after 1st round of matching process using SSD-based approach for *Flower Garden*: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



(a)

(b)

(c)

(d)

Figure 5.7: Disparity maps after 1st round of matching process using Gabor-based approach for *Tsukuba*: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.

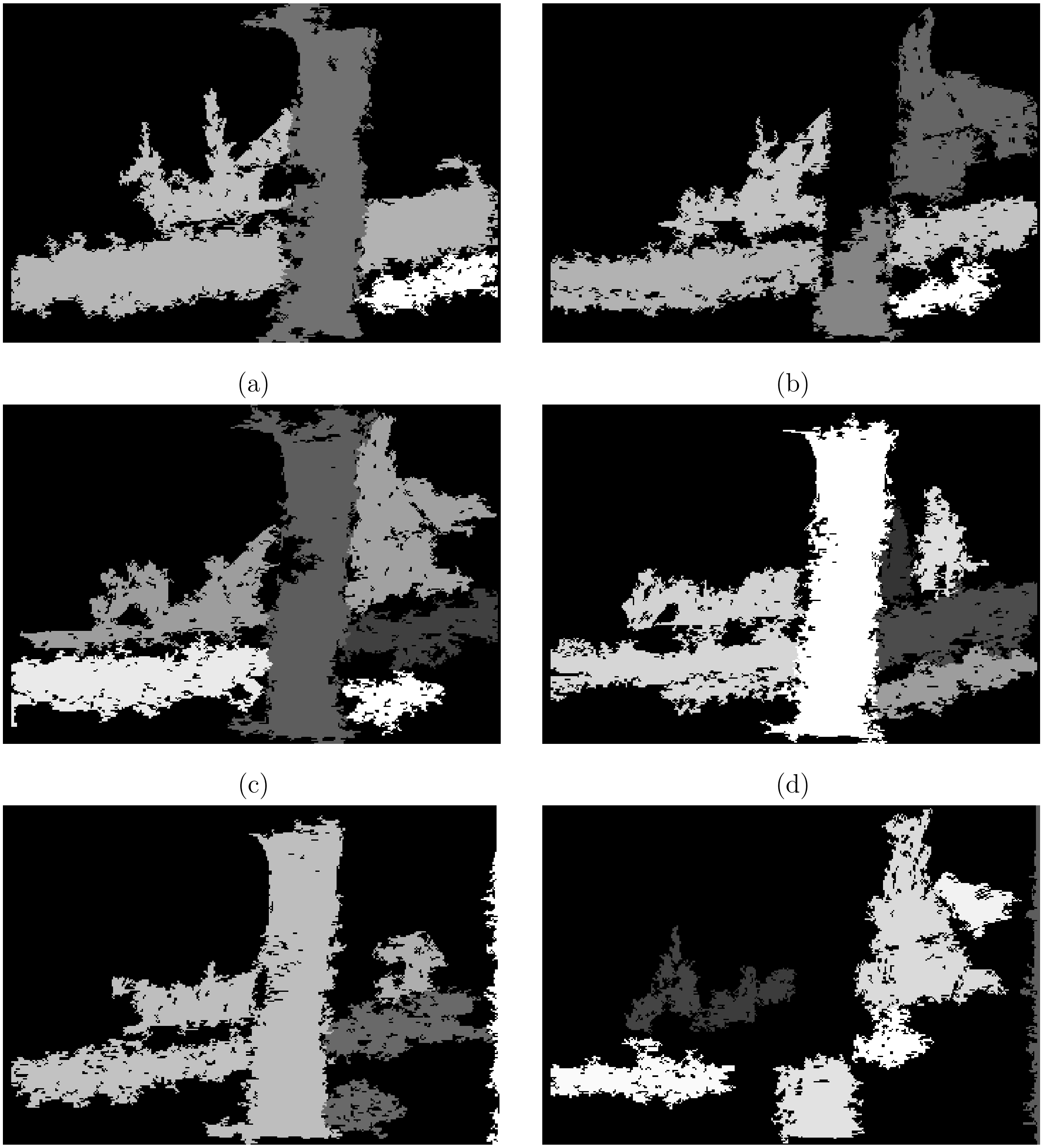
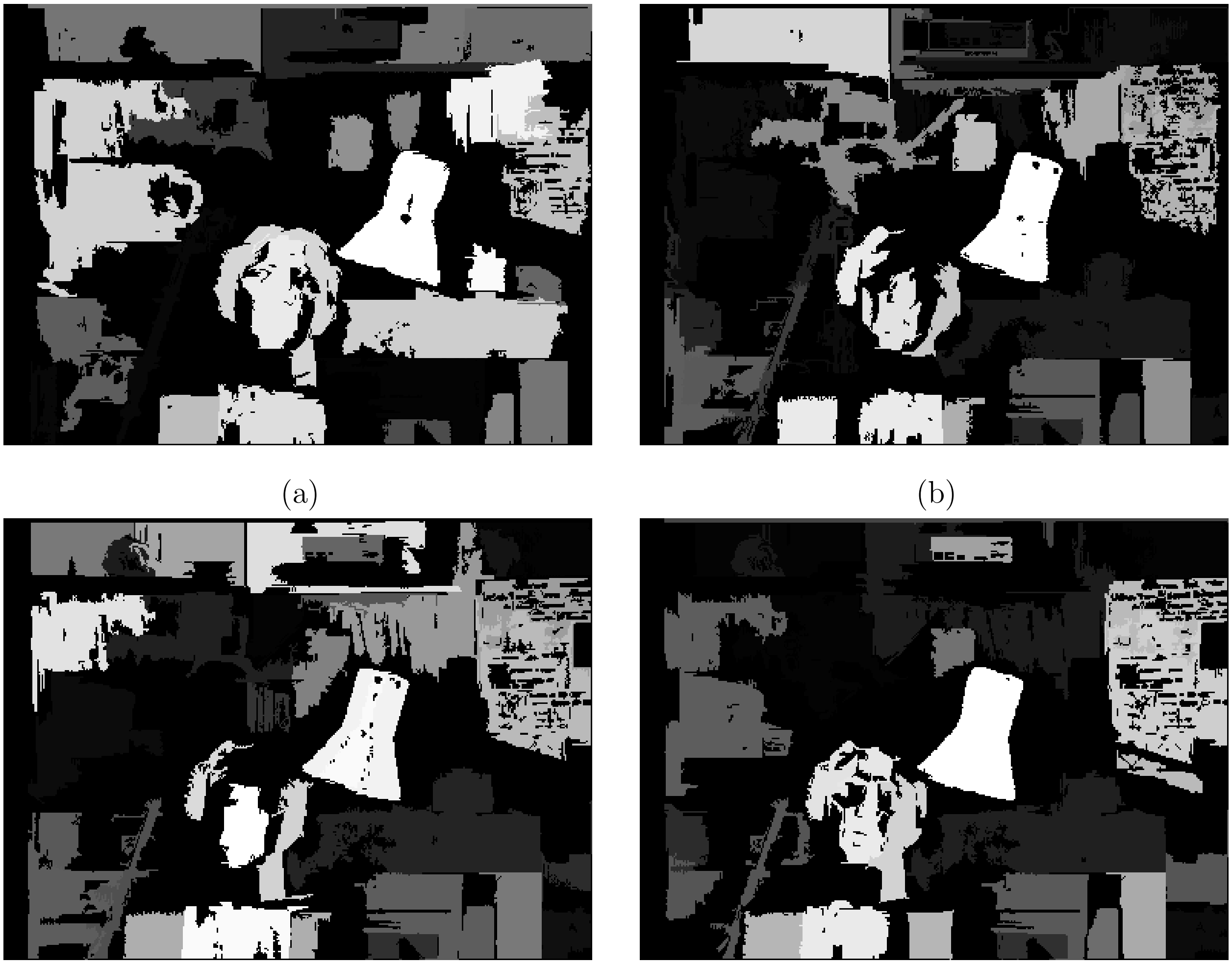


Figure 5.8: Large region labels for *Flower Garden* in Gabor-based approach: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



(a)

(b)

(c)

(d)

Figure 5.9: Large region labels for *Tsukuba* in Gabor-based approach: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.

Table 5.1: Ego-motion parameters for *Flower Garden* in Gabor-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *garden0*→*garden1* | 0.988393 | -0.062613 | 0.000234 |
| *garden1*→*garden2* | 0.978613 | -0.015231 | 0.000114 |
| *garden2*→*garden3* | 1.000874 | -0.003362 | 0.000098 |
| *garden3*→*garden4* | 1.004249 | 0.002557 | 0.000001 |
| *garden4*→*garden5* | 1.001691 | 0.003529 | 0.000007 |

tional image sequences, we also assume the main motion of the camera is translational, with slight rotations around $x$ and $z$ axis. For the two consecutive images $I_{t-1}$ and $I_t$, under the above assumption and thus the same approximations as in Eq. (2.38), we have for the image displacements:

$$\begin{aligned} x_{t-1} - T_x D_{t-1}(x_{t-1}, y_{t-1}) &= \cos(\alpha)x_t - \sin(\alpha)y_t \\ y_{t-1} - b &= \sin(\alpha)x_t + \cos(\alpha)y_t \end{aligned} \tag{5.1}$$

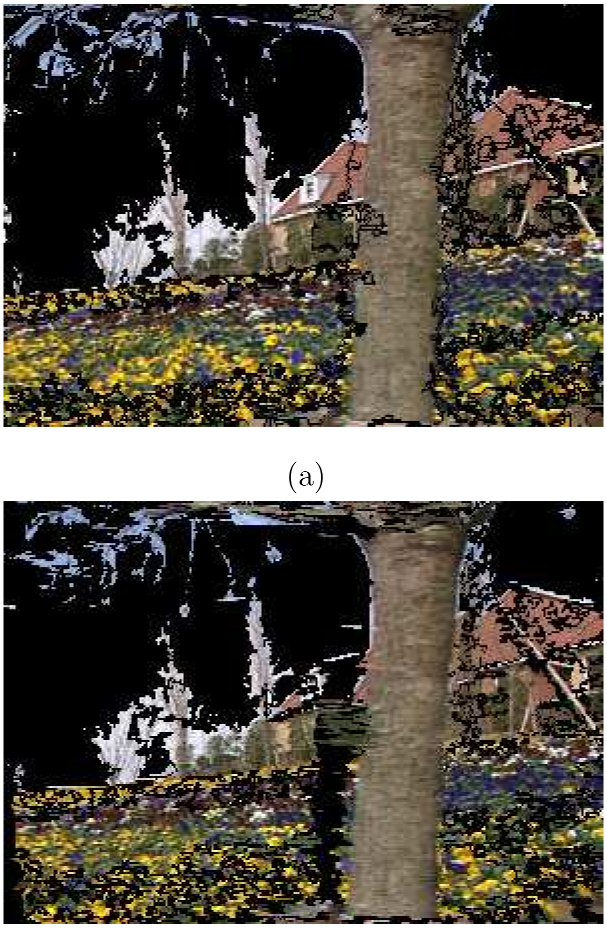
where $T_x$ represents the horizontal translation, $D_{t-1}$ is the normalized disparity which is the disparity value divided by $T_x$ for $I_{t-1}$, $x_{t-1}$ and $y_{t-1}$ are image coordinates of $I_{t-1}$, and $x_t$ and $y_t$ are image coordinates of $I_t$. Similar to (2.38), $\alpha$ denotes the rotation around the $z$ axis, and the rotation around the $x$ axis is approximated by a uniform vertical translation $b$. Also, with small rotation around the $z$ axis, we have $\cos(\alpha) \approx 1$ and $\sin(\alpha) \approx \alpha$.

We use the same formulae as in (2.39) – (2.41) for the estimation of the ego-motion parameters $T_x$, $b$ and $\alpha$. The estimated parameters for each consecutive set of positions are shown in Table 5.1 for *Flower Garden*, and in Table 5.2 for *Tsukuba*.

To show the quality of the ego-estimation, we show the rendered image of *garden0* from OpenGL after setting up the 3D model using the depth map for the position of *garden0*, and the rendered image of *garden5* after transforming the separate 3D model of *garden5* to the position of *garden0*, in Fig. 5.10 (without the sky regions). We can see that both for the object with large displacement (the tree), and the object with small displacement (the

Table 5.2: Ego-motion parameters for *Tsukuba* in Gabor-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *Tsukuba0*→*Tsukuba1* | 1.000224 | 0.003139 | -0.000007 |
| *Tsukuba1*→*Tsukuba2* | 1.000542 | 0.003490 | -0.000012 |
| *Tsukuba2*→*Tsukuba3* | 0.999933 | 0.000178 | 0.000010 |



(a)

(b)

Figure 5.10: An example of homogeneous transform after the ego-motion estimation: (a) direct rendering of *garden0*; (b) rendering of *garden5* transformed to the position of *garden0*.

