Prime Component Decomposition of Images and its Applications in an Image Understanding System

Maciej Macieszczak

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

February 1997

0-612-25929-3

ABSTRACT

Prime Component Decomposition of Images and its Applications in an Image Understanding System

Maciej Macieszczak, Ph.D.
Concordia University, 1997

A reliable and flexible model of a low-level processing stage is one of the most crucial requirements in the development of an image understanding system (IUS). In this thesis, a model for the low-level processing stage based on a new scheme of prime component decomposition is proposed. This model is then used to develop a knowledge-based image understanding system that is capable of solving many image processing problems without employing complex algorithms.

A scheme for the prime component decomposition that utilizes the maximum size geometrical polygons is devised. It is shown that the optimal decomposition element in the continuous metric space has a circular shape. The decomposition operator is also optimized in the discrete metric space to deal with the actual implementation of the prime component decomposition operator, yielding square decomposition elements. The derived decomposition operator is used to extract shape elements of the objects contained in input scenes and to produce their intermediate object descriptions. In the proposed approach of shape extraction, the prime component decomposition technique is used to partition the object's interior, while a modified Sobel operator is used to detect the object's edges. The typical errors of a shape extraction process such as noise sensitivity, description errors of diagonal objects and the description errors caused by a small sampling frequency are reduced using a shape equalization approach that is based on Fourier descriptors and nonlinear interpolation.

In the development of an image understanding system, a hierarchical approach of constructing the intermediate object representation is used to represent the knowledge within the system. The knowledge base of the IUS is developed as a relational multidimensional tree structure that dynamically changes the relational links among its elements. The dynamical process of creating and transforming the knowledge base is controlled by a feedback with the low-level processing stage that reduces the memory requirements of the IUS. The traditional data type definitions are extended to include the base and derived data types. These extensions effectively represent and process the time-varying knowledge of the system and increase its overall efficiency. The high-level processing stage of the IUS is implemented based on the black-board architecture with a specialized control mechanism - the agenda-based control. This control mechanism reduces the number of computational steps within the high-level processing stage by employing a selective focusing mechanism.

The functional behaviour of the proposed prime component decomposition scheme and the model of the image understanding system is experimented with several application examples including the isolation and identification of stationary and time-varying objects.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Dr. M. O. Ahmad, for providing support and guidance that made this work possible. I am grateful for his patience and critical review of my thesis, and for his inspiration throughout this work. I feel privileged for having the opportunity to work with him. I would also like to thank the members of the examination committee for their most helpful comments and suggestions.

Chcialbym takze podziekowac wszystkim moim najblizszym przyjaciolom za ich pomoc i wyrozumialosc. W szczegolnosci skladam serdeczne podziekowania dla wszyskich bliskich i drogich mojemu sercu.

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\odot$	Convolution operator in a continuous metric space
$\circledast$	Convolution operator in a discrete metric space
$\ominus$	Minkowski subtraction operator
μ_{pq}	Central moment of order $(p+q)$
A_0, a_n, b_n	Fourier coefficients
A_{sc}	Association between two prime components
$\mathcal{C}$	Partition of a set
CH	Chain of skeleton points
D	Data type
$\mathcal{DEC}[\cdot]$	Decomposition operator
$\mathcal{D}_o$	Relational description for an object
$d(\cdot,\cdot)$	Distance (metric) function
$E(m,n)$	Edge map
$\mathcal{F}(\xi_1,\xi_2)$	Fourier Transformation for a function $f(x,y)$
F_{in}	Input transformation
fd	Fourier descriptor
G, G'	Gray level
$h()$	Impulse response of a system
I_{im}	Inner image representation
KS	Knowledge source
l	Shape element
M	Edge detection operator
m_{pq}	Moment of order $(p+q)$
$\mathcal{OS}$	Shape descriptor
$O_n(x,y)$	Object map
P_{sc}	Prime component
$\mathcal{POW}(X)$	Number of elements in set X
$\mathcal{R}$	Set of real numbers
$\mathbf{R}_L$	Relation, set of rules between prime components
$\mathcal{ROT}[\cdot]$	Rotation operator
$\mathcal{R}^n$	Region in an inner image representation

$\mathcal{R}_t^n$	Translation of the region $\mathcal{R}^n$
$SK(X)$	Skeleton of the object X
SP	Skeleton point
S_r	Rotationally and Spatially invariant subset of a world scene
$skf(z)$	Skeleton function
$s_n(x,y)$	Skeleton map
$sup(\mathcal{SQR})$	Largest square-shaped prime component in the object
$\mathcal{U}_\mathcal{A}$	Set of all associations (links) among prime components
$\mathcal{U}_\mathcal{I}$	Set of all possible inner image representations
$\mathcal{U}_\mathcal{P}$	Set of all prime components
$\mathcal{U}_\mathcal{W}$	Set of all possible world scenes
u_a	Set of associations between prime components
u_p	Set of prime components
V_m^L	Elementary set of prime components generated by a rule $\mathbf{R_L}$
$\mathcal{V}_{sc}^L$	Set of all elementary groupings V_m^L of prime components
W_{sc}	World scene
X, Y, X', Y'	Spatial coordinates
Z_{sc}	Result of the decomposition of an inner image representation

LIST OF ACRONYMS

AI	Artificial Intelligence
BB	Black Board
DFT	Discrete Fourier Transformation
DSP	Digital Signal Processing
FFT	Fact Fourier Transformation
IUS	Image Understanding System
VCR	Video Cassette Recorder
VLSI	Very Large Scale Integration
1-D	One Dimensional
2-D	Two Dimensional

Chapter 1

INTRODUCTION

An image understanding system (IUS) is essentially a knowledge transformation system utilizing many information processing paradigms. Since majority of the developments for image understanding systems have been achieved in the field of image processing and artificial intelligence, an IUS is also known as the knowledge-based DSP architecture, computer vision system or simply a knowledge-based system. The problems encountered in an IUS are extremely difficult and involve many complex processes and representation dilemmas. Some of them are related to the description of the knowledge in the physical world, while others are the ones dealing with the knowledge transformation schemes. These problems can be more explicitly stated as two major tasks but related to one another: (a) the knowledge representation, and (b) the knowledge transformation.

1.1 Knowledge Representation

Solving the problem of a knowledge representation is one of the most difficult tasks faced by the researches wanting to build a reliable IUS. Apart from the strict theoretical considerations of the knowledge representation, one has to deal with the practical issues concerning the implementation of a given theoretical model. Therefore, some of the solutions that promote logically complex and computationally expensive algorithms might be rejected. On the other hand, the "hardware" oriented approaches with simple models might be too simple to effectively represent the complex relations between the objects in the physical world.

There have been several attempts to develop a formal structure for the representation of knowledge for various IUS. Among the most successful ones are the production systems [61], the procedural systems [43], and the state-space representation [67], [99]. Each of these systems has been used either in its original form or with some variations as required by the application problems. In computer vision, the knowledge representation may require to have the following features.

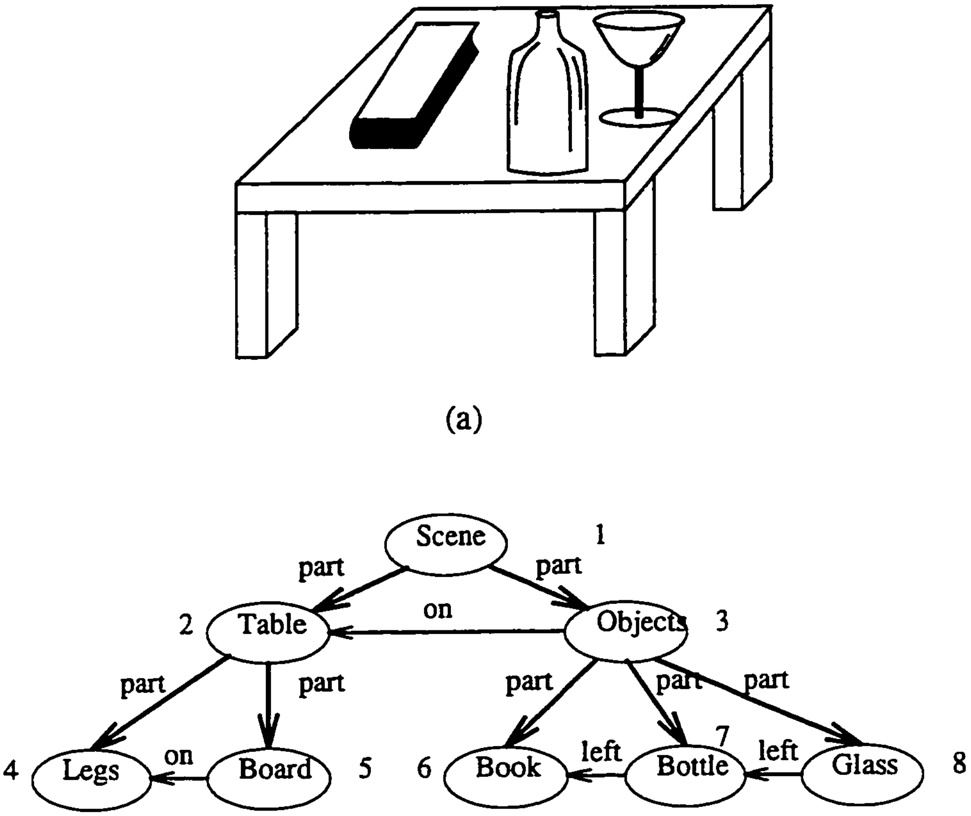
(i) The flexibility and consistency of a knowledge representation scheme to which one can add or delete the needed pieces of knowledge in a systematic fashion.

