Université d'Ottawa • University of Ottawa

University of Ottawa
Faculty of Engineering
School of Information Technology and Engineering
Ottawa-Carleton Institute for Computer Science

The Disparity Pyramid: An Irregular Pyramid Approach for Stereoscopic Image Analysis

M.C.S. Thesis

By

Rimon Elias

Supervised by

Dr. Robert Laganière

January 1999

0-612-46566-7

Abstract

The disparity pyramid is a proposed solution for both problems of disparity estimation and object detection using a stereoscopic pair of images. This pyramid can be classified within the class of irregular pyramids. The hierarchy of the pyramid is achieved by successive calculations of a set of levels. These calculations are done over a disparity range to get the difference in intensities between the two images. A cell can survive at the next level if it has the minimum difference within its neighbourhood. This minimum difference plays the role of an indicator of how near this cell, in terms of intensity, to another cell in the other image. This means that the disparity pyramid takes into account the information of intensity of the images under consideration; in other words, it is a data-dependent structure. A cell is a root, if it cannot be linked to any of its neighbours because it is far enough from all cells in its neighbourhood. Unlike some other pyramids, and according to the input values and/or the distance between disparity vectors and/or number of objects contained in the scene, the top level of this pyramid may consist of more than one cell. Thus, in terms of geometry, its shape looks like an incomplete pyramid with a flat top level and irregular sides.

Keywords

Irregular pyramid, multiresolution, multiscaling, disparity estimation, object extraction, stereo vision.

Acknowledgments

I want to express my gratitude to my supervisor Dr. Robert Laganière for his continuous encouragement. He was very helpful to me over the period of my study. When I began my study in this area, I felt like a rat in a maze!! But he knew how to give the information step by step and at the right time. Also, I would like to thank Dr. John Oommen, from Carleton University. He always encourages me to do my best in research. Thanks to both of them.

Contents

List of Figures

List of symbols

α	A factor used to compute the contrast between regions.	67
μ_{ijh}	Mean of the receptive field of the cell (i, j, h) at level h.	60
$Brothers_{ijh}$	A list of the brothers of the cell (i, j, h) $\equiv$ Set of neighbours of (i, j).	50
$Children(\rho)$	List of children of cell ρ.	22
$color_{ijh}$	Intensity (or colour) of the cell (i, j, h)	95
$contrast_{ijh}$	Minimun contrast between the cell (i, j, h) and its surrounding as a function of s_{ijh}.	104
$contDst_{ijh}$	Minimun distance between the disparity vectors of the cell (i, j, h) and its surrounding as a function of s_{ijh}.	104
$(disp_x, disp_y)_{ijh}$	A 2D disparity vector associated with F_{ijh}.	49
$(dx_{min}, dx_{max}, dy_{min}, dy_{max})$	The disparity range.	65

F_{ijh}	The minimum disparity value associated with the cell (i, j, h) i.e. at level h.	49
$F_{xy}(dx, dy)$	The disparity value for the cell (x, y) using the disparity vector (dx, dy).	42
$f_n(x, y)$	The intensity of the cell (x, y) in the image n.	42
I_h	The image I at level h of the pyramid.	19
(i, j, h)	Region characterized by the cell (i, j) at the level h of the pyramid.	48
L_1	Relation L_1 or L_2 defined between two cells ρ and β on different levels.	22
$minDif_{ijh}$	Minimum difference between this cell and the least contrasted brother.	103
min_cont	Minimun contrast needed to distinguish between two regions.	69
min_dist	Minimun distance between two disparity vectors needed to distinguish between two regions.	69
min_size	Minimun size of the receptive field needed to decide if the cell is a root.	69
$\mathcal{N}$	The set of integer numbers.	19
$\mathcal{P}$	Symbol of a plane in 3D space.	36
P	The symbol of a pyramid.	19

$Parents(\rho)$	List of parents of cell ρ.	22
p_{ijh}	A state variable indicates if the cell (i,j,h) will survive at level $h+1$ or if it is a root.	49
$parent_{ijh}$	The parent of the cell (i,j,h), i.e. pointing to level $h+1$.	22
q_{ijh}	A state variable indicates if the cell (i,j,h) is a good candidate to survive at level $h+1$.	49
R	Relation R defined between two neighbour cells ρ and β.	21
$Receptive_{ijh}$	A list of all descendants of the cell (i,j,h) at the level h.	24
r	Reduction factor used in the bottom-up process.	26
$Support_{ijh}$	Support of $(i,j,h) \equiv$ Set of neighbours of $(i,j,h) \cup (i,j,h)$ itself.	48
s_{ijh}	Size of the receptive field of the cell (i,j,h) at level h.	67
w	Number of links between a parent and its children.	26
(x,y)	A cell, a node or, in general, a pixel specified by x and y coordinates.	19
x_ρ	A random variable associated with the cell ρ.	32

Chapter 1

Introduction

Vision works as a set of integrated processes that detect and extract information from the visual field, construct hypotheses, weigh hypotheses, build descriptions and control motor processes. All these processes may not be performed accurately without good understanding of the three-dimensional geometry of the scene. In the meantime, it is not possible to deduce the three-dimensional geometry of a scene from a single image. But on the other hand, as soon as one has multiple images taken from slightly different viewing angles, it is possible to determine the relative position of points in at least two images and the depth of every object included in the scene. At this point, one of the most important topics arises in computer vision, examining the disparity between two perspective views for the same scene, and how to estimate the disparity value between every pair of matching points.

1.1 Disparity estimation

The disparity estimation is the search for corresponding image points in a stereoscopic image pair, where corresponding image points are projections of the same object point into both image planes. This is a very attractive research area and many methods have been developed in this area.

Over the last two decades, all methods developed in this area can be classified into two major groups (Chapter 2 is another classification survey on many of such methods):

1. Intensity-based matching: where the algorithms proposed attempt to find good matching among different gray levels of perspectives under consideration assuming that there is some similarity among them [BB82, GGB84, Han89, Fua93, CC90]. However, this assumption may be applicable for the textured regions and may be wrong at smooth regions and occlusion areas.

2. Feature-based matching: where the algorithms proposed first extract some important features, such as edge segments or contours, to match them between the two images. Each image can be implemented as a graph with features representing the nodes and some geometric relations representing the links. In this category, some heuristics may be introduced to reduce the complexity. The algorithms of this category are usually fast as only a small subset of the pixels are processed. However, as these methods depend mainly on feature detection, failing to do so efficiently may cause the whole process to fail [Ull79, SH81, BT80, CH84, Rad84, MW87, HS89, WAH92].

Disparity estimation has been proved to be useful in:

1. Understanding of the three-dimensional structure of a scene using two different perspectives [Fau96, DA89];
2. Stereo coding [TGS96];
3. Synthesis of new perspectives [PPD98].

Practically, the most important processes of vision including the one mentioned above, the disparity estimation, deal with huge amounts of raw data. At this level, the data contained in the image have not been processed or reduced to more meaningful tokens, e.g. curves, lines, edges, corners or textural features. So such a process needs very sophisticated computer system that:

- should be very fast to handle these quantities of raw data in a short time, and the idea of parallel processing satisfies this demand;
- should provide a structure that simplifies the transformation of such data into curves, edges, lines, corners and textural features.

1.2 The pyramidal architecture

The "pyramid" is a parallel hierarchical structure which satisfies those demands. At the same time, it is considered a very powerful computation tool providing appropriate data structure (graphs, arrays, lists and trees) to handle this amount of raw data. It suggests new and powerful ways to solve the complicated mathematical problems of vision. Pyramids are therefore important to computer vision not only as a fast architecture, but also as model of computation, bringing new methodology to the algorithm-design activity.

Pyramids could be called a computer scientist's approach to vision-integrating two-dimensional arrays or graphs with trees. These trees permit lots of interesting data structures like binary trees, quad trees, linked lists, arrays and graphs. And yet pyramids have more general appeal. In this work, we will use this powerful, fast, hierarchical structure, the pyramid, to solve the disparity estimation problem by segmenting the scene into parts according to their disparity values, i.e., the goal of this work is to achieve not only a good disparity map or an accurate point correspondence but a reasonable segmentation of the scene into layers of constant disparity values.

1.3 Organization of the thesis

In order to understand the topic of stereoscopic vision, a literature review seems to be necessary. Chapter 2 presents a survey on previous methods introduced to achieve a good image understanding using a stereo pair of images for the same scene. These methods are classified into many different groups. Each group has its own strategy but sometimes more than one strategy are grouped together in order to achieve better results.

Chapter 3 of this thesis will discover different types of pyramid architectures pointing to their advantages and disadvantages and finally selecting an appropriate architecture to be modified for the task under consideration.

The second important point mentioned above, the disparity estimation, is the subject of Chapter 4. This chapter begins with the basic scientific description of image formation. It then proceeds to explain how the binocular vision works and the disparity estimation problem appeared as a result of it. This chapter ends up with a suggested formula to calculate the disparity.

In Chapter 5, we will present an irregular pyramid model which solves the disparity estimation problem, and with the same technique, different objects contained in a scene can be extracted. This technique, named the disparity incomplete pyramid, is explained in detail. Some practical problems appeared during the implementation are presented. Then the parameters to the algorithm along with their roles in controlling the process are discussed.

Chapter 6 presents the experimental results of the suggested model. Different types of stereoscopic scenes are presented. Also this chapter shows that the technique can work on mono images.

Chapter 7 discusses all the work which has been done. This chapter also presents the limitations of the proposed pyramid, comments on the results and suggests ideas for a future work.

Finally, Appendix A is a detailed algorithm to build this model.

Chapter 2

Stereoscopic Vision: A Survey

2.1 Introduction

Since the 70's, a lot of work has been done in understanding stereoscopic vision, achieving good corresponding points matching algorithms, estimating the disparity between a pair of stereo images and suggesting different techniques. A very good review can be found in [NP84, NA91]. Bibliographic reviews of this subject can be found in [Fau96, Lus87, SV86, Tos87, LL86, Rho84]. In what follows, we will give a short survey on different methods proposed and their constraints.

2.2 Neurophysiology

Human vision represents a biological solution to the stereoscopic vision problem so as the vision of certain animals. Some of the biological processes have been studied in neurophysiology. This can be found in [HW70, HW73].

The idea to simulate the biological process should face the human brain and its visual system and their extraordinary complexity taking into account that the brain has more than ten billion neurons and more than a thousand connections per neuron and the retina contains more than a hundred and fifty million light receptors. Even with the aid of the most sophisticated tools that we have, and even if the biological vision were completely understood, this would not necessarily lead to an artificial solution.

2.3 Correlation

This type of algorithms was first applied in 1977 [KMM77]. The matching process was done by searching for correlation peaks between regions. The authors used epipolar (see Section 4.3.1), geometric similarity[1], uniqueness[2] and continuity[3] constraints.

Another attempt, [WM88], supposed that a single pair of images is not a reliable source of depth information and one solution is to integrate the information from several image pairs. They developed a stereo algorithm based on correlation matching that is suited to being integrated into a proposed feedback loop.

In [Nis83], what is called the PRISM system was developed. This system obtains image signs from zero-crossing extraction at different resolutions

[1] The fact that there is a small separartion between the two cameras leads to geometric similarity.

[2] The uniqueness constraint assumes that a point of image I_1 has only one corresponding point in image I_2.

[3] If an object can be represented by sufficiently dense geometric primitives, and its surfaces are sufficiently regular, then this object can be represented by subsets of neighbouring primitives in a space which can be traversed continuously.

and correlates them. This correlation method are also used in [Mor80, Gen80, Tsa83, Tsa86, Nev76]

2.4 Dynamic programming

A dynamic programming technique was presented in [OK85] performing searching for the corresponding points within the same scanline "intra-scanline search" which can be treated as the problem of finding matching path on a 2D search plane whose axes are the right and the left scanlines. Another search is "inter-scanline search" which is a stack of the 2D search planes.

Another dynamic programming algorithm is presented in [OTI87] to find the optimal correspondence between the left and right intervals delimited by edges.

A dynamic algorithm which uses the order[4] constraint was presented in [BB81]. The final solution may be sub-optimal [NA91].

2.5 Prediction and verification

In [NA91], N. Ayache presented an algorithm to reconstruct real objects of the scene. This algorithm proceeds in three different phases; The first is a prediction where a certain number of connected components of the graph, and those associated with real objects in the scene, are marked by hypothesis prediction. The second stage is a propagation where each of the previous marked components is checked. This results in a set of real and

[4] This constraint assumes the preservation of homologous points along both epipolar lines.

virual matches. The third stage is the validation where the matches are compared amongst themselves.

A derived algorithm from the previous is presented in [Hen87]. It was developed at the University of Pennsylvania for binocular stereo experiments. Another development of the previous algorithm was presented in [Kro87]. This results in an active stereo system which controls many camera parameters.

2.6 Region-based matching

A different type of algorithms was presented in [XKT] based on matching regions. The algorithm consists of four main steps: segmenting the images into regions, calculating the area and perimeter for each region then matching the regions and assigning disparities.

2.7 Sub-graph isomorphism

In this type of algorithms, each image is represented as a graph. This was introduced in [SH87b, SH87a]. The algorithm looks for sub-graph isomorphism to achieve matching. The results obtained are good but the algorithm is relatively slow.

2.8 Relaxation

The first attempt using this type of algorithms was introduced in [MP76]. To achieve their goal, the authors used the epipolar, uniqueness and continuity constraints. This algorithm is initialized by the matches that satisfy the epipolar constraint. Then a weight is associated to each match and an iterative algorithm is applied: for every iteration and for every match, if the match violates the uniqueness constraint, an inhibitory process is applied to reduce its weight. On the contrary, an excitatory process is applied to increase the weights of matches with similar disparities. After a few iterations, this algorithm converges to a solution. However, in this algorithm, matching errors occur in regions that violate the continuity constraint.

An improved parallel implementation version of the previous algorithm was introduced in [DP86]. This version uses the order constraint on the *Connection Machine* to achieve stereo matching in a very short period of time (about one second).

An algorithm was introduced in [Gri81] uses the contour points obtained from zero-crossings and the concept of multiresolution (see Section 2.9) where matches conflicts at finer scales are resolved by those obtained at coarse scales. The author used epipolar, geometric, uniqueness and continuity constraints. This brings us to an important approach, named the pyramids.

2.9 Pyramids

A few attempts were done to use the concepts of the regular pyramids (see Section 3.4 on Page 24) in order to estimate the disparity between a pair of stereo images. S. Barnard introduced a technique [Bar89] to apply the quad-pyramid (Figure 3.2 on Page 20) to both images in order to get low-resolution versions of them so that the disparity estimation at the top levels would be easier. Then he used coarse-to-fine strategy to refine the results towards the base of the pyramid.

A similar idea is presented in [WAH88]. The pair of images is preprocessed to generate a collection of attribute images (e.g., edges, corners). The quad-pyramid is applied to these attribute images where blurring occurs through the bottom-up process. At the highest level, the disparity (called displacement) is computed. Then the refinement or coarse-to-fine approach through the top-down process (see Section 3.1.2.2 on Page 14) is applied until we get the final results at the original resolution.

The same top-down process applied to the quad-pyramid (Figure 3.2 on Page 20) could be found in [HPB92] where what is called a coarse-to-fine relaxation algorithm is implemented. In this algorithm a value calculated for a cell at level h is passed to its four children at the lower level $h-1$ through an interpolation process then this value is enhanced using observations from the original image through another process called relaxation process (see Section 2.8).

A very recent attempt using the quad-pyramid is presented in [SK98]. The mapping between the pair of images is computed at each level through the top-down process. The number of candidate mappings at each level is constrained by using the mapping at the upper level of the hierarchy.

Another recent attempt to estimate the disparity map using hierarchical basic functions is presented in [WBH98]. Although the results presented indicate a good disparity map estimation, the method does not introduce any scene segmentation according to disparity values.

In Chapter 5, an irregular pyramid technique is presented to estimate the disparity between a pair of stereo images where the rigidity of the regular pyramids presented above is avoided. For more discussion on the pyramids, refer to Chapter 3.

2.10 Trinocular stereo vision

This refers to the use of three cameras instead of two to view a scene. This will force an additional epipolar constraint as a result of the third camera. This technique is preferred by many groups. N. Ayache states the advantages of using three cameras in [NA91] and answered the question "Why not four cameras?". A review can be found in [PH87].

2.11 Summary

This chapter summarized many methods developed in understanding human stereo vision and in suggesting computational systems. Many constraints including, epipolar, geometric, uniqueness, continuity and order were important factors and played significant roles in developing such algorithms. The most important type, for our work, is the last binocular one mentioned above, the pyramids. The pyramid is a hierarchical structure which has many advantages and forms. This is discussed, in detail, in Chapter 3.

Chapter 3

The Irregular Pyramid

3.1 Introduction to hierarchical processing

3.1.1 What is a hierarchy?

A *hierarchy* is defined as a set of layers, each of them corresponding to a given level of abstraction of the information being processed [JR94]. From the computer vision point of view, usually, the bottom layer of the hierarchy (called the *base*) is fed with the input image. Continuing gathering more global informations as we go to the top of the hierarchy until reaching the highest layer, the top layer (called the *apex*) is related to more global abstraction levels such as the interpretation of a scene. Information flows up and down in the hierarchy and is transformed and combined between layers. Because there are two directions of the flow of information, there are two different kind of processes: $bottom - up$ and $top - down$.

3.1.2 Types of processes

3.1.2.1 Bottom-up process

A $bottom - up$ or $fine - to - coarse$ is a hierarchical propagation of information from bottom to the top of the hierarchy. For instance, this information may consist of only one value, as in multiresolution applications (see Section 3.2).

The goal of the bottom-up process is the *extraction* and *detection* of the most important features or patterns[1] contained in the image. More generally, it allows the transformation of *local* (based on sub-image values) and *distributed* (one cell for each sub-image) sets of data, into *global* (based on the entire image) and *centralized* (one cell, the apex, for the entire image) data. Local feature values are recursively fused, yielding new feature values. Usually the resulting local map of features will be smaller than the previous one, reflecting the degree of abstraction of information that the transformation has affected.

3.1.2.2 Top-down process

A $top - down$ process is a hierarchical propagation of feature (or pattern) values from the top to the bottom of the hierarchy, i.e., transfer the global features extracted using the bottom-up process downwards to the bottom. At the same time, and using the local information at the lower levels, it refines the feature values leading to smoother results. So the goal of such

[1] A feature is some measurable characteristic of the input which has been found to be useful for recognition e.g. intensity or disparity of a given input. In many image processing systems, the pattern takes the form of a set of cells holding the input information, and over these cells, the values of the features are calculated and recorded.

a process can be of two types, *delineation* and *refinement*.

The *delineation* of the previously extracted features is the transformation of a global and centralized set of values into global distributed sets of values. Notice that the features remain global because they are extracted based on large parts of the original image.

The second approach is called the *refinement* or *coarse − to − fine* strategy. According to C.R. Dyer [Dye87], coarse-to-fine strategy can be defined as the operation which uses feature results obtained at coarse levels during the bottom-up process to either localize or focus attention and refine or verify these and other features at finer scales. The first attempt toward hierarchical processing in computer vision is due to Kelly [Kel71]. He applied a planning technique defined by Minsky for the problem solving purposes [Min63] on the edge detection problem. Basically, a smaller image first built from an original image (a bottom-up process). Edges are located in this smaller image and are used as a plan for finding edges in the original image by recursively refining the edge location.

The main disadvantage is that features may be missed at a coarse level even if they are at the first level. As a matter of fact, the technique proposed by Kelly does not detect all the edges in an image of a human face, but only the "main edge", i.e., the outlines of the head.

In the coarse-to-fine strategy, the higher levels which have access to global informations, propose hypotheses which are related to features values. Then the refinement strategy is used to locally verify this guess by recursively taking into account finer, i.e., more local, information.

3.2 Hierarchical multiresolution

One way to increase the speed of computer vision algorithm is to reduce the volume of the input data. This approach is related to the concept of *multiresolution*. However, this is not the only reason this problem has been extensively studied in the last two decades. Rosenfeld and Thurston first noticed in their early work on edge detection that multiresolution must be used in order to achieve better results [RT71].

Indeed, consider the segmentation of an image as a transformation from the original image data to a symbolic description. (In that sense, edge detection can be regarded as a segmentation process.) This description is of a compound nature; it refers to properties of sets of pixels and also to their positions. For instance, visible edges in a scene are often associated with abrupt changes in average gray level. Such changes can vary greatly in degree of "localness". At one extreme, two adjacent pixels having different gray levels define a $micro - edge$, while at the other extreme, two large coarsely textured regions having different average gray levels can also define a sharp edge if the transition between them is abrupt.

Unfortunately, these aspects are "incompatible". This is the fundamental source of uncertainty in segmentation process [WS88]. In general, precision in region property values is increased (e.g. by smoothing, different gray levels are easily to detect) at the cost of decreased precision in position of region boundaries because smoothing usually causes blurring to edges. Refer to Figure 3.1 on Page 17. Among the traditional segmentation techniques, pixel classification is concerned with properties (e.g. the intensity, the mean or the variance) and ignores location, while edge detection local-

Figure 3.1: Blurring of edges as a result of smoothing.

izes region boundaries but ignores (constant) values within regions, i.e., the main concern in edge detection problem is to determine if a certain edge is existing at a certain location regardless the properties of the regions it separates.