Table 5.3: Ego-motion parameters for *Flower GArden* in SSD-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *garden0*→*garden1* | 0.984231 | -0.091047 | 0.000256 |
| *garden1*→*garden2* | 0.989103 | -0.003425 | 0.000062 |
| *garden2*→*garden3* | 0.996749 | -0.003728 | 0.000087 |
| *garden3*→*garden4* | 1.001183 | -0.000138 | 0.000017 |
| *garden4*→*garden5* | 1.001676 | 0.005113 | -0.000003 |

Table 5.4: Ego-motion parameters for *Tsukuba* in SSD-based approach

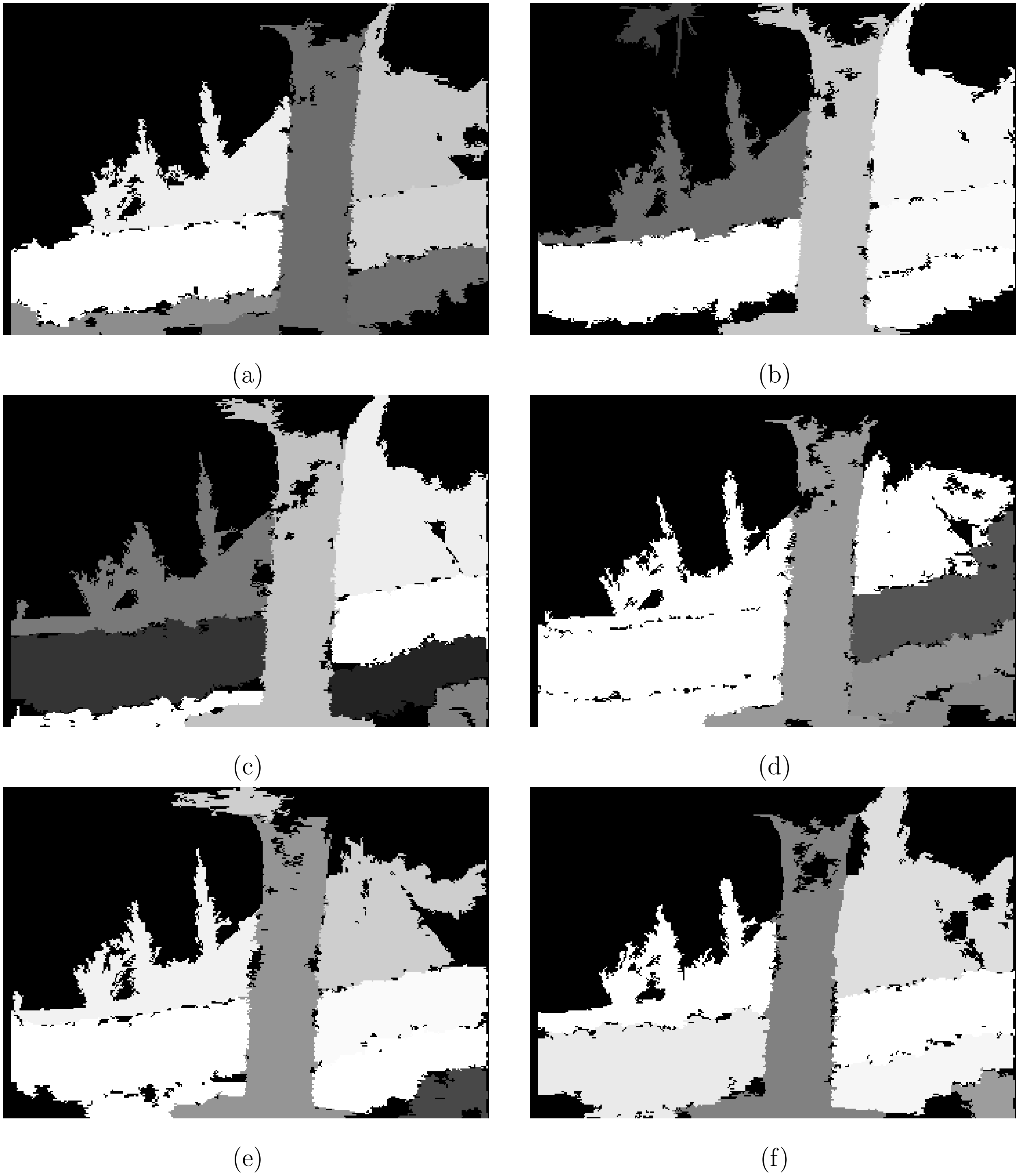
| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *Tsukuba0*→*Tsukuba1* | 0.998399 | -0.001166 | 0.00010 |
| *Tsukuba1*→*Tsukuba2* | 1.000423 | 0.012033 | -0.000023 |
| *Tsukuba2*→*Tsukuba3* | 0.999868 | 0.000395 | 0.000016 |

house), the alignment of the two images after the homogeneous transformation is very good which can be seen by comparing vertically different parts of the tree and the house along the borders (e.g., each part along the left borders of the tree are almost on the same vertical line between the two images in Fig. 5.10(a) and (b)).

### 5.3.2 SSD-Based Approach

Similar to the cases in Gabor-Based Approach, for the approach starting from SSD (Fig. 4.8), we first pick out the large regions in each image, which are shown in Fig. 5.11 for *Flower Garden*, and in Fig. 5.12 for *Tsukuba*.

Using the same ego-motion estimation algorithm as in the Gabor-based approach ((2.39) – (2.41)), the estimated parameters $T_x$, $b$ and $\alpha$ are shown in Table 5.3 for *Flower Garden*, and in Table 5.4 for *Tsukuba*.



(a)

(b)

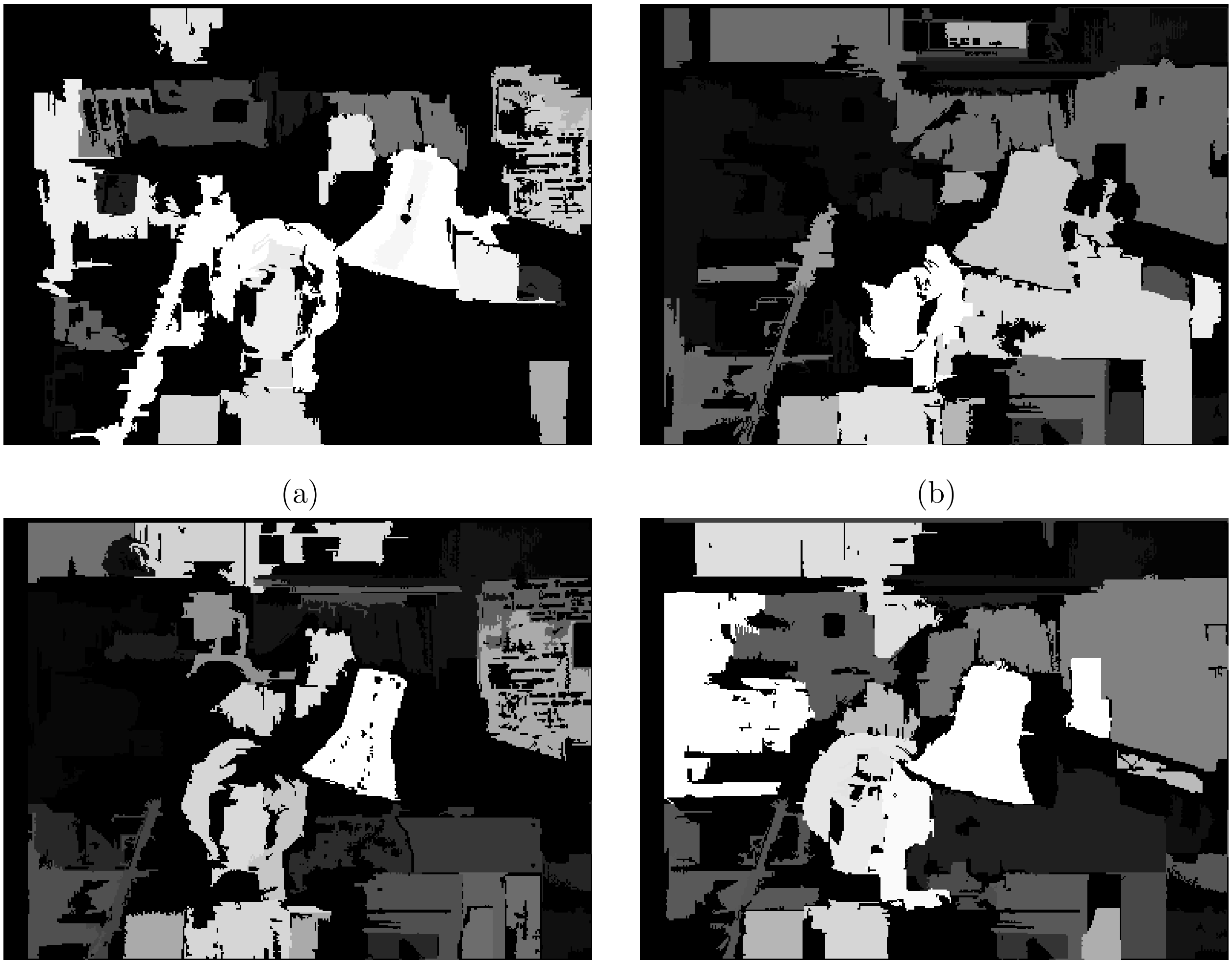
(c)

(d)

(e)

(f)

Figure 5.11: Large region labels for *Flower Garden* in SSD-based approach: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



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

Figure 5.12: Large region labels for *Tsukuba* in SSD-based approach: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.

## 5.4 Refinement for the Shape of Large Regions

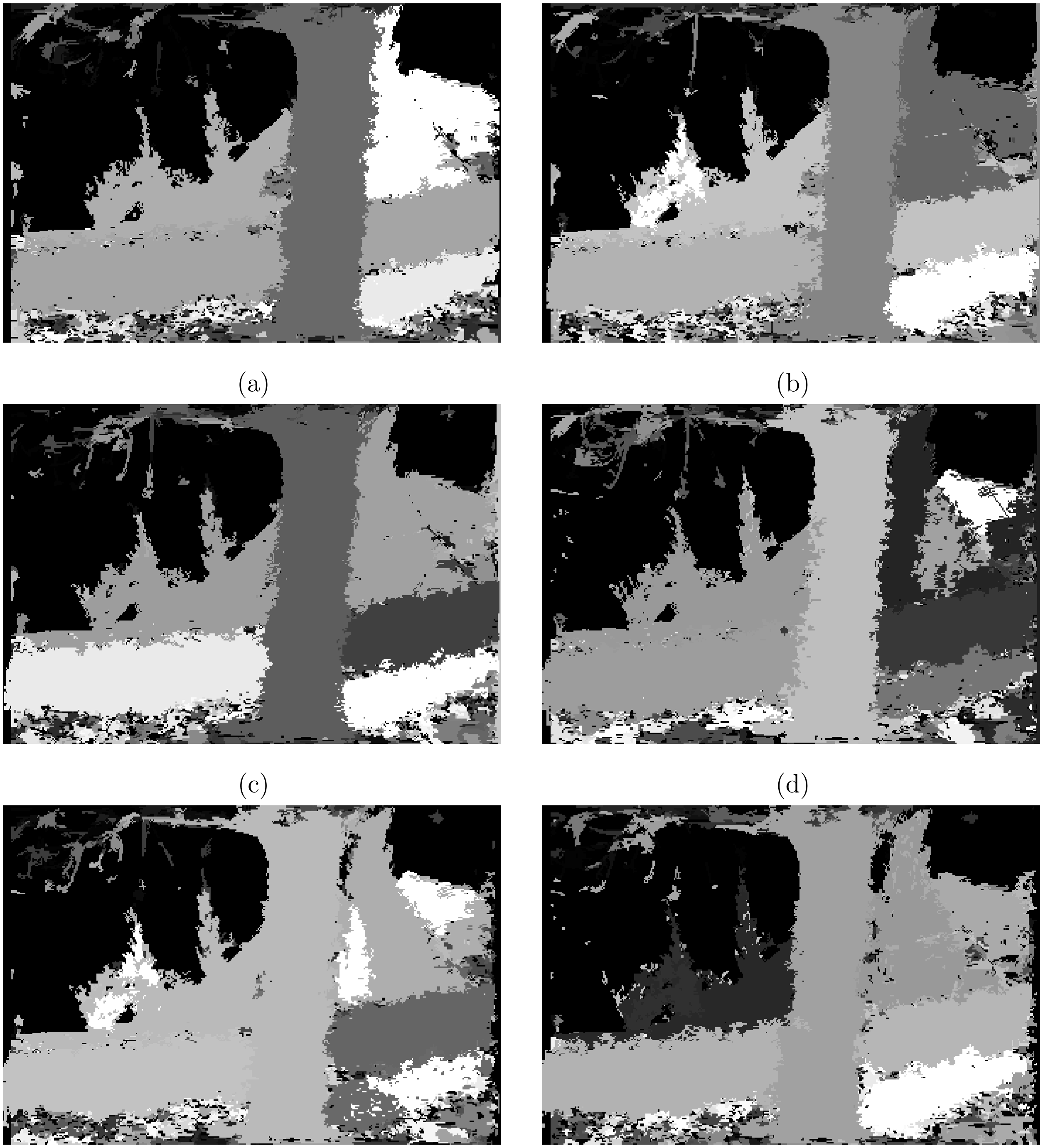
Once we have the parameters of the homogeneous transformations among all the image locations, we can propagate the large regions in one image to the locations of other images, and compare the correspondence relations among them. If a large region in one image is determined to be the corresponding region for another large region in another image by comparing the amount of overlapping pixels between them, then the shapes of these two regions as well as the areas they occupy can be used to compensate each other along with their associated depth information. Assume we need to refine a large region in image $I_A$ with label value $A$ using the large region label $LP_{I_B}$ propagated from image $I_B$, and a large region in $LP_{I_B}$ with label value $B$ is determined as the corresponding region for large region $A$ in $I_A$. The large region label of $I_A$ is $L_{I_A}$, and the disparity maps are $d_A$ for $I_A$ and $d_BProp$ for $d_B$ of $I_B$ propagated to $I_A$. Then the procedure to regine large region $A$ in $I_A$ can be summarized as follows:

```
For each pixel $(x, y)$ in $I_A$, go through following steps:
    (1) If $L_{I_A}(x, y) = 0$ and $LP_{I_B}(x, y) = B$, then set $L_{I_A}(x, y) = A$
    (2) If $L_{I_A}(x, y)$ not equal to zero, $L_{I_A}(x, y)$ not equal to $A$ and $LP_{I_B}(x, y)$
        equal to $B$, then check:
        (a) If $d_A(x, y)$ is less than $d_BProp(x, y)$, then set $L_{I_A}(x, y) = A$
        (b) If $d_A(x, y)$ is larger than $d_BProp(x, y)$ and $(x, y)$ belongs to a region
            less than 10 pixels, then set $L_{I_A}(x, y) = A$
```

This procedure is very useful in eliminating the very small regions inside a large region (usually resulting in outliers in the depth map).

### 5.4.1 Gabor-Based Approach

For the Gabor-based approach, we show the new region labels including all the large regions for each image after the above refinement procedure which is done by propagating the large regions of all the other images into its location in Fig. 5.13 for *Flower Garden*. Comparing



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.13: Refined region labels for *Flower Garden* after homogeneous transformation using Gabor-based approach: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.

Fig. 5.13 with Fig. 5.8, we can see that most of the false small regions in each of the large regions are filled by large regions propagated from other image locations.

Similarly, for *Tsukuba*, the new region labels including the refined large region masks are shown in Fig. 5.14.

### 5.4.2 SSD-Based Approach

For the approach starting from SSD, the new region labels including the refined large region masks are shown in Fig. 5.15 for *Flower Garden*, and Fig. 5.16 for *Tsukuba*. Comparing Fig. 5.15 and Fig. 5.16 with Fig. 5.13 and Fig. 5.14, we can see that the large regions for SSD-based approach are more clean than the large regions for the Gabor-based approach. This is because the SSD give initial disparity maps with fewer outliers than the Gabor-based approach.

## 5.5 Second Round of Matching for all the Image Locations

After the shapes of large regions are refined to get rid of the false small regions, we put *all* the regions to a second round of matching process.