(ii) A facility for representing factual information (predicates, functions or relations) and procedures which could extract information on request from the input data

(iii) An inference scheme, which could generate new reasoning strategies.

The development of an efficient model for an IUS would require the knowledge representation model to have these features.

Almost all researches agree that an effective scheme for the knowledge representation should posses a mechanism to represent two major groups of relations between the objects of the physical world: the spatial relations and the time relations. Because of the complexity of the problem, most of the efforts have concentrated on finding an optimal knowledge representation scheme of the relations between the objects only in restricted domains. As an example, a typical knowledge representation model may describe only the spatial relations between the objects of the physical world. This approach seems to be easier than the one that models the spatial and time relations between the objects at the same time. The model of the knowledge representation together with the information contained therein is usually called the knowledge base or the data base. In a simple case, a knowledge representation scheme may be based on the approach that couples the objects from the physical world to the concepts and relations among the objects to the rule bases. Consequently, the information in this knowledge representation scheme can be expressed as relations among the concepts, and then it can be listed as rule bases. This representation is known as a semantic (syntactic) representation of the knowledge and it is usually modeled by graphs, networks or frames. Concepts are defined as the sets or fuzzy sets of objects, observations, or actions. For example, in the knowledge representation model described by a semantic network, the nodes of the network represent specific concepts and the links among the nodes represent specific relations among the concepts. Figure 1.2(a) shows a simple example of a world-scene, and Figure 1.1(b) shows its semantic network representation.



(a)

(b)

Figure 1.1: Knowledge representation using a semantic network.(a) An example of a simple scene. (b) Its semantic network representation.

The semantic network, that represents the knowledge is usually expressed in a high-level formal language. The coupling between the high-level formal language and the physical world is not unique in the sense that one formal language can represent many physical worlds and one physical world can be represented by many formal languages. Therefore, one of the biggest problems in this type of the description is to define and to understand the connections between the syntax and semantics of the formal language to the physical world.

Another problem related to the modeling of a knowledge representation scheme concerns constructing the inference rules and reasoning strategies within the model of the knowledge representation. Since a formal language can exist and define itself within its syntax, it is very difficult to construct and develop appropriate heuristics and rules within the logical system of the language itself. This is particularly difficult when one wants to develop a self learning system with ability to act in an adaptive and intelligent manner similar to human behavior. Trying to overcome these difficulties by the construction of artificial built-in heuristics and inference rules can be very misleading. In this case, the complexity and dynamical changes of the physical world very often result in the generation of syntactically correct but false entailed relations to the physical world (false rules). Therefore, the built-in inference rules and the reasoning mechanisms should be chosen very carefully and a proper knowledge transformation scheme, including learning strategies, should determine the final form of the heuristics and rules in the model.

1.2 Knowledge Transformation

The development of the knowledge transformation model is another very important task belonging to an IUS. Formally, the knowledge transformation is a process that generates (produces) new forms of the output knowledge representation (knowledge description) from the input knowledge representation. The transformation process in an IUS is based on applying rules (productions), described by the semantic of the formal language and its syntax (grammar) to the input knowledge representation. Built into the knowledge base of an IUS, a set of productions and inference rules constitutes a base from which the next production sets are generated. Conceptually, the generation process is similar to the process which translates the formal symbolic language. Therefore, the knowledge transformation model is sometimes known as the knowledge compilation or knowledge translation model.

Similar to the knowledge representation, the knowledge transformation model should not only be based on a some theoretical foundation, but should also take into consideration its practical implementation. It is specially important, since the computational complexity of the compilation process increases rapidly with the increase of the complexity of the knowledge representation. There are many strategies that have been developed to decrease the amount of information generated by the various stages of the knowledge transformation module. Some of them utilize built in adaptive rules [82] while others are based on fixed search algorithms (heuristics) [17]. Ideally, the generation process should make use of the contextual information contained in the knowledge base which is similar to the human reasoning strategies. This idea that is very easily accomplished by humans is not easily convertible into the knowledge transformation model. The meaning of the context itself does not have a formal description and many interpretations are possible [97].

There have been a few attempts to propose models of the knowledge transformation based on the contextual description. Among them, the two most important ones are the conception of frame model introduced by Minsky [57] and the conception of meaning condition model introduced by Winograd [99]. In the approach presented by Minsky, the knowledge transformation is performed by algorithms described by relational data-structures known as frames. The algorithms can perform two main kinds of hierarchical transformation: *bottom-up* and *top-down*. In the bottom-up transformation, features from the image are extracted and grouped in some way, with no knowledge of the structure of the object or the scene. A top-down transformation, on the other hand, starts with the hypothesis that the image contains a particular object or it can be categorized as a particular type of scene. This leads to entering deeper into the set of hypotheses about the parts of the object present in the scene. The model introduced by Winograd is based on the bottom-up approach of the knowledge transformation and the concepts of meaning conditions are similar to the concepts of frames. In his model, the knowledge is represented and then transformed by data structures whose terminal nodes consist of *slots* which are attributes, and *fillers* which are the values of the attributes. In fact, a filler may be the name of a procedure that must be executed to produce a new value. The data structures composed of the slots and fillers are then connected similar to the objects in the link-list.

In order to formalize the contextual description, one can assume that the "meaning", that is, the entailed relations to the world of particular knowledge agents, can have many dimensions. Consequently, a particular dimension can be chosen depending on the context of the input information. This mechanism allows to switch the context in which the input information is analyzed. However, there are many problems associated with this type of context switching. One of the most important ones is the classification mechanism for mapping the context of the scene into one or more dimensions. The other problems concern the practical implementation of the whole classification process i.e. its computational complexity and associated with it the huge amount of the data to be stored and transferred. Even the most simple task performed by humans effortlessly require extremely complicated and computationally very expensive algorithms. The typical and the most difficult task of the knowledge transformation in an IUS is to identify the objects contained in a scene and to produce the description of their spatial-time interrelations such as shape analysis, stereo vision and motion estimation. Identification is generally done on the basis of inner knowledge (knowledge base of the system) and on the knowledge extracted from the image or image sequences (information extraction).

1.3 Image Understanding Systems

In order to deal with the complicated and involved issues as discussed in Sections 1.1 and 1.2, many models of image understanding architectures have been proposed [7]. Most of these models are based on the approaches which divide the knowledge compilation process into smaller sequential processes and group them into levels of hierarchical transformations. The hierarchical transformation is defined by a predefined hierarchy of ordering of the procedures that performs the required analysis task. Typically, the models based on the hierarchical approaches imply at least two levels of processing which are usually called the low-level processing (or early stage vision) and the high-level processing. In the most common scenario, a main procedure calls on a set of preprocessing routines from the low-level processing stage of the IUS, that convert the original image to a form most suitable for extracting the primitive features. This stage of transformation involves the classical image processing techniques such as contrast enhancement or edge detection. Next, a feature extraction routine locates the features of interest and constructs a symbolic description called the intermediate object description of these features and their attributes and interrelations.

1.3.1 Intermediate Object Description

One of the major drawbacks that is present in most of the existing image understanding systems is that they lack an effective mechanism which reduces the redundant input information and produces an efficient intermediate object description suitable for further processing. This mechanism should be built into the low-level processing stage and it should utilize in the most optimal way the available resources of the IUS. Typically, an intermediate object description is a very complex data structure whose elements represent the time-space relations between the objects contained in the input world-scene. Generally, there is little use of the past input information in the existing systems. Moreover, the adaptability of the low-level processing stage to the changing environment of the existing system is very limited. This makes high-level processing difficult, slow and complicated. In addition, the organization of the commonly used data structures [92] for the intermediate object description and their syntax makes later processing even more complicated. The formal description language of data structures, syntax and its semantic relations are essential elements in order to develop an affective and flexible intermediate object description model.

1.3.2 Decomposition Operator in a Low-Level Processing Stage

The key idea in low-level processing is to find a method of the description of the image and image sequences which minimizes the computational complexity of transformations and simplifies the required algorithms for high-level processing. Usually, this goal is achieved by performing several stages of transformations such as preprocessing, feature extraction and mapping the extracted features into an intermediate object description. The feature extraction is one of the most important and difficult tasks to model [35]. In an IUS the feature extraction mechanism should effectively represent three groups of relations present in world-scenes: (a) specialized measurement relations, (b) visual relations, and (c) pattern synthesizing relations. The first group of relations describes a particular set of outstanding measurements derived from the initial pattern measurements - outstanding in the sense that they more strongly characterize and discriminate the classes than the raw measurements obtained by the receptors. The second group of relations represents visual (topological and geometrical) characteristics of patterns; for example, closed curves, forks, corners, straight segments, bays, etc. Finally, the third group of relations describe subpatterns whose superposition results in a given pattern. The feature extraction algorithm that extracts this group of relations is equivalent to determining a minimal set of features which are sufficient to synthesize all the patterns found in world-scenes. Because of the relational constrains, the typical feature extraction algorithms are based on the polygonal decomposition approach [32] and they utilize sophisticated hierarchical data structures.