To reconcile the conflicting needs for accuracy in both properties and positions, a proposed solution is the multiscale representation and analysis of an image. This approach is confirmed by studies of the human vision. Indeed, one of the basic laws of perception does not depend on the scale of the retinal image (over a limited but large range of scales). It is thus conjectured that the visual system makes use of a multiresolution representation of the input signal [JR94]. i.e., an image can be represented using different resolutions without loss in properties of the objects contained in the image e.g. their colours and without changing the relative positions of the objects unlike the case of smoothing mentioned above. The idea behind that comes from the human vision system. The human perception is working using the recognition cone through which we perceive images (or data) from the surrounding media. These images are transferred from the base of the cone (objects real plane) to the top (the retina) in multiresolution sequence, i.e., the resolution decreases in the direction of the retina. From this important concept comes the idea of simulation of the recognition cone and the pyramid appears to play the role of the artificial recognition cone of the machine.

3.3 The pyramid architecture

The concept of a *pyramid* was introduced into the field of image analysis by Tanimoto and Pavlidis [TP75], who developed an idea suggested by Kelly [Kel71]. It is worth nothing here that this first attempt was intended as a solution to edge detection (as Kelly did in his proposed hierarchical planning techniques).

The pyramid architecture tries to be a useful compromise between the need of a parallel architecture which allows local-distributed to global-centralized communication, and the need for a hierarchical representation of an image at several resolutions, not necessarily to be processed by more than one processor . Since 1970's, much work has been done in this area toward the development of the theoretical foundations of this model.

The pyramidal model we are focusing on now is characterized by:

1. a tessellation of the image plane;
2. a set of (elementary) processors;
3. a hierarchical communication network (the set of the links among the processors).

3.3.1 The elements of a pyramid

We shall first introduce some definitions related to the image analysis field according to [JR94].

A *pixel* is a pair (x, y) where $x \in X$ and $y \in Y$; X and Y, are subsets of $\mathcal{N}$, are the sets of coordinates defining a tessellation . The set of pixels defined by $I = X \times Y$ is called an *image* . Most of the time (but not in our implementation), $X = Y = [0, 2^n - 1]$ where n is a small integer number and 2^n characterizes the side length of the image and is also called the diameter.

A pyramid P is a list of images, $I_h, h \in [0, n]$.h is a *level* and n , which is log_2(diameter), is the *height* of the pyramid. An element of this pyramid is called a *node* or *cell* and is characterized by (x, y, h) where (x, y) is a pixel of the image I_h. If, for every level h, the set of coordinates is $X_h = Y_h = [0, 2^{n-h} - 1]$ then the number of nodes in the pyramid
$= 1 + 2 \times 2 + 4 \times 4 + \ldots + 2^{n-1} \times 2^{n-1} + 2^n \times 2^n$
$= 1 + 4 + 4^2 + 4^3 + \ldots + 4^{n-1} + 4^n = \frac{4^{n+1}-1}{3}$ elements.

The smallest image I_n is called the *apex* of the pyramid and is composed of only one node, the largest image, I_0 is the *base* or the bottom of the pyramid. Usually, this image is the input image to be processed. An example of such a pyramid is shown in Figure 3.2 on Page 20.

In what follows the cells will be regarded as independent processors, each having its own local memory, processing unit (the methods used to implement each cell) and a set of links to other cells. These links consti-

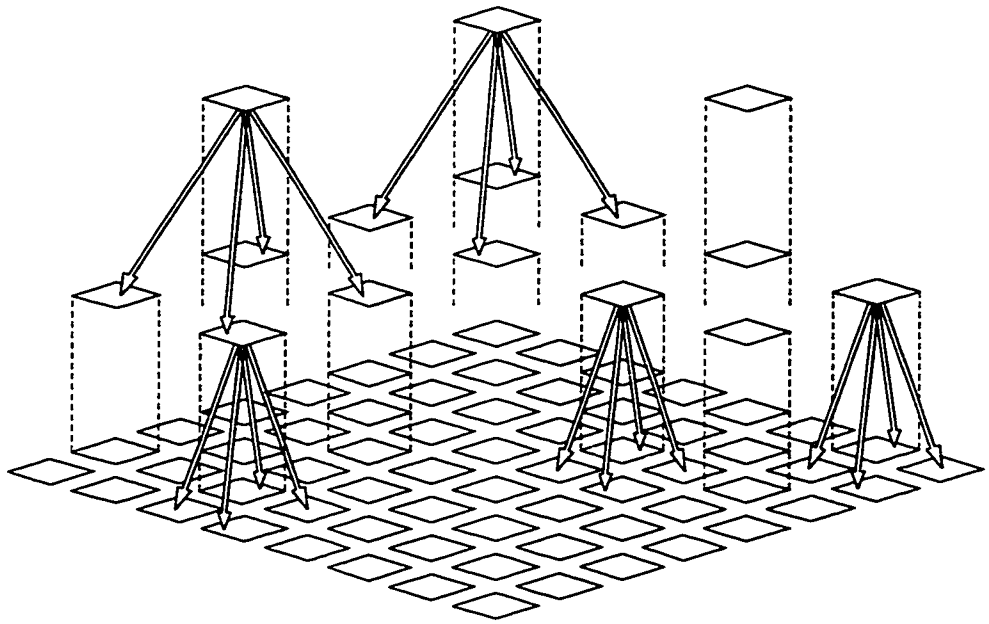


Figure 3.2: An example of quad-pyramid

tute the communication network which is the most important part of the architecture.

3.3.2 Thc communication network

The internal communication network is made up of two different kinds of links: $intra-level$ or horizontal links (a mesh) and $inter-level$ or vertical links (a quadtree, in the case of the quad-pyramid shown in Figure 3.2 on Page 20)

3.3.2.1 The intra-level network

Every node of the pyramid (x, y, h) can be linked to its neighbours in the image I_h. This set of neighbours depends on which connectivity is chosen (4 or 8). Consider Figure 3.3 on Page 22. The nodes belonging to the border of the image are special cases with incomplete neighbourhood. These neighbourhoods can be made complete if we assume that the image is a torus. This, of course, must be compatible with the nature of the image. It is the case when the image contains objects on a uniform background. In the following, we shall denote the set of neighbours of a given node ρ by $Brothers(\rho)$.

Let R be the relation defined on $I_h \times I_h$ by
$R = \{(\rho, \beta) | \ \rho$ is a neighbour of $\beta\}$
R is non-reflexive (i.e., ρ canNOT be a neighbour of ρ), symmetric (i.e., if ρ is a neighbour of β then β is a neighbour of ρ), and non-transitive (i.e., if ρ is a neighbour of β and β is a neighbour of δ then β may NOT be a neighbour of δ). Note that every level I_h can be associated with a graph (I_h, E) where E is the set of links among neighbour cells or nodes.

	x $y-1$	
$x-1$ y	x y	$x+1$ y
	x $y+1$	

$x-1$ $y-1$	x $y-1$	$x+1$ $y-1$
$x-1$ y	x y	$x+1$ y
$x-1$ $y+1$	x $y+1$	$x+1$ $y+1$

Figure 3.3: The 4-connectivity and 8-connectivity based neighbourhoods

3.3.2.2 The inter-level network

Every node (x, y, h) is linked to nodes on levels $h-1$, its *children*, and node(s) on level $h+1$, its *parent(s)* . Each node except those on the base, has at least one child and each node except the apex, has at least one parent. The number and nature of these links characterize the hierarchical structure. In the following, we shall refer to $Parents(\rho)$ and $Children(\rho)$ as the set of parents and children of a given node ρ, respectively.

Let L_1 be the relation defined on pyramid P by
$L_1 = \{(\rho, \beta) |\ \rho \text{ is a parent of } \beta\}$
L_1 is non-reflexive (i.e., ρ canNOT be a parent of ρ), anti-symmetric (i.e., if ρ is a parent of β then β is NOT a parent of ρ), and non-transitive (i.e., if ρ is a parent of β and β is a parent of δ then β is NOT a parent of δ). The pyramid P could thus be associated to a directed graph (P, E_1) where E_1 is the set of links among the cells and their parents.

Similarly, let L_2 be the relation defined on pyramid P by
$L_2 = \{(\rho, \beta) |\ \rho \text{ is a child of } \beta\}$

L_2 is non-reflexive, anti-symmetric and non-transitive. The pyramid P could thus be associated to a directed graph (P, E_2) where E_2 is the set of links among the cells and their children.

We can define the *ancestors* and *descendants* of a given node [JR94] as follows:

If ρ and β are cells contained at levels h and k respectively then ρ is an ancestor of β, if and only if

1. $h > k$;
2. There exists a set of nodes, $\rho_m, \rho_m \in I_m, m = k+1, \ldots, h-1$ such that
   - $L_1(\rho, \rho_{k+1})$
   - $L_1(\rho_m, \rho_{m+1})$ for $m = k+1, \ldots, h-1$
   - $L_1(\rho_{h-1}, \beta)$

In the same way,

If ρ and β are cells contained at levels h and k respectively then ρ is a descendant of β, if and only if

1. $h < k$;
2. There exists a set of nodes, $\rho_m, \rho_m \in I_m, m = h+1, \ldots, k-1$ such that
   - $L_2(\rho, \rho_{k-1})$
   - $L_2(\rho_{m+1}, \rho_m)$ for $m = k-2, \ldots, h+1$
   - $L_2(\rho_{h+1}, \beta)$

Finally, the *receptive field* of a given node is defined as the set of all descendants located on the base of the pyramid.

3.4 Regular vs irregular pyramids

Essentially, we can identify two board classes: *regularly* sampled pyramids, and *irregularly* sampled pyramids. The former class includes the "bin pyramid" and "quad pyramid"; the later includes "stochastic," and "adaptive" pyramids.

3.4.1 Regular pyramids

3.4.1.1 Bin- vs Quad-pyramid

The architecture is called $bin - pyramid$ if every node has two children. In order to achieve complete coverage of an image, we must alternate row-children and column-children (see Figure 3.4 on Page 25).

The most common architecture is the $non-overlapped\ quad-pyramid$, (see Figure 3.2 on Page 20), where each node (x, y, h) has four children: $(2x,\ 2y,\ h-1)$, $(2x+1,\ 2y,\ h-1)$, $(2x,\ 2y+1,\ h-1)$, $(2x+1,\ 2y+1,\ h-1)$ and one parent $(2x\ div\ 2,\ 2y\ div\ 2,\ h+1)$ where *div* stands for integer division.

The main advantage of the bin-pyramid is the simplicity of the nodes because of the small number of links (see Table 3.1 on Page 28 [JR94]). However, it is costly because it results in $2^{2n+1}-1$ nodes as opposed to $\lfloor \frac{4}{3} \times 2^{2n} \rfloor$ in quad-pyramid, and the number of levels is also bigger, $2n+1$ instead of $n+1$.

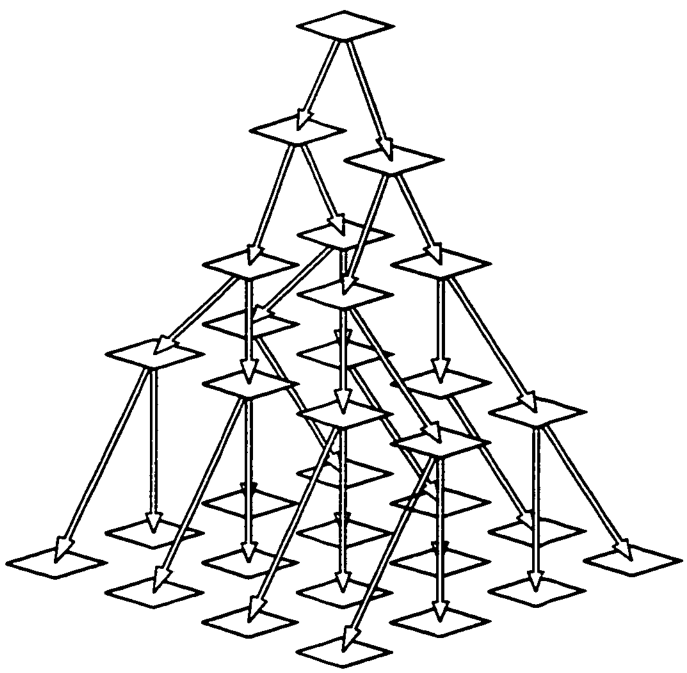


Figure 3.4: The Bin-Pyramid. Alternate row- and column- children.

3.4.1.2 Overlapped quad pyramid

Overlapped [2] pyramids are built by using a reducing factor r smaller than the number of links w which links a parent to its children. An *overlapped quad − pyramid* is obtained if we add the neighbours of the four central children (using 8-connectivity). Thus, a very common situation, for any internal node, is the case in which a parent has 16 children ($w = 16$) arranged as 4×4 block and a child has 4 possible parents, reflecting the fact that 4×4 blocks overlap horizontally and vertically by 50%. For a cell (x, y, h), the parents are (x *div* 2, y *div* 2, $h+1$) the "direct parent[3]" through the main links, and (x *div* 2, y *div* 2 +2(y *mod* 2)-1 , $h+1$), (x *div* 2+2(x *mod* 2)-1 , y *div* 2, $h+1$), (x *div* 2+2(x *mod* 2)-1 , y *div* 2 +2(y *mod* 2)-1 , $h+1$) through the secondary links. The children are $(u, v, h-1)$ where u=$2x-1, \ldots, 2x+2$ and v=$2y-1, \ldots, 2y+2$. However, this is not valid for the border nodes (see Figure 3.5 on Page 27). The resulting reduction factor r is 4.

By looking at Table 3.1 on Page 28, we see that the smallest number of links is thus 3 (in the case of the bin-pyramid) and the largest number is 28 (in the case of 8-connected overlapped quad-pyramid)[JR94].

3.4.1.3 Other forms

Another form of overlapping is generated by the "dual pyramid" proposed by Kropatsch [Kro85]. This pyramid grows by powers of 2 rather than powers of 4. (Recall that, in the classical pyramid architecture, the size of

[2] Overlapping is a modification of an existing architecture e.g. quad pyramid but it is not an architecture by itself.

[3] Direct parent is the original parent of the cell before overlapping.

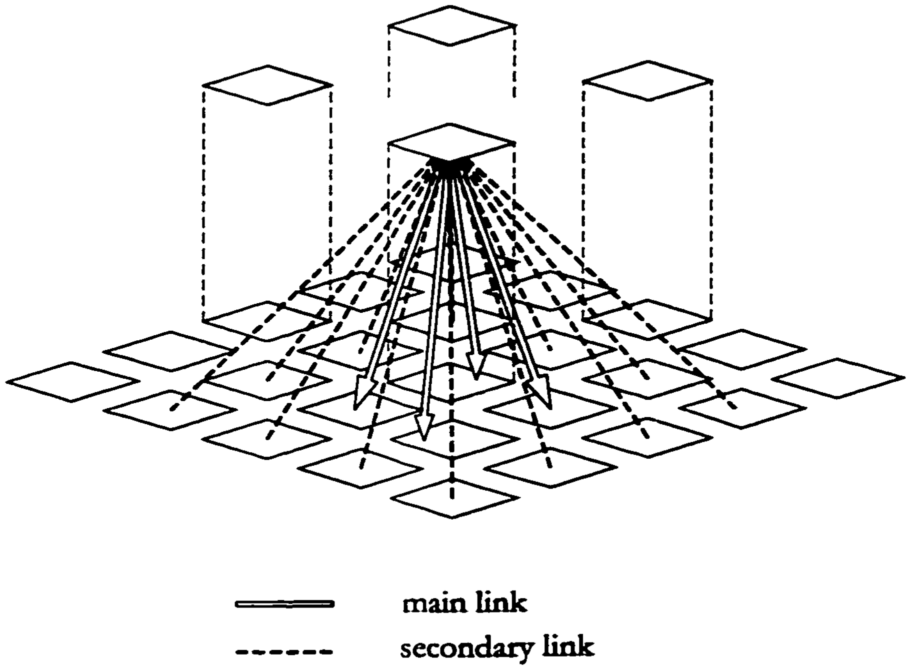


Figure 3.5: An example of overlapped quad-pyramid showing that each parent has up to 16 children and as a consequence each child has up to 4 parents.

Architecture	Parents	Children	Neighbours	Links
a	1	2	0	3
b	1	2	4	7
c	1	2	8	11
d	1	4	0	5
e	1	4	4	9
f	1	4	8	13
g	4	16	0	20
h	4	16	4	24
i	4	16	8	28

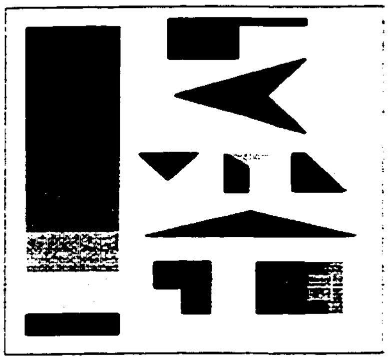
Table 3.1: The number of links for every internal node and the number of parents, children and neighbours of the most classical pyramid: (a) Bin-pyramid (b) 4-connected bin-pyramid (c) 8-connected bin-pyramid (d) Non-overlapped quad-pyramid (e) 4-connected Non-overlapped quad-pyramid (f) 8-connected Non-overlapped quad-pyramid (g) Overlapped quad-pyramid (h) 4-connected overlapped quad-pyramid (i) 8-connected overlapped quad-pyramid

the receptive field of a node in level h is four times the size of the receptive field in $h-1$.) In this case, a recursive method generates a pyramid by rotating the meshes by 45° at the alternate levels. This is obtained by considering half the diagonal of a square as the edge of the square of the next level, this produces a degree of overlapping equal to 50%. The top-down generation of the dual pyramid proceeds by replacing a 2×2 square over the centre of the parent; the length of the edge of each subsquare is half the diagonal of the parent. The resulting pyramid has a variable resolution, a variable reduction factor and a fixed number of children = 4. Thus, each node has a maximum of two parents and exactly 4 children [CF94].

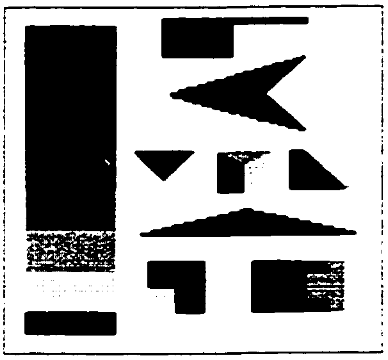
It is possible to build pyramids with different architectures by changing the characteristics of the original model. As an example, Hartman and Tanimoto [HT84] proposed a hexagonally shaped pyramid with a triangular tessellation. The apex is made of six triangles arranged in a hexagon. Each triangle within a level has four children and a parent.

3.4.1.4 Advantages and disadvantages

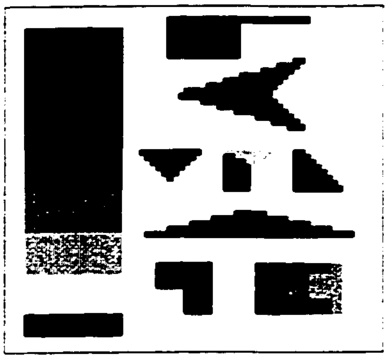
As shown in section 3.2, the main property of an image pyramid (a hierarchical structure) is fast data gathering across the input image. Moreover, the amount of noise, i.e., everything which is not useful for the task under consideration, can be significantly decreased as a result of the integration of larger and larger areas of the input. Specially, for the purposes of object extraction, the rigidity of the architecture results in theoretical inconveniences such as feature distortion and shift-variance. In the regular pyramid architecture, a cell at level h, represents a $2^h \times 2^h$ square in the input



(a)



(b)



(c)

Figure 3.6: Distortion problem: (a) Original image; (b) At level 1, magnified 2 times; (c) At level 2, magnified 4 times.

image. Thus the root, calculated at upper level cannot exactly represents an object but only an approximation between the largest square included in the object and the smallest square containing the object. A root must represent an irregular region. If the shape of the object is followed throughout the bottom-up process, the distortion problem will appear step by step. An example of such a problem is shown in Figure 3.6 on Page 30.

One proposed solution is to make the structure adaptive to the process being performed on the data. This is equivalent to looking for the optimal hierarchical architecture in a particular context through the design of a hierarchical communication network. This brings us to the second board class of pyramids, "the irregular pyramids".

3.4.2 Irregular pyramids

All the pyramid architectures considered before make use of a decimation process. Very few attempts to design an adaptive hierarchical architecture have been made in the recent past. Among them, Peleg *et al.* [PFH86] proposed that the generation of the sampling grids can be adapted to the weights of the cells. This was the bridge between the regular and irregular pyramids. They used the term "Custom-Made" pyramid to denote a type of geometric construction that resamples or reduces a regular $m \times n$ grid into a regular $k \times l$ smaller one.

In this pyramid, the resampling (or reduction) can be uniform or weighted, but in any case, a cell in any level contributes a finite quantity to its parent(s) on the next level. The hierarchy of this pyramid is more variable than those mentioned above, i.e., the regular pyramids. In this architecture, the choice of surviving cells on the next level can be made dependent on the image information using an interest operator which influences the weights. They used a measure of the "busyness" to determine the degree of reduction: a smoothed value gives a clue of the regularity of the region. The coarser the region, the more the support used in building the next level. In other words, this means that busier regions would have less resolution reduction than smoother regions of the input, this makes areas richer in contrast, are sampled more finely. In a hierarchical reasoning environment, this would mean that the number of parents would depend on the volume of data sent by a set of children on the level below.