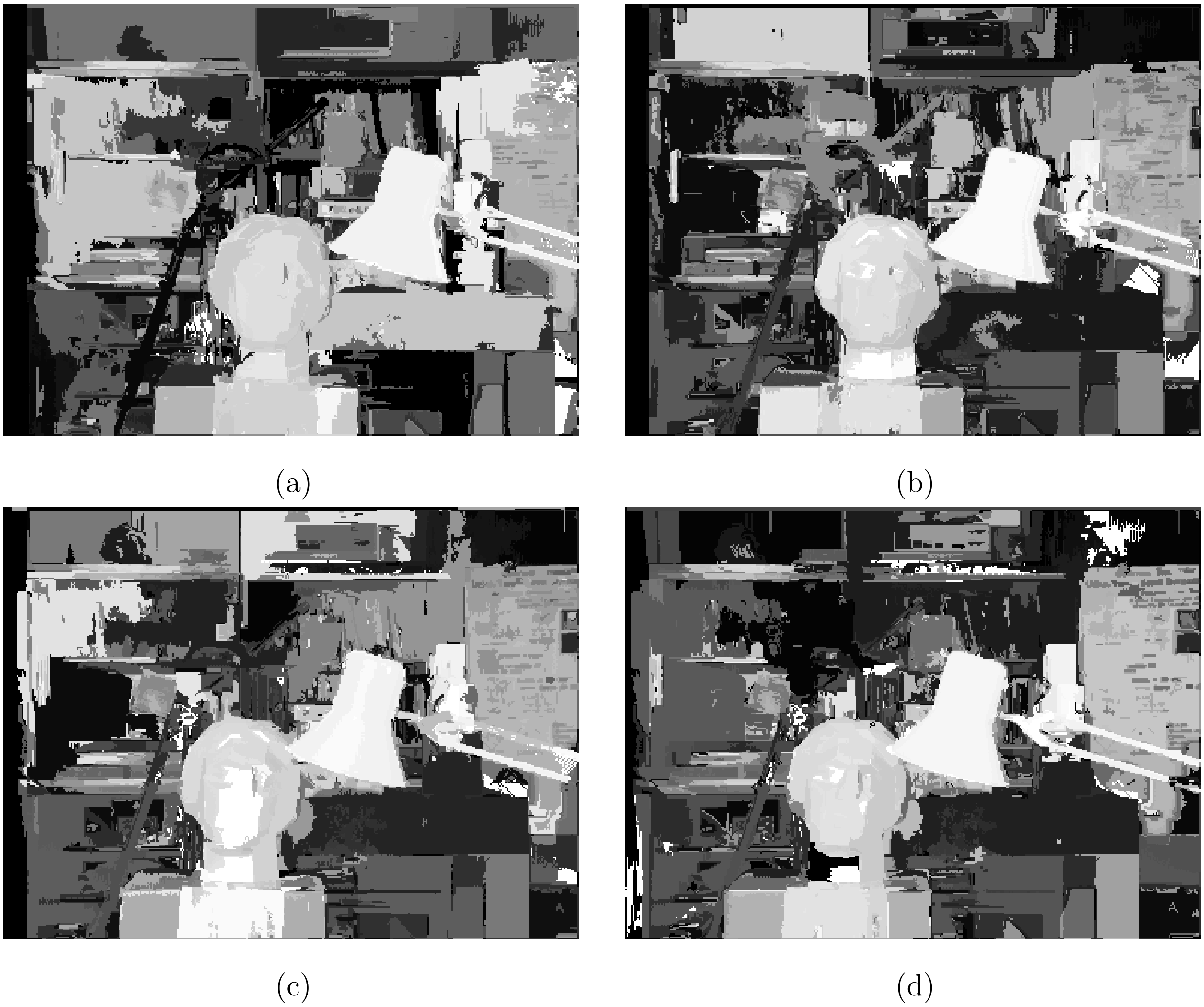
### 5.5.1 Gabor-Based Approach

The results of the second round of disparity estimation using the Gabor-based approach are shown in in Fig. 5.17 for *Flower Garden*, and in Fig. 5.18 for *Tsukuba*.

We can see that most of the outliers are eliminated.

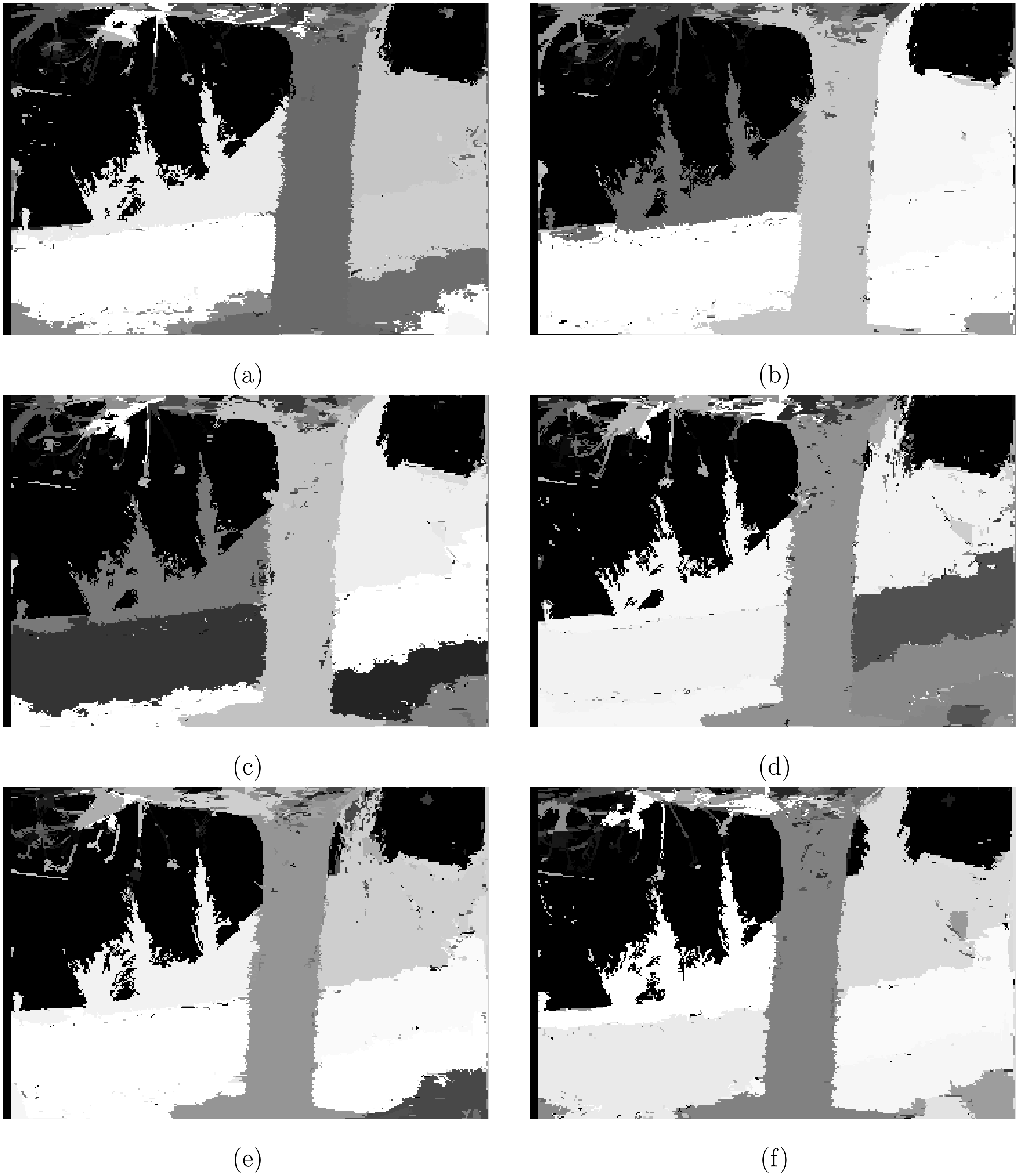
### 5.5.2 SSD-Based Approach

The results of the second round of disparity estimation using the Gabor-based approach are shown in in Fig. 5.19 for *Flower Garden*, and in Fig. 5.20 for *Tsukuba*.



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

Figure 5.14: Refined region labels for *Tsukuba* using Gabor-based approach: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.



(a)

(b)

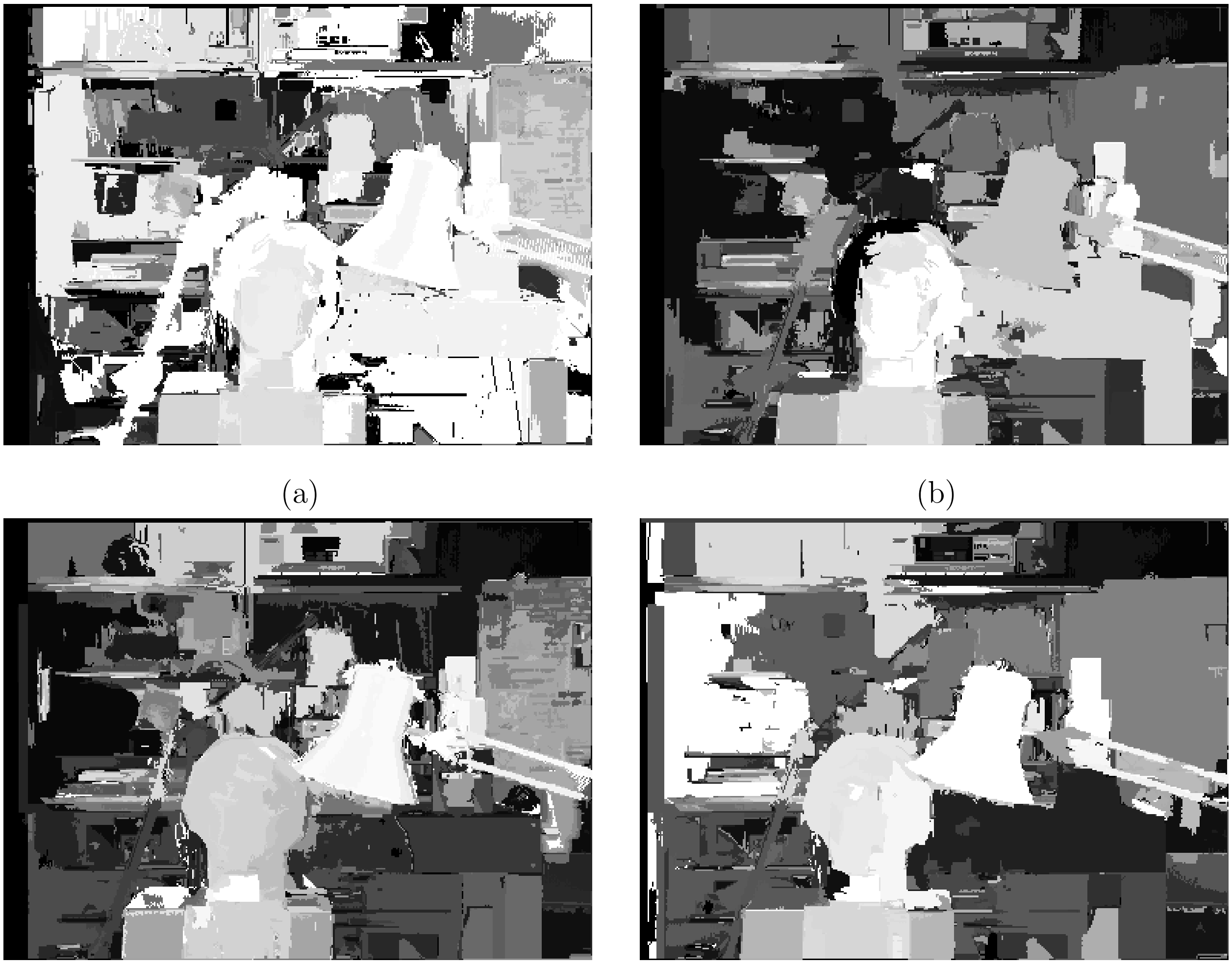
(c)

(d)

(e)

(f)

Figure 5.15: Refined region labels for *Flower Garden* after homogeneous transformation using SSD-based approach: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



(a)

(b)

(c)

(d)

Figure 5.16: Refined region labels for *Tsukuba* using SSD-based approach: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.