In order to construct an efficient feature extraction algorithm in an IUS, many models of polygonal decomposition have been proposed [14], [32], [93]. Some of them are based on the interior decomposition of the objects contained in the scene while others concentrate on describing the object's boundaries [95]. The group of polygonal decompositions based on the description of the object's interior includes quad-tree decomposition, symmetric axis transformation and 2-D C-string representations [39], [74]. The transformations that describe the object's boundaries are Fourier descriptors, B-spline decomposition, moment invariant transformations and others [2], [23], [53]. Both groups of polygonal decomposition can utilize many different data structures. Among the most popular ones are linear lists, hierarchical structures, graph structures, complex recursive structures [85].

There are many problems in the existing decomposition approaches used in the low-level stage of an IUS. First, the existing decomposition models [34] do not combine the description of the object's interior with the description of the object's boundaries. This fusion is extremely important and it significantly simplifies the context-dependent transformations in the high-level processing stage of an IUS. Second, most of the existing polygonal decomposition models [10] are computationally very expensive and difficult to implement, including parallel implementation of algorithms and optimal utilization of system's resources. Third, the existing decomposition models are too restrictive to efficiently decompose typical time-varying events. Time-varying situations occur when an IUS is involved in the analysis of image sequences containing objects that change their form or shape with time. Fourth, most of the proposed decomposition models are very rigid in the sense that they can describe only very simple input data such as regular and symmetrical polygons.

1.4 Scope and Organization of the Thesis

The objective of this thesis is to propose an efficient model for a knowledge-based image understanding system that can reduce the weaknesses of the existing models. The model of the low-level processing stage is based on developing a new prime component decomposition scheme that can produce an efficient intermediate representation of the objects within the input world-scenes and facilitate an effective development of the models of the knowledge base and the high-level processing stage of the IUS.

The thesis is organized as follows. Chapter 2 presents a general overview of an IUS and gives the necessary background material. The model of image formation in an IUS, transformations of the input image and their properties are discussed.

Since in majority of the cases, the essential relations are not those between elementary objects (i.e. pixels) but those between the groups of elementary objects, one would like to deal with a scene decomposed into groups of elementary objects and relations among them. In Chapter 3, a decomposition operator based on an approach that reduces the redundant information by an effective extraction of the repeated elements of the scene and by grouping together the sets of pixels having similar relational properties in the space and time domains is developed. In order to increase the implementation efficiency, the decomposition model is optimized to yield polygonal prime components keeping in mind many independent parameters related to the properties of the real world-scenes and to the processing environment.

In Chapter 4, the scheme developed for polygonal decomposition is then used to model the shape extraction process and to obtain a relational description of the patterns and their clusters in world-scenes. The shape extraction process performs polygonal decomposition of the interior of the objects in the scene and detects their boundaries. The interior decomposition is done using squared polygons (prime components). The boundaries of the objects (edges) are determined using a modified Sobel operator. In order to decrease the computational complexity of the shape extraction process and to eliminate the geometrical distortions of diagonal, overlapped and neighborhood objects, the shape extraction process also includes a shape equalization procedure. This procedure is based on a smoothing the parametrically represented boundary of the extracted object by a 1-D adaptive filter in the Fourier descriptor domain. After the decomposition of the objects' interior and detection of their boundaries, a discrete set of points representing the edges and coordinates of prime components are fused (clustered) together into a full discrete skeleton of an object. The final stage of the extraction is to produce an intermediate object description. This process is carried out by matching the B-spline curves to the whole discrete skeleton of the object. The resulting intermediate object description is composed of a set of extracted shape elements (set of B-spline curves) and interrelations among them.

Chapter 5 presents models for the low-level processing stage, high-level processing stage and the knowledge base of the IUS. These models are optimized to reduce the complexity of the intermediate object description, and the amount of data for knowledge representation. The reduction in the complexity of the intermediate object description is achieved by a feedback mechanism with the knowledge base that utilizes the contextual description and statistical properties of the objects in the world-scene now represented by the polygonal prime components and their clusters. This feedback mechanism with the knowledge base acts as a "filter" which further eliminates the redundant information from the intermediate object description and it helps to perform top-down reasoning process in the IUS. The knowledge representation in the IUS is based on the model of a multidimensional stochastic tree graph structure. With this type of data representation, one particular context of the intermediate object description corresponds to one dimension of the stochastic tree graph structure. In this manner, switching between contexts is accomplished by switching the dimensions in the multidimensional data representation, which is very effective and easy to implement. In order to efficiently represent the time-varying events described by the intermediate object description the knowledge representation scheme has two additional timing mechanism. The first mechanism reduces the size of the knowledge-base by eliminating "unused" or "rarely used" information. The second one incorporates into the knowledge base the time relations between the objects of a world-scene. The model of the IUS composed of interacting low-level, high-level processing stages and the knowledge base is shown in Figure 1.2. Additional blocks that are included in the IUS are an image synthesis stage and input-output devices. The low-level processing stage is designed to include the elements of adaptability and self-optimization.

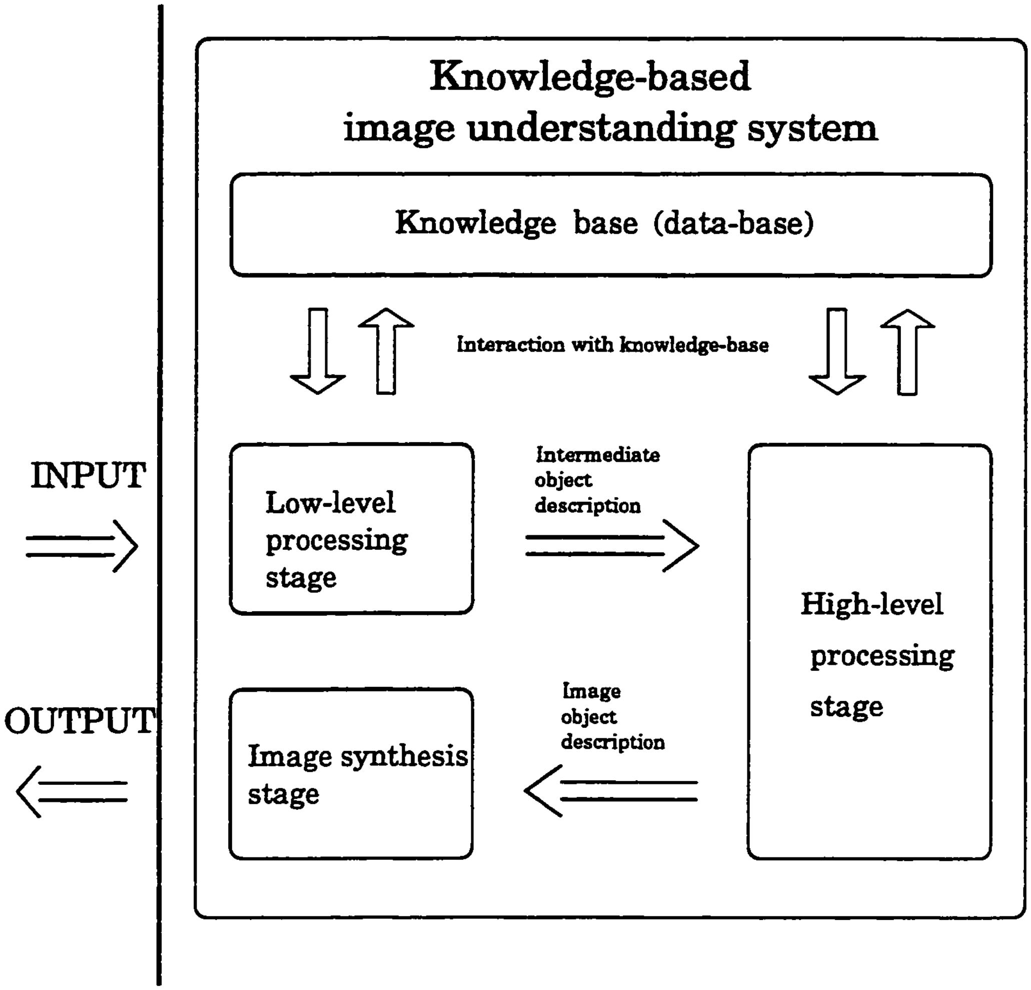

Knowledge-based
image understanding system
Knowledge base (data-base)
Interaction with knowledge-base
INPUT
Low-level processing stage
Intermediate object description
High-level processing stage
OUTPUT
Image synthesis stage
Image object description

Figure 1.2: The proposed model of the image understanding system.

The task of the high-level processing stage is to generate from the intermediate object description, the description of the image events and relations among them, and it is based on the interaction with the knowledge base of the IUS. The generated description represents information about complex two-dimensional world-scenes, that can be used to form symbolic representation of events. The high-level processing stage incorporates many techniques of artificial intelligence (AI) such as symbolic reasoning, Black-Board (BB) models and heuristic algorithms.

In order to demonstrate the effectiveness of the proposed prime component decomposition scheme and the model of the image understanding system, in Chapter 6, several examples of low-level and high-level processing applications are considered.

In Chapter 7, the results of this study are summarized and some suggestions for future investigation are made.