3.4.2.1 The stochastic pyramid

The stochastic pyramid is another approach in which the levels are obtained through random processes instead of deterministically. Let ρ be a cell on a level I_h and x_ρ an outcome of a random variable allocated to that cell. The site of ρ remains on the next level I_{h+1} if and only if x_ρ is greater than all the outcomes of x_β where β is a brother of ρ. This process is iterated till every node of I_h either is a surviving node or has at least one brother which is surviving node. The next level I_{h+1} is made up of the surviving nodes. The neighbourhood of the surviving node is the union of the neighbourhoods of all its brothers which are linked to it on the level below. So the number of parents, children and brothers are no longer constants. The mean decimation ratio (size of I_h, size of I_{h+1}) has been proved to be 5.44 [JR94] and so is a little greater than in a classical regular pyramid (where the ratio is 4). As a consequence, the height of a stochastic pyramid is slightly smaller than that of a regular pyramid. Note that the selection of the nodes in the upper levels is only half of the process in generating the hierarchy. The second half consists of establishing hierarchical links between parents and children.

3.4.2.2 The adaptive pyramid

The random election process of the stochastic pyramid has no relation at all to the contents of the image in the base of the pyramid. To obtain more realistic hierarchy, J.M. Jolion and A. Montanvert [JM91] introduced the "adaptive pyramid". The generation rules for an adaptive pyramid are similar to the generation of a stochastic pyramid, and the same rules are used to build the disparity pyramid (chapter 5). The decimation procedure is based on two principles [JM91]:

- Two neighbours at a given level cannot both survive at the next level;
- For each nonsurviving cell , there exists at least one surviving cell in its neighbourhood.

In such a pyramid, the decimation process is carried out by letting a cell survive if it is a local extreme of an *interest operator* . For example, in a segmentation task, it would be useful to isolate regions of pixels with a similar gray levels. The chosen operator in this case (the interest operator) is the variance of gray level in the region, and the surviving cells are those where the variance is the minimum. The decimation process is locally defined on the basis 3 variables:

- Two binary variables p and q;
- An interest operator, the variance of the gray level v.

A complete example of such a process is shown in Figure 3.7 on Page 34.

The cell will survive at the next level if the binary variable p is set during the decision process. Practically, a lot of cells will be left nonsurviving (i.e., p=0) although they have no surviving cell in their supports (the second principle). This case happens when a cell has an interest operator greater than one of its neighbours whose interest operator is, again, greater than another neighbour's (Figure 3.7b on Page 34). To overcome this problem,

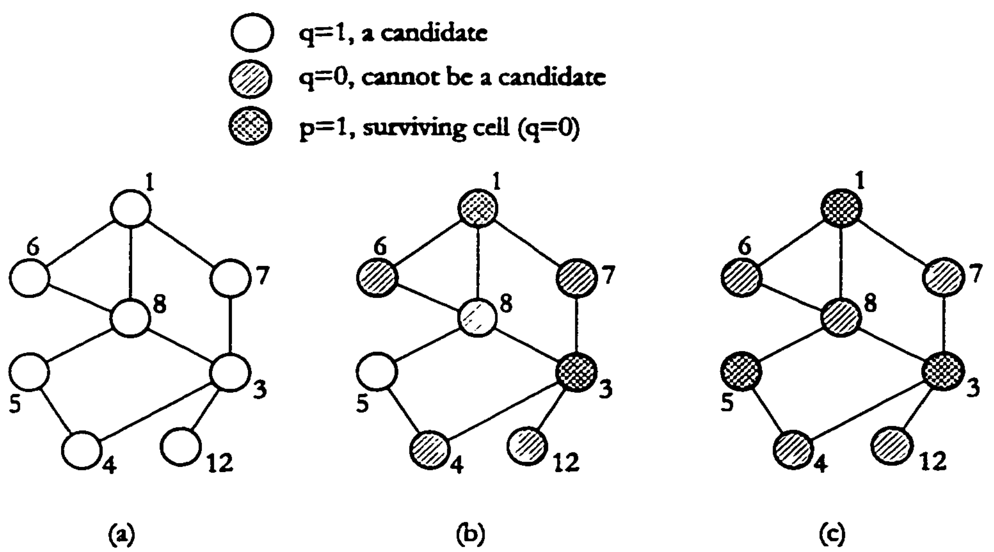


Figure 3.7: The decision process: (a) initial graph with v_{ijh} values; (b) global minima are extracted; (c) local minima are added.

the decision process is repeated until there is no new cells can survive (see Figure 3.7c on Page 34).

The links are established with the parent according to a criterion on the gray level contrast: the least contrasted cell in the support is the best parent and therefore is chosen. The decision process is to be repeated over the cells on the new level (the parents) to extract the surviving cells on the next level, then the links are established among the parents and the children. The whole process is repeated until there is exactly one cell left for each object on the base, this cell is called the root of this object.

The adaptive pyramid is slightly larger than its stochastic counterpart (i.e., has less decimation ratio) and was used successfully on the same type of problems with comparable results. However, all the samples presented by Jolion and Montanvert [JM91] followed the restriction of the square base image, i.e., the input image is a square of a side length 2^n.

3.5 Summary

The foregoing discussion dealt with the fundamental concepts of different pyramidal architectures. A formal definition of the hierarchy was given along with different types of processes associated with such structures. The multiresolution has been defined as a technique to reduce the input data and to enhance the processing speed. The characteristics of the pyramid have been explained including the elements of the pyramid and the types of communication networks. Finally, a classification of the pyramids has been discussed and different types of pyramids have been explained in detail.

Chapter 4

Disparity Estimation

4.1 Introduction

In this chapter, we will try to explore an important concept in the field of computer vision, named the disparity estimation. To reach this goal, we have to introduce some basic points in calibrating the cameras used as our viewpoints to the scenes.

4.2 Image modeling

The concept behind the image formation is depicted in Figure 4.1 on Page 37. It consists of two planes $\mathcal{P}_1$ and $\mathcal{P}_2$ and an object B in the space. If a small hole is punched in plane $\mathcal{P}_1$ and through it some rays of light are emitted from the object B. This will form an inverted image of B on the plane $\mathcal{P}_2$ [Fau96].

From the previous idea, we can construct the camera model. We consider this hole (called *pinhole*) as the lens of the camera which has a centre point

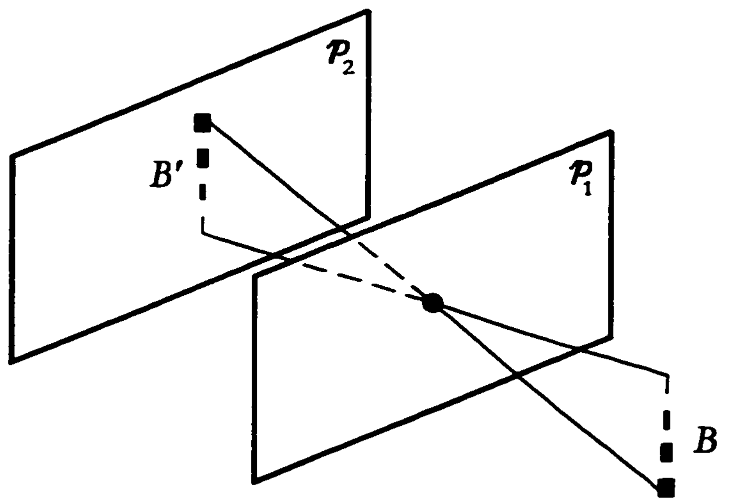


Figure 4.1: Image Formation: B is the object in the space, B' is the image of B passed through the pinhole

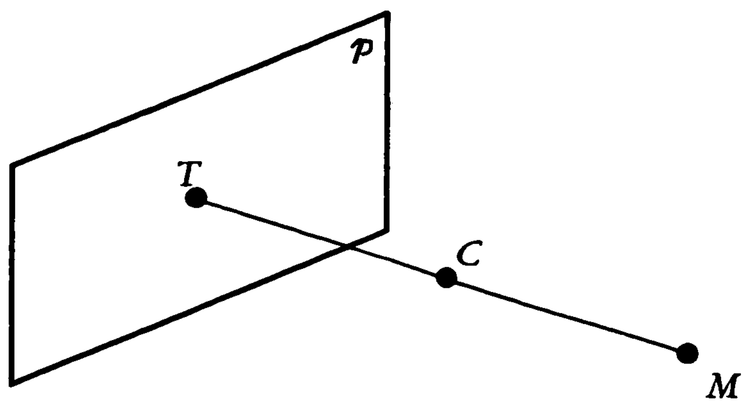


Figure 4.2: Image Model: M is point in the space, T is the image of M on the retinal plane $\mathcal{P}$ passed through the optical centre C and MCT is the optical axis.

C (called *optical centre*) and the plane $\mathcal{P}_2$ as the image plane (called *retinal* or *image plane*). A point M in the observed space is projected into the camera retina at the image point T. Point T is the intersection of the line MC with the retinal plane and MCT is called the *optical axis* . This is shown in Figure 4.2 on Page 38.

4.3 Disparity in image modeling

4.3.1 Multiple perspectives for one scene

With the same previous idea, if we use two cameras to view a point in the space, we will get two images for this point, one on each retinal plane. This is shown in Figure 4.3a on Page 40. Given a point M in the space and two cameras with optical centres C_1 and C_2, T_1 is the image of M formed through C_1 on the retinal plane $\mathcal{P}_1$ and T_2 is the image of the same point on $\mathcal{P}_2$. So, given an image point T_1, its matching point T_2 belongs to the line segment formed by intersecting the plane MC_1C_2 with the retinal plane $\mathcal{P}_2$. This line is called the *epipolar line* associated with T_1, while the epipolar line associated with T_2 is the line formed by intersecting the plane MC_1C_2 with $\mathcal{P}_1$. These two lines are called the *conjugate epipolar lines.*

As shown in Figure 4.3a the intersection of the plane MC_1C_2 with $\mathcal{P}_1$ and $\mathcal{P}_2$ is forming the conjugate epipolar lines. Any point on the first epipolar line has its matches on the second and vice versa. Finally, the points N_1 and N_2 are the intersection of the line C_1C_2 with $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively and they are on the epipolar lines of T_1 and T_2.

4.3.2 Disparity

For an image point T_1, to reach the matching point T_2 on $\mathcal{P}_2$, we can compute the epipolar line containing T_2 and the exact position of T_2 can be measured relative to the position of T_1 using a single parameter e.g. the difference between N_2T_2 and N_1T_1 (see Figure 4.3a on Page 40). This brings us to the definition of disparity.

Disparity : The disparity between a pair of points $(T_1, T_2)$is defined by

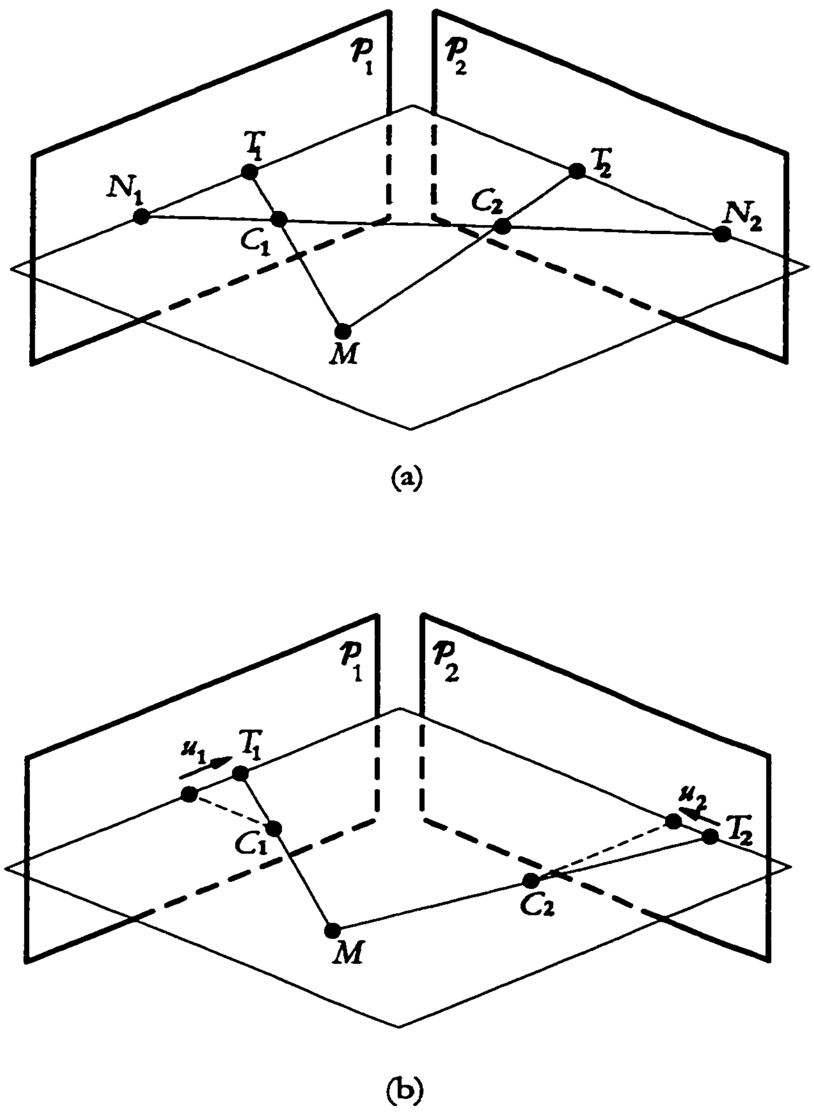


Figure 4.3: (a)Using two cameras to view a point in the space: M is a point in the space, C_1 and C_2 are the optical point, T_1 and T_2 are the images of M on the retinal planes $\mathcal{P}_1$ and $\mathcal{P}_2$. (b) Disparity and distance: Possible choice for u_1 and u_2.

difference [NA91]:

$$\delta = u_2 - u_1 \tag{4.1}$$

Finding the matching point T_2 now depends on the determination of the disparity vector δ with T_1, where u_1 and u_2 are 2D vectors representing the difference between the orthogonal projection of the points C_1 on $\mathcal{P}_1$ and T_1 for the first camera and C_2 on $\mathcal{P}_2$ and T_2 for the second (see Figure 4.3b on Page 40). The possible values of δ are limited because:

1. The observed points are necessarily in the space situated in the front of the image planes and the optical centre of each camera;
2. The dimensions of the image are limited;
3. Most of the time, the dimensions of the observed scene are limited.

4.4 Disparity estimation

Disparity Estimation concerns finding pairs of point features in two perspective views of a scene such that each pair corresponds to the same scene point. Usually, the relationship between the viewpoints used to obtain the two images is *unknown*. Further, the objects in the scene may be stationary or moving. However, the last case, i.e., moving objects, is not a stereo case which we handle here.

In general, the two subsets of points are locally intensity matched and preserve the interpoint geometrical structure. As is true for all matching algorithms, the underlying principle for matching is geometrical similarity and/or intensity (or colour) similarity [HA94], which we used.

4.5 Intensity-based matching

The goal here is to find pairs of points from each image such that they correspond to the same scene points.

Applying to all the points in the image, a *difference measure* is computed over a window W. This window is determined by a disparity range $(dx_{min}, dx_{max}, dy_{min}, dy_{max})$ which is a parameter to the algorithm. Let $f_1(x, y)$ be the image intensity at the point (x, y) in the first image and $f_2(x, y)$ be the image intensity at the point (x, y) in the second image. Then, over all the points in the window W determined by the range $(dx_{min}, dx_{max}, dy_{min}, dy_{max})$, if we apply the difference formula

$$F_{xy} = f_1(x, y) - f_2(x + dx, y + dy) \tag{4.2}$$

where $dx \in [dx_{min},\ dx_{max}]$ and $dy \in [dy_{min},\ dy_{max}]$ respectively, we can determine which point in the second image is the closest one to (x, y) in terms of intensity value which may be slightly affected by camera position. However, we may end up with more than one candidate, that is expected because the intensity value may be repeated many times within the same window. So we should not apply this difference formula on individual points, however, we should apply it over another window. We applied it over a 3×3 window centered on the point, then we should calculate this one-to-one cell difference over the sum of the intensities of this window. This leads us to

$$F_{xy}(dx, dy) = \sum_{i,j=-1}^{i,j<2} \{f_1(x + i, y + j) - f_2(x + i + dx, y + j + dy)\}^2 \tag{4.3}$$

where F_{xy} is the minimum disparity value associated with (x, y) using the disparity vector (dx, dy). Note that, F_{xy} is the squared value of the difference, this would magnify it and avoid getting negative results. Here, we should keep track of the values of dx and dy which result in this minimum difference. The last formula is the base in determining the interest operator in our work as we will see later.

4.5.1 Example

A complete example of the previous procedure is illustrated in Figure 4.4 on Page 45. Figures 4.4 a, b are the original intensity values of the pair of images under consideration. For example, if we want to calculate F_{11} value for the cell included in the dotted square in the first image (Figure 4.4a), then we have to consider the 3×3 neighbourhood centered on the cell included in the dashed square (Figure 4.4a). Assume that the disparity range is (0, 1, 0, 1) (this window is shown in Figure 4.4b), and that each cell outside the group shown, has an intensity value of 1.

Now, to calculate F_{11} value, we have to consider the whole possible disparity range:

- If $(dx, dy)^T = (0, 0)^T$

  Intensity difference =

9	8	9
8	7	9
9	9	2

2	3	1
2	8	7
1	9	8

$\Longrightarrow$

7	5	8
6	-1	2
8	0	-6

$$F_{11}(0,0) = 7^2 + 5^2 + 8^2 + 6^2 + (-1)^2 + 2^2 + 8^2 + 0^2 + (-6)^2 = 279;$$

- If $(dx, dy)^T = (0,1)^T$

  Intensity difference =

$$\begin{array}{|c|c|c|}\hline 9&8&9\\\hline 8&7&9\\\hline 9&9&2\\\hline\end{array}\quad\begin{array}{|c|c|c|}\hline 2&8&7\\\hline 1&9&8\\\hline 1&9&8\\\hline\end{array}\quad\Longrightarrow\quad\begin{array}{|c|c|c|}\hline 7&0&2\\\hline 7&-2&1\\\hline 8&0&-6\\\hline\end{array}$$

  $F_{11}(0,1) = 7^2 + 0^2 + 2^2 + 7^2 + (-2)^2 + 1^2 + 8^2 + 0^2 + (-6)^2 = 207;$

- If $(dx, dy)^T = (1,0)^T$

  Intensity difference =

$$\begin{array}{|c|c|c|}\hline 9&8&9\\\hline 8&7&9\\\hline 9&9&2\\\hline\end{array}\quad\begin{array}{|c|c|c|}\hline 3&1&3\\\hline 8&7&8\\\hline 9&8&9\\\hline\end{array}\quad\Longrightarrow\quad\begin{array}{|c|c|c|}\hline 6&7&6\\\hline 0&0&1\\\hline 0&1&-7\\\hline\end{array}$$

  $F_{11}(1,0) = 6^2 + 7^2 + 6^2 + 0^2 + 0^2 + 1^2 + 0^2 + 1^2 + (-7)^2 = 172;$

- If $(dx, dy)^T = (1,1)^T$

  Intensity difference =

$$\begin{array}{|c|c|c|}\hline 9&8&9\\\hline 8&7&9\\\hline 9&9&2\\\hline\end{array}\quad\begin{array}{|c|c|c|}\hline 8&7&8\\\hline 9&8&9\\\hline 9&8&2\\\hline\end{array}\quad\Longrightarrow\quad\begin{array}{|c|c|c|}\hline 1&1&1\\\hline -1&-1&0\\\hline 0&1&0\\\hline\end{array}$$

  $F_{11}(1,1) = 1^2 + 1^2 + 1^2 + (-1)^2 + (-1)^2 + 0^2 + 0^2 + 1^2 + 0^2 = 6;$

$F_{11} = \min\{F_{11}(0,0), F_{11}(0,1), F_{11}(1,0), F_{11}(1,1)\}$=6;

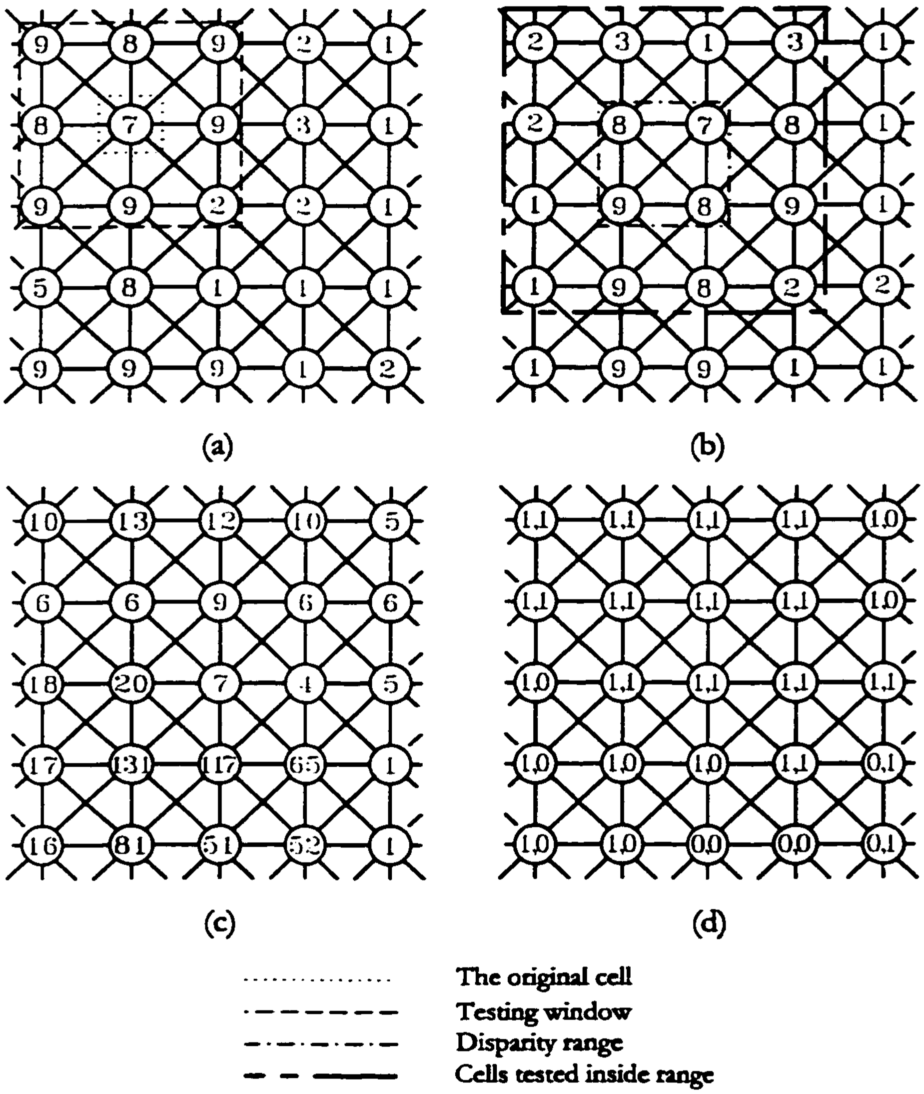


Figure 4.4: F Calculation: (a) The first image with intensity values. (b) The second image with intensity values. (c) F value for each cell. (d) Associated disparity vector.