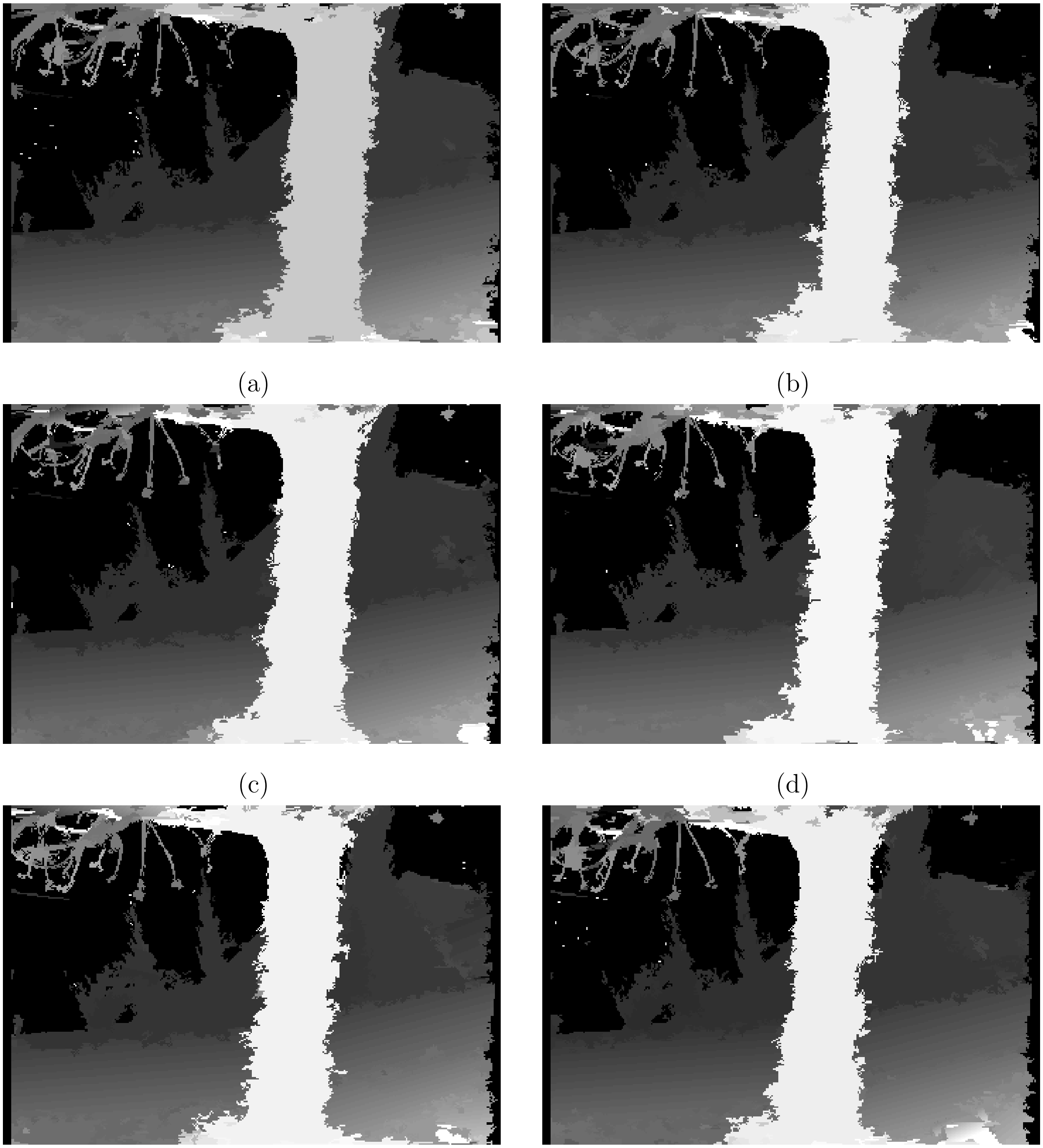
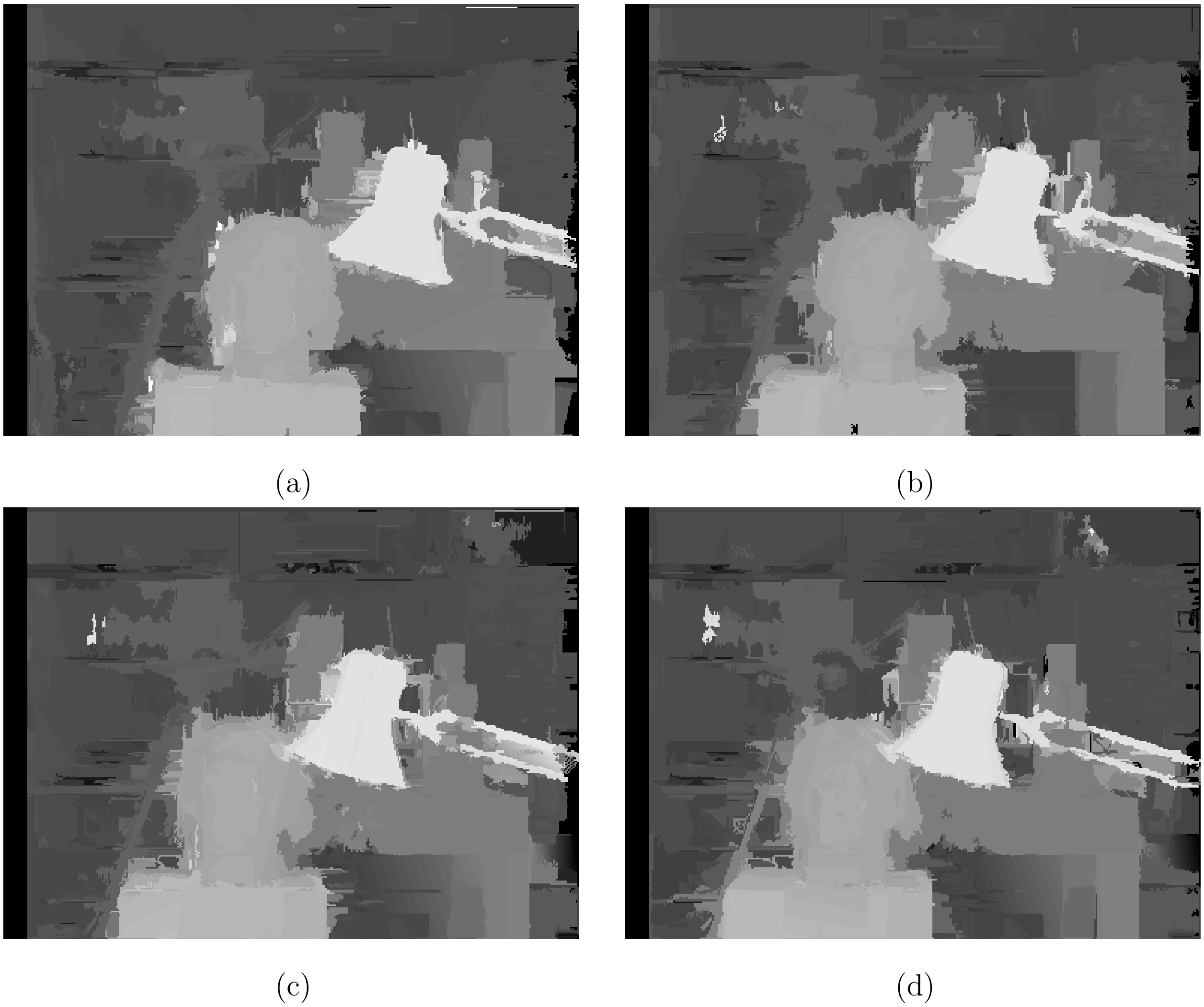

(a)
(b)
(c)
(d)
(e)
(f)

Figure 5.17: Disparity maps after 2nd round of matching process using Gabor-based approach for *Flower Garden*: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



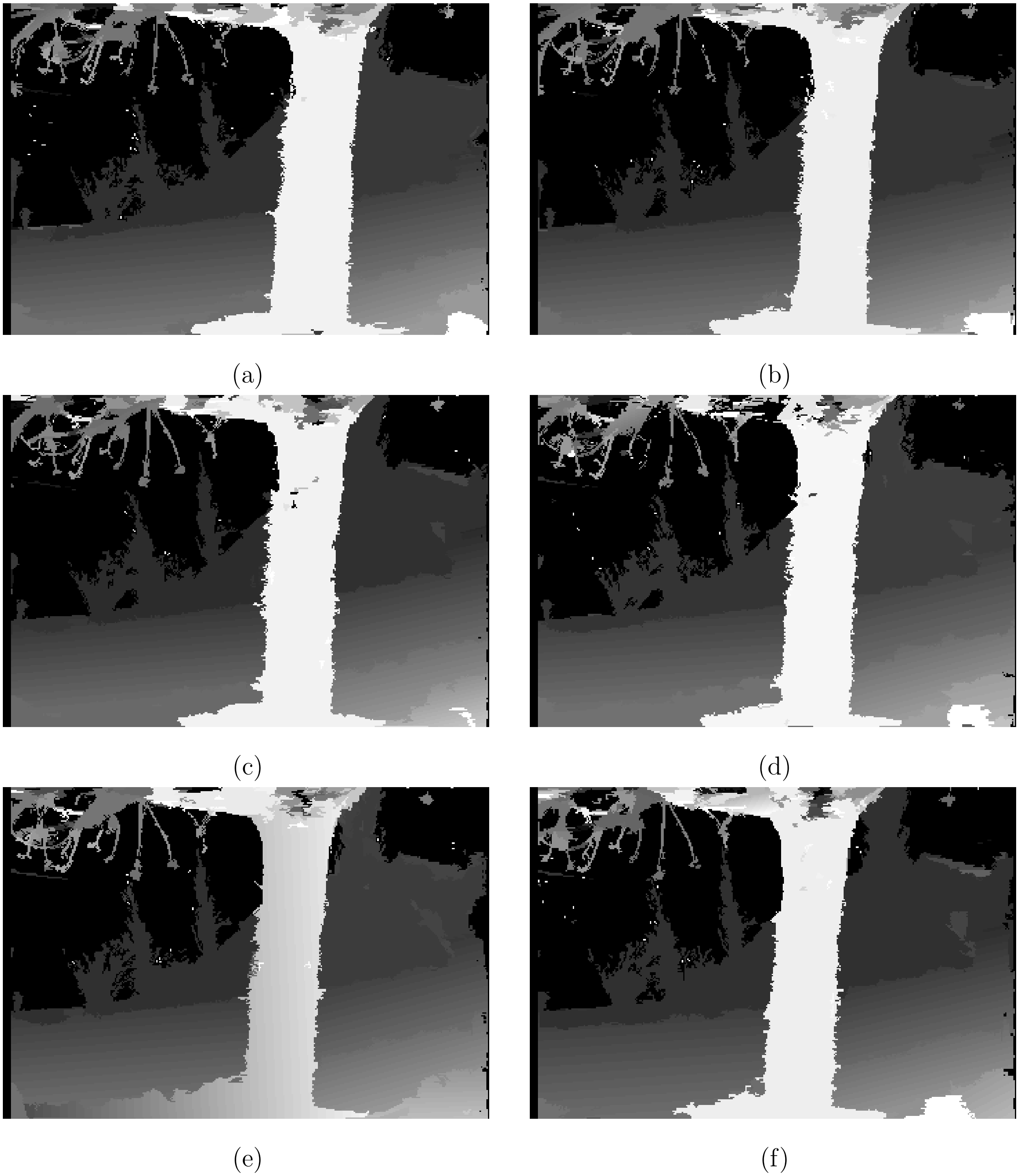
(a)

(b)

(c)

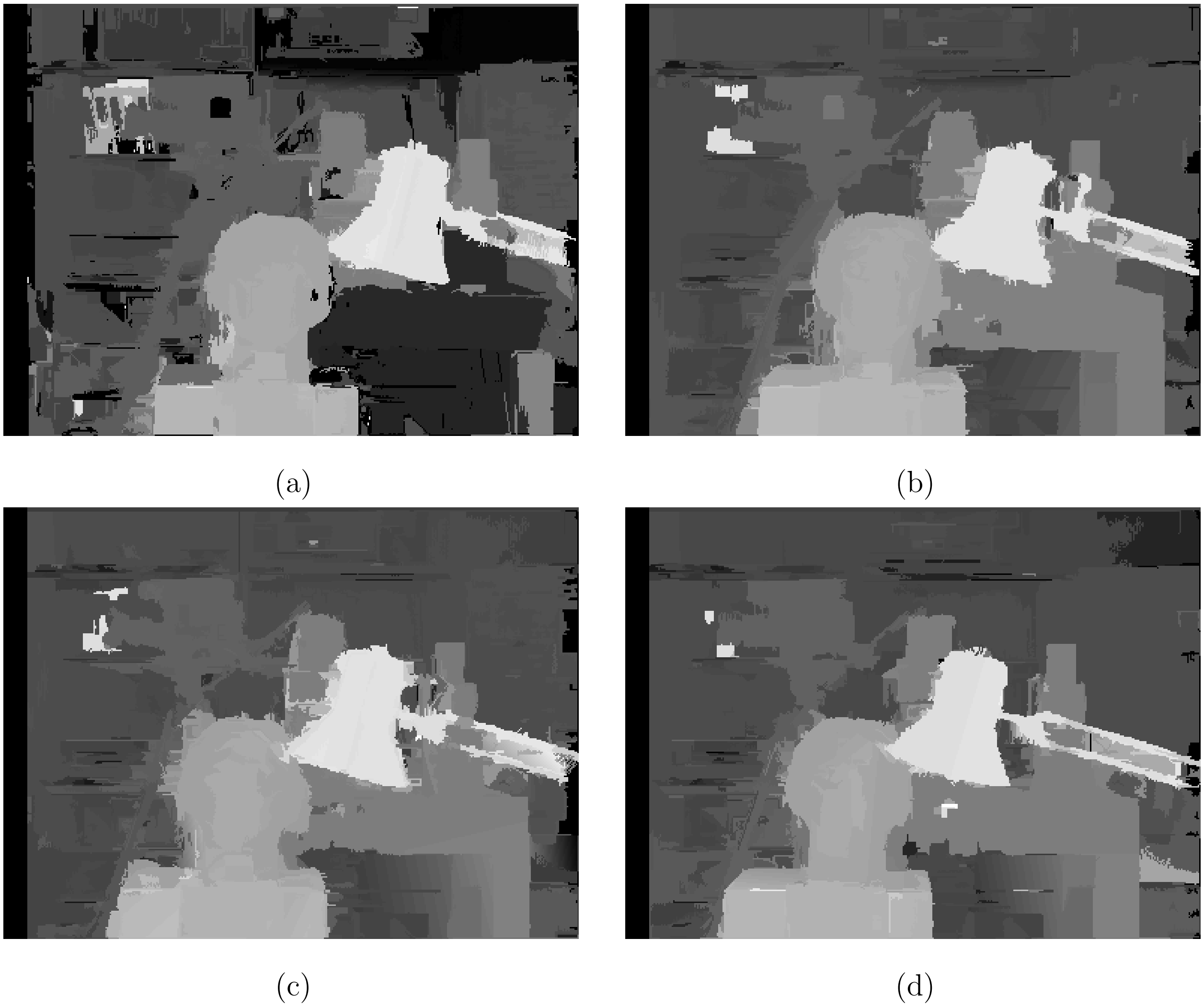
(d)

Figure 5.18: Disparity maps after 2nd round of matching process using Gabor-based approach for *Tsukuba*: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.



(a) (b) (c) (d) (e) (f)

Figure 5.19: Disparity maps after 2nd round of matching process using SSD-based approach for *Flower Garden*: (a) *garden0*; (b) *garden1*; (c) *garden2*; (d) *garden3*; (e) *garden4*; (f) *garden5*.



(a)

(b)

(c)

(d)

Figure 5.20: Disparity maps after 2nd round of matching process using SSD-based approach for *Tsukuba*: (a) *tsukuba0*; (b) *tsukuba1*; (c) *tsukuba2*; (d) *tsukuba3*.

Table 5.5: Second set of ego-motion parameters for *Flower Garden* in Gabor-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *garden0*→*garden1* | 0.995553 | -0.053821 | 0.000183 |
| *garden1*→*garden2* | 1.000509 | 0.000100 | 0.000050 |
| *garden2*→*garden3* | 1.002317 | -0.001033 | 0.000070 |
| *garden3*→*garden4* | 1.002498 | 0.002248 | 0.000011 |
| *garden4*→*garden5* | 1.001633 | 0.005674 | -0.000012 |

Table 5.6: Second set of ego-motion parameters for *Tsukuba* in Gabor-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *Tsukuba0*→*Tsukuba1* | 0.995898 | -0.014878 | 0.000131 |
| *Tsukuba1*→*Tsukuba2* | 0.956393 | -0.195677 | -0.000431 |
| *Tsukuba2*→*Tsukuba3* | 1.001717 | 0.111050 | -0.000542 |

## 5.6 Second Round of Ego-Motion Estimation

With the new depth maps for each image location in which most of the outliers are removed from the large regions, we can perform a second round of ego-motion estimation. The procedure is the same as in section 5.3.

The newly estimated ego-motion parameters using Gabor-based approach are shown in Table 5.5 for *Flower Garden*, and in Table 5.6 for *Tsukuba*.

The newly estimated ego-motion parameters using SSD-based approach are shown in Table 5.7 for *Flower Garden*, and in Table 5.8 for *Tsukuba*.