Chapter 2

IMAGE REPRESENTATION

One of the most fundamental and first operation of an image understanding system is the task of the forming an internal image representation of a world-scene. The quality of this operation directly determines many important characteristics of the IUS such as memory requirements, real-time performance and the system's hardware and software resources. Moreover, the understanding of the image formation process helps to develop efficient internal data structures and to better optimize the subsequent stages of the IUS. particularly the low-level processing stage and the construction of intermediate object representation. The construction of an accurate intermediate object representation, in turn, can avoid the use of many computationally expensive algorithms within the low-level and high-level processing stages of an IUS.

2.1 Image Formation Model

The image formation model of a vision system can be symbolically presented as in Figure 2.1. The function $W_{sc} : X' \times Y' \rightarrow G'$ represents a real world-scene, whereas $I_{im} : X \times Y \rightarrow G$ represents the inner-image formation, where X', Y', G' denote the spatial coordinates and gray level of the world-scene, and X, Y, G denote these quantities for the inner-image representation. The function W_{sc} is defined as

$$W_{sc}(x',y') = \{(x',y',g') : x' \in X' \wedge y' \in Y' \wedge g' \in G'\}, \tag{2.1}$$

and similarly, I_{im} is defined as

$$I_{im}(x,y) = \{(x,y,g) : x \in X \wedge y \in Y \wedge g \in G\}, \tag{2.2}$$

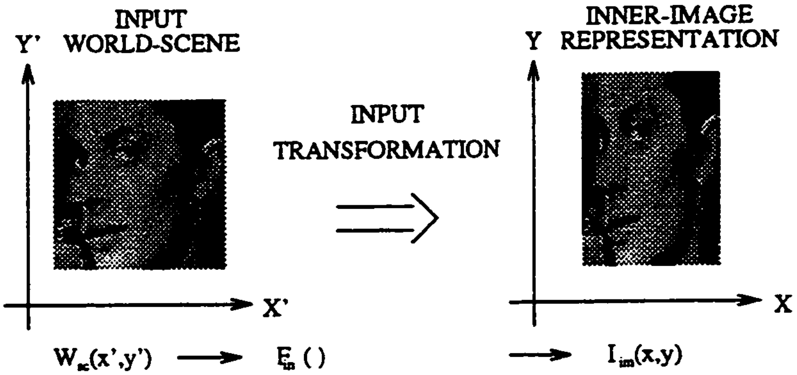

INPUT
WORLD-SCENE
Y'
X'
INPUT
TRANSFORMATION
INNER-IMAGE
REPRESENTATION
Y
X
$W_{sc}(x',y') \longrightarrow F_{in}() \longrightarrow I_{im}(x,y)$

Figure 2.1: Model of the image formation in a vision system.

where X', Y', G', X, Y, G $\subset \Re$, and $\Re$ denotes the set of all real numbers. The transformation function

$$F_{in} : X' \times Y' \times G' \times X \times Y \rightarrow G,$$

defined as

$$F_{in}(x',y',g',x,y) = \{(x',y',g',x,y,g) : (x',y',g') \in W_{sc} \wedge (x,y,g) \in I_{im}\}, \qquad (2.3)$$

transforms the world-scene onto the inner-image representation in the visual system. The spatial coordinates X', Y', X, Y are related by the relations CXYX' : $X \times Y \rightarrow X'$ and CXYY' : $X \times Y \rightarrow Y'$., which are defined by

$$CXYX'(x.y) = \{(x,y,CXYX'(x,y)) : (x,y) \in X \times Y \wedge CXX'(x,y) \in X \times Y \times X'\}. \qquad (2.4)$$

and

$$CXYY'(x,y) = \{(x,y,CXYY'(x,y)) : (x,y) \in X \times Y \wedge CXYY'(x,y) \in X \times Y \times Y'\}. \qquad (2.5)$$

In this manner, we obtain

$$\begin{aligned} I_{im}(x,y) &= F_{in}(x',y',g',x,y) \\ &= F_{in}(x',y',W_{sc}(x',y'),x,y) \\ &= F_{in}(CXYX'(x,y),CXYY'(x,y),W_{sc}(CXYX'(x,y),CXYY'(x,y)),x,y). \end{aligned} \qquad (2.6)$$

The transformation given by (2.6), in the general case, is very complex. In order to simplify the analysis, one can assume that the functions CXYX', CXYY' are just simple substitutions that is, CXY$X'(x,y) = x$ and CXY$Y'(x,y) = y$. In addition, let $x' = x$ and $y' = y$. After this simplification, the transformation F_{in} yields

$$I_{im}(x,y) = F_{in}(x,y,W_{sc}(x,y)).$$

Additional simplification can be achieved by linearizing (2.6), which in many cases is a good approximation of " word-scene → inner-image " relation.

Definition 2.1 :
The transformation

$$I_{im}(x,y) = F_{in}(x,y,W_{sc}(x,y))$$

is said to be linear in respect to domain G ($W_{sc} \subset X \times Y \times G$) iff $(\forall \alpha, \beta \in G) \wedge$ $(\forall W1_{sc}, W2_{sc} \subset X \times Y \times G)$

$$F_{in}(x,y,(\alpha W1_{sc}(x,y) + \beta W2_{sc}(x,y))) = \alpha F_{in}(x,y,W1_{sc}(x,y)) + \beta F_{in}(x,y,W1_{sc}(x.y)).$$

When the input is the two-dimensional delta function $\delta(\cdot,\cdot)$ at $(x_1,y_1) \in X \times Y$, $\delta(\cdot,\cdot) = \delta(x-x_1, y-y_1)$, the output in coordinates $(x,y) \in X \times Y$ can be expressed as

$$h(x,y,x_1,y_1) = F_{in}(x-x_1, y-y_1, \delta(x-x_1,y-y_1)). \tag{2.7}$$

Now, for this linear system, the inner-image function $I_{im}(x,y)$ can be written as

$$I_{im}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y,x_1,y_1) W_{sc}(x_1,y_1)\, dx_1\, dy_1. \tag{2.8}$$

Equation (2.8) represents spatially-varying point-spread transformation of the world-scene. A spatially-varying transformation becomes spatially invariant or shift invariant, if a translation in the domain of the world-scene W_{sc} causes the same translation in the domain of the inner-image representation I_{im}. Alternatively, (2.8) becomes spatially invariant if the function $h(x,y,x_1,y_1)$ depends only on x and y.

Definition 2.2:
The transformation

$$I_{im}(x,y) = F_{in}(x,y,W_{sc}(x,y))$$

is said to be spatially invariant iff $(\forall x, x_1 \in X) \wedge (\forall y, y_1 \in Y) \wedge (\forall W_{sc} \subset X' \times Y' \times G')$

$$I_{im}(x - x_1, y - y_1) = F_{in}(x - x_1, y - y_1, W_{sc}(x - x_1, y - y_1))$$

$$\Updownarrow \tag{2.9}$$

$$\begin{aligned} h(x, y, x_1, y_1) &= F_{in}(x - x_1, y - y_1, \delta(x - x_1, y - y_1)) \\ &= h(x - x_1, y - y_1, 0, 0) \\ &= h(x - x_1, y - y_1) \end{aligned}$$

Consequently, for the shift-invariant image formation model, the inner-image representation can be obtained by performing the convolution operation of the world-scene W_{sc} with the impulse response of the the system $h(x, y)$ as given by

$$I_{im}(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x, y) W_{sc}(x_1, y_1)\, dx_1\, dy_1. \tag{2.10}$$

For a shift-invariant image formation model, the convolution operation given by (2.10) can be performed by utilizing the properties of the Fourier transformation, i.e.

$$I_{im}(x, y) = h(x, y) \odot W_{sc}(x, y) \Leftrightarrow \mathcal{I}_{im}(\xi_1, \xi_2) = H(\xi_1, \xi_2) \mathcal{W}_{sc}(\xi_1, \xi_2). \tag{2.11}$$

where $\mathcal{I}_{im}(\xi_1, \xi_2)$, $H(\xi_1, \xi_2)$ and $\mathcal{W}_{sc}(\xi_1, \xi_2)$ are the two-dimensional Fourier transforms of the $I_{im}(x, y)$, $h(x, y)$ and $W_{sc}(x, y)$, respectively, and the symbol $\odot$ represents the convolution operation. The two dimensional Fourier transform $\mathcal{F}(\xi_1, \xi_2)$ of the function $f(x, y)$ is defined by the linear transformation as

$$\mathcal{F}(\xi_1, \xi_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(x\xi_1 + y\xi_2)} dx\, dy. \tag{2.12}$$

The inverse Fourier transform of $\mathcal{F}(\xi_1, \xi_2)$ is given by

$$f(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{F}(\xi_1, \xi_2) e^{j2\pi(x\xi_1 + y\xi_2)} dx_{\xi_1}\, dy_{\xi_1}. \tag{2.13}$$

The function $H(\xi_1, \xi_2)$, which is the Fourier transform of the impulse response $h(x, y)$, is also called the frequency response of the image formation.