4.6 Summary

The foregoing chapter dealt entirely with a very important and interesting area of research in computer vision and stereo image analysis, the disparity estimation. This chapter began with a brief basic discussion on monocular image formation. Then it moves to a discussion on binocular image formation and the disparity estimation problem formulated as a result of it. The last point in this chapter was a discussion on the intensity-based matching and difference measures. This chapter ends up with a suggested difference equation to be applied over a 3×3 window to get a value of an interest operator which will be used in Chapter 5.

Chapter 5

The Disparity Incomplete Pyramid

5.1 Introduction

This chapter will deal almost entirely with a proposed technique among a class of techniques that seem to play a key role in the early stages of visual process-namely, techniques for segmenting the image into distinctive parts. Indeed, when we look at a scene we do not perceive an array of brightnesses; usually, we see a collection of regions separated by more or less well-defined edges. In computer vision, processes that decompose a scene into parts are called *segmentation techniques* .

It should be emphasized that there is no single standard approach to segment a scene [JR94]. Many different types of image or scene parts can serve as the segmentation depend on which descriptions are based, there are many ways in which one can attempt to extract these parts from the image. However, we are interested in tools that rapidly provide useful information and we claim that pyramid based techniques are such tools. One proposed technique using the concepts of disparity and irregular pyramids is presented in this chapter, (a detailed algorithm is presented in appendix A). The issue of how to integrate segmentation techniques into a comprehensive vision system is still an open problem.

5.2 Pyramidal architecture rules

A pyramidal architecture is completely defined if we specify how a new level is built and how a parent is linked to its children. The generation rules of this pyramid are similar to those of a stochastic and adaptive pyramid. The decimation process is based on two principles [JM91]:

- Two neighbours at a given level cannot both survive at the next level;
- For each nonsurviving cell , there exists at least one surviving cell in its neighbourhood.

5.3 Building a new level

5.3.1 Initialization

Let I_h be the image at level h in the pyramid . The neighbours of a cell (i,j,h) is the list of cells, $Brothers_{ijh}$, sharing links with it (see Figure 5.1 on Page 50). The support, $Support_{ijh}$, of a cell (i,j,h) is defined as the set of all the neighbours or brothers of (i,j,h) in addition to the cell itself, i.e., $Brothers_{ijh} \cup (i,j,h)$ [JM91]. (See Figure 5.1b on Page 50.) At the base of the pyramid (I_0 is the input image), $Support_{ij0}$ is the 3×3 square array centered on the cell. Recall that, the receptive field of a cell

(i, j, h), $Receptive_{ijh}$, is the list of all descendants located on the base of the pyramid. Each cell is associated with four important variables:

- Two state variables p_{ijh} and q_{ijh} ;
- The output F_{ijh} of the *interest operator* which is calculated as discussed earlier (see equation 4.3 in Section 4.5 on Page 42), where the lower the value F_{ijh}, the higher the interest of this cell, i.e., this cell is a good candidate to survive. Note that in a stochastic pyramid, F_{ijh} is the outcome of a random variable uniformly distributed between 0 and 1 while in the adaptive pyramid, F_{ijh} is the variance of the gray levels in the receptive field;
- The last variable is a 2D vector $(disp_x, disp_y)^T_{ijh}$ which determines the disparity values associated with F_{ijh} in both directions.

We recall that the interest operator F_{ijh} of a cell (i, j, h) is the square of the difference between the intensities or the gray levels of the support of the current cell and the intensities or the gray levels of the support of another cell $(i + disp_x,\ j + disp_y,\ h)$ in the second image, (see equation 4.3 on Page 42). For the first level ($h = 0$), I considered the receptive field $Receptive_{ij0}$ of the cell $(i, j, 0)$ to be the cell itself, i.e., equation 4.3 is applied only once for this cell. This is not the case for the upper levels where this formula is applied as many times as the number of cells of the receptive field.

5.3.2 The decimation process

The decimation process is the procedure by which the surviving cells are selected. We can define this process on the basis of the first three variables

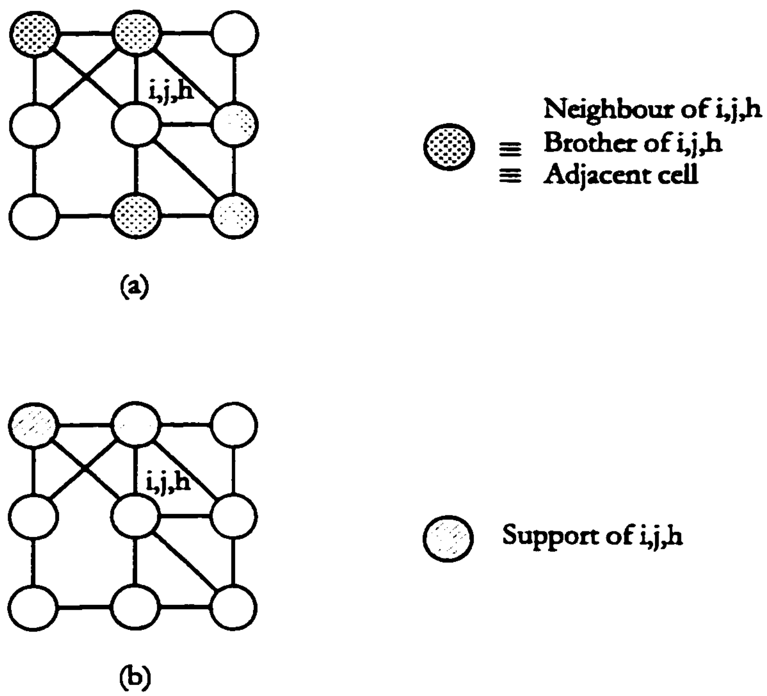


Figure 5.1: (a) $Brothers_{ijh}$ ≡ Brothers of a node (i, j, h). (b) $Support_{ijh}$ ≡ $Brothers_{ijh} \cup (i, j, h)$.

mentioned above, i.e., p, q, and F. The previous example in Section 4.5.1 is re-used to illustrate this process. Refer to Figure 5.2 on Page 52.

A cell is retained for the next level $(h+1)$ if and only if the variable p_{ijh} is set to 1 during the decimation process (initially, p_{ijh} =0 and q_{ijh} =1 for all the cells as shown in Figure 5.2a on Page 52). The state p_{ijh} =1 indicates that the cell (i, j, h) is the best candidate to survive over all its support $Support_{ijh}$, i.e., the value of F_{ijh} is the local minimum with respect to its support (see Sections A.4 and A.5 on Page 98). Repeating this process for all the cells extracts the global minima of the level (see Figure 5.2b). This is the same rule applied for the adaptive pyramid:

$$\begin{aligned} p_{ijh} &= 1 && \text{if and only if} \\ F_{ijh} &= \min F_{mnh} && \text{for}(m, n, h) \in Support_{ijh}. \end{aligned} \tag{5.1}$$

$$\begin{aligned} q_{ijh} &= 1 && \text{if } \forall (m, n, h) \in Support_{ijh},\ p_{mnh} = 0 \\ q_{ijh} &= 0 && \text{otherwise} \end{aligned} \tag{5.2}$$

In other words, once the cell (i, j, h) is selected to survive, its p and q variables are changed (p_{ijh} flips from 0 to 1 and q_{ijh} flips from 1 to 0). At the same time, q values of $Brothers_{ijh}$ flip from 1 to 0 as well.

The first principle is verified (two neighbours cannot both survive, see Section 5.2 on Page 48) because the procedure always skips the cells which cannot be good candidates, where q_{ijh}=0 (see Figures 5.2b and 5.2c on Page 52). A practical problem arises here, in most cases the second principle is not verified (there must be a surviving cell in every neighbourhood, see Section 5.2 on Page 48) after the first iteration (see Figure 5.2b on Page 52). In other words, after one iteration the procedure cannot select all

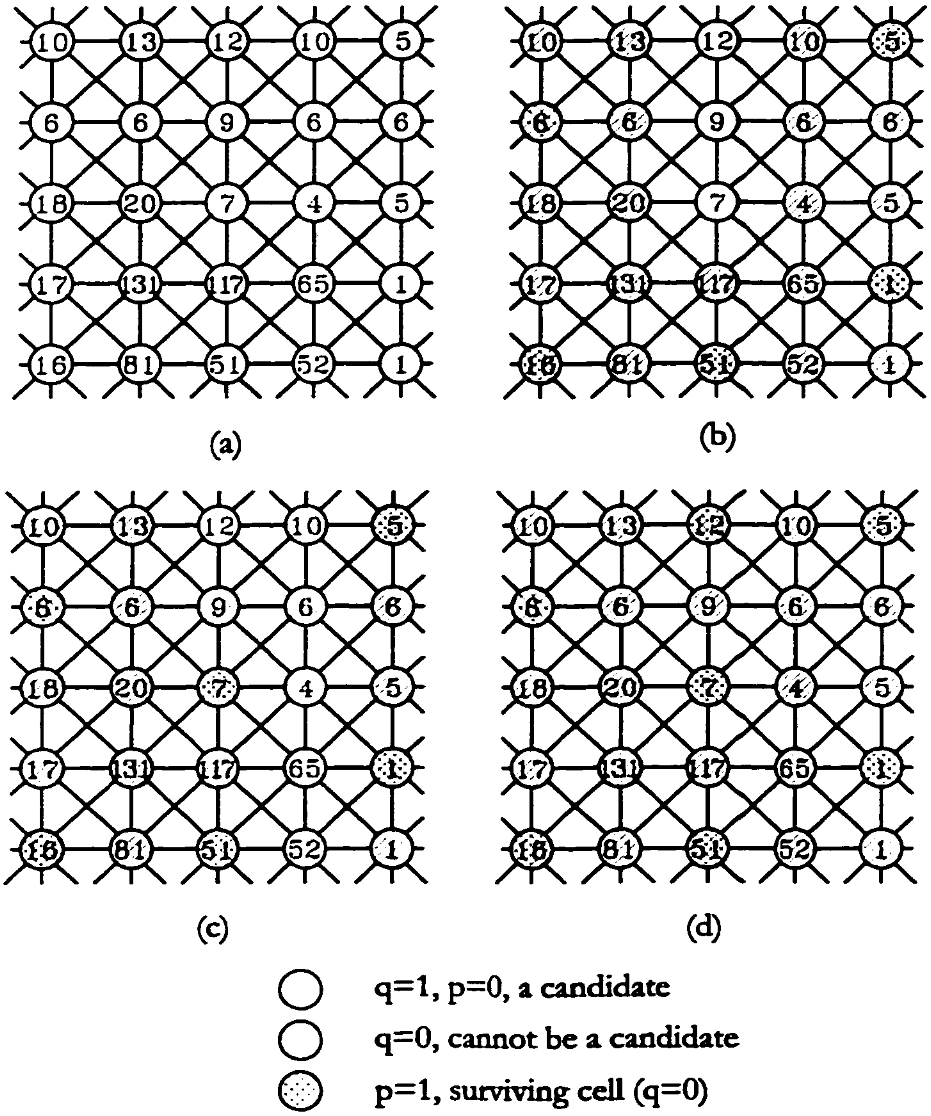


Figure 5.2: Decimation process: (a) initial graph with F_{ijh} values; (b) First iteration: global minima are extracted; (c) Second iteration: local minima are added; (d) Third iteration: local minima are added. *Note*: if q_{ijh}=0, this means that the cell cannot be chosen because either it has been chosen at a previous iteration or it is a brother of a surviving cell.

surviving cells. Note that this is also true if F_{ijh} is an outcome of a random variable (the stochastic pyramid) or data-based interest operator (variance in the adaptive pyramid and disparity in our work) and more particularly for the lower levels of the pyramid.

These cells, which have not been decided yet to survive or not, are indicated by the second state variable q_{ijh} and the local minima of these cells are extracted (see Sections A.6 and A.7 on Page 98):

$$\begin{aligned} q_{ijh} &= 1 \quad \text{if} \forall (m,n,h) \in Support_{ijh}, p_{mnh} = 0 \\ q_{ijh} &= 0 \quad \text{otherwise} \end{aligned} \tag{5.3}$$

$$\begin{aligned} p_{ijh} &= 1 && \text{if and only if} \\ F_{ijh} &= \min F_{mnh} && \text{for } (m,n,h) \in Support_{ijh}, \\ & && \text{such that } q_{mnh} = 1. \end{aligned} \tag{5.4}$$

Steps 5.3 and 5.4 are iteratively repeated until there is no surviving cell (see Figure 5.2d on Page 52 and Section A.7 on Page 99). At this point, the second principle (Section 5.2 on Page 48) is verified.

In our case, with a nonrandom interest operator, the number of iterations depends on the operator and on the data, i.e., the disparity range and the gray levels in the image. So there is no fixed number of iterations. Indeed, if F_{ijh} is constant over a region, there will be no real minimum, e.g. if we have the same background in both images. Thus we must randomly decide which cell becomes a surviving cell. Of course, this particular case increases the number of iterations.

5.4 Linking parents with children

A cell at level $h+1$ in the pyramid (the parent) is linked to a set of cells at level h (the children). In the regular pyramid , the number of children is the same for all cells and all levels, as mentioned in chapter 3. In the disparity incomplete pyramid , as in the stochastic and the adaptive pyramid, the number of children depends on the information provided (random variable in the case of the stochastic pyramid or data-based in the case of the adaptive pyramid).

5.4.1 Which cell to choose?

According to the second principle (see Section 5.2 on Page 48), if (i, j, h) is a nonsurviving cell, there always exists at least one surviving cell in the support $Support_{ijh}$. Here we need a contrast measure to determine which two cells should be linked together. Note that, in the adaptive pyramid, the cell (i, j, h) is linked to the least contrasted surviving cell in its support. In the disparity pyramid, we used two different contrast criteria which are:

- In the case of disparity estimation: The disparity variation is used, i.e., the distance between disparity vectors of the two cells under consideration is calculated to help decide whether to choose the cell or not. So, the cell is linked to the nearest surviving cell (in terms of disparity). Here, the Euclidean's distance[1] between their respective disparity vectors is computed.

[1] Euclidean's Distance is defined as:
$ED \equiv \sqrt{(dx_1 - dx_2)^2 + (dy_1 - dy_2)^2}$ where $(dx_1, dy_1)^T$ and $(dx_2, dy_2)^T$ are the two disparity vectors.

o In the case of object extraction: The average gray level absolute difference along the common border of the regions could be used as the contrast measure.

A complete example of this procedure is shown in Figure 5.3 on Page 56. Figure 5.3a shows the disparity vectors associated with each cell, Figure 5.3b shows minimum values of F calculated using equation 4.3 as shown in example 4.5.1 and the choice of surviving cells accordingly, while Figure 5.3c determines the Euclidean's distance between each cell and the nearest surviving cell in its neighbourhood. Finally, Figure 5.3d shows receptive field boundaries for each surviving cell.

The same procedure is repeated in Figure 5.4 on Page 57 where the intensity values of example 4.5.1 are re-used. Figure 5.4a shows the disparity vectors associated with each cell, Figure 5.4b shows minimum values of F and the choice of surviving cells accordingly, while Figure 5.4c determines the minimum average gray level difference (intensity at the base and mean at upper levels). Finally, Figure 5.4d shows receptive field boundaries for each surviving cell.

However, at this point, we may face a lot of cases in which we have to break these criteria. In the example shown in Figure 5.5 on Page 59, assume that we need to estimate the disparity and the values shown are the disparity differences between this particular cell and the nearest surviving cell (in terms of disparity). The nonsurviving cell, included in the dashed square, cannot be linked to the surviving cell at the right, which is the only

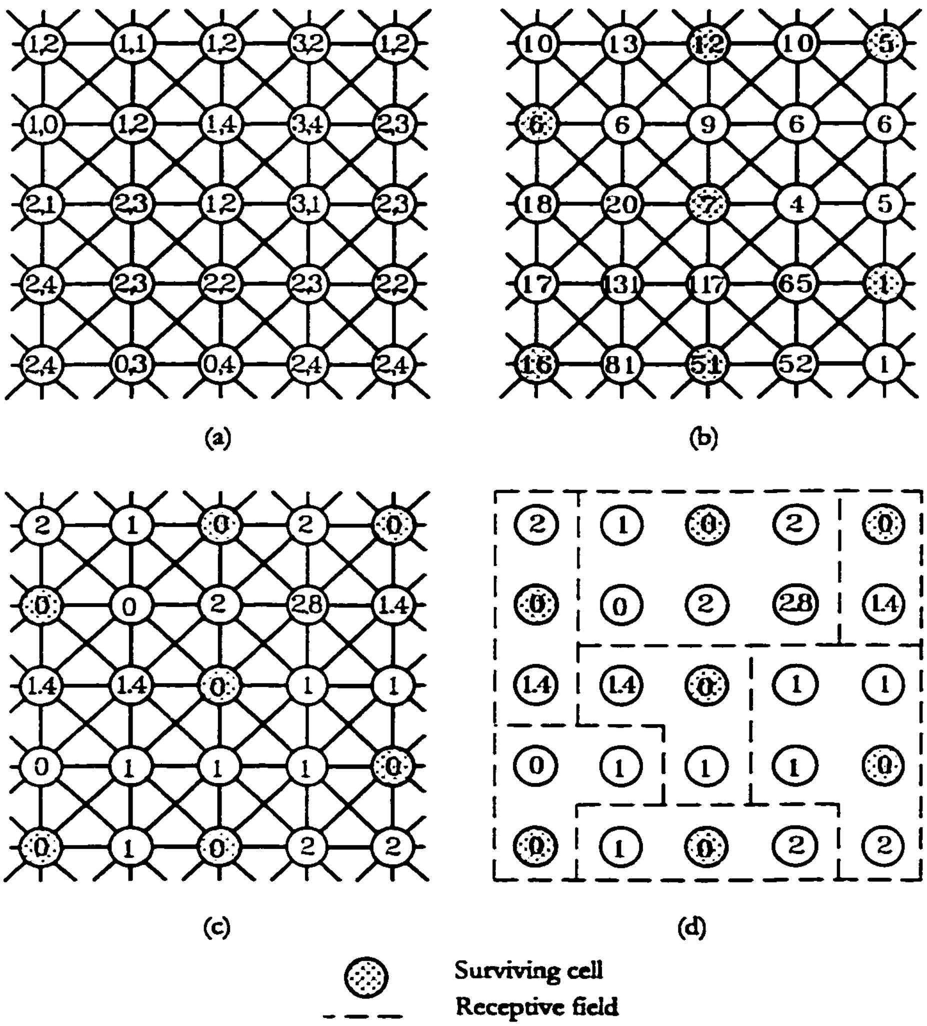


Figure 5.3: Disparity estimation: Which cell to choose? (a) Disparity vector values; (b) Minimum F values calculated as shown before and the choice of surviving cells is marked using the hatched circles; (c) Distance between each cell and the nearest surviving cell (in terms of disparity) in its neighbourhood; (d) Receptive field boundaries.