Table 5.7: Second set of ego-motion parameters for *Flower Garden* in SSD-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *garden0*→*garden1* | 0.981548 | -0.096068 | 0.000282 |
| *garden1*→*garden2* | 0.998241 | 0.000659 | 0.000041 |
| *garden2*→*garden3* | 0.997694 | -0.002068 | 0.000057 |
| *garden3*→*garden4* | 1.000623 | -0.000024 | 0.000015 |
| *garden4*→*garden5* | 0.999254 | 0.003270 | 0.0 |

Table 5.8: Second set of ego-motion parameters for *Tsukuba* in SSD-based approach

| Position | $T_x$ | $b$ | $\alpha$ |
|---|---|---|---|
| *Tsukuba0*→*Tsukuba1* | 0.999258 | -0.018133 | 0.000185 |
| *Tsukuba1*→*Tsukuba2* | 0.983966 | -0.056794 | -0.000928 |
| *Tsukuba2*→*Tsukuba3* | 1.002299 | 0.106683 | -0.000581 |

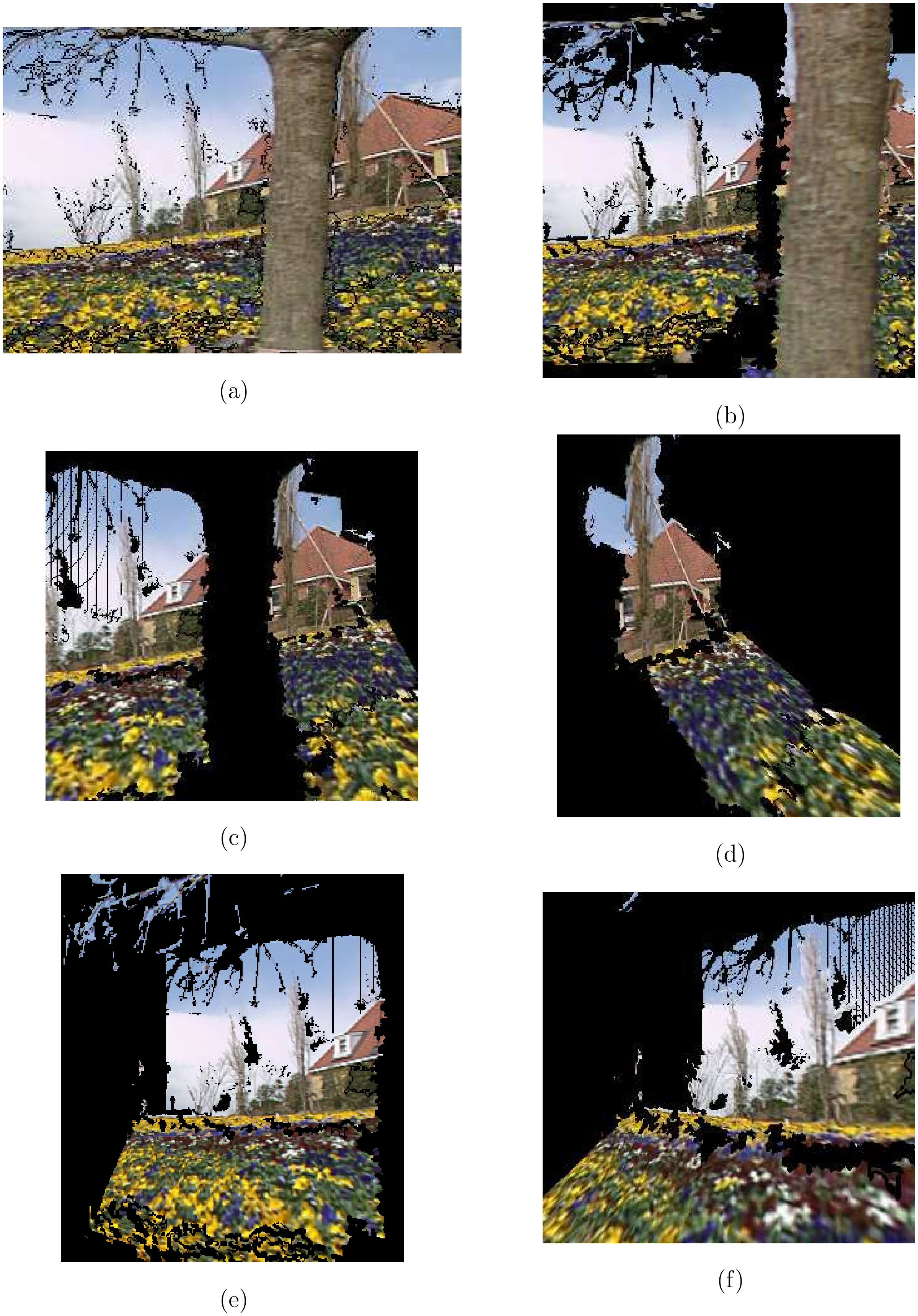
## 5.7 Merging of 3D Models

Finally, with the much improved separate 3D models and the new set of ego-motion parameters, we reach the final step to merge these separate 3D models into one combined model. The method we used to achieve this purpose can be classified as the surface approach (as opposed to the volumetric approach). In the existing methods for surface approach, surfaces from separate 3D models with triangular meshes are put together in 3D space. Then, if two or more separate surfaces are determined to belong to one whole surface, the overlapping and non-overlapping parts among them are determined. To stitch these surfaces together, some strategies need to be developed on how to deal with the overlapping parts (keeping one of the surfaces and abandoning others, or making an average among these overlapping surfaces). For the non-overlapping parts, usually the triangles connecting them to the overlapping parts are abandoned, and a new triangulation process should be applied on those vertices along the gap of overlapping and non-overlapping parts [55][56]. Therefore, we can see that the existing surface approach is a tedious and computational expensive process. In our integration procedure, we do such surface stitching only based on pixels in each regions with triangulation process applied to the finally merged surfaces.

### 5.7.1 Results for Gabor-Based Approach

We first show the rendering from the separate 3D model at the location of *garden1* in Fig. 5.21. The black lines and dots on the sky region in the left part of Fig. 5.21(c)(e)(f) are caused by the fact that the shape of the left sky region is very complex, and the triangulation process failed on it. In order to present the whole image, we put 3D point array instead of triangular mesh for this sky region. Thus, when we zoom into the scene, this sky region will split.

Our merging procedure still starts from large regions in each image. Once the corresponding relations for all the large regions in all images are determined, then to set up one combined model in one reference image location, e.g., the location of *garden1*, the large regions from all other images are propagated to the location of *garden1*, and each large region in *garden1*



(a)

(b)

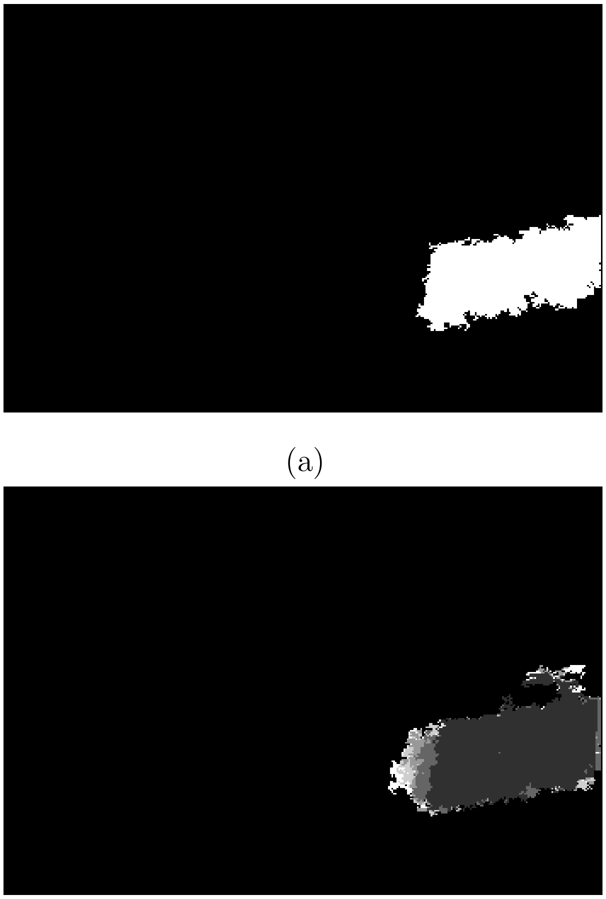
(c)

(d)

(e)

(f)

Figure 5.21: Rendering from separate 3D model at *garden1*: (a) original viewpoint; (b) – (f) novel viewpoints.



(a)

(b)

Figure 5.22: An example of large region expansion: (a) An original large region in *garden1*; (b) Corresponding large regions from other images.

is expanded by the corresponding large regions from all other images. We show this with one example in Fig. 5.22 using Gabor-based approach, in which Fig. 5.22(a) is one original large region in *garden1*, and Fig. 5.22(b) is its corresponding large regions propagated from other images. We can see the expansion from Fig. 5.22(a) – (b), and this expansion will bring some geometry and texture information behind the tree for the location of *garden1*.

This kind of procedure is applied to all the large regions of *garden1*, and each expanded large region is triangulated with textures from all images while setting up the final whole 3D model for the location of *garden1*. After dealing with large regions, we put the triangulated small regions of *garden1* with textures and this gives us one combined 3D model for the location of *garden1*. Six rendered images after the models of *garden0* – *garden5* are combined at the reference location of *garden1* are shown in Fig. 5.23.

Comparing Fig. 5.23 with Fig. 5.21, we can find that more surfaces and textures behind the tree are filled (the house and slope areas on the right side of the tree). However, there are obvious artifacts along the stitched borders in which two regions belonging to the house are stitched together with a vertical shift. This is due to the inaccuracy of the estimated ego-motion parameters.

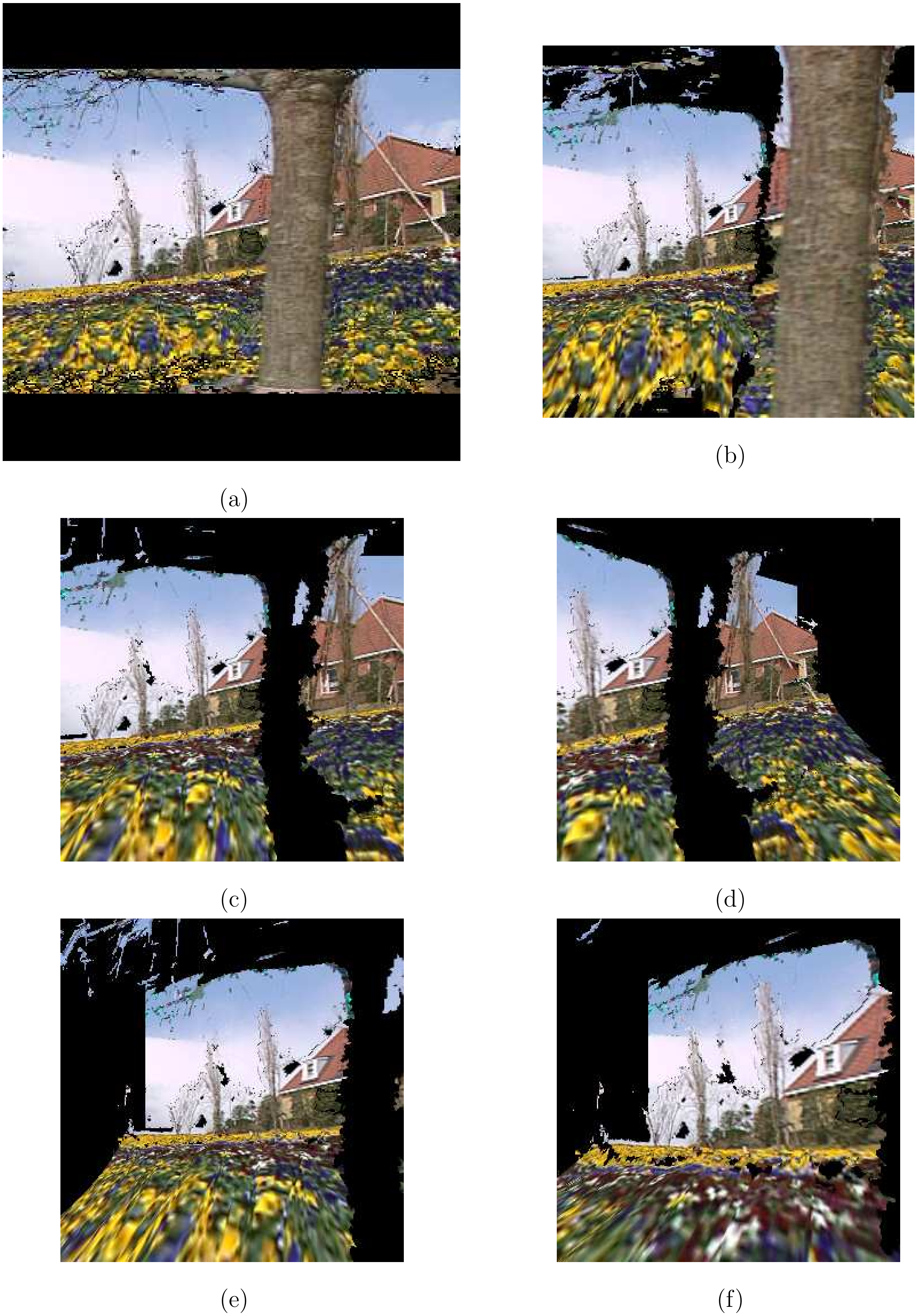
We also generated the images at the locations of *garden0*, *garden2* and *garden3*, and obtained the PSNR results by comparing them with the original images. The results are shown in Fig. 5.24.

Six rendered images for *Tsukuba* are shown in Fig. 5.25 for the image location of *tsukuba1* after the separate 3D models for *tsukuba2* – *tsukuba4* are transformed to the location of *tsukuba1* and merged with the model of *tsukuba1*. We can see the integrated 3D model for *Tsukuba* is not as good as for the *Flower Garden*. This is because *Tsukuba* has more occluded areas around the lamp and the head, and many of them are not large regions. Since our algorithm on 3D model integration so far only works for large regions, so the visual effects of the finally combined 3D model are worse than that of the *Flower Garden*, and we do not calculate the PSNR values for *Tsukuba*.

### 5.7.2 Results for SSD-Based Approach

We first show the rendering from the separate 3D model at the location of *garden0* in Fig. 5.26.