2.2 Distortions and Geometrical Transformations

Equation (2.10) has been derived by setting the coordinate transformation functions (operators) CXYX', CXYY' as identity operators, i.e., no geometrical distortion occurs. In a practical IUS, the inner-image representation is constructed by transforming the physical stimulus (world-scene W_{sc}) into the internal data structures of the system. The transformation process is usually performed by different types of the input devices having their own nonlinear or linear input-output characteristics. These characteristics are the major factors in various types of geometrical distortions such as in displacement (translation), dialation (magnification), rotation, perspective distortions. Considering the case in which the geometrical distortion occurs, one should take into account the functions CXYX' and CXYY' as well. Typical cases can be described as

(a) Identity operation:

$$\begin{bmatrix} CXYX'(x,y) \\ CXYY'(x,y) \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.14}$$

(b) Translation:

$$\begin{bmatrix} CXYX'(x,y) \\ CXYY'(x,y) \end{bmatrix} = \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix}. \tag{2.15}$$

(c) Reflection:

$$\begin{bmatrix} CXYX'(x,y) \\ CXYY'(x,y) \end{bmatrix} = \begin{bmatrix} x_1 - x \\ y \end{bmatrix}. \tag{2.16}$$

(d) Magnification:

$$\begin{bmatrix} CXYX'(x,y) \\ CXYY'(x,y) \end{bmatrix} = \begin{bmatrix} x \,/\, x_1 \\ y \,/\, y_1 \end{bmatrix}. \tag{2.17}$$

(e) Rotation through positive angle Θ:

$$\begin{bmatrix} CXYX'(x,y) \\ CXYY'(x,y) \end{bmatrix} = \begin{bmatrix} x\cos\Theta \ - \ y\sin\Theta \\ x\sin\Theta \ + \ y\cos\Theta \end{bmatrix}. \tag{2.18}$$

Taking into account the geometrical transformations of spatial coordinates, (2.10) becomes

$$I_{im}(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\delta(CXYX'(x,y) - x_1, CXYX'(x,y) - y_1)W_{sc}(x_1,y_1)\,dx_1\,dy_1. \tag{2.19}$$

In addition, some distortions may also occur due to the diffraction of the optical systems used as input devices, the motion between the detector and the objects present in a world-scene, atmospheric turbulence or rectangular scanning aperture. This type of the geometrical distortions can be described by the spatially invariant image formation models having different impulse responses as given below [10], [11], [43].

(f) Limited coherent diffraction (with rectangular aperture)

$$h(x,y) = ab\,\mathrm{sinc}(ax)\,\mathrm{sinc}(by) \Leftrightarrow H(\xi_1,\xi_2) = \mathrm{rect}(\frac{\xi_1}{a},\frac{\xi_2}{a}) \tag{2.20}$$

(g) Limited incoherent diffraction (with rectangular aperture)

$$h(x,y) = ab\,\mathrm{sinc}^2(ax)\,\mathrm{sinc}^2(by) \Leftrightarrow H(\xi_1,\xi_2) = \mathrm{tri}(\frac{\xi_1}{a},\frac{\xi_2}{a}) \tag{2.21}$$

(h) Horizontal motion

$$h(x,y) = \frac{1}{\alpha_0}\mathrm{rect}(\frac{x}{\alpha_0} - \frac{1}{2})\delta(y) \Leftrightarrow H(\xi_1,\xi_2) = \mathrm{sinc}(\xi_1\alpha_0)e^{(-j\pi\xi_1\alpha_0)} \tag{2.22}$$

(i) Atmospheric turbulence

$$h(x,y) = e^{-\pi\alpha^2(x^2+y^2)} \Leftrightarrow H(\xi_1,\xi_2) = \frac{1}{\alpha^2}e^{\frac{-\pi(\xi_1^2+\xi_2^2)}{\alpha^2}} \tag{2.23}$$

(j) Rectangular scanning aperture

$$h(x,y) = \mathrm{rect}(\frac{x}{\alpha},\frac{y}{\beta}) \Leftrightarrow H(\xi_1,\xi_2) = \alpha\beta\,\mathrm{sinc}(\alpha\xi_1)\,\mathrm{sinc}(\beta\xi_2) \tag{2.24}$$

In (2.20) and (2.21), the function rect(.,.) and tri(.,.) are defined as

$$\mathrm{rect}(\xi_1,\xi_2) = \begin{cases} 1 & |\xi_1|,|\xi_2| < \frac{1}{2} \\ 0 & |\xi_1|,|\xi_2| > \frac{1}{2}, \end{cases} \tag{2.25}$$

and

$$\mathrm{tri}(\xi_1,\xi_2) = \begin{cases} (1-|\xi_1|)(1-|\xi_2|) & |\xi_1|,|\xi_2| < 1 \\ 0 & |\xi_1|,|\xi_2| > 1. \end{cases} \tag{2.26}$$

2.3 Summary

In this chapter, the necessary background material concerning image-formation model in an IUS has been presented. In the model presented, the inner image representation I_{im} is constructed by applying the input transformation F_{in} to the real world-scene W_{sc}. The input transformation F_{in} has been linearized in order to obtain a simple and convenient expression that is suitable for analysis. In addition, the image formation model has been refined to take into account the geometrical transformations and distortions due to the diffraction occurring in optical systems.

Chapter 3

PRIME COMPONENT DECOMPOSITION

As mentioned in Chapter 1, one of the most important tasks of the low-level processing stage in an IUS is to generate an intermediate object description of a scene [84], [90]. An intermediate object description is obtained as a result of a transformation of the given scene into set(s) of objects and "relations" (associations) among them. Because of the physical properties of real world-scenes, the most natural relations are those between the scene objects and the groups of geometrical objects represented as two-dimensional polygons. Consequently, most of the transformations that produce intermediate object description are based on geometrical transformations of input world-scenes. Conceptually, such transformations can be equivalent to a geometrical (polygonal) decomposition of the scene into groups of pixels having similar relational properties [94]. In majority of the cases, the typical result of the polygonal decomposition operation on a world-scene followed by additional processing (such as shape equalization, edge detection etc.) in the low-level stage is the intermediate object representation known as semantic network.

The process of generating an intermediate object description is extremely important part of the knowledge transformation in an IUS. The structure of the intermediate object description and its complexity characterizes the properties and the capability of the IUS in relation to its processing power. Therefore, the process of modeling the image decomposition operator should be performed very carefully, keeping in mind many independent parameters related to the properties of the decomposition operator developed and its implementation.

In this chapter, he decomposition operator is developed and optimized by the analysis of the properties of world-scenes. First, the most general relations between the prime components are investigated. Next, the basic relations among the objects found in a scene are identified and their implications on the process of modeling of the decomposition operator are determined. The optimization process of the decomposition operator is performed in the discreet metric space for its practical implementation. The results of the mathematical analysis are supported through a statistical simulation employing a family of world-scenes.

3.1 Prime Components and their Associations

In order to derive the basic properties of the decomposition, let us first assume that the input transformation defined by (2.6) is an identity function (operator), that is,

$$\left(I_{im}(x,y) \; = \; W_{sc}(x^{'},y^{'}) \left| \begin{array}{l} \\ x \; = \; x' \\ y \; = \; y' \end{array} \right. \right) \Longrightarrow (G \; = \; G^{'}). \tag{3.1}$$

Choosing the representation of a word-scene as a function $W_{sc}^{n} : X' \times Y' \rightarrow G'$, where n (the frame number) belongs to the set of natural numbers ($\mathcal{N}$), we can construct a set of all possible word-scenes $\mathcal{U}_{\mathcal{W}}$, whose members are functions of the form

$$\begin{aligned} \mathcal{U}_{\mathcal{W}} &= \{\ldots, W_{sc}^{n-1}, W_{sc}^{n}, W_{sc}^{n+1}, \ldots\} \\ &= \{W_{sc}^{n} : n \in \mathcal{N}\}. \end{aligned} \tag{3.2}$$

Similarly, we can construct a set of all possible inner-image representations $\mathcal{U}_{\mathcal{I}}$ as

$$\begin{aligned} \mathcal{U}_{\mathcal{I}} &= \{\ldots, I_{im}^{n-1}, I_{im}^{n}, I_{im}^{n+1}, \ldots\} \\ &= \{I_{im}^{n} : n \in \mathcal{N}\}. \end{aligned} \tag{3.3}$$

Because of (3.1), these two sets are equal, i.e., $\mathcal{U}_{\mathcal{W}} = \mathcal{U}_{\mathcal{I}}$. We can now introduce a decomposition operator $\mathcal{DEC}[\cdot]$ as $\mathcal{DEC} : \mathcal{U}_{\mathcal{I}} \rightarrow \mathcal{U}_{\mathcal{P}} \times \mathcal{U}_{\mathcal{A}}$, which maps the set of all inner-image representations $\mathcal{U}_{\mathcal{I}}$ into a set of all objects called the prime components $\mathcal{U}_{\mathcal{P}}$ and a set of associations (links) $\mathcal{U}_{\mathcal{A}}$ among them. Consequently, the decomposition $\mathcal{DEC}[\cdot]$ can be defined as

$$\mathcal{DEC}[I^n_{im}] = \{(W^n_{sc}, Z^n_{sc}) \,:\, I^n_{im} \in \mathcal{U}_{\mathcal{I}} \wedge Z^n_{sc} \in \mathcal{U}_{\mathcal{P}} \times \mathcal{U}_{\mathcal{A}},\, n \in \mathcal{N}\}\,, \tag{3.4}$$

where Z^n_{sc} denotes the resulting decomposition of the inner image representation W^n_{sc}. For a finite and countable set $\mathcal{U}_{\mathcal{P}}$ having r elements P^k_{sc} $\{\, k, r \in \mathcal{N}$ and $k < r < \infty\}$, one can write

$$\begin{aligned} \mathcal{U}_{\mathcal{P}} &= \{P^0_{sc}, P^1_{sc}, P^2_{sc}, \ldots, P^{r-1}_{sc}\} \\ &= \{P^k_{sc} \,:\, k \in \mathcal{N} \wedge k < r\}\,. \end{aligned} \tag{3.5}$$