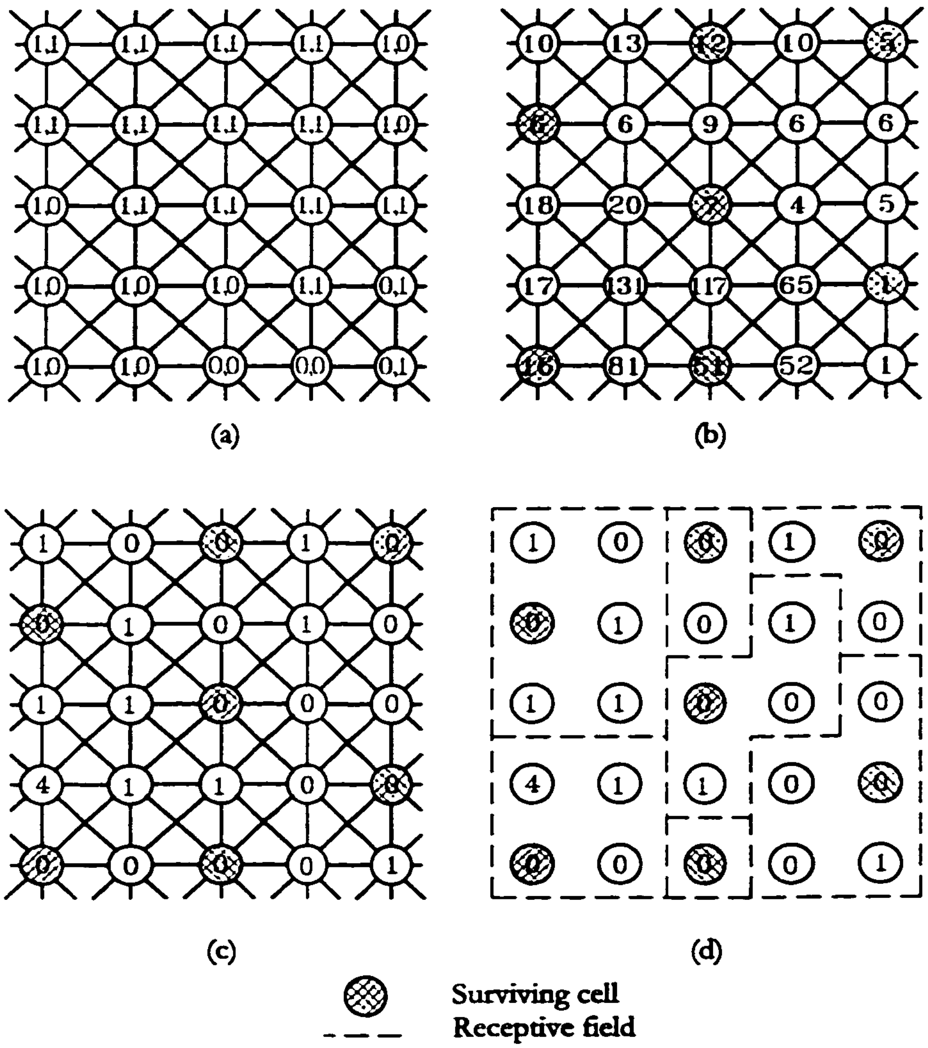


Figure 5.4: Object extraction: Which cell to choose? (a) Disparity vector values; (b) Minimum F values; (c) Minimum average gray level difference; (d) Receptive field boundaries.

surviving cell in its neighbourhood unless *min_dist* parameter[2] ≥ 8 (the difference 9-1) but if $min\_dist < 8$ then we consider it as noise and should be connected to the parent of the nearest brother (in terms of disparity). Again we may face another problem, if the parent of this brother is already created or not. That's why these locations, i.e., the noise, are indicated by the state variable $q = 2$ if the parent exists and $q = 3$ otherwise.

The same is valid in the case of object extraction if the values shown are the mean of the receptive field. Again, the nonsurviving cell, included in the dashed square, cannot be linked to the surviving cell at the right, unless *min_cont* parameter[3] ≥ 8 otherwise we consider it as noise and should be connected to the parent of the nearest brother (in terms of mean).

To link a nonsurviving cell (m, n, h) to (i, j, h)
In the case of object extraction,

$$\begin{aligned} &\text{if } \mid \mu_{ijh} - \mu_{mnh} \mid \leq min\_cont \\ &\quad \text{if } parent_{ijh} \text{ exists} \\ &\quad \text{then } q_{mnh} = 2 \\ &\quad \text{else } q_{mnh} = 3; \end{aligned} \tag{5.5}$$

In the case of disparity estimation,

$$\begin{aligned} &\text{if } dist(ijh, mnh) \leq min\_dist \\ &\quad \text{if } parent_{ijh} \text{ exists} \\ &\quad \text{then } q_{mnh} = 2 \\ &\quad \text{else } q_{mnh} = 3; \end{aligned} \tag{5.6}$$

[2] *min_dist* is a parameter to the algorithm needed to distinguish regions with different disparity vectors (see Section 5.10 on Page 69).

[3] *min_cont* is a parameter to the algorithm needed to distinguish regions with different gray levels (see Section 5.10 on Page 69).

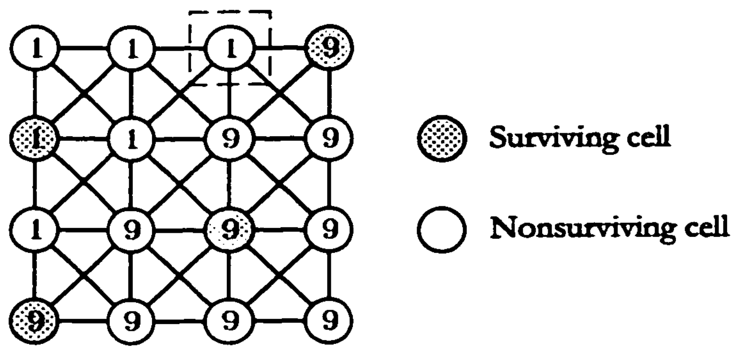


Figure 5.5: An example of a case where a nonsurviving cell (i, j, h) has no surviving cell (m, n, h) having a mean difference $< min\_cont \in Support_{ijh}$ if the values shown are the ones of the intensity (mean for the upper levels) or has no surviving cell (m, n, h) having disparity difference $< min\_dist \in Support_{ijh}$ if the values shown are the ones of the disparity differences.

5.4.2 Linking cells

As mentioned before, we used the absolute difference between the mean gray level of the regions (in the case of object extraction) and the disparity variation (in the case of disparity estimation) to be the contrast measure (see Section A.15 on Page 103). So, if (i, j, h) is a nonsurviving cell and (m, n, h) is the least contrasted surviving cell in its support $Support_{ijh}$, then:
In the case of object extraction,

$$\begin{aligned} &(i, j, h) \longrightarrow (m, n, h) \\ &\text{if and only if } \mid \mu_{ijh} - \mu_{mnh} \mid = \min \mid \mu_{ijh} - \mu_{klh} \mid \\ &\text{for } (k, l, h) \in Support_{ijh} \text{ such that } p_{klh} = 1 \end{aligned} \tag{5.7}$$

where
μ_{ijh} is the mean value of the receptive field of (i, j, h), i.e., $Receptive_{ijh}$.
In the case of disparity estimation,

$$\begin{aligned} &(i, j, h) \longrightarrow (m, n, h) \\ &\text{if and only if } dist(ijh, mnh) = \min dist(ijh, klh) \\ &\text{for } (k, l, h) \in Support_{ijh} \text{ such that } p_{klh} = 1 \end{aligned} \tag{5.8}$$

where
$dist(ijh, mnh)$ is the minimum Euclidean's distance between $(disp_x, disp_y)_{ijh}$ and $(disp_x, disp_y)_{klh}$.

Note that (m, n, h), the surviving cell is *not* linked to any other surviving cell at the same level. Always, this cell has at most one parent $(m, n, h+1)$ and at least one child $(m, n, h-1)$ (see Section A.17 on Page 104). Of course, there are no children for the cells in the base of the pyramid and no parents for those at the highest level. A cell always carries the statistical parameters (size, mean, disparity values) of its receptive field. Recall that, the set of cells in the base of the pyramid linked to a surviving cell (m, n, h) is the receptive field, $Receptive_{mnh}$, of (m, n, h).

5.5 New neighbourhood construction

One important question may rise up here, after choosing the surviving cells for the next new level, how can we determine the new neighbourhood for each cell? Recall that as a main property of all the irregular pyramids, we cannot know the neighbours of a surviving cell before the other surviving cells have been identified. This is in contrast with the case of regular pyramids in which we can use the coordinates of the cell to know its neighbourhood at each level. In the case of disparity pyramid, the decision of surviving a cell depends on the gray level or the intensity of the image and the input data as well (see Section 5.10 on Page 69). There are two methods to construct the new neighbourhood, the first uses the coordinates of receptive field cells, and the other uses the children of the surviving cell.

5.5.1 Receptive-field-based neighbourhood

In this method, we should check the locations of the receptive field cells of both cells under consideration. For example, if we want to check if (i, j, h) is a brother or a neighbour of (m, n, h) and suppose they have cells $(I, J, 0)$ and $(M, N, 0)$ in their receptive fields such that $(I, J, 0)$ belongs to $Receptive_{ijh}$ and $(M, N, 0)$ belongs to $Receptive_{mnh}$ then (see Sections A.9

and A.18 on Pages 100 and 108 respectively):

$$\begin{aligned}&\text{When } (I,J,0) \in \textit{Receptive}_{ijh}\\&\text{and } (M,N,0) \in \textit{Receptive}_{mnh}\\&\text{if } (|I-M| \leq 1) \text{ OR } (|J-N| \leq 1)\\&\text{then } (i,j,h) \text{ is a brother of } (m,n,h);\end{aligned} \tag{5.9}$$

5.5.2 Children-based neighbourhood

The idea beyond this method takes the advantage of knowledge of the linking information between the cell at the new level, the parent, and its children at the lower level. For example, if we want to check if (i, j, h) is a brother or a neighbour of (m, n, h) and suppose they have cells $(I, J, h-1)$ and $(M, N, h-1)$ as children such that $(I, J, h-1)$ is a child of (i, j, h) and $(M, N, h-1)$ is a child of (m, n, h) then:

$$\begin{aligned}&\text{When } (I,J,h-1) \text{ is a child of } (i,j,h)\\&\text{and } (M,N,h-1) \text{ is a child of } (m,n,h)\\&\text{if } (I,J,h-1) \text{ is a brother of } (M,N,h-1)\\&\text{then } (i,j,h) \text{ is a brother of } (m,n,h);\end{aligned} \tag{5.10}$$

Because "$(I, J, h-1)$ is a brother of $(M, N, h-1)$" means that they have a common border between each other, even if this common border is only one pixel long. So, if the small regions characterized by $(I, J, h-1)$ and $(M, N, h-1)$ included in bigger ones, namely (i, j, h) and (m, n, h) respectively, this means that the two bigger regions have a common border, i.e., they are neighbours or brothers. This idea is illustrated in Figure 5.6 on Page 63.

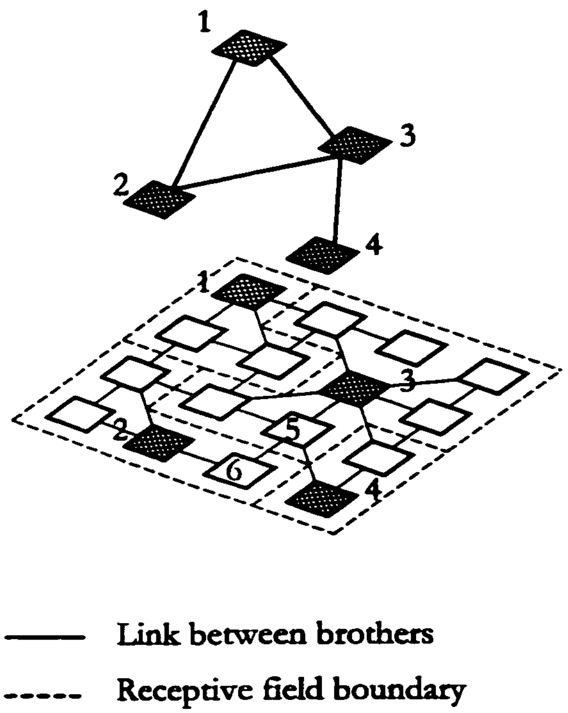


Figure 5.6: If a child is a brother of another then their parents are brothers. In this example cell 2 is a brother of cell 3 because cell 5 is a brother of cell 6 while cell 2 is not a brother of cell 4 because they do not have such a property.

5.6 State variables adjustment

After constructing the new neighbourhood $Brothers_{ijh}$ for each surviving cell (i, j, h) at the new level, we must recompute the values of the the state variables p_{ijh} and q_{ijh} for (i, j, h) according to the new graph construction. So, for every cell (i, j, h) at the new level, we should check its neighbourhood $Brothers_{ijh}$. Suppose that (m, n, h) is a neighbour of $(i, j, h,)$ then:
In the case of object extraction,

$$\begin{aligned} &\text{if } \mid \mu_{ijh} - \mu_{mnh} \mid < min\_cont \\ &\text{then } p_{ijh} = 0 \text{ and } q_{ijh} = 1 \\ &\text{else } p_{ijh} = 2 \text{ and } q_{ijh} = 0; \end{aligned} \tag{5.11}$$

In the case of disparity estimation,

$$\begin{aligned} &\text{if } dist(ijh, mnh) < min\_dist \\ &\text{then } p_{ijh} = 0 \text{ and } q_{ijh} = 1 \\ &\text{else } p_{ijh} = 2 \text{ and } q_{ijh} = 0; \end{aligned} \tag{5.12}$$

where

- min_cont, (see Section 5.10 on Page 69), is a parameter to the algorithm and its value is essential to distinguish between two different objects in the scene.
- min_dist, (see Section 5.10 on Page 69), the same as the previous entry, it is a parameter to the algorithm and its value is essential to distinguish between two different disparity vectors in the scene.
- μ_{ijh} is the mean value of the receptive field of (i, j, h), $Receptive_{ijh}$.
- $p_{ijh} = 2$ means that (i, j, h) is a root (see Section 5.8.1 on Page 66) and it should not be connected to any other cell as a child except itself.

This idea is presented in Section A.10 on Page 100.

5.7 Statistical computation

The algorithm presented can operate either on the original input image at the base, or on images, at the upper levels, generated from the input using local operations in which the new value of a cell depends on the old values of it and the set of its neighbours. We are concerned with two classes of statistical computations, namely the mean and the disparity values.

5.7.1 Computing the average value

Linking parents with children should, as expected, change the statistical parameters associated with the cell. Because of the change in the receptive field of the parent (i, j, h), the parameters s_{ijh}, its size and μ_{ijh}, its mean should change accordingly, i.e., for every cell (m, n, h) belonging to $Receptive_{ijh}$:

$$\begin{aligned} &\text{For } (m,n,h) \in Receptive_{ijh} \\ &s_{ijh} = s_{ijh} + s_{mnh}; \\ &\mu_{ijh} = \mu_{ijh} + \mu_{mnh} \times s_{mnh}; \end{aligned} \tag{5.13}$$

This should be followed by the mean adjustment:

$$\mu_{ijh} = \frac{\mu_{ijh}}{s_{ijh}}; \tag{5.14}$$

This idea is presented in Section A.11 on Page 101.

5.7.2 Computing the disparity value

The second main class of the statistical parameters to be computed is the disparity values of the surviving cells. Based on the disparity range $(dx_{min}, dx_{max}, dy_{min}, dy_{max})$ (see Section 5.10 on Page 69) and the disparity equation 4.3 (see Section 4.5 on Page 42), for every cell (i, j, h), we compute:

$$
\begin{aligned}
&\text{For } dx \in [dx_{min}, dx_{max}] \text{ and } dy \in [dy_{min}, dy_{max}] \\
&\quad \text{For } (m, n, h) \in Receptive_{ijh} \\
&\qquad F(dx, dy) = \textstyle\sum_{(m,n,h) \in Receptive_{ijh}} F_{mnh}; \qquad (5.15) \\
&F_{ijh} = \min{}_{dx \in [dx_{min}, dx_{max}], dy \in [dy_{min}, dy_{max}]} F(dx, dy); \\
&(disp_x, disp_y)^T_{ijh} = (dx, dy)^T;
\end{aligned}
$$

where

- $F(dx, dy)$ is the sum of all disparity values associated with all cells $\in Receptive_{ijh}$. dx and dy indicate the disparity vector used to compute this value.
- $(disp_x, disp_y)^T_{ijh}$ is the new disparity vector associated with the cell(i, j, h) computed using the minimum value of the previous item, $F(dx, dy)$, over the whole disparity range.

5.8 Object detection

5.8.1 What is the root?

In the disparity pyramid , as in the adaptive pyramid [JM91] the estimation of a disparity or the detection of an object in a scene is exactly the detection of one cell at an appropriate level. Indeed, from level h to level $h + 1$ the number of cells associated with a disparity vector or contained in a particular object is strictly a decreasing function. Then, we must control the decimation procedure to be sure that there will be one cell for each vector or object at the highest level of the pyramid. This is the root extraction process . The *root* can be defined as a surviving cell all the way from the base turning to nonsurviving cell that verifies a root predicate.

5.8.2 Root decision criteria

This root predicate must take into account different factors. First, a cell is a root if all the contrast measures with the surviving cells of its support are too large. However, we must not take into account local variation due to the noise, i.e., small regions, so the size of the root must be large enough and that's why we have to determine a minimum size, *min_size*, as a parameter to the algorithm to take care of this decision (see Section A.2 on Page 95). A small region must compensate its low size by a high contrast with all its neighbours. We used the following procedure, represented in [JM91] (see Section A.16):

Let (i, j, h) be a nonsurviving cell and (m, n, h) be the most similar surviving cell in its support. The cell (i, j, h) is called a root if and only if:

$$| \mu_{ijh} - \mu_{mnh} | > M(s_{ijh}) \tag{5.16}$$

in the case of object extraction and,

$$dist(ijh, mnh) > D(s_{ijh}) \tag{5.17}$$

in the case of disparity estimation.

where

- s_{ijh} is the size of (i, j, h) receptive field ($Receptive_{ijh}$);
- Function M is defined as:

$$M(s) = \begin{cases} min\_cont & \text{if } s > min\_size \\ min\_cont \times e^{\alpha(min\_size - s)} & \text{otherwise} \end{cases} \tag{5.18}$$

- Function D is defined as:

$$D(s) = \begin{cases} min\_dist & \text{if } s > min\_size \\ min\_dist \times e^{\alpha(min\_size - s)} & \text{otherwise} \end{cases} \tag{5.19}$$

5.8.3 Stopping criteria

All the above modules should be repeated iteratively until satisfying a certain stopping criterion. This is done by checking all surviving cells if they are turned into roots. This is an important property to this pyramid that all cells at the highest level should be roots, so for every cell (i, j, h) at the current level, we should check:

$$\begin{array}{l}\text{if } p_{ijh} \neq 2 \\ \text{then break the search and continue processing;}\end{array} \tag{5.20}$$

This is presented in Section A.12 on Page 101.

5.9 Practical problems

5.9.1 No decreasing function

One practical problem may rise up here if the input values cannot converge into one stable state, e.g. if *min_cont* and/or *min_dist* are not big enough to help decide whether the cell is a root or not, or to merge it with another cell, this could lead to recomputing the same cells for the next level. Because we cannot control the inputs as they should vary according to the gray levels of the objects contained in the scene and/or camera locations, we should force the processing to stop if the number of roots does not decrease. Recall that from level h to level $h+1$ the number of cells contained in a particular object is strictly a decreasing function [JM91].

5.9.2 More than one root

Even with this process, we may end up with more than one root for a unique object if all cells represent this object disappear at the same level and/or the *min_cont* or *min_dist* is too small to distinguish the whole object, rather it is distinguishing subparts from it. We decide that root is always a surviving cell, so the first principle is not always verified [JM91] (see Section 5.2 on Page 48).

Note that, as it must be, the roots are detected at many levels on the basis of the size of the objects they represent. At the highest level of the pyramid, each root is associated with one component, i.e., object or disparity vector, of the scene. Thus, we have segments of the scene. This segmentation could, of course, high depend on the criteria we used.

5.10 Parameters to the algorithm

We have 5 essential parameters for our algorithm (see Section A.2 on Page 95):

- The value of α which is computed given the value of the size 1 (the highest contrast corresponds to the smallest size). In our experiments, we used α=0.2;
- The values of the disparity range (dx_{min}, dx_{max}, dy_{min}, dy_{max}) which specify the comparison window between both images;
- The value of the *min_dist* needed to distinguish between two different regions having different disparity vectors;
- The value of the *min_cont* needed to distinguish between two different regions having different gray levels;
- The value of the *min_size* needed to decide if or not this region is a real object in the scene or just a noise.

The second and the third parameters, i.e., the disparity range and the minimum distance depend on camera locations, i.e., the view points of the scene, while the last two parameters, i.e., minimum contrast and minimum size depend on the gray level and the size of the image. Also, these parameters could be location dependent, e.g. *min_cont* could be satisfied at a location but not at the other.

5.11 Types of nonsurviving cells

One last point should be mentioned here is that there are three kinds of nonsurviving cells.

- First, the root characterizing a particular region, extracted as presented in Section 5.8. Recall that the *root* is a surviving cell all the way from the base turning to nonsurviving cell at certain level (once all the cells representing the region have been collected together). The height of this level depends on the area of the region represented by this root in the scene.
- Second, the subparts of a component. They may represent sufficiently large regions but are not sufficiently contrasted with their neighbours. They are linked to the most similar surviving cell. This is the case of Figure 5.3c on Page 56 if $min\_dist \geq 2.8$.[4] Also, the case of Figure 5.4c on Page 57 if $min\_cont \geq 4$.[5]
- The last kind of nonsurviving cells are the highly contrasted small regions. We decide that they correspond to nonsignificant objects or

[4] provided that the size of the receptive field $\geq min\_size$.

[5] provided that the size of the receptive field $\geq min\_size$.

to a noise effect. they are also linked to the most similar surviving cell or to the parent of the most similar nonsurviving cell in their neighbourhood (see Figure 5.5 on Page 59).[6]

5.12 Summary

This chapter dealt with a proposed hierarchical technique to segment the scene into layers of constant disparity. The same technique could be used to segment the scene according to different gray levels involved. An irregular pyramid, the disparity pyramid, is implemented using two basic pyramidal rules. A cell is retained for the next level if and only if it is a local minimum among its support. A nonsurviving cell (a child) at a level is linked to the nearest surviving cell (a parent) at the upper level. The measure of this distance could be the Euclidean's distance as in the case of disparity estimation or the absolute gray level difference as in the case of object extraction. The whole process should stop if all cells turn into roots, i.e., all cells at the highest level should be far enough from each other. Each root at the highest level represents a disparity vector in the case of disparity estimation or an object in the case of object extraction.

[6]provided that the size of the receptive field $\leq$ *min_size*.