Then we transform and combine the models at *garden1* – *garden5* and combine with the model at *garden0*. The six rendered images are shown in Fig. 5.27. Comparing Fig. 5.27 with Fig. 5.26, like in the case of Gabor-based approach in Fig. 5.23, we can find that more surfaces and textures behind the tree are filled (the house and slope areas on the right side of the tree). However, unlike in the Fig. 5.23, there are no obvious artifacts along the stitched borders in which two regions belonging to the house are aligned together without



(a)

(b)

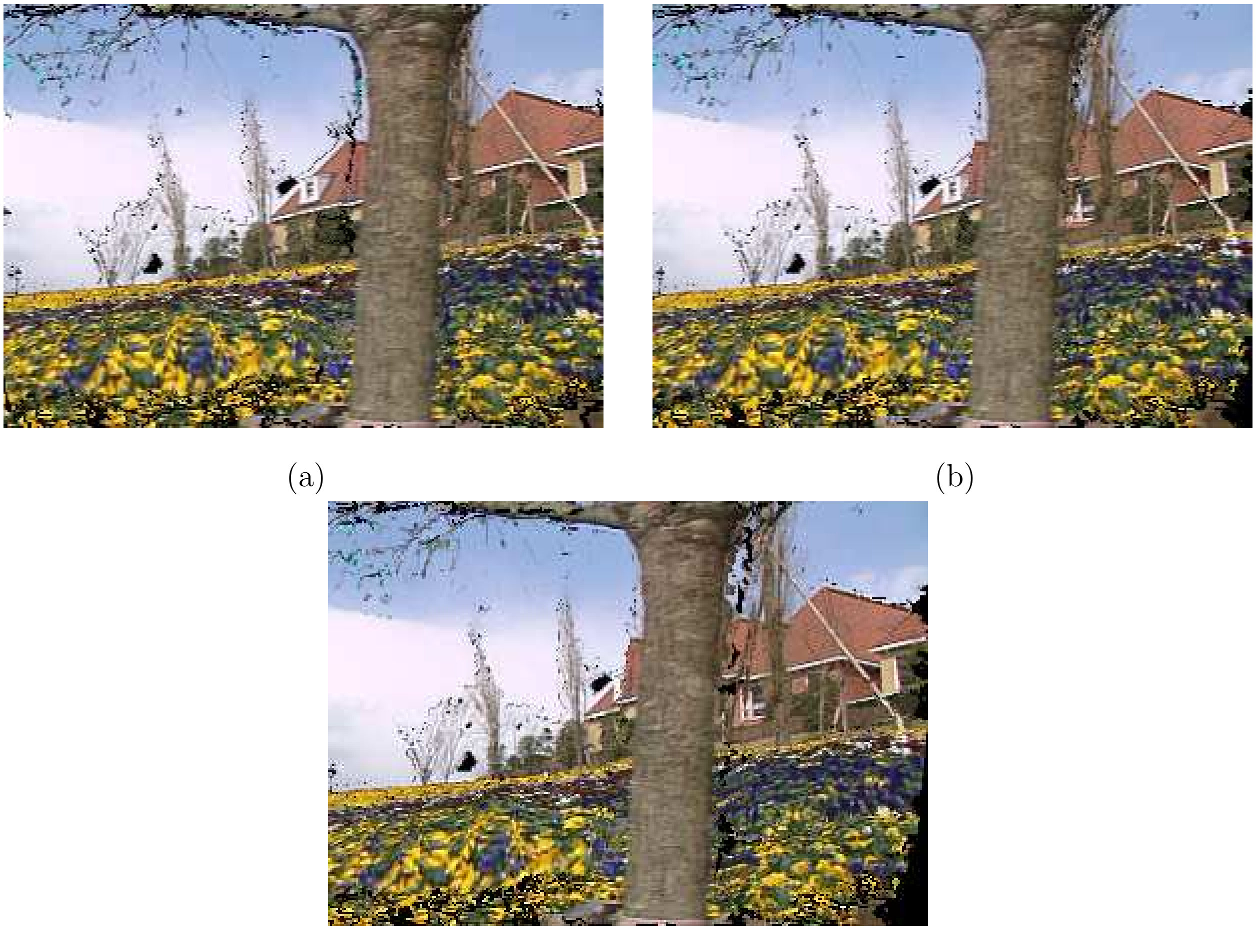
(c)

(d)

(e)

(f)

Figure 5.23: Rendering from combined 3D models at *garden1*: (a) original viewpoint; (b) – (f) novel viewpoints.

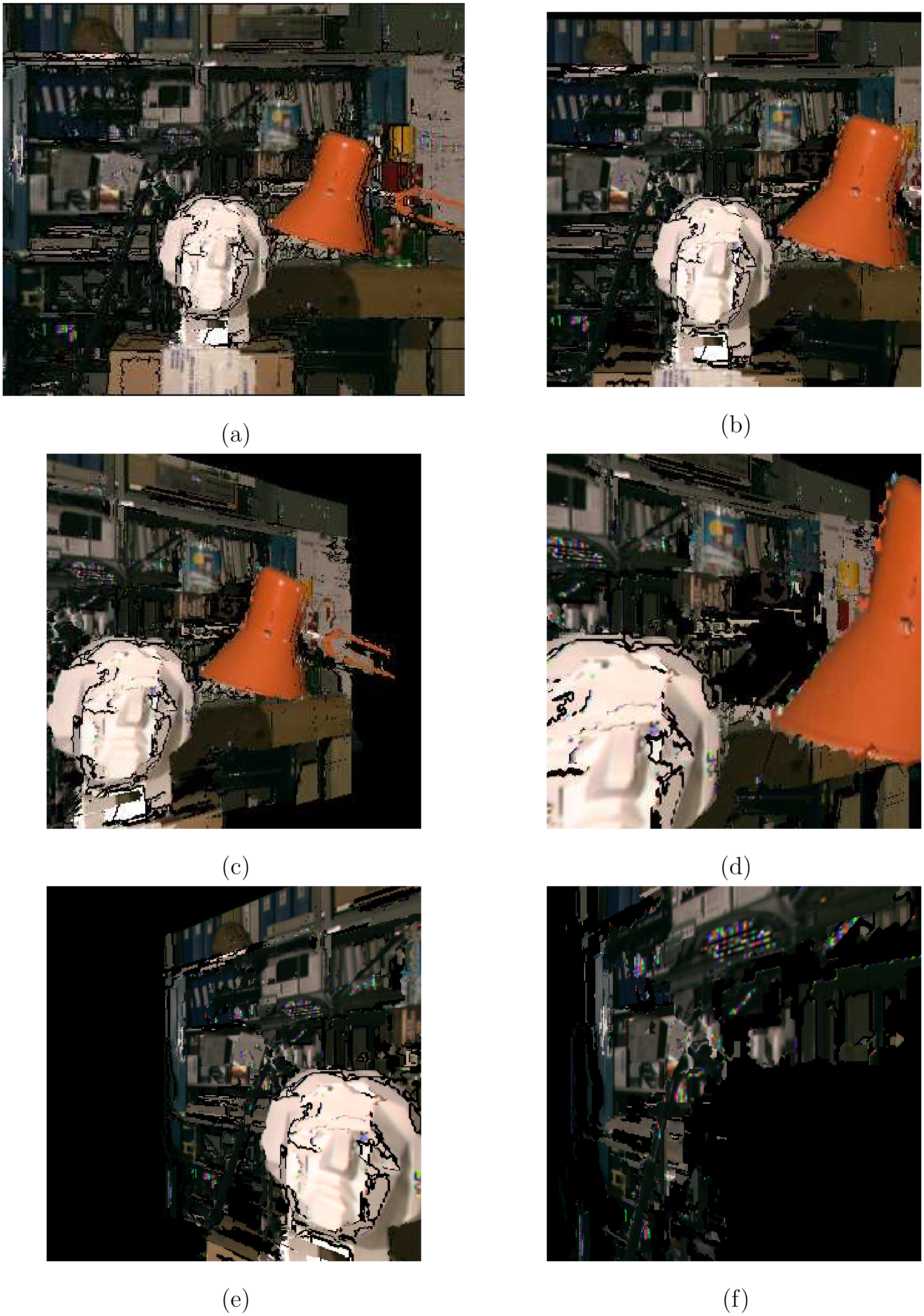


(a)

(b)

(c)

Figure 5.24: PSNR results from the final 3D model at *garden1* for Gabor-based approach: (a) *garden0*, 10.21dB; (b)*garden2*, 9.34dB; (c) *garden3*, 8.48dB.



(a)

(b)

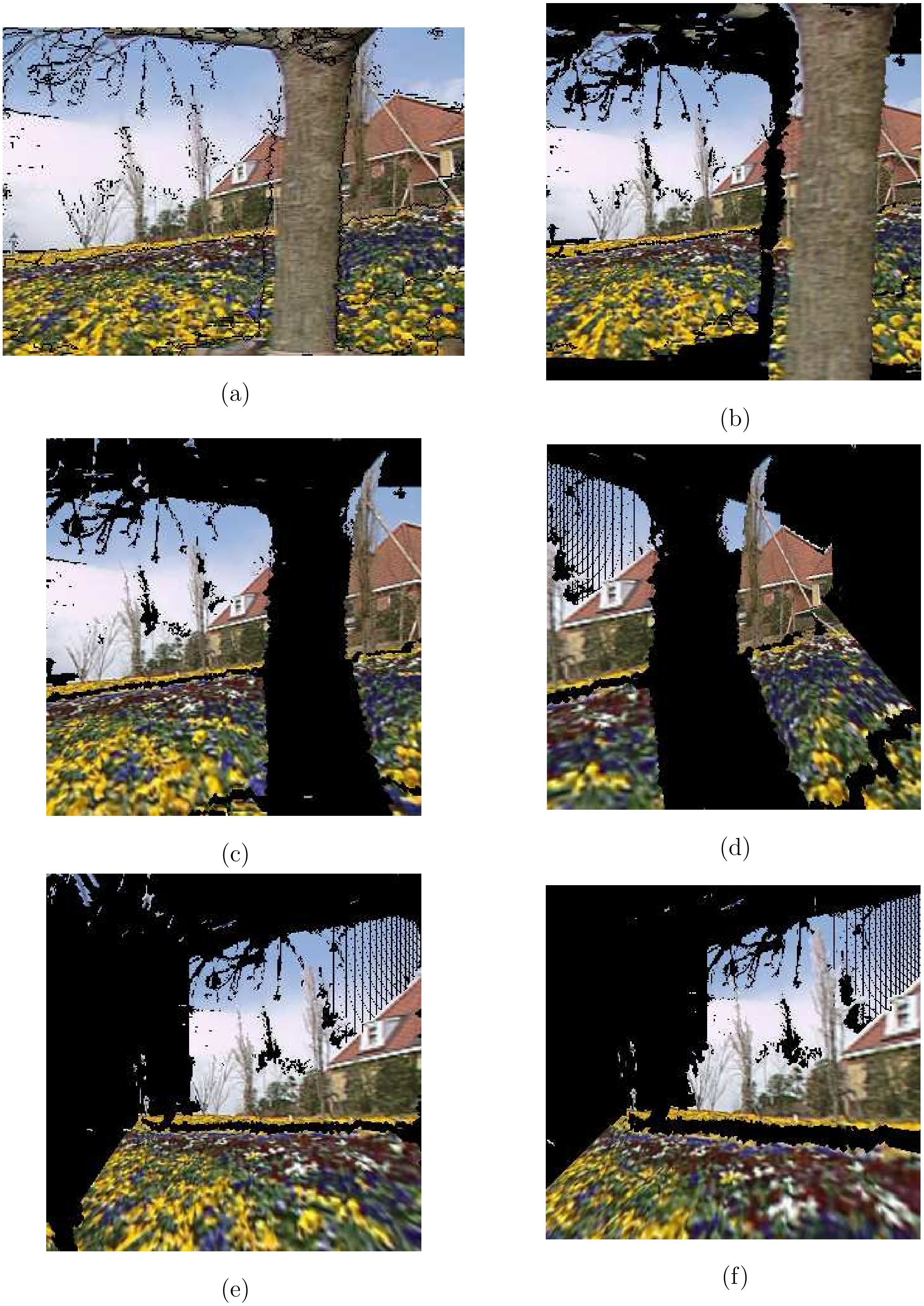
(c)

(d)

(e)

(f)

Figure 5.25: Rendering from combined 3D models at *tsukuba1*: (a) original viewpoint; (b) – (f) novel viewpoints.



(a)

(b)

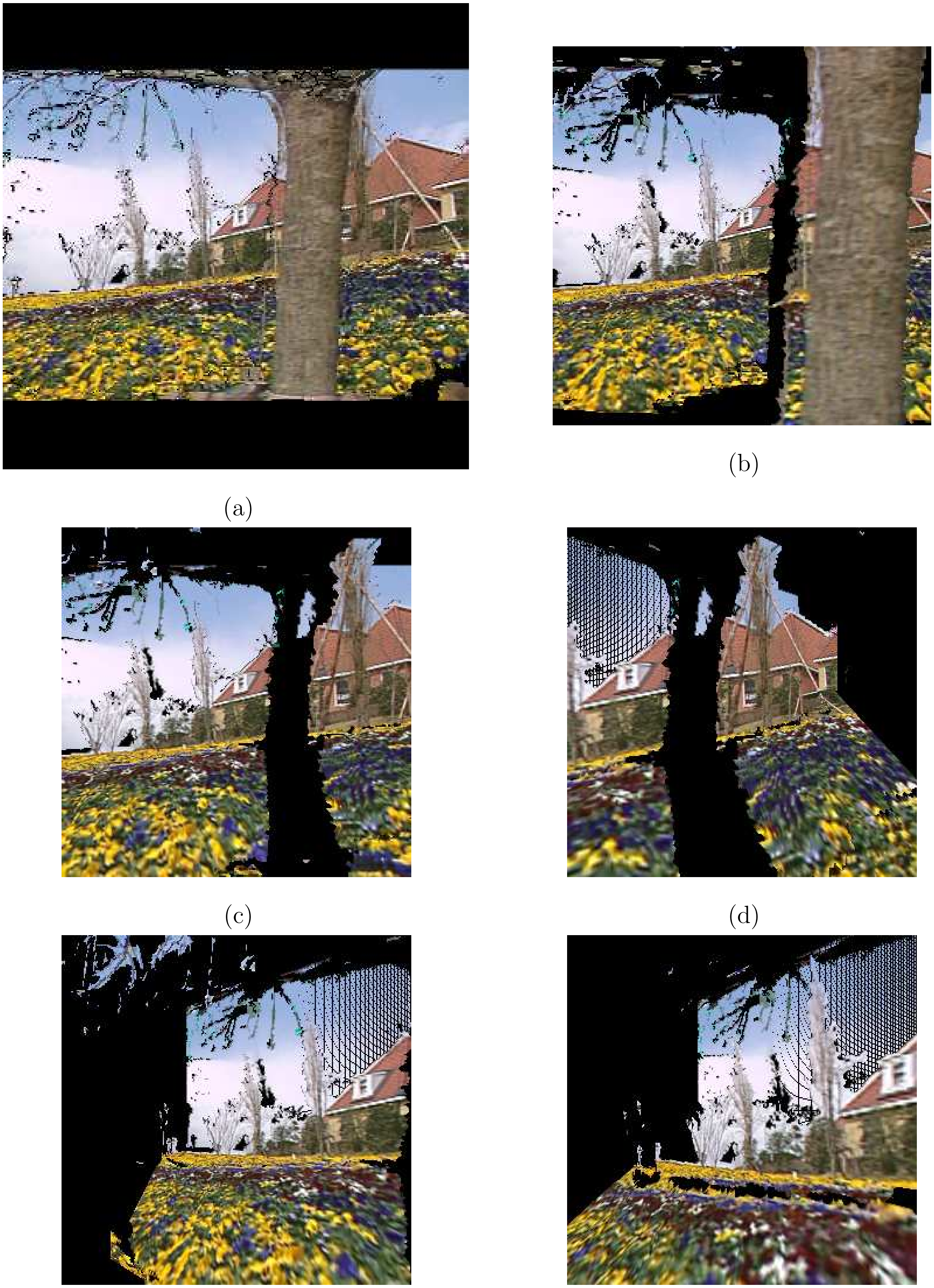
(c)

(d)

(e)

(f)

Figure 5.26: Rendering from separate 3D model at *garden0*: (a) original viewpoint; (b) – (f) novel viewpoints.