Similar to the set of all prime components $\mathcal{U}_{\mathcal{P}}$, the countable set $\mathcal{U}_{\mathcal{A}}$ of all associations having s elements A^{mk}_{sc} $\{\, m, k \in \mathcal{N}$ and $m, k < r < \infty\}$, where superscript "mk" denotes association between elements P^m_{sc} and P^k_{sc}, can be expressed as

$$\mathcal{U}_{\mathcal{A}} = \{A^{mk}_{sc} \,:\, m, k \in \mathcal{N} \wedge m, k < r < \infty\}\,. \tag{3.6}$$

The association $A^{mk}_{sc} \in \mathcal{U}_{\mathcal{A}}$, $\{\, m, k \in \mathcal{N}$ and $m, k < r < \infty\}$ is:
(i) reflective $\iff$ (m = k),
(ii) symmetric $\iff$ $(A^{mk}_{sc} \equiv A^{km}_{sc})$.
The above two properties limit the number of distinct elements in $\mathcal{U}_{\mathcal{A}}$ by

$$s \leq \begin{cases} r^2 & \text{if all associations are allowed.} \\ r^2 - r & \text{if the reflective elements are excluded} \\ \frac{r^2-r}{2} & \text{if the reflective and symmetric elements are excluded.} \end{cases}$$

Figure 3.1 shows an example of 4 prime components together with their associations. The associations (links) should be understood as a collection of rules (operators) that connect (unify) the given set of prime components to the equivalent description set of the world-scene. By an equivalent description of the word-scene, it is meant that the given description preserves all the information of the input world-scene W_{sc}.

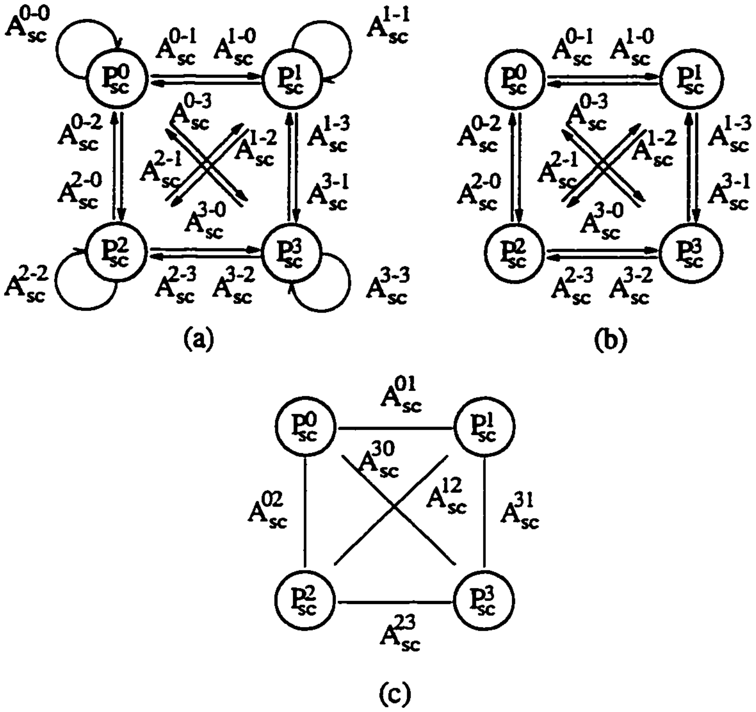


Figure 3.1: Prime component representation of a world-scene. (a) allowing all associations A_{sc}^{mk}, (b) excluding reflective associations, (c) excluding reflective and symmetric associations.

3.2 Relations between Prime Components

In the previous section, the relation between prime components have been defined as a general set of rules known as associations that combine together the set of prime components. In order to proceed further, the theoretically defined set of associations has to be described in terms of the rules and relations from the real physical world.

Let $I_{im}(x, y)$ represent the inner image representation of the real world-scene $W_{sc}(x', y')$ obtained by the applying the transformation (3.1). All functions I_{im}^{n} are projections from the physical world, from which we can conclude that physical relations among the objects will have their natural reflection in the relations belonging to the inner image representation I_{im}. Let $\mathbf{R_L}$ be a relation giving rise to members $V_m^L \subseteq X \times Y \times G$, $m \in \mathcal{N}$. Now, for the given relation $\mathbf{R_L}$, the equivalence set $\mathcal{V}_{sc}^L \subseteq X \times Y \times G$ can be formed. The fact that every set $V_m^L \subseteq X \times Y \times G$, $m \in \mathcal{N}$ (member of $\mathcal{V}_{sc}^L \subseteq X \times Y \times G$) is composed of the elements

$$i_{im,k}^L = (x_k, y_k, g_k) \in X \times Y \times G = (x_k, y_k, I_{im}(x_k, y_k)),\ k \in \mathcal{N} \wedge (x_k, y_k) \in X \times Y, \quad (3.7)$$

implies a relation

$$i_{im,k}^L \in I_{im} \in V_m^L \in X \times Y \times G\,.$$

Generally, the relation $\mathbf{R_L}$ that "extracts" the member V_m^L from I_{im} can be very broad [30], [70], [87], [98] . It could be a set of elements having the same gray level $g \in G$, or a set of elements having "the same shape" or a set having some special properties. Some examples of relations are presented in Figure 3.2. Figure 3.3 shows one particular example of the relation $\mathbf{R_L}$ belonging to an inner image representation I_{im}, which groups together the elements $I_{im}(x, y) = i_{im}^\alpha$, $\alpha \in \mathcal{N}$ having the same gray level g'. In this case 4-gray level 128 × 128 image has been chosen.

Definition 3.1:

A relation $\mathbf{R_L}$ on a set $X \times Y \times G$ ($\mathbf{R_L} \subseteq (X \times Y \times G) \times (X \times Y \times G)$ is an *equivalence relation* iff it has the following properties for all (x_1, y_1, g_1), (x_2, y_2, g_2), (x_3, y_3, g_3) in $X \times Y \times G$:

(1) $((x_1, y_1, g_1), (x_1, y_1, g_1)) \in \mathbf{R_L}$

(2) if $((x_1, y_1, g_1), (x_2, y_2, g_2)) \in \mathbf{R_L}$ then $((x_2, y_2, g_2), (x_1, y_1, g_1)) \in \mathbf{R_L}$.

(3) if $((x_1, y_1, g_1), (x_2, y_2, g_2)) \in \mathbf{R_L}$ and $((x_2, y_2, g_2), (x_3, y_3', g_3)) \in \mathbf{R_L}$ then $((x_1, y_1, g_1), (x_3, y_3, g_3)) \in \mathbf{R_L}$.

It is easy to check that the relations $\mathbf{R_1}$, $\mathbf{R_2}$, and $\mathbf{R_3}$ shown on the Figure 3.2 as well as the relation $\mathbf{R_L}$ shown in the Figure 3.3 are *equivalence relations* on a set $X \times Y \times G$.

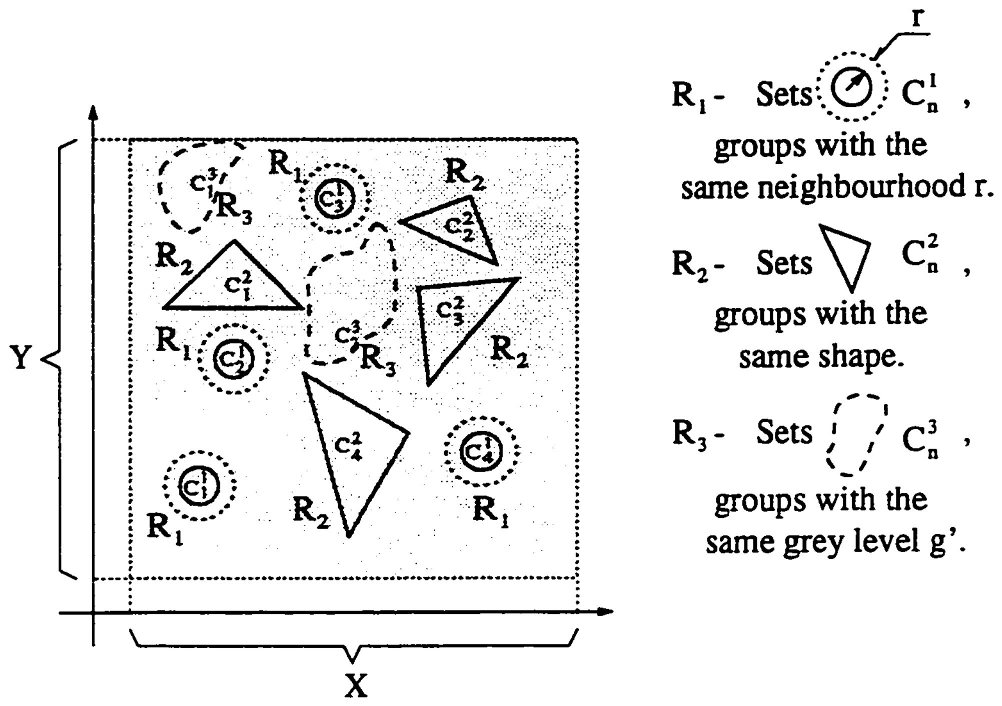


Figure 3.2: Examples of the relations $\mathbf{R_L}$ in an inner image representation I_{im}.