Chapter 6

Experimental Results

6.1 Introduction

In this chapter, experimental results will be presented. The disparity pyramid is implemented to go through two streams to achieve two goals. The first goal is to estimate the disparity between a stereoscopic pair of images and to segment the scene accordingly. The second is to extract objects from a scene according to their average gray level using two stereoscopic images.

6.2 Disparity estimation

To estimate the disparity between a stereoscopic pair of images, the parameter *min_dist* is used to link regions with different disparities under its value, i.e., with disparity differences less than or equal to *min_dist*.

The results of the disparity estimation process are presented in two ways. The first shows different point correspondences and the second presents a segmentation process of the scene according to the disparity values. Different types of stereo images are used including synthetic, outdoor and indoor scenes.

6.2.1 Point correspondence results

Figures 6.1, 6.2 and 6.3 on Pages 74, 76 and 77 respectively show the final result of the estimation process using the disparity pyramid. The test points are marked by the dotted white squares on both images.

Figure 6.1 on Page 74 shows two examples of synthetic scenes. Notice that the second example represents a special case where the background is exactly the same in both images, i.e., the disparity vector of any point included in this background is $(0, 0)^T$. However, this is not the case of real world scenes. The only difference between both images is the location of the small square which makes some area of the background in the first image hidden in the second. Similarly, some area of the background in the second image is hidden in the first. This area is called an *occlusion area* where there is no enough information about it in both images. Generally, this kind of area, the occlusion area, is a source of errors in image understanding. This is the case of our example, where we find an error occurred in this area.

Consider Figure 6.2 on Page 76. The first pair, the house, represents an outdoor scene where some ambiguities take place. In this example, there is no distinction between the house and the trees located behind its ceiling. In other words, there is no enough features to distinguish between those two objects. This case is repeated again between the house and the fence

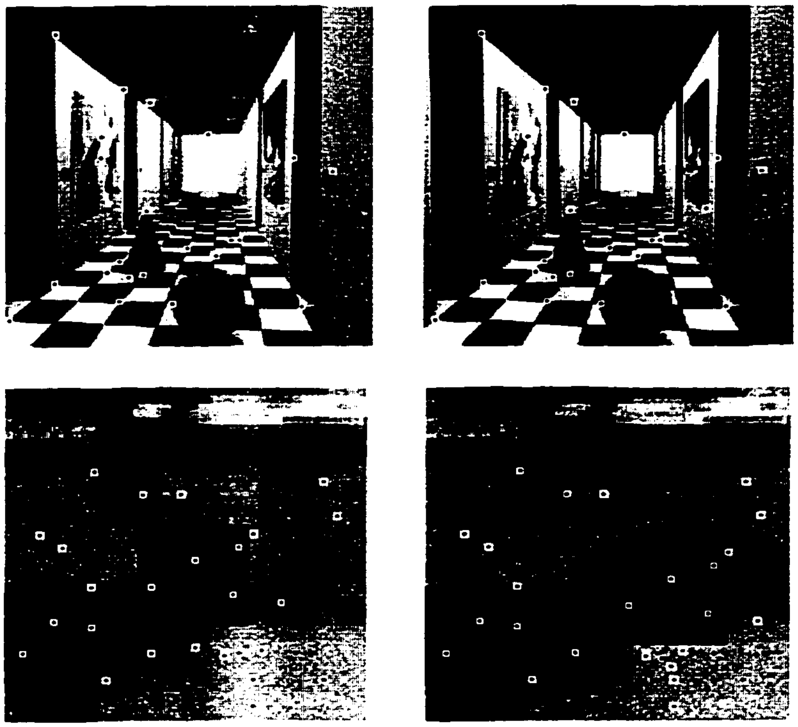


Figure 6.1: Disparity estimation: Synthetic scenes, the dotted white squares show the original points in the first image and the same points (disparity estimated) in the second. Notice the error occurred in the occlusion area in the second pair.

in front of it which makes the fence as if it were a part of the elevation of the house. In disparity estimation, our model does not distinguish between different real objects included in the scene as long as their disparity values are the same, or they have a disparity difference but it is less than or equal to *min_dist*. Again, we might expect some errors to appear in these areas.

In Figure 6.3 on Page 77, two pairs of indoor scenes are presented. The most important note to be mentioned here is the lack of features in some areas. Consider the first pair, we notice that the background is overlighted which makes, for example, some areas of the book shelves appear as a continuity of the wall beside them. This results in a wrong perception and reduces the information needed to understand the scene.

Consider the second pair, the chair and the desk, where there is no enough features on the wall behind the chair or on the back of the desk. As the previous case, we might expect some ambiguities in recognizing those areas.

One last note should be mentioned here is that, at the highest level of the pyramid, each root represents only one distinctive disparity vector or, more accurately, it may represent more than one disparity vector with differences less than or equal to *min_dist*.

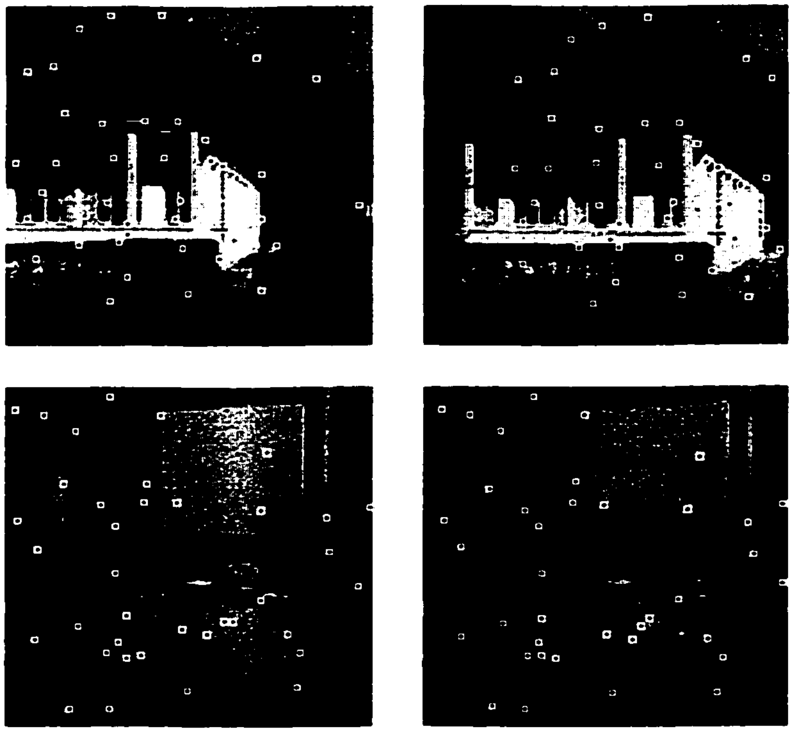


Figure 6.2: Disparity estimation: Outdoor and Indoor scenes. (These images are INRIA-Syntim © copyright.)

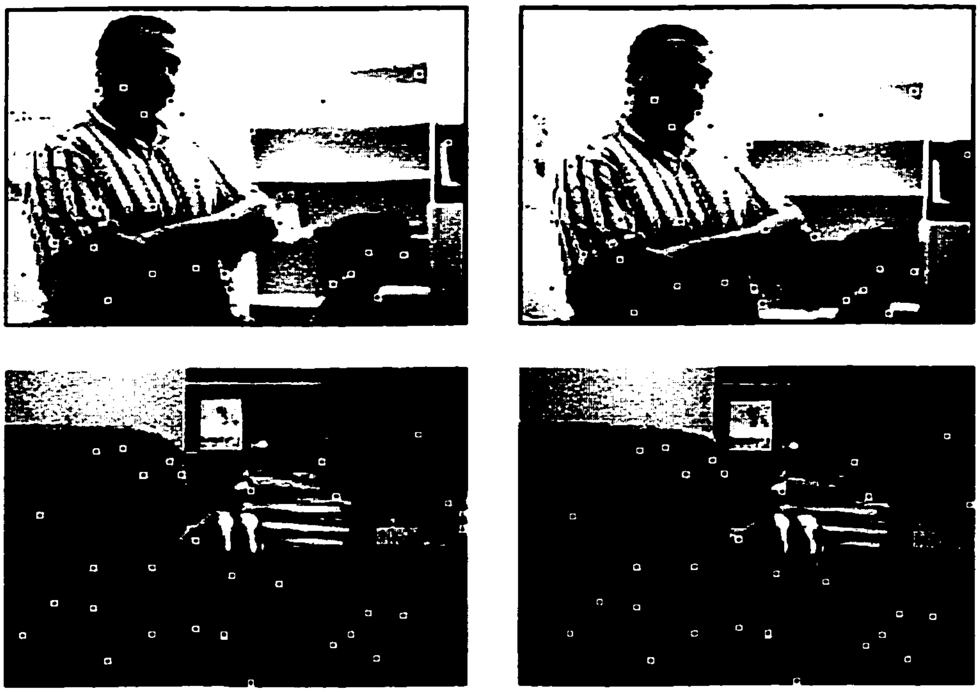


Figure 6.3: Disparity estimation: Indoor scenes were taken in KAML lab.

6.2.2 Segmentation results

In order to get clear understanding of disparity estimation process, we segmented each scene into layers of constant disparity values, or with disparity differences which are less than or equal to *min_dist*. Each layer represents the receptive field of a root at the highest level of the pyramid. Recall that each root represents a disparity vector.

A very important note should be mentioned here is that, as the disparity pyramid segmenting the image according to the disparity differences, each root may include in its receptive field more than one real object in the scene. However, all objects contained under one root have disparity differences less than or equal to *min_dist* between each other. Recall that the values of *min_dist* and disparity vectors are fully dependent on the information contained in the scene or, more specifically, camera locations.

Figures 6.4, 6.5, 6.6, 6.7 and 6.8 on Pages 79, 80, 81, 82 and 83 respectively show the receptive field of each root at the highest level of the pyramid.

Consider Figure 6.4 on Page 79. The second pair shows the result of disparity estimation using the disparity pyramid where each image of this pair represents the receptive field of a root or, in other words, the receptive field of a disparity vector. The disparity vector of the left is $(3,0)^T$ and of the right is $(2,0)^T$.

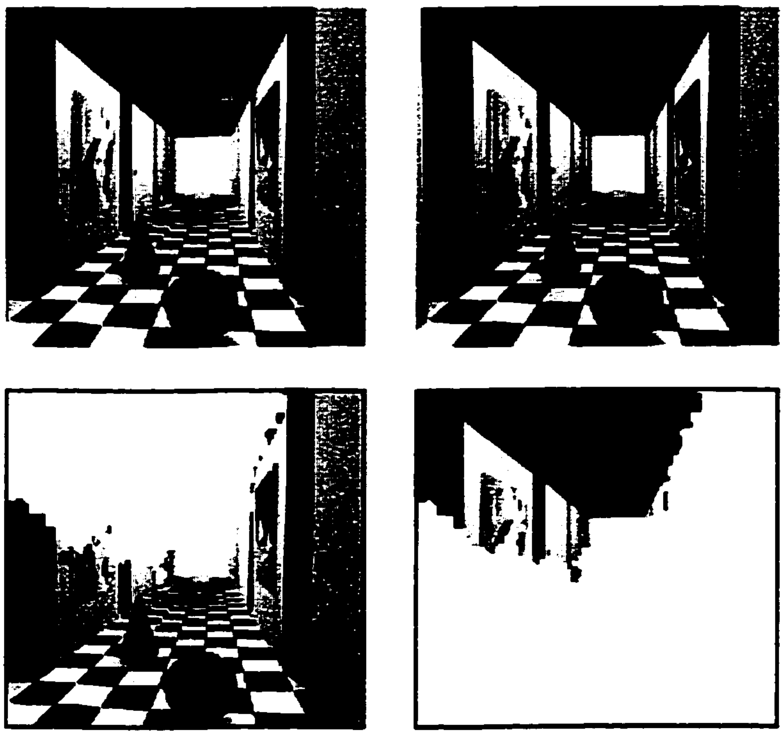


Figure 6.4: Disparity estimation: The second pair shows the result of disparity estimation using the disparity pyramid where each image of this pair represents the receptive field of a root, or in other words, the receptive field of a disparity vector. The disparity vector for the left is $(3,0)^T$ and for the right is $(2,0)^T$.

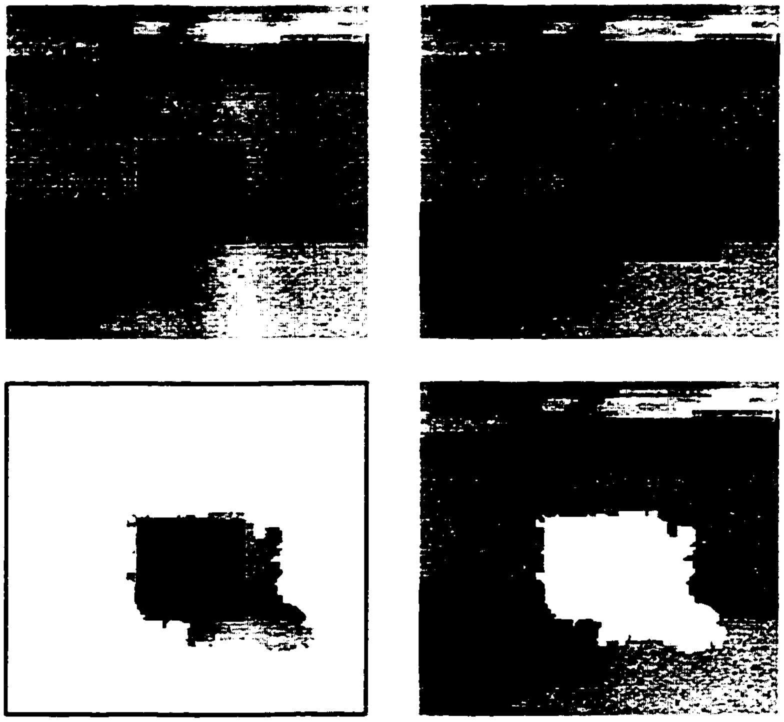


Figure 6.5: Disparity estimation: Receptive field of the roots. The disparity vector for the left is $(20, 9)^T$ and for the right is $(0, 0)^T$. Note the error occurred in the occlusion area.

Figure 6.5 on Page 80 shows the other synthetic example, the square. Note that the segmentation process results in two layers of constant disparity values, as expected, where the disparity vector of the square is $(20, 9)^T$ and of the background is $(0, 0)^T$. The problem with the occlusion area is very clear in this example. A part of this area is linked to each of the two layers obtained.

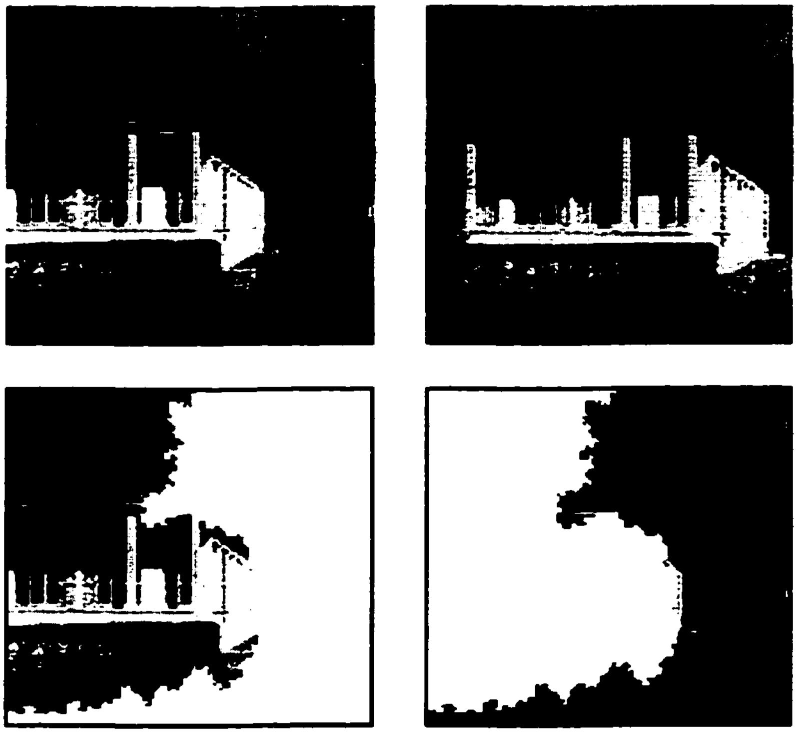


Figure 6.6: Disparity estimation: Receptive field of the roots. The disparity vector for the left is $(27,2)^T$ and for the right is $(30,1)^T$.

Figure 6.6 on Page 81 presents the segmentation process for the previous house example. Note that, as mentioned in Section 6.2.1, due to the lack of features, the fence and the tree area behind the ceiling are considered having the same disparity (or with a difference less than or equal to min_dist) as the house. The segmentation process results in a disparity vector, $(27,2)^T$, for the house and another disparity vector, $(30,1)^T$, for the background.

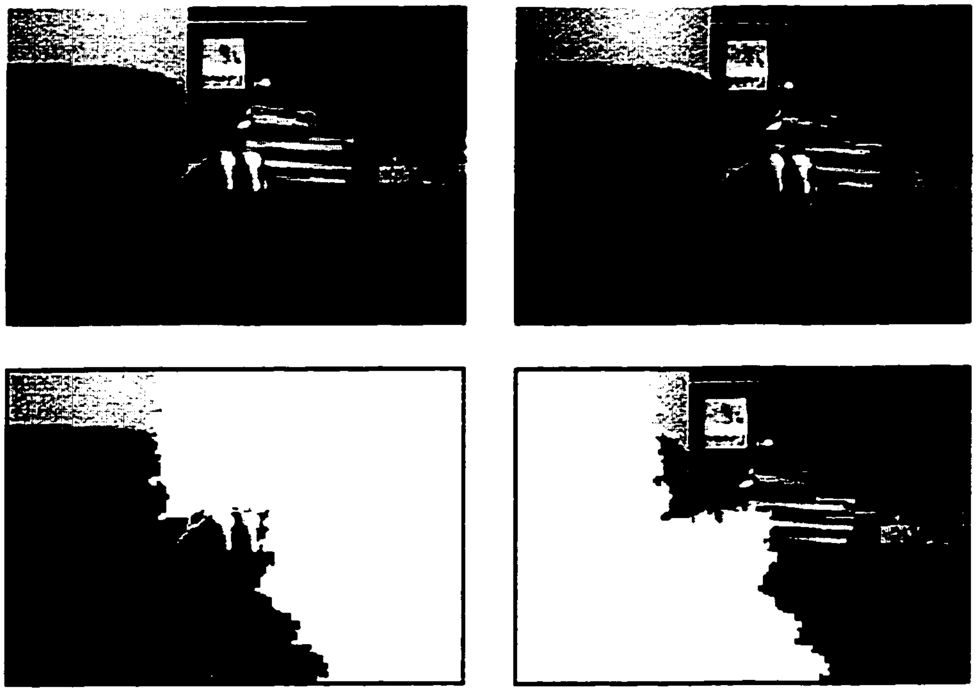


Figure 6.7: Disparity estimation: Receptive field of the roots. The disparity vector for the left is $(7,0)^T$ and for the right is $(7,1)^T$.

Figure 6.7 on Page 82 represents the segmentation process of the chair example. As expected, due to a lack of features, some areas belonging to the wall and to the desk are incorrectly merged with the chair.

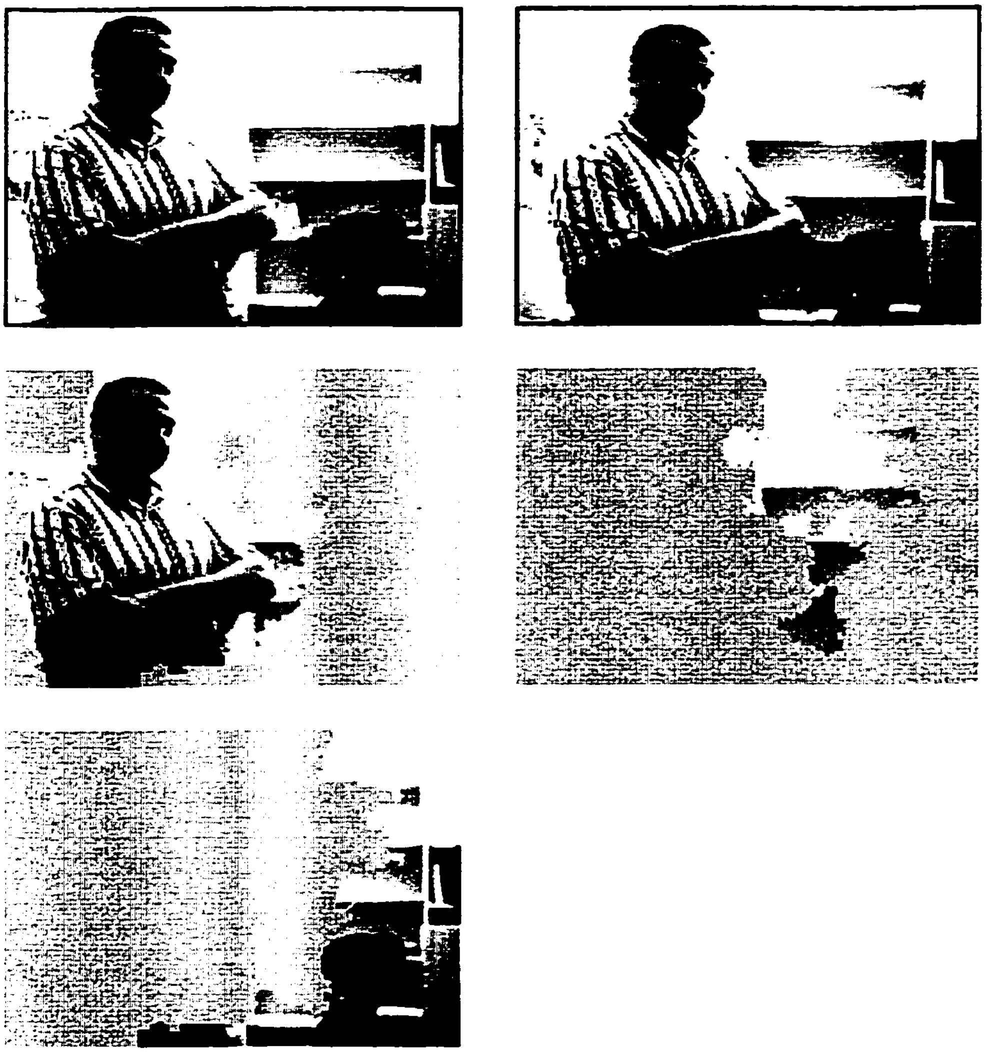


Figure 6.8: Disparity estimation: Receptive field of the roots. The disparity vector for the upper left (of the root) is $(7,5)^T$ and for the right is $(-5,6)^T$ and for the lower left is$(-5,7)^T$.