(a)

(b)

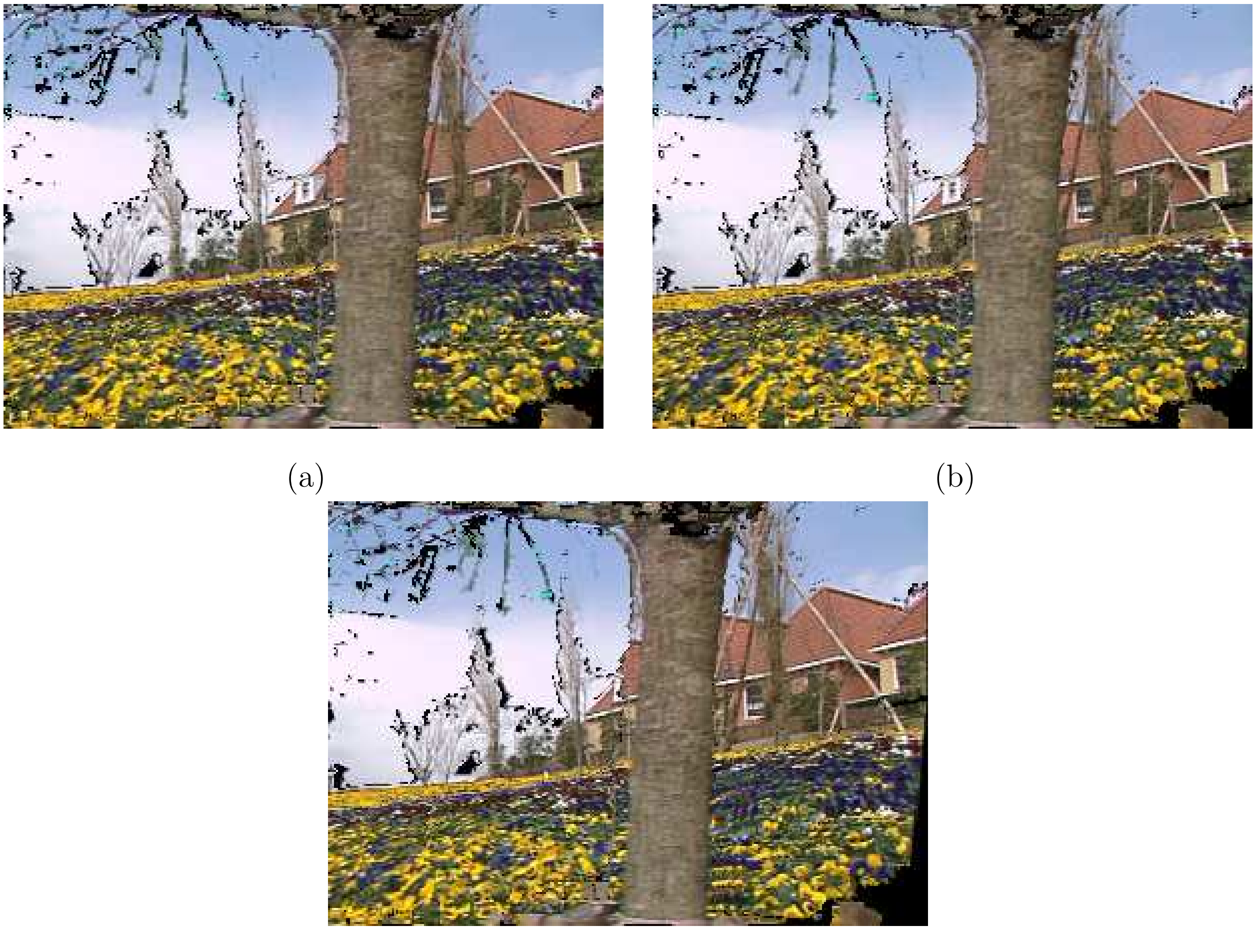
(c)

(d)

(e)

(f)

Figure 5.27: Rendering from combined 3D models at *garden0*: (a) original viewpoint; (b) – (f) novel viewpoints.



(a) (b)

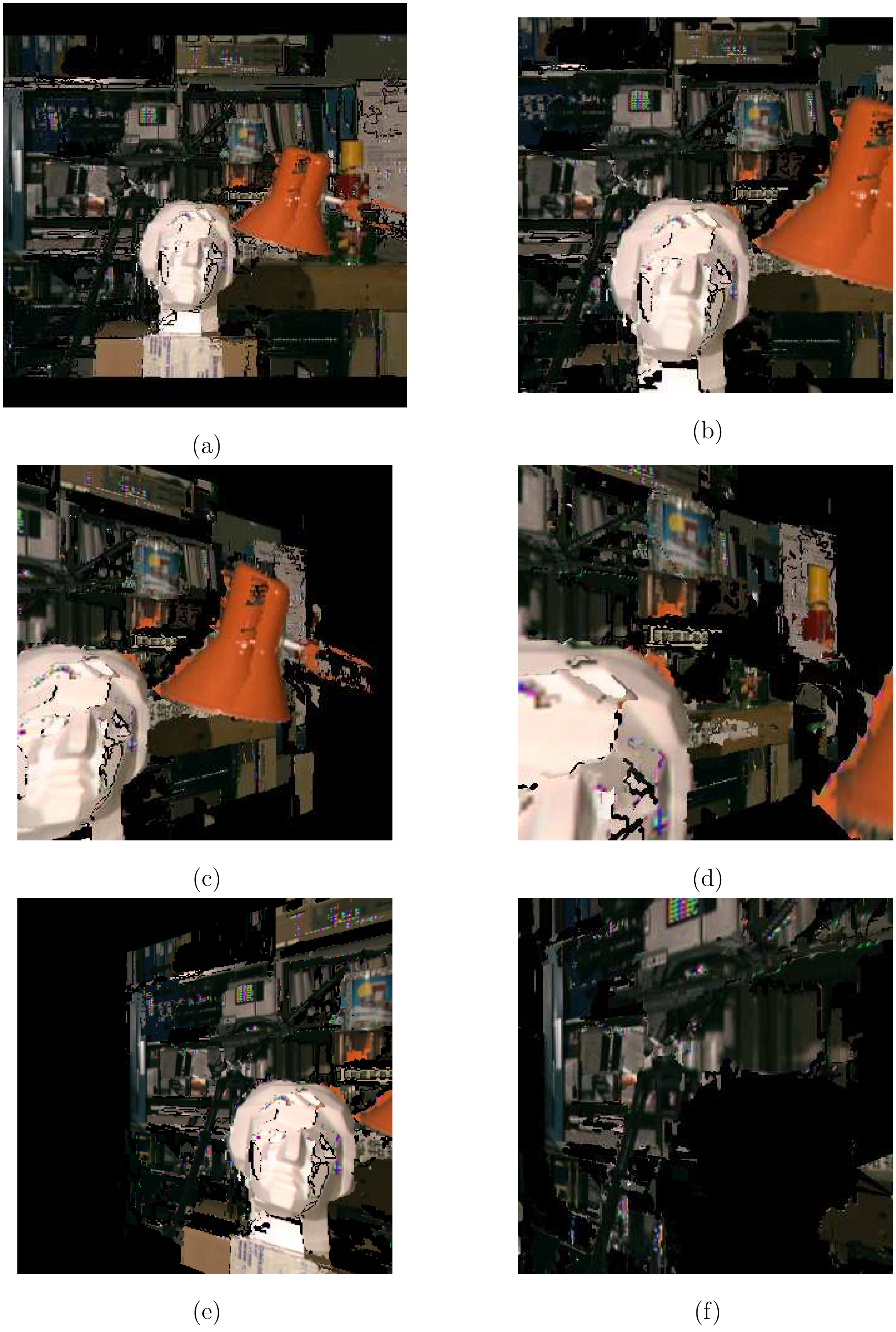
(c)

Figure 5.28: PSNR results from the final 3D model at *garden0* for SSD-based approach: (a) *garden1*, 8.71dB; (b)*garden2*, 8.56dB; (c) *garden3*, 7.85dB.

obvious vertical shift. Such results indicate that our ego-motion estimation algorithm works reasonably well. This indicates that the estimated ego-motion parameters in the SSD-based approach are better than that of in the Gabor-based approach.

We also generated the images at the locations of *garden0*, *garden2* and *garden3*, and obtained the PSNR results by comparing them with the original images. The results are shown in Fig. 5.28

Six rendered images for *Tsukuba* are shown in Fig. 5.29 for the image location of *tsukuba1* after the separate 3D models for *tsukuba2* – *tsukuba4* are transformed to the location of *tsukuba1* and merged with the model of *tsukuba1*. We can see that, similar to the reasons for



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.29: Rendering from combined 3D models at *tsukuba1*: (a) original viewpoint; (b) – (f) novel viewpoints.

*Tsukuba* in Gabor-based approach, the integrated 3D model for *Tsukuba* is not as good as for the *Flower Garden*. In some areas (e.g. the lamp), the quality of the integrated textures are even lower than in the separate 3D model cases.

## 5.8 Summary

In this chapter, we showed the procedure for the estimation of ego-motion parameters as well as the integration of separate 3D models using both of Gabor- and SSD-based approaches. These procedures work well for *Flower Garden*, but not as well for *Tsukuba*, since the integration procedure is mainly designed for large regions.

From the rendered images using the combined 3D models for *Flower Garden*, the SSD-based approach gives better visual effects as well as more accurate alignment of separate 3D models based on the estimated ego-motion parameters than that of the Gabor-based approach. This is mainly due to the fact that the initial disparity maps estimated by SSD give more complete region information and fewer outliers. However, the Gabor-based approach gives better PSNR results, but this is mainly because some small sky regions in the left part of the image have not been put to the triangulation and rendering in the SSD-based approach, since the shape of the sky region in that part is very complex.

# Chapter 6

# Conclusions and Future Work

This chapter summarizes the results and conclusions of the thesis along with contributions that we have achieved, and outlines the remaining problems and possible directions for future work.

## 6.1 Summary of Algorithms and Conclusions

In this thesis we have presented a set of approaches for the challenging problem of 3D reconstruction from images and videos, for the application of image-based virtual environment using explicit geometry with texture mapping. These approaches can be classified into two major parts: the matching with disparity estimation for the determination of separate 3D models at each image location, and the integration of separate 3D models into one model for the whole environment.

Matching is an ill-posed problem. At the beginning, we tried to handle this problem by looking into the existing methods for disparity estimation – SSD, variational regularization, graph cuts, etc. We found, as we showed in Chapter 2, that the method using blocking matching through SSD usually distorts object boundaries. Although we can reduce such problems by adjusting the block size of SSD, this makes the matching process not robust to different kinds of scenes. The method of graph cuts is mainly suitable for objects with fronto-parallel surfaces, and its complexity is very high. In order to obtain a robust matching algorithm, we used the Gabor transform, which was motivated by the work of Fleet [26][27], with the main difference that we use a cost function involving the absolute difference of Gabor filter outputs between the left and the right images, rather than comparing phase components as in the work by Fleet. Once we have a coarse disparity map estimated using the Gabor transform, we put it through variational regularization for further refinement. The refinement process is carried out by applying the gradient descent method to the Euler-Lagrange equation associated with the cost function including a data fidelity term and a regularization term. In the experiments using this kind of Gabor and variational refinement scheme, although we obtained improved disparity estimates, we found it very hard to obtain a satisfactory disparity map suitable for the purpose of 3D reconstruction. One difficult issue is that it is impossible to identify the sky region while retaining small front objects (like the twigs and sky in *Flower Garden*). Another hard problem is how to keep some object boundaries sharp and crisp. Although we can use some edge-perserving regularization functionals, as stated in Chapter 3, such functionals could not completely distinguish the real object boundaries. Thus, with the already distorted object boundaries from the coarse disparity map, there are still distortions left along the real object boundaries after the refinement process. These difficulties are due to the ill-posed nature of disparity estimation, and we believe that they could not be completely solved by improving any associated cost functions; rather we felt they should be solved by making use of region information.

Exploiting region information means that we have to segment the image first, and such segmentation should give us surfaces of the objects rather than segmenting a surface full of texture into several regions. However, without 3D information, identifying and segmenting the real surface with abundant texture inside it for one single image is another ill-posed problem. Therefore, we need to start from color-based segmentation, even if it results in the over-segmentation of some surfaces. Then, with the help of 3D information obtained from disparity maps, we can manipulate the segmented regions so that several regions under one real surface (usually with same disparity values in SSD- or Gabor-based coarse disparity maps) can be merged together in order to form one surface.

We first tried an existing color-based segmentation algorithm, mean shift [4]. Combining the obtained region information with a kind of disparity histogram analysis on each region brings us improved disparity maps in which most of the object contours remain crisp, and some large regions with zero displacement (like the sky regions) can be identified. In such disparity histogram analysis, the disparities come from a hybrid disparity estimation approach, in which pixel-based approaches and region matching are used together. The pixel-based approaches are implemented using the SSD- or Gabor-based coarse disparity estimation followed by variational refinement. The intermediate disparity maps obtained from the pixel-based approaches are then put through a region matching scheme, and the disparities finally obtained are used for the disparity histogram analysis on each region. However, one main drawback for the mean shift algorithm is that it might miss some tiny objects (like the twigs of a tree) and merge them with the surrounding or background regions. Another issue is that in order to obtain one complete region for a surface full of textures, we need to merge all the over-segmented regions by such a color-based segmentation algorithm inside the surface. These two issues led to the following reasoning: if we need to solve the over-segmentation issue for the regions coming from a color-based segmentation algorithm which could miss some tiny objects, then why not start from some simple segmentation scheme which could retain these tiny objects and solve the over-segmentation later?

The above consideration leads to our quantization-grouping process applied on the image intensity values to replace the mean shift segmentation algorithm, as described in Chapter 4. In this process, the image intensities are quantized, and the adjacent pixels with the same quantized intensity values are grouped together to form one region. In addition to the replacement of the mean shift algorithm with this quantization-grouping process, we also changed the zero displacement detection scheme in which a variational regularization process is applied to the whole image with the initial disparity for all the pixels starting from zero. For the disparities obtained this way, most of the disparitiy values for the regions with zero displacement will remain around zero, and this can be used to determine if a region has zero displacement. After such regions with zero displacement are identified, the remaining regions are put through a region merging process in which the adjacent regions under the same disparity (the initial coarse disparity maps coming from the SSD- or Gabor-based disparity estimation) masks are merged together. Then a region matching is applied to the regions after the region merging process (except for those regions determined as having zero displacement) with initial disparity values coming from the SSD- or Gabor-based coarse disparity maps. The disparities obtained using the above procedure can give us disparity maps with good visual quality in which most of the object boundaries remain crisp with the tiny objects being retained, while the regions with zero displacement (like sky regions) can be identified. Based on such disparity maps together with the region information, we can set up separate 3D models with good quality using Delaunay triangulation on each region.