For the equivalence relations $\mathbf{R_L}$ on a set $X \times Y \times G$, we define *equivalence class* (with respect to equivalence relation $\mathbf{R_L}$) of $(x_1, y_1, g_1) \in X \times Y \times G$ to be the set

$$E_{(x_1,y_1,g_1)} = \{(x_2, y_2, g_2) : ((x_2, y_2, g_2), (x_1, y_1, g_1)) \in \mathbf{R_L}\}.$$

Definition 3.2:
A *partition* of the set $X \times Y \times G$ is a collection C of nonempty subsets of $X \times Y \times G$ such that:

(1) each $(x_1, y_1, g_1) \in X \times Y \times G$ belongs to some subset A$\in C$, and

(2) if A, $B \in C$ and $A \neq B$, then $A \cap B = \emptyset$.

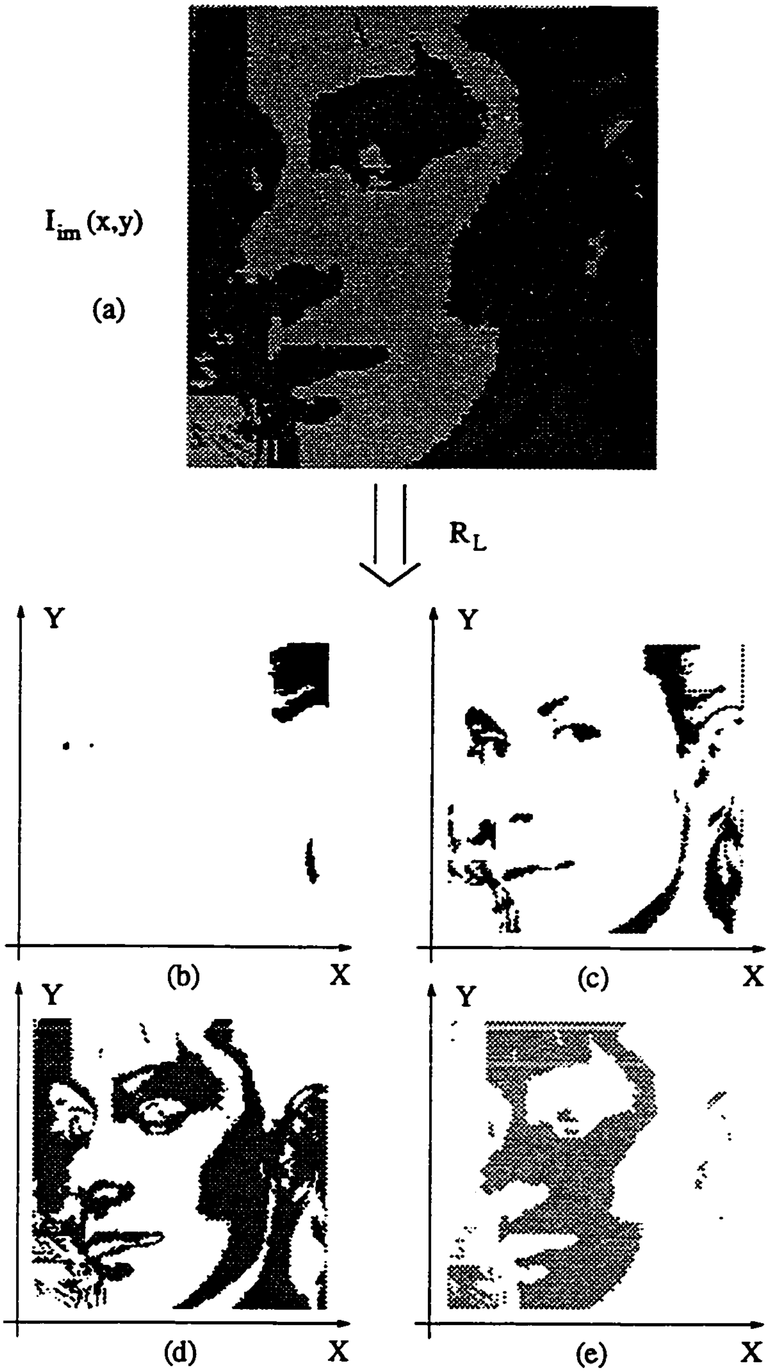


Figure 3.3: Decomposition of a scene by a relation of groups of pixels with the same gray level. (a) the scene. (b-e) members of the equivalence set V_{sc}^{L}.

It can be easily verified that for an equivalence relation $\mathbf{R_L}$ on the set $X \times Y \times G$, the set

$$\mathcal{C} = \{E(x_2, y_2, g_2) : ((x_2, y_2, g_2), (x_1, y_1, g_1)) \in \mathbf{R_L}\}$$

is a partition of $X \times Y \times G$. Conversely, if $\mathcal{C}$ is a partition of $X \times Y \times G$ using a relation $\mathbf{R_L}$, then $\mathbf{R_L}$ is an equivalence relation and the set of corresponding equivalence classes is the same as $\mathcal{C}$.

According to the definitions of equivalence class and partition, it follows that if a relation $\mathbf{R_L}$ is an equivalence relation then the set $\mathcal{V}_{sc}^L$ is a partition (collection) of the set $X \times Y \times G$ and its each *member* $V_m^L \subseteq X \times Y \times G$, $m \in \mathcal{N}$ stands for an equivalence class of the relation $\mathbf{R_L}$ "corresponding" to the set $E_{(x_1, y_1, g_1)}$ defined above. Thus, we will have

$$V_m^L = E_{(x_1, y_1, g_1)} \subseteq \mathcal{V}_{sc}^L, \ m \in \mathcal{N}.$$

Consequently, the sets determined by relations $\mathbf{R_n}$ $(\mathrm{n} = 1, 2, 3)$ in Figure 3.2 can be grouped as

$$\begin{aligned}
\mathcal{V}_{sc}^1 &= \{V_1^1, V_2^1, V_3^1, V_4^1\} = \{C_1^1, C_2^1, C_3^1, C_4^1\}, \\
\mathcal{V}_{sc}^2 &= \{V_1^2, V_2^2, V_3^2, V_4^2\} = \{C_1^2, C_2^2, C_3^2, C_4^2\}, \\
\mathcal{V}_{sc}^3 &= \{V_1^3, V_2^3\} = \{C_1^3, C_2^3\} \\
\mathcal{V}_{sc}^4 &= X \times Y \times G \setminus \{\mathcal{V}_{sc}^1 \cup \mathcal{V}_{sc}^2 \cup \mathcal{V}_{sc}^3\} \\
\mathcal{C} &= \{\mathcal{V}_{sc}^1, \mathcal{V}_{sc}^2, \mathcal{V}_{sc}^3, \mathcal{V}_{sc}^4\}.
\end{aligned}$$

Having formally identified the relations in a scene, we can now define one of the most important properties of the decomposition operator.

Proposition I :

The decomposition $\mathcal{DEC} : \mathcal{U_I} \rightarrow \{\mathcal{U_P}, \mathcal{U_A}\}$ is *context invariant* under the equivalence relation $\mathbf{R_L} \subseteq (X \times Y \times G) \times (X \times Y \times G)$ iff

(a) for all equivalence classes V_m^L $(n \in \mathcal{N})$ with respect to $\mathbf{R_L}$

$$\bigcap_{V_m^L \in I_{im}^n} \mathcal{DEC}[I_{im}^n(x, y)] = PA_m^L = PA^L = \{\mathcal{P}_n^L, \mathcal{A}_n^L\} \qquad (3.8)$$

(b) $\mathcal{DEC}[I^n_{im}(x,y)] \cap PA^L = PA^L$, $n \in \mathcal{N} \Rightarrow \exists m \in \mathcal{N}$ such that $V^L_m \in I^n_{im}$

The set $PA^L = \{\mathcal{P}^L_{n}, \mathcal{A}^L_{n}\}$ is then called the *representative* of the relation $\mathbf{R_L}$.

If the inner image representation $I^n_{im}(x,y)$, $n \in \mathcal{N}$, includes at least one equivalence class V^L_m with respect to $\mathbf{R_L}$ then we can say that relation $\mathbf{R_L}$ is present in the inner image representation $I^n_{im}(x,y)$

3.3 Properties of a Decomposition Operator

Generally speaking, specific properties of a decomposition operator $\mathcal{DEC}[\cdot]$ is difficult to enunciate. Only the most general ones that characterize the decomposition can be stated as follows.

(I) Decomposition should preserve all the input information.

(II) Decomposition should decompose information uniquely under the given context.

(III) Decomposition should be context invariant under the equivalence relation $\mathbf{R_L}$.

(IV) Decomposition should significantly reduce the complexity of the description of a given scene.