6.3 Object extraction

To extract an object from a scene using the disparity pyramid, the parameter *min_cont* is used to merge regions with different average values under its value, i.e., with average difference less than or equal to *min_cont*. The following figures show the final result of the extraction process using the disparity pyramid.

At the highest level of the pyramid, each root represents one object using the variation of the mean value of its gray level and mean value of the surrounding media. A note should be mentioned here is that, as the disparity pyramid segmenting the image according to the gray level average, each root may include in its receptive field more than one different gray area. However, all the areas included under one root have gray level difference less than or equal to *min_cont* between each other.

Figures 6.9 and 6.10 on pages 85 and 86 respectively show the receptive field of each root at the highest level of the pyramid. Figure 6.11 on page 88 shows a special case of using the disparity pyramid where only one image is used to perform object extraction with a disparity range of (0, 0, 0, 0).

Figure 6.9 on Page 85 shows a synthetic scene. The goal here is to extract the dollar image from the surrounding constant background. In this example, as the intensity value of the background does not change, the interest operator calculated to choose the surviving cells should be constant as well throughout the whole background (except for a small frame around the dollar whose width is determined by the disparity range entered as a parameter). Our rules require a surviving cell to have a minimum value

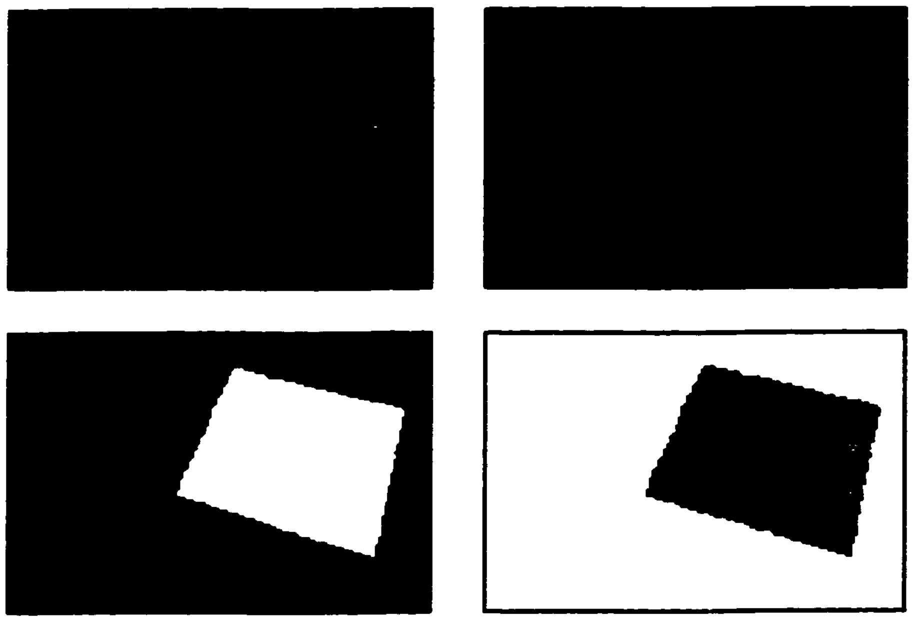


Figure 6.9: Object extraction. (These images are INRIA-Syntim © copyright.)

for the interest operator over its whole support. Clearly, the value of the interest operator for all cells included in the background should be 0 (the minimal value we can ever get). In this case, we should randomly choose a cell to survive taking into account the rules mentioned in Section 5.2 on Page 48. As the result shows, this random selection does not affect the whole process.

In Figure 6.10 on Page 86, although the background does not have a constant intensity value, the variations are too small. Besides, many intensity values are repeated throughout the background which forces the interest

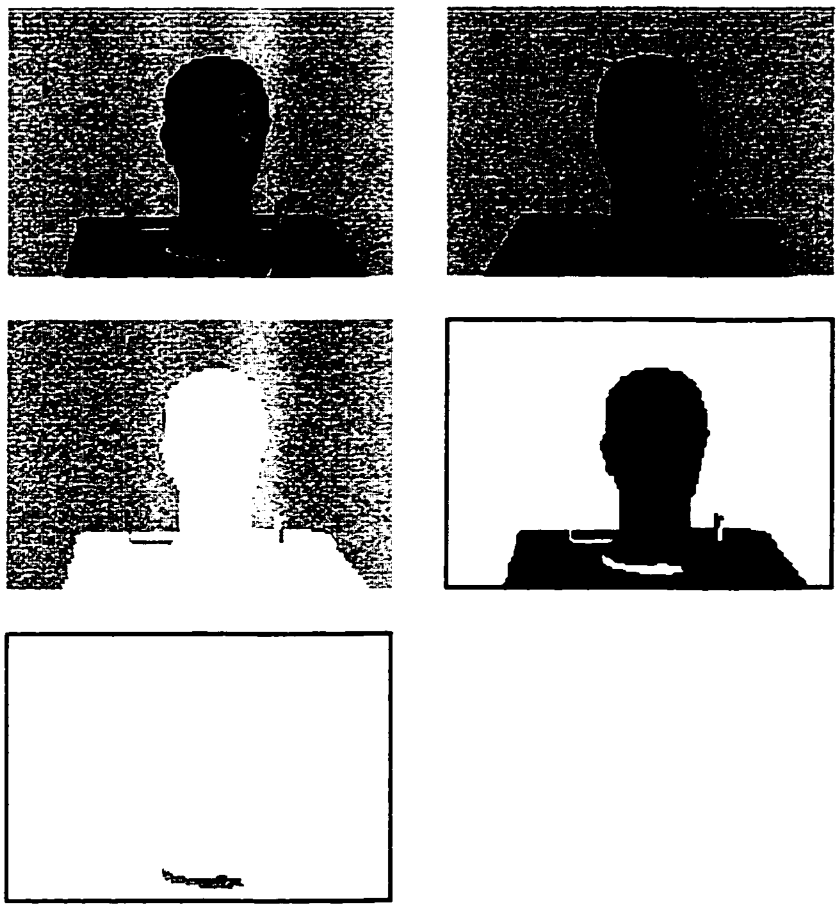


Figure 6.10: Object extraction. (These images are INRIA-Syntim © copyright.)

operator to have identical repeated values (some of those values reach no difference level, i.e., a value of 0). We treat this case as the previous one. In other words, we randomly choose a cell to survive from its support taking into account the rules mentioned in Section 5.2 on Page 48.

Figure 6.11 on Page 88 shows a special case where only one image is used, i.e., this case is not a stereo case. Because there is only one image, there is no disparity range required. In other words, the disparity range should be (0,0,0,0) in this case. Obviously, the interest operator value will not take any value but 0 for each and every cell in the image. Again, we randomly choose a cell to survive considering the rules stated in Section 5.2 on Page 48.

6.4 Summary

This chapter presented the experiments we implemented on the disparity pyramid. Those experiments could be divided into two major groups, disparity estimation and object extraction results. Each one of those groups could be divided into two sub-groups. For the disparity estimation, we divided the results into point correspondence and segmentation sub-groups. For the object extraction, we divided the results into stereo and mono sub-groups. Different types of scenes, including synthetic, indoor and outdoor scenes, were presented.

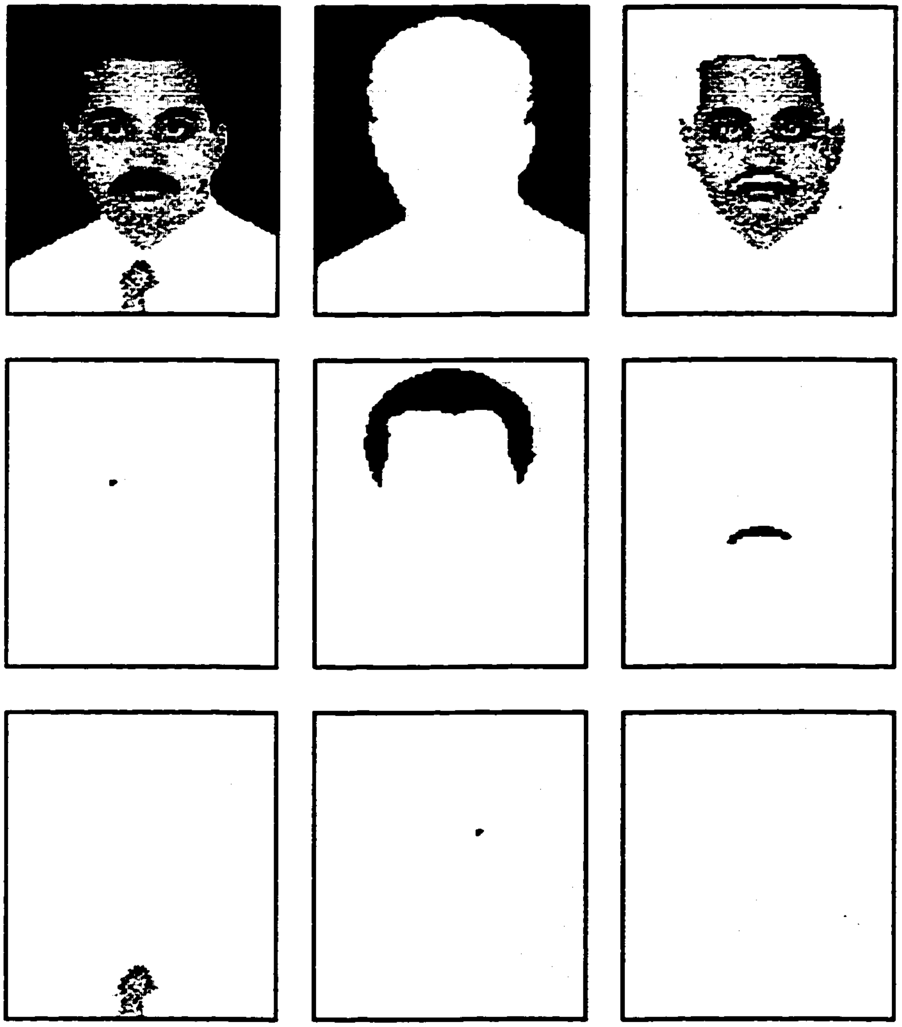


Figure 6.11: Object extraction: special case where one image is used with a disparity range (0,0,0,0).

Chapter 7

Conclusion and Future Work

The study of this thesis focused on exploring a new method to segment a scene into layers of constant disparity values using a stereoscopic pair of images, to estimate the locations of corresponding points and to extract objects with different gray level values from the scene.

Chapter 1 gave a general introduction to the topics under consideration including the stereoscopic vision and disparity estimation on one side and the concept of hierarchical processing and pyramids architecture on the other.

In order to grasp the whole picture of the problem, Chapter 2 discussed a literature review of previous methods developed in the area of stereo vision. Those methods were divided into many groups, each of them has its own view to deal with the problem. Sometimes, more than one of those methods are merged together to get better results. This chapter ends up with discussion on the regular pyramids and their use in stereo vision. This is an important point to our research because none of those pyramidal methods, as far as we could reach, tried to explore the behaviour of the irregular pyramid in stereo vision. This is the core point of this thesis research.

At this point, we had to study the characteristics of the pyramids. This is the topic of Chapter 3 which discussed the fundamental concepts of different pyramidal architectures including regular and irregular types. The definitions of hierarchy and multiresolution were given. Different types of processes were discussed as well as possible types of communication networks.

Chapter 4 dealt entirely with the disparity estimation problem. A discussion on monocular and binocular image formation was given. The disparity estimation problem was discussed. A basic difference equation based on intensity matching was suggested. This equation was used again in Chapter 5.

In Chapter 5, we proposed an irregular technique called, the disparity pyramid, in order to segment the scene into layers of constant disparity. We used the same technique to segment the scene into layers according to different gray level values. Two important pyramidal rules were used. The levels of this pyramid were built by successive extraction of surviving cells. The choice of a surviving cell was controlled by the value of an interest operator using the intensity difference values between the pair of images under consideration. This operation was limited within a disparity range including the whole possible set of disparity vectors of the scene objects. To build the inter-level network, a nonsurviving cell was linked to the nearest surviving cell in its support. However, the distance between these two cells could be measured in terms of disparity variations as the Euclidean's distance in the case of disparity estimation or in terms of gray level difference as the absolute gray level difference in the case of object extraction. At the last step all cells should turn to roots. A root, associated with each component of the scene, should be far enough from all the surviving cells in its neighbourhood. The bottom-up process was implemented to extract the receptive field of each root.

The results presented in Chapter 6 showed that the proposed technique, the disparity pyramid, is suitable to estimate the disparity between a stereo pair of images, to segment the scene according to different disparity values and to extract objects of different gray levels included in the scene.

The results showed that the same technique could work on individual images (mono case), considering the disparity range reaching its zero value, i.e., the disparity range becomes (0,0,0,0). As expected in this case, the value of the interest operator was always 0 and we had to randomly decide which cell would survive to the next level. However, this random choice did not affect the overall process.

As a matter of fact, the results showed that there are three cases in which we have to randomly decide which cell will survive. Besides the above-mentioned case, we have to do the same when the background has a constant intensity value or different intensity values which repeat continuously.

Many experiments implemented showed that a slight difference in parameter values would result in a significant difference in the output. The parameter *min_dist* is a crucial entry which controls the whole process of disparity estimation. Variations in order of thousands (if the distance between two successive pixels in the original image represents the unit) could produce completely different results. The same is valid for the parameter *min_cont* which controls the process of object extraction. These two parameters depend on some factors like camera locations and lighting conditions of the scene which are beyond the scope of the proposed technique.

Also the results showed that, as in the case of disparity estimation, some areas were incorrectly linked together, this resulted in inaccurate estimation for some points in the scene. This could happen at the object boundaries specially in the areas where there is no enough features included. One point should be mentioned here is that, equation 4.3 on Page 42 suggests that the neighbours of a cell in the first image are neighbours of the same cell in the second. This assumption is generally true for cells located inside object boundaries but usually false at the boundaries and occlusion areas.

Thus as a suggestion for a future work, the geometrical properties included in the scene are to be considered. For example, if the image is preprocessed for edge detection before considering disparity estimation, this could eliminate the existence of such a problem and enhance the accuracy of the results.

Finally if we analyze our work, we will find that the most costly step in the whole algorithm is the construction of the new neighbourhood when using the algorithm mentioned in Section 5.5.1. This brings the overall cost to a quadratic time searching for neighbours in the receptive field. This can be overcome, if we used a depth search technique with the algorithm mentioned in Section 5.5.2.

Appendix A

How to Build the Disparity Pyramid

This appendix presents a detailed algorithm to build the disparity pyramid. Section A.1 of this appendix introduces the notation used. Section A.2 states the main steps and initial values fed to the pyramid. The rest of this appendix presents the algorithms of different modules called in order to construct this structure.

Note that, we will build only one pyramid (*bottom* $-$ *up* process) for the first image while using the second image to compute the interest operator and disparity values. There is no need to build another pyramid for the second image.

A.1 Notation

(i, j, h): Region characterized by the cell (i, j) at the level h of the pyramid. Each cell (i, j, h), or region at upper levels, is associated with (Refer to the list of symbols for explanation):

- F_{ijh} : The calculated disparity value for the cell (i, j, h).
- $(disp_x, disp_y)_{ijh}$: The disparity vector associated with F_{ijh}.
- $Brothers_{ijh}$: The list of all neighbours of (i, j, h).
- $Support_{ijh}$: The list of the support of (i, j, h).
- $Receptive_{ijh}$: The list of all descendant of (i, j, h) at the level $h = 0$.
- s_{ijh} : Size of the receptive field of (i, j, h), at level h.
- μ_{ijh} : Mean of the receptive field of (i, j, h), at level h.
- $contrast_{ijh}$: Minimun contrast between (i, j, h) and its brothers.
- $contDst_{ijh}$: Minimum distance between $(disp_x, disp_y)_{ijh}$ and the surrounding vectors.
- $minDif_{ijh}$: Minimum difference between (i, j, h) and the least contrasted brother.
- $color_{ijh}$: Intensity (or colour) of (i, j, h).
- $parent_{ijh}$: The parent of (i, j, h), i.e. pointing to level $h + 1$.

A.2 Initialization

Function : This is the main module where the images are fed to the algorithm and the initial values are determined.

1. Input two perspective views, $(image_1,\ image_2)$, for the same scene, one of them will become the base of the pyramid;

2. Input the data: minimum contrast, (*min_cont*), and minimum number of cells allowed in the receptive field, (*min_size*), and the value of α;

3. Input the disparity range: $dx1$, $dx2$ and $dy1$, $dy2$.

4. $h = 0$; h is the level in the pyramid;

5. FOR all cells $(i, j, 0)$, initialize:

    ◦ s_{ij0} : size of the receptive field, $\equiv 1$ at the base;

    ◦ μ_{ij0} : mean of the receptive field, $\equiv color_{ij0}$ at the base;

    ◦ $Brothers_{ij0}$: a list of 8 brothers (using 8-connectivity) for the interal node, 5 brothers for the node at the edge of the image, and only 3 for the one at the corner;

    ◦ $Receptive_{ij0}$: the receptive field consists of only one entry ($\equiv$ the cell itself), at the base;

    ◦ F_{ij0} : The minimum disparity value computed for $(i, j, 0)$ after comparing the base and the other image ;

    ◦ $(disp_x, disp_y)_{ij0}$: disparity vector associated with F_{ij0};

6. FOR all the cells $(i, j, 0)$, set p_{ij0}=0;

7. FOR all the cells $(i, j, 0)$, set q_{ij0}=1;

8. Build the lowest level of the pyramid as a graph using the above information for each cell, taking into account that the pyramid is a list of such a graph;

9. Call *Build Levels*, (section A.3);

A.3 Build Levels

Function : Allocating and building different levels of the pyramid through calling different other modules.

do{

1. Call *ExtractLocalMinima*, (section A.4);
2. do{
   (a) Call *Adjust Q*, (section A.6);
   (b) Call *Extract SubMinima*, (section A.7);

   }while(there are new surviving cells);
3. Allocate memory for the new level;
4. Call *Linking*, (section A.8);
5. Call *Adjust Brothers*, (section A.9);
6. Call *Adjust P and Q*, (section A.10);
7. Call *Adjust Mean F*, (section A.11);
8. Call *Check Roots*, (section A.12);
9. Call *Final Adjust*, (section A.13);
10. Add the new level to the pyramid, the list of levels;

}while(we haven't reached the apex);

A.4 Extract Local Minima

Function : Setting the state variable p to 1 if the cell can survive.

FOR every cell $(i, j, h) \in$ the current level:

1. Call *Check Survive*, (section A.5);
2. IF the cell (i, j, h) can survive
   THEN p_{ijh}=1;

A.5 Check Survive

Function : Checking whether or not the cell can survive by examining the state variable q and the value of F for the whole support.

IF $q_{ijh} = 1$

THEN

$\forall\ (m, n, h) \in Support_{ijh}$

IF $(F_{ijh} > F_{mnh})$ AND $(q_{mnh} = 1)$

THEN

The cell cannot survive

ELSE

The cell can survive

ELSE

The cell cannot survive;

A.6 Adjust Q

Function : Setting the state variable q to 1 if the cell is a good candidate, or resetting it to 0 otherwise.

FOR every cell $(i, j, h) \in$ the current level:

1. Call *Check Candidate*, (section A.14);
2. IF the cell (i, j, h) is a good candidate
   THEN q_{ijh}=1
   ELSE q_{ijh}=0;

A.7 Extract SubMinima

Function : Checking the cells which have not been decided yet to survive or not and changing their p variables accordingly.

FOR every cell $(i, j, h) \in$ the current level:

IF (p_{ijh}=0)

THEN

(a) Call *Check Survive*, (section A.5);

(b) IF the cell (i, j, h) can survive
    THEN p_{ijh}=1;

A.8 Linking

Function : Linking every nonsurviving cell to the nearest surviving cell in the upper level.

FOR every cell $(i, j, h) \in$ the current level:

1. Call *Choose To Link*, to get (m, n, h) (section A.15);
2. Call *Calculate minimum Contrast*, (section A.16);
3. Call *Link Cell* to link $(i, j, h) \longrightarrow (m, n, h)$, (section A.17);

4. Many special cases may arise here;

A.9 Adjust Brothers

Function : Adjusting the list of neighbours for every cell.