For the final 3D model integration stage, we first performed ego-motion estimation using the correspondence relations of large regions between each pair of two consecutive images, since the matching results for large regions are usually more reliable than those of the small regions. Doing ego-motion estimation in this way is different from the usual bundle adjustment and ICP algorithms in which the 3D information (or 3D points) are used explicitly in the associated cost function. In our ego-motion estimation approach, we use image intensities in the cost function along with disparities rather than 3D points. Therefore, we achieved ego-motion estimation in image space with implicit 3D informaiton, rather than in 3D space using explicit 3D points. This method increases the estimation efficiency and avoids the ambiguities coming along with the noise in disparity maps (which will result in outliers in 3D space), and the latter point is particularly useful for image matching-based modeling since the matching results usually contain more noise than, e.g., a laser scanned 3D model.

After the ego-motion parameters are estimated, we can transform the separate 3D models to a reference image location and combine these models with the 3D model at that image location. In our algorithm at this stage, the integration process is only applied to large regions. With the correspondence information, the shape of each large region in the image at the reference location can be adjusted and expanded, and this procedure can eliminate most of the outliers inside the large region. Then, with the shape of each large region adjusted, all the regions in the reference image are put through a second round of matching to obtain an improved disparity map. Using such a procedure, with the disparity maps of all the images updated, we perform a second stage ego-motion estimation. Then, with the separate 3D models transformed to the reference location again, each large region is merged with their correspondent large regions and the Delaunay trangulation is applied on these merged large regions. The texture mapping is also achieved by first determining, for each large region, which parts of pixels are from which images, and then the textures can be mapped on this large region from the respective images.

From the rendered images, the SSD-based approach gives better visual quality, for separate 3D models of both *Flower Garden* and *Tsukuba*, in which there are less outliers and the regions are closer to the actual object shapes. For combined 3D models, since the SSD-based approach has better region information, the ego-motion parameters are more accurate than those of the Gabor-based approach and therefore give better alignment for separate models. Thus, our conclusion for Gabor- and SSD-based approaches is that the Gabor-based approach might give disparity maps with better visual quality at the initial pixel-based stage, while the SSD-based approach usually gives better results after region matching and the whole procedure of our algorithm. This is mainly because the coarse disparity maps estimated by SSD contain fewer outliers and give better region information. Although there are two main problems for SSD (blocking effect and distortion of object boundaries), they can be corrected by our segmentation and region manipulation processes.

## 6.2 Thesis Contributions

1. We developed disparity estimation method using Gabor filters with a new cost function. This method is robust to scenes with different complexities, in the sense that we do not need to predetermine any block size like in the case of SSD. Related publications are:

    - X. Huang and E. Dubois, *"Disparity estimation for the intermediate view interpolation of stereoscopic images"*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, pp. II-881–II-884, March 2005.

- X. Huang and E. Dubois, *"Three-view dense disparity estimation with occlusion detection"*, Proc. IEEE Int. Conf. Image Processing, pp. III-393–III-396, Sept. 2005.

2. A whole set of procedures defining a hybrid algorithm for disparity estimation. The basic idea of our algorithm is to combine the pixel-based and region-based matching schemes in order to obtain disparity results with quality that neither pixel-based nor region-based matching method could reach. Matching is a long standing problem, and we believe that it should be solved in a joint analysis approach utilizing both pixel and region information, rather than depending on one or two functionals from only pixel-based or region-based method. There are two versions of our algorithm depending on the segmentation methods:

   (A) Using segmentation algorithm 'mean shift', we developed a hybrid disparity estimation algorithm in which the previous disparity results obtained from the pixel-based procedures are combined with segmentation results for further region matching as well as for disparity histogram analysis within each region, so that the disparity in each of the segmented regions can be jointly refined and hence eliminate the outliers in each region. Most of the background regions with occlusions can also be identified through histogram analysis in order that the true disparity values for such regions can be restored from the results of pixel-based procedures. Related publications are:

      – X. Huang and E. Dubois, *"3D reconstruction based on a hybrid disparity estimation algorithm"*, IEEE Int. Conf. Image Processing, Oct. 2006.

      – X. Huang and E. Dubois, *"Region-based motion analysis and 3D reconstruction for a translational video sequence"*, Int. Symposium on 3D Data Processing, Visualization, and Transmission, Jun. 2006.

   (B) We proposed a new and simple quantization-grouping process to replace a color-based segmentation algorithm (mean shift) in order that the very small regions can be retained after segmentation. Then a region merging and manipulation process is applied on the regions, which gives a better region matching results with fewer outliers and thus a better 3D model. In addition, most of the sky regions can also be detected.

3. We used an ego-motion estimation method performed in image space with implicit 3D information, rather than the traditional bundle adjustment and ICP which is performed in 3D space. Thus we can perform ego-motion estimation with more efficiency, and avoid the ambiguities caused by most of the outliers from separate 3D models if performing ego-motion estimation using bundle adjustment or ICP. Here we want to state that our idea of using large region to perform ego-motion estimation in image space came out at the same period when Rav-Acha's technical report was published [50], therefore we just implemented our idea by making use of the cost functions in [50].

4. We proposed a 3D model integration method utilizing large regions with texture mapping, in which corresponding large regions from different 3D models can be stitched together. The stitching process is achieved mainly using pixel information without the manipulation of triangular meshes which come along with different 3D models. This is a main difference between our method and the existing common methods and thus the efficiency of stitching process is also increased.

In summary, we have contributions in both areas of disparity estimation and the 3D integration process. Our algorithms contain novel ideas that have dramatic differences with current approaches in both areas. This thesis can be seen as the initial implementation with initial results for our algorithms.

## 6.3 Remaining Problems and Future Work

### 6.3.1 Disparity Estimation and Matching

Our hybrid disparity estimation algorithm only works for two consecutive images. A better matching result is expected if we can extend our matching procedure to multiview matching. For example, we estimated the disparity maps between each pair of consecutive images from *garden5* to *garden0*. Based on these disparity maps, if we do the disparity estimation between *garden0* and *garden5* directly, then more details should be coming out (e.g., the small tree with slanted posts right in front of the house, they should show different disparty values after the disparity between *garden0* and *garden5* is estimated), and thus give more detailed 3D information.

In addition, if we want to capture the 3D structure of an environment by taking unconstrained videos, we should also solve the problem of extracting 3D information from a zooming video sequence. For doing this, we should use stereo cameras since in a zooming sequence from a single camera, the pixels on objects which lie around the centre of the zooming might give zero motion values, and thus the associated 3D information is ambiguous. Thus, if we combine the 3D information from stereo image pairs in each position of the zooming, we could solve such ambiguities. Therefore, combining our disparity estimation with 2D motion estimation for arbitrary stereo video sequences is another important direction.

### 6.3.2 3D Model Integration

From the model integration results on *Tsukuba*, we can see that our integration scheme needs to be further improved. At this stage, our integration scheme only work for large regions. We should extend it to smaller regions. In addition, our integration scheme can handle the expansion of large regions which could extend a large region to behind the occluding object (like filling the house area behind the tree in *Flower Garden*), but could not handle the situation once the two large should be connected to form one region (like the connection between the right and left areas of the tree and completely filling the pixels behind the it in

*Flower Garden*). This problem should be solved by identifing more situations in the model merging process.

# Bibliography

[1] H. Shum and S. Kang, "A review of image-based rendering techniques," in *Proc. SPIE Visual Communications and Image Process.*, pp. 2–13, 2000.

[2] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 11, pp. 1–18, 2001.

[3] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. IEEE Int. Conf. Computer Vision*, vol. 2, pp. 508–515, July 2001.

[4] D. Comaniciu and P. Meer, "Mean-shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, 2002.

[5] J. Ohm and E. Izquierdo, "An object-based system for stereoscopic viewpoint synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 801–811, 1997.

[6] M. Lhuillier and L. Quan, "Image-based rendering by joint view triangulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 1051–1063, 2003.

[7] M. Lhuillier and L. Quan, "Match propagation for image-based modeling and rendering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 1140–1146, 2002.

[8] X. Sun and E. Dubois, "A matching-based view interpolation scheme," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, 2005.

[9] A. Mancini and J. Konrad, "Robust quadtree-based disparity estimation for the reconstruction of intermediate stereoscopic images," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 3295, pp. 53–64, 1998.

[10] M. Ouali, D. Ziou, and C. Laurgeau, "Inaccurate phase-based disparities removal," in *Proc. 6th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 973–976, 1999.

[11] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH'98*, pp. 231–242, 1998.

[12] C. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. SIGGRAPH'99*, pp. 291–298, 1999.

[13] S. Seitz and C. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Intern. J. Comput. Vis.*, vol. 35, pp. 151–173, 1999.

[14] P. Narayanan, P. Rander, and T. Kanade, "Constructing virtual worlds using dense stereo," in *Proc. IEEE Int. Conf. Computer Vision*, pp. 3–10, 1998.

[15] S. Se and P. Jasiobedzki, "Instant scene modeler for crime scene reconstruction," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 3, pp. 123–130, 2005.

[16] O. Faugeras and S. Laveau, "Representing three-dimensional data as a collection of images and fundamental matrices for image analysis," in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 689–691, 1994.

[17] P. Havaldar, M.-S. Lee, and G. Medioni, "View synthesis from unregistered 2-D images," in *Graphics Interface*, pp. 61–69, May 1996.

[18] R. Koch, M. Pollefeys, and L. V. Gool, "Automatic 3D model acquisition from uncalibrated image sequences," in *Proc. Computer Graphics Int.*, pp. 597–604, 1998.

[19] X. Huang and E. Dubois, "Disparity estimation for the intermediate view interpolation of stereoscopic images," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, pp. 881–884, 2005.

[20] X. Huang and E. Dubois, "Three-view dense disparity estimation with occlusion detection," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 393–396, 2005.

[21] L. Alvarez, R. Deriche, J. Sanchez, and J. Weickert, “Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach,” *Journal of Visual Communication and Image Representation*, vol. 13, pp. 3–21, Jan. 2002.

[22] L. Robert and R. Deriche, “Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities,” in *Proceedings of the 4th European Conference on Computer Vision*, Apr. 1996.

[23] C. Strecha, T. Tuytelaars, and L. V. Gool, “Dense matching of multiple wide-baseline views,” in *Proc. IEEE Int. Conf. Computer Vision*, vol. 2, pp. 1194–1201, 2003.

[24] H. Kim and K.Sohn, “Hierarchical disparity estimation with energy-based regularization,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 14–17, 2003.

[25] H. Kim and K.Sohn, “3D reconstruction from stereo images for interactions between real and virtual objects,” *Signal Process., Image Commun.*, vol. 20, pp. 61–75, 2005.

[26] D. Fleet, Jepson, and M. Jenkin, “Phase-based disparity measurement,” *CVGIP: Image Understanding*, vol. 53, pp. 198–210, 1991.

[27] D. Fleet, “Disparity from local weighted phase-correlation,” in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 48–56, 1994.

[28] J. Goulermas, P. Liatsis, and T. Fernando, “A constraint nonlinear energy minimization framework for the regularization of the stereo correspondence problem,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 550–565, 2005.

[29] G. Wei, W. Brauer, and G. Hirzinger, “Intensity- and gradient-based stereo matching using hierarchical Gaussian basis functions,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 1143–1160, 1998.

[30] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proc. of Alvey Vision Conf*, pp. 147–151, 1988.

[31] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[32] E. Vincent and R. Laganière, “Models from image triplets using epipolar gradient,” in *Vision, Video, Graphics*, pp. 143–150, July 2003.

[33] A. Brint and M. Brady, “Stereo matching of curves,” *Image and Vision Computing*, vol. 8, pp. 50–56, 1990.

[34] M. Nasrabadi, “A stereo vision technique using curve-segments and relaxation matching,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 566–572, 1992.

[35] L. Robert and O. Faugeras, “Curve-based stereo: Figural continuity and curvature,” in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 57–62, 1991.

[36] J. McIntosh and K. Mutch, “Matching straight lines,” *Comput. Vis. Graph. Image Process.*, vol. 43, pp. 386–429, 1998.

[37] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” in *Microsoft Technical Report, MSR-TR-2001-81*, 2001.

[38] H.-H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 565–593, 1986.

[39] M. Snyder, “On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 1105–1114, 1991.

[40] P. Boggs, “Sequential quadratic programming,” *Acta Numerica*, vol. 1, pp. 1–52, 1995.

[41] B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[42] B. Lucas and T. Kanade, "An iterative image registration technique with application to stereo vision," in *Proc. DARPA Image Understanding Workshop*, pp. 121–130, 1981.

[43] C. Slama, *Manual of Photogrammetry.* Falls Church, VA: American Society of Photogrammetry, 1980.

[44] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proc. Int. Workshop on Vision Algorithms: Theory and Practice*, pp. 298–372, 1999.

[45] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from `http://www.ics.forth.gr/~lourakis/sba`.

[46] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 239–256, 1992.

[47] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Int. Conf. 3D Digital Imaging and Modeling*, pp. 145–152, 2001.

[48] D. Mount and S. Arya, "ANN: library for approximate nearest neighbor searching," in *Proc. Center for Geometric Computing Second Ann. Workshop Computational Geometry*, 1997.

[49] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

[50] A. Rav-Acha, G. Engel, and S. Peleg, "Minimal aspect distortion (MAD) mosaicing of long scenes," Tech. Rep. HUJI-CSE-LTR-2007-111, Hebrew University, Jun. 2007.

[51] G. Roth and E. Wibowo, "An efficient volumetric method for building closed triangular meshes from 3-d image and point data," in *Proc. Graphics Interface (GI)*, pp. 173–180, 1997.

[52] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. SIGGRAPH*, pp. 303–312, 1996.

[53] M. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus surfaces for modeling 3D objects from multiple rangeimages," in *Proc. IEEE Int. Conf. Computer Vision*, pp. 917–924, 1998.

[54] W. Lorenen and H. Cline, "Marching cubes: a high resolution 3D surface reconstruction algorithm," in *Proc. SIGGRAPH*, vol. 21, pp. 163–169, 1987.

[55] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proc. SIGGRAPH*, vol. 26, pp. 311–318, 1994.

[56] R. Pito, "Mesh integration based on co-measurements," in *Proc. IEEE Int. Conf. Image Processing*, pp. 397–400, 1996.

[57] X. Huang and E. Dubois, "3D reconstruction based on a hybrid disparity estimation algorithm." Int. Conf. Image Processing, October 2006.

[58] X. Huang and E. Dubois, "Region-based motion analysis and 3D reconstruction for a translational video sequence," in *3DPVT*, 2006.

[59] Y. Wei and L. Quan, "Region-based progressive stereo matching," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 1, pp. 106–113, 2004.

[60] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," in *Proc. IEEE Int. Conf. Computer Vision*, pp. 1073–1080, 1998.

[61] `http://www.caip.rutgers.edu/~comanici/segm_images.html`.

[62] J. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, pp. 348–365, 1995.

[63] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. European Conf. Computer Vision*, vol. 4, pp. 25–36, 2004.

[64] J. Kim and T. Sikora, “Hybrid recursive energy-based method for robust optical flow on large motion fields,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 129–132, 2005.