Corresponding to property (I), the decomposition operator $\mathcal{DEC}[\cdot]$ must posses its inverse operator $\mathcal{DEC}^{-1} : \mathcal{U_P} \times \mathcal{U_A} \rightarrow \mathcal{U_I}$, such that the given input inner image representation I^n_{im} can be recovered. Further, because of properties (I) and (II), the decomposition operator $\mathcal{DEC}[\cdot]$ is a one-to-one mapping function (bijective).

Let X be a non-empty set. A metric on X is a real function $d : (x,y) \rightarrow z \in \mathcal{R}$ which satisfies the following three conditions

(1) $d(x,y) \geq 0$, and $d(x,y) = 0 \iff x = y$

(2) $d(x,y) = d(y,x)$

(3) $d(x,y) \leq d(x,z) + d(z,y)$.

The function $d(x, y)$ is called the distance between elements x and y. A metric space consists of two objects: a non-empty set X and a metric d on X. Let A be subset of X $(A \subseteq X)$. If $x \in X$, then the distance from x to A is defined by

$$d(x, A) = min\{d(x, a) : a \in A\}. \tag{3.9}$$

Similarly, let A and B be two subsets of X $(A, B \subseteq X)$. Then, the distance between A and B is defined by

$$d(A, B) = min\{d(a, b) : (a, b) \in A \times B\}. \tag{3.10}$$

According to (2.18), a rotation operator $\mathcal{ROT}[\cdot]$ on a region $\mathcal{R}^n(x, y) \subseteq I^n_{im}(x, y)$ can be defined as

$$\mathcal{ROT}[\mathcal{R}^n(x, y)(\Theta)] = \{((x, y), g) : \mathcal{R}^n_{\mathrm{rot}}(x, y) = \mathcal{R}^n(CXYX(x, y), CXYY(x, y)) \in I^n_{im}(x, \tag{3.11}$$

where by $\mathcal{R}^n$ we denote any region (subset) of I^n_{im} $(\mathcal{R}^n(x, y) \subseteq I^n_{im}(x, y),\ n, k \in \mathcal{N})$ and functions $CXYX(x, y)$ and $CXYY(x, y)$ are defined consistent to (2.4) and (2.5). Usually, each set W^n_{sc} is a frame from the physical world; therefore, one can assume that the following properties hold.

(a) Probabilities of occurrence of a region $\mathcal{R}^n(x, y)$ and the region $\mathcal{R}^n_t(x, y)$, which is a translation of the region $\mathcal{R}^n(x, y)$ in an inner image representation $I^n_{im}(x, y)$ $(\mathcal{R}^n_t(x, y) = \mathcal{R}^n(x - x_0, y - y_0))$, are the same, that is

$$P(\mathcal{R}^n(x-x_0, y-y_0) \subseteq I^n_{im}(x, y)) = P(\mathcal{R}^n(x, y) \subseteq I^n_{im}(x, y)),\ \ (x-x_0, y-y_0) \in X \times Y. \tag{3.12}$$

(b) Probabilities of occurrence of a region $\mathcal{R}^n(x, y)$ and the region $\mathcal{R}^n_t(x, y)$, which is a rotated version of the region $\mathcal{R}^n(x, y)$ through an angle Θ , in an inner image representation $I^n_{im}(x, y)$ $(\mathcal{R}^n_t(x, y) = \mathrm{ROT}[\mathcal{R}^n(x, y)(\Theta)])$ are the same, that is

$$P(\mathcal{R}^n(x, y) \subseteq I^n_{im}(x, y)) = P(\mathcal{R}^n_t(x, y) \subseteq I^n_{im}(x, y)),\ \ (x, y) \in X \times Y. \tag{3.13}$$

(c) Given two disjoint regions $\mathcal{R}^n(x,y)$ and $\mathcal{R}_t^n(x,y)$ $(\mathcal{R}^n(x,y) \cap \mathcal{R}_t^n(x,y) = \emptyset)$, belonging to the same *equivalence class* $E_{(x,y,g)}$ with respect to equivalence relation $\mathbf{R_L}$ representing sets with the same gray level g_0

$$\mathcal{R}^n(x,y),\ \mathcal{R}_t^n(x,y) \subseteq E_{(x,y,g)} \subseteq \mathcal{R}_{\mathcal{L}},$$
$$\mathbf{R_L} = \{(x,y,g_0) : I_{im}^n(x,y) = g_0\},$$

the conditional probability of occurrence of the region $\mathcal{R}^n(x,y)$, given the region $\mathcal{R}_t^n(x,y)$, depends on the distance $d(\mathcal{R}^n, \mathcal{R}_t^n)$ between the region $\mathcal{R}(x,y)$ and the region $\mathcal{R}_t^n(x,y)$, that is,

$$P(\mathcal{R}^n(x,y) \subseteq I_{im}^n(x,y) \,/\, \mathcal{R}_t^n(x,y) \subseteq I_{im}^n(x,y)\,) \neq P(\mathcal{R}^n(x,y) \subseteq I_{im}^n(x,y)). \quad (3.14)$$

Consistent with (3.8), property (III), and taking into consideration the physical properties (a), (b), and (c) of the objects in a real world-scene, it can be deduced that operator $\mathcal{DEC}$ is spatially invariant and it decomposes the inner image representation $I_{im}^n(x,y)$ into sets (regions) $\mathcal{R}_{k_m}^n(x,y)$ and spatial relations among them. This type of decomposition can be written as

$$I_{im}^n(x,y) = \mathcal{R}_{k_1}^n(x,y) \cup \mathcal{R}_{k_2}^n(x,y) \cup \mathcal{R}_{k_3}^n(x,y) \cup \ldots, \quad (3.15)$$

In this manner, each region $\mathcal{R}_{k_m}^n(x,y)$ corresponds to a particular prime component $P_{sc}^{k_m}$ and the spatial relations among the regions correspond to the sets of associations $A_{sc}^{k_i,k_j}$. The set of all prime components $\mathcal{U}_{\mathcal{P}}$ having r elements P_{sc}^k $\{\ k, r \in \mathcal{N}$ and $k < r < \infty\}$ can be understood as a basis from which any given inner image representation $I_{im}^n(x,y)$ (and the word scene W_{sc}^n) can be reconstructed.

3.4 Derivation of Decomposition Operator

One of the most important parameters that characterizes the decomposition operator $\mathcal{DEC}$ is the number of elements P_{sc}^k (prime components) required for the reconstruction of the inner image representation $I_{im}^n(x,y)$. The operator which generates statistically fewer number of elements P_{sc}^k for the description of the inner image representation is considered to be a better operator. Let the subset $Is_{im}^{n_k}$ of a given inner image representation $I_{im}^n(x,y)$ represent areas with the same specific gray level g_0, as given below

$$Is_{im}^{n_k}(x,y) = \{(x,y) : I_{im}^n(x,y) = g_0\,\}.$$

The decomposition operator $\mathcal{DEC}$ decomposes the subset $Is_{im}^{n_k}(x,y)$ into a set of prime components $u_p^{n_k}$ and a set of associations $u_a^{n_k}$,

$$\mathcal{DEC}[Is_{im}^{n_k}(x,y)] = \{u_p^{n_k}, u_a^{n_k}\} \subset \{U_P, U_A\}.$$

Next, let us consider another inner image representation $I_{im}^{n_2}$ that contains the subset $I's_{im}^{n_2}(x,y)$ which is a translation and rotation of the subset $Is_{im}^{n_k}(x,y)$ as given by

$$I's_{im}^{n_2}(x,y) = \mathcal{ROT}[Is_{im}^{n_k}(x-x_1, y-y_1)(\Theta)] \subset I_{im}^{n_2} \neq I_{im}^{n}.$$

In this case, the operator $\mathcal{DEC}$ decomposes the subset $I's_{im}^{n_2}(x,y)$ into a set of prime components $u_p'^{n_2}$ and a set of associations $u_a'^{n_2}$,

$$\mathcal{DEC}[I's_{im}^{n_2}(x,y)] = \{u_p'^{n_2}, u_a'^{n_2}\} \subset \{U_P, U_A\}.$$

Following this procedure many different subsets $\{u_p'^{n_k}, u_a'^{n_k}\}$ can be constructed. All the subsets obtained by applying this method are said to be mutually *congruent*. According to the postulated properties of a world-scene, probabilities of finding the sets $I's_{im}^{n_k}(x,y)$, $n_k = 1, 2, \ldots.$, are equal. Consequently, the number of elements of the sets $u_p^{n_k}$, $u_p'^{n_2}$, ... should be the same. This requirement in conjunction with the minimum number of elements of the basis can be achieved only for the sets that are *equivalent by finite decomposition*, which means, that for all the pairs $Is_{im}^{n_k}(x,y)$ and $Is_{im}^{n_l}(x,y)$, there exist the sets $P_{sc}^{k_0}, \ldots, P_{sc}^{k_n}$ and $P_{sc}^{l_0}, \ldots, P_{sc}^{l_n}$ such that the following holds

(a) $Is_{im}^{n_k}(x.y) = P_{sc}^{k_0} \cup \ldots \cup P_{sc}^{k_n}$ and $Is_{im}^{n_l}(x,y) = P_{sc}^{l_0} \cup \ldots \cup P_{sc}^{l_n}$.

(b) $P_{sc}^{k_i} \cap P_{sc}^{k_j} = \emptyset$ and $P_{sc}^{l_i} \cap P_{sc}^{l_j} = \emptyset$