FOR every cell $(i, j, h) \in$ the current level:

FOR every cell $(m, n, h) \in$ the current level:

IF $(i, j, h) \neq (m, n, h)$

THEN Call *Is Brother*, (section A.18);

IF (i, j, h) is a brother of (m, n, h)

THEN add (m, n, h) to $Brothers_{ijh}$;

A.10 Adjust P and Q

Function : Checking average difference or disparity distance between every two neighbours and setting p and q values accordingly.

FOR every cell $(i, j, h) \in$ the current level:

IF$(p_{ijh} \neq 2)$

THEN FOR every cell $(m, n, h) \in Brothers_{ijh}$:

IF $|\mu_{ijh} - \mu_{mnh}| < min\_cont$

$[dist(ijh, mnh) < min\_dist]$

THEN $p_{ijh} = 0$; $q_{ijh} = 1$;

ELSE $p_{ijh} = 2$; $q_{ijh} = 0$;

ELSE $q_{ijh} = 0$;

A.11 Adjust Mean F

Function : Adjusting the statistical parameters μ and F for every cell.

FOR every cell $(i, j, h) \in$ the current level:

1. $s_{ijh} = \mu_{ijh} = 0$;
2. FOR every cell $(m, n, h) \in Receptive_{ijh}$:

   $s_{ijh} = s_{ijh} + s_{mnh}$;

   $\mu_{ijh} = \mu_{ijh} + \mu_{mnh} \times s_{mnh}$;
3. $\mu_{ijh} = \frac{\mu_{ijh}}{s_{ijh}}$;

FOR every cell $(i, j, h) \in$ the current level:

1. FOR($dx : dx1 \Rightarrow dx2$)

   FOR($dy : dy1 \Rightarrow dy2$)

   FOR every cell $(m, n, h) \in Receptive_{ijh}$:

   Call *Compare Cell*, (section A.19);

   Calculate $F_{dx,dy} = \sum F_{mnh}$;
2. F_{ijh}=min$F_{dx,dy}$;
3. $(disp_x, disp_y)_{ijh} = (dx_{min\ mnh}, dy_{min\ mnh})$;

A.12 Check Roots

Function : Stopping the whole process if all cells convert to roots.

FOR every cell $(i, j, h) \in$ the current level:

IF($p_{ijh} \neq 2$)

THEN still more to process;

A.13 Final Adjust

Function : Adjusting the state variables p and q after checking the neighbourhood and receptive field size of each cell.

FOR every cell $(i, j, h) \in$ the current level:

IF($s_{ijh} > min\_size$)

THEN FOR every cell $(m, n, h) \in Brothers_{ijh}$:

IF($|\mu_{ijh} - \mu_{mnh}| < min\_cont$) AND ($s_{mnh} > s_{ijh}$)

THEN Call *Is Brother*, (section A.18);

IF (i, j, h) is a brother of (m, n, h)

THEN $p_{ijh} = 0$; $q_{ijh} = 1$;

ELSE $p_{ijh} = 1$; $q_{ijh} = 0$;

A.14 Check Candidate

Function : Determining whether or not the cell is a good candidate.

IF $p_{ijh} = 0$

THEN

$\forall\ (m, n, h) \in Support_{ijh}$

IF ($p_{mnh} = 0$)

THEN

The cell is not a good candidate

ELSE

The cell is a good candidate

ELSE

The cell is not a good candidate;

A.15 Choose To Link

Function : Determining what cells should be linked together.

FOR every cell $(i, j, h) \in$ the current level:

IF ($p_{ijh} = 0$)

THEN FOR every cell $(m, n, h) \in Brothers_{ijh}$:

IF ($p_{mnh} = 2$) AND ($q_{mnh} \neq 2$)

THEN

$minDif_{ijh} = |\mu_{mnh} - \mu_{ijh}|$;

[$minDst_{ijh} = dist(mnh, ijh)$;]

Choose (m, n, h);

ELSE

IF ($p_{mnh} = 1$) AND ($q_{mnh} \neq 2$) AND

($|\mu_{mnh} - \mu_{ijh}| < minDif_{ijh}$)

[$dist(mnh, ijh) < minDst_{ijh}$)]

THEN

$minDif_{ijh} = |\mu_{mnh} - \mu_{ijh}|$;

[$minDst_{ijh} = dist(mnh, ijh)$;]

Choose(m, n, h);

ELSE

(i, j, h) will survive;

FOR every cell $(m, n, h) \in Brothers_{ijh}$:

IF $|\mu_{mnh} - \mu_{ijh}| < minDif_{ijh}$

[$dist(mnh, ijh) < minDst_{ijh}$]

THEN

$minDif_{ijh} = |\mu_{mnh} - \mu_{ijh}|$;

[$minDst_{ijh} = dist(mnh, ijh)$;]

A.16 Calculate Minimum contrast

Function : Using the size of the receptive field and the value of α to determine the contrast measure for every cell.

FOR every cell $(i, j, h) \in$ the current level:

IF($s_{ijh} > min\_size$)

THEN

$contrast_{ijh} = min\_cont;$

[$conDist_{ijh} = min\_dist;$]

ELSE

$contrast_{ijh} = min\_cont \times e^{\alpha(min\_size - s_{ijh})};$

[$contDist_{ijh} = min\_dist \times e^{\alpha(min\_size - s_{ijh})};$]

A.17 Link Cell

Function : Detailing the steps required to link a nonsurviving cell to its nearest surviving cell in its support.

To link (i, j, h) to (m, n, h):

IF (m, n, h) was not built before in the upper level:

IF ($minDif_{ijh} > min\_cont$)

[($minDst_{ijh} > min\_dist$)] AND ($p_{ijh} = 0$)

THEN (i, j, h) is a noise effect

FOR every cell $(I, J, h) \in Brothers_{ijh}$:

Check the difference $\mu_{ijh} - \mu_{IJh}$;

[Check the distance (ijh, IJh);]

Choose (I, J, h) which makes the least difference [distance;]

$parent_{ijh} = (I, J, h + 1);$

IF $(\mu_{ijh} - \mu_{IJh} > min\_cont)$

$[dist(ijh, IJh) > min\_dist)]$ AND $(s_{ijh} > min\_size)$

THEN $parent_{ijh} = (i, j, h+1)$;

IF $(parent_{ijh} \neq (i, j, h+1))$ OR $(s_{ijh} < min\_size)$

IF $((I, J, h+1)$ exists) AND $((I, J, h+1) \notin Receptive_{ijh})$

AND $(i \neq I)$ OR $(j \neq J)$

THEN

$parent_{ijh} = parent_{IJh}$;

IF$((q_{parent_{ijh}} \neq 3)$ AND $(q_{IJh} \neq 3)$ AND $(q_{ijh} \neq 3))$

OR$((|\mu_{IJh} - \mu_{ijh}| < min_cont)$ AND $(s_{ijh} \geq min\_size)$

$[(dist(IJh, ijh) < min\_dist))]$

THEN

add $Receptive_{ijh} \longrightarrow Receptive_{parent_{ijh}}$;

$q_{ijh} = 3$;

ELSE $parent_{ijh} = (i, j, h+1)$;

ELSE

IF NOT($(parent_{IJh}$ exists) AND $((i, j, h) \notin Receptive_{parent_{IJh}}))$

THEN search for a surviving cell $\in Brothers_{IJh}$ s.t.

$|\mu_{thiscell} - \mu_{IJh}| \leq min\_cont$

$[dist(thiscell, IJh) \leq min\_dist]$

IF found

THEN

IF $(parent_{ijh}$doesnot exist) AND $(q_{ijh} \neq 3)$

THEN

$Recptive_{ijh} \longrightarrow Receptive_{IJh}$;

IF$(q_{IJh} = 3)$

THEN $parent_{ijh} = (i, j, h+1)$;

ELSE $parent_{ijh} = (I, J, h);\ q_{ijh} = 3;$

ELSE

IF $(q_{IJh} \neq 2)$

THEN $parent_{ijh} = (i, j, h + 1);\ q_{ijh} = 2;$

ELSE

IF$(q_{ijh} \neq 3)$

THEN $parent_{ijh} = (i, j, h + 1);$

ELSE

$Recptive_{ijh} \longrightarrow Receptive_{IJh};$

$parent_{ijh} = (I, J, h);$

IF $(q_{ijh} = 0)$

THEN $parent_{ijh} = (m, n, h);$

ELSE (m, n, h) was not built before in the upper level, should be created:

IF$(minDif_{ijh} < contrast_{ijh})$ $[(minDst_{ijh} < contDist_{ijh})]$

IF$(minDif_{ijh} > min_cont)$ AND $(p_{ijh} = 0)$

$[(minDst_{ijh} > min\_dist)]$

AND $(i \neq m)$ AND $(j \neq n)$

THEN (i, j, h) is a noise effect

FOR every cell $(I, J, h) \in Brothers_{ijh}$:

Check the difference $\mu_{ijh} - \mu_{IJh};$

[Check the distance $dist(ijh, IJh)$;]

Choose (I, J, h) which makes the least difference [distance];

$parent_{ijh} = (I, J, h + 1);$

IF $(\mu_{ijh} - \mu_{IJh} > min\_cont)$ $[dist(ijh, IJh) > min\_dist]$

THEN $parent_{ijh} = (i, j, h + 1);\ q_{ijh} = 2;$

IF $(parent_{ijh} \neq (i, j, h + 1))$

IF ($parent_{IJh}$ exists) AND ($q_{ijh} \neq 3$)

AND ($parent_{IJh} \notin Receptive_{ijh}$)

THEN

$parent_{ijh} = parent_{IJh}$;

$Receptive_{ijh} \longrightarrow Receptive_{parent_{ijh}}$;

$q_{ijh} = 2$;

ELSE

IF($q_{ijh} \neq 3$)

THEN

$Receptive_{ijh} \longrightarrow Receptive_{IJh}$;

$q_{ijh} = 2$;

$q_{IJh} = 3$;

$parent_{ijh} = (I, j, h + 1)$;

ELSE (i, j, h) is not a noise effect

$parent_{ijh} = (i, j, h + 1)$;

$q_{ijh} = 3$;

ELSE This ia a root..

IF($s_{ijh} \geq min\_size$)

THEN

$parent_{ijh} = (i, j, h + 1)$;

$q_{ijh} = 3$;

$p_{parent_{ijh}} = 2$;

$parent_{ijh} = (i, j, h + 1)$;

ELSE

FOR every cell $(I, J, h) \in Brothers_{ijh}$:

Check the difference $\mu_{ijh} - \mu_{IJh}$;

[Check the distance $dist(ijh, IJh)$;]

Choose (I, J, h) which makes the least difference [distance];

IF($parent_{IJh}$ exists)

THEN

$parent_{ijh} = parent_{IJh}$;

$Receptive_{parent_{ijh}} \longrightarrow Receptive_{ijh}$;

$q_{ijh} = 2$;

ELSE i.e. there is no parent

IF ($q_{ijh} \neq 3$)

THEN

$Receptive_{IJh} \longrightarrow Receptive_{ijh}$;

$parent_{ijh} = (I, J, h)$, build it;

$q_{ijh} = 2$;

ELSE

IF ($p_{ijh} = 2$)

THEN $parent_{ijh} = (i, j, h + 1)$;

$Receptive_{ijh} \longrightarrow Receptive_{parent_{ijh}}$;

A.18 Is Brother

Function : Checking two cells for being neighbours.

For cells (i, j, h) and (m, n, h):

FOR every cell $(I, J, 0) \in Receptive_{ijh}$:

FOR every cell $(M, N, 0) \in Receptive_{mnh}$:

IF ($|I - M| \leq 1$) OR ($|J - N| \leq 1$)

THEN (i, j, h) is a brother of (m, n, h);

A.19 Compare Cell

Function : Calculating the interest operator F using a 3×3 comparison window.

For a cell (i, j, h):

FOR $(I : i - 1 \Rightarrow i + 1)$

FOR $(J : j - 1 \Rightarrow j + 1)$

$F_{ijh} = \{f_1(i, j) - f_2(i + I, j + J)\}^2$;

Bibliography

[Bar89] Stephen Barnard. Stochastic stereo matching over scale. *International Journal for Computer Vision*, 3:17–32, 1989.

[BB81] H. Baker and T. O. Binford. Depth from edge and intensity based stereo. *Proceedings 7th Joint Conferenceon Artificial Intelligence, Vancouver, Canada*, pages 631–636, August 1981.

[BB82] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, N.J., 1982.

[BT80] S. T. Barnard and W. B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):333–340, July 1980.

[CC90] C. H. Chou and Y.C. Chen. Moment-preserving pattern matching. *Patteren Recognition*, 23(5):461–474, 1990.

[CF94] Virginio Cantoni and Marco Ferretti. *Pyramidal Architectural for Computer Vision.* Plenum Publishing corporation, 1994.

[CH84] J. K. Cheng and T. S. Huang. Image registration by matching relational structures. *Patteren Recognition*, 17(1):149–159, 1984.

[DA89] R. U. Dhond and J. K. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, December 1989.

[DP86] M. Drumheller and T. Poggio. On parallel stereo. In *International Conference on Robotics and Automation*, pages 1439–1448, 1986.

[Dye87] C.R. Dyer. Multiscale image understanding. *Parallel Computer Vision*, pages 171–213, edited by Leonard Uhr, Academic Press, Inc. 1987.

[Fau96] Olivier Faugeras. *Three-Dimentional Computer Vision, A Geometric Viewpoint.* The MIT Press, Cambridge, Massachusetts, 1996.

[Fua93] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Application*, 6(1), Winter 1993.

[Gen80] D. B. Gennery. Modelling the environment of an exploring vehicle by means of stereo vision. *PhD Thesis, Stanford University*, 1980.

[GGB84] A. Goshtasby, S. H. Gage, and J. F. Bartholic. A two-stage cross correlation approach to template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):374–378, May 1984.

[Gri81] W. E. L. Grimson. *From Images to Surfaces.* The MIT Press, Cambridge, Massachusetts, 1981.

[HA94] Xiaoping Hu and Narendra Ahuja. Matching point features with ordered geometric, rigidity and disparity constrains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1041–1049, October 1994.

[Han89] M. J. Hannah. A system for digital stereo image matching. *Photogrammetric Engineering and Remote Sensing*, 55(12):1765–1770, December 1989.

[Hen87] K. Henriksen. Line based stereo matching. Technical Report MS-CIS-87-52, University of Pennsylvania, Computer Science Department, Philadelphia, 1987.

[HPB92] F. Heitz, P. Perez, and P. Bouthemy. Multiscale minimization of global energy functions in some visual recovery problems. *Image Understanding*, 59(1):125–134, 1992.

[HS89] R. Horaud and Th. Skordas. Stereo correspondance through feature grouping and maximal cliques. *IEEE Transactions on Patteren Analysis and Machine Intelligence*, 11(11):1168–1180, 1989.

[HT84] N.P. Hartman and S. Tanimoto. A hexagonal pyramid data structure for image processing. *IEEE Trans. Syst., Man, Cybern*, 14:247–256, 1984.

[HW70] D. H. Hubal and T. N. Weisel. Cells sensitive to binocular depth in area 18 of the macaque monkey cortex. *Nature*, (225):41–42, 1970.

[HW73] D. H. Hubal and T. N. Weisel. A reexamination of stereoscopic mechanisms in the cat. *J. Physiol.*, (232):29–30, 1973.

[JM91] J.M. Jolion and A. Montavert. The adaptive pyramid: A framework for 2d image analysis. *CVGIP: Image Understanding*, 55(3):339–348, 1991.

[JR94] J.M. Jolion and A. Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, 1994.

[Kel71] M.D. Kelly. Edge detection in pictures by computer using planning. *Machine Intelligence*, 6:397–409, 1971.

[KMM77] R. E. Kelly, P. R. H. McConnell, and S. J. Mildenberger. The gestalt photomapper. *Photogrammetric Engineering and remote sensing*, 43:1407–1417, 1977.

[Kro85] W.G. Kropatsch. A pyramid that grows by power of 2. *Pattern Recognition Letters*, 3(9):315–322, 1985.

[Kro87] E. Krotkov. Exploratory visual sensing for determining spatial layout with an agile stereo camera system. *PhD Dissertation MS-CIS-87-29, Computer Science Department, University of Pennsylvannia, Philadelphia*, 1987.

[LL86] P. Limozin-Long. Vision stéréoscopique appliquée à la robotique. *Thèse, Université de Nice*, Octobre 1986.

[Lus87] F. Lustman. Vision stéréoscopique et perception du mouvement en vision artificielle. *PhD thesis, Paris-Orsay*, 1987.

[Min63] M Minsky. Steps toward artificial intelligence. *Computers and Thoughts*, pages 406–450, 1963.

[Mor80] H. P. Moravec. Obstacle avoidance and navigation in the real world by a sensing robot rover. *PhD Thesis, Stanford Artificial*

Intelligence Laboratory, also as Stanford Artificial Memo 340, 1980.

[MP76] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[MW87] H. Maitre and Yifeng Wu. Improving dynamic programming to solve image registration. *Patteren Recognition*, 20(4):443–462, 1987.

[NA91] translated by Peter Sander Nicholas Ayache. *Artificial Vision for Mobile Robots, Stereo Vision and Multisensory Perception.* The MIT Press, Cambridge, Massachusetts, 1991.

[Nev76] R. Nevatia. Depth measurement by motion stereo. *Computer Vision, Graphics and Image Processing*, 5:203–214, 1976.

[Nis83] K. Nishihara. Prism: a practical real-time imaging stereo matcher. *Proceedings 3rd International Conference on Robot Vision and Sensory Controls*, 1983.

[NP84] K. Nishihara and T. Poggio. Stereo vision for robotics. *Robotics Research, The First International Symposium*, pages 489–505, 1984.

[OK85] Yuichi Ohta and Takeo Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, March 1985.

[OTI87] Yuichi Ohta, Kenji Takano, and Katsuo Ikeda. A highspeed stereo matching system based on dynamic programming. *IEEE Conference*, pages 335–342, 1987.

[PFH86] S. Peleg, O. Federbusch, and R.A. Hummel. Custom-made pyramids. *Parallel Computer Vision*, pages 125–146, 1986.

[PH87] M. Pietikainen and D. Harwood. Progress in trinocular stereo. In *NATO Advanced Workshops on Real-time Object and Environment Measurement and Classification*, pages 1439–1448, 1987.

[PPD98] C.L . Pagliari, M. M. Perez, and T. J. Dennis. Reconstruction of intermediate views from stereoscopic images using a rational filter. *IEEE International Conference on Image Processing, ICIP*, October 4-7 1998.

[Rad84] B. Radig. Image sequence analysis using relational structures. *Patteren Recognition*, 17(1):161–167, 1984.

[Rho84] K. Ben Rhouma. Analyse de scènes par une méthode k2d en robotique avancée. *Thèse de Docteur Ingénieur, Université de Paris-Sud, Orsay*, 1984.

[RT71] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Trans. Computers*, 20:562–569, 1971.

[SH81] L. G. Shapiro and R. M. Haralick. Structural description and inexact matching. *IEEE Transactions on Patteren Analysis and Machine Intelligence*, 3:504–519, September 1981.

[SH87a] T. Skordas and R. Horaud. Groupement robuste de caractéristiques image. Technical Report RR676-I- 63, LI-

FIA, IMAG, Laboratoire d'informatique Fondamentale et d'Intelligence Artificielle, Septembre 1987.

[SH87b] T. Skordas and R. Horaud. Stereo correspondence through feature grouping and maximal cliques. Technical Report RR677-I-64, LIFIA, IMAG, Laboratoire d'informatique Fondamentale et d'Intelligence Artificielle, September 1987.

[SK98] Yoshihisa Shinagawa and Tosiyasu Kunii. Unconstrained automatic image matching using multiresolutional critical-point filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):994–1010, September 1998.

[SV86] A. R. Saint-Vincent. Perception et modélisation de l'environnement d'un robot mobile: une approach par stéréovision. *Thèse, Université Paul Sabatier, Toulouse*, 1986.

[TGS96] D. Tzovaras, N. Gammalidis, and M. G. Strintzis. Disparity field and depth map coding for multiview image sequence compression. *IEEE International Conference on Image Processing, ICIP*, pages 887–890, September 16-19 1996.

[Tos87] G. Toscani. Système de calibration optique et perception du mouvement en vision artificielle. *PhD Thesis, Paris-Orsay*, 1987.

[TP75] S. Tanimoto and T. Pavlids. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4:104–119, 1975.

[Tsa83] R. Tsai. Multiframe image point matching and 3d surface reconstruction. *Report 9249, IBM US Research Centerss*, May 1983.

[Tsa86] R. Tsai. Multiframe image point matching and 3d surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):159–174, 1986.

[Ull79] S. Ullman. *The Interpretation of Visual Motion.* The MIT Press, Cambridge, Massachusetts, 1979.

[WAH88] Juyang Weng, Narendra Ahuja, and Thomas Huang. Two-view matching. *ICCV'88*, pages 64–73, 1988.

[WAH92] J. Weng, N. Ahuja, and T. S. Huang. Matching two perspective views. *IEEE Transactions on Patteren Analysis and Machine Intelligence*, 14(8):806–825, August 1992.

[WBH98] Guo-Qing Wei, Wilfried Brauer, and Gerd Hirzinger. Intensity- and gradient-based stereo matching using hierarchical gaussian basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1143–1160, November 1998.

[WM88] Terry Weymouth and Saied Moezzi. Wide base-line dynamic stereo: Approximation and refinement. *IEEE Conference*, pages 183–188, 1988.

[WS88] R. Wilson and M. Spann. *Image Segmentation and Uncertainty.* Wiley, New York, 1988.

[XKT] Gang Xu, Hideki Kondo, and Saburo Tsuji. A region-based stereo algorithm. *Vision and Robotics*, pages 1661–1666.

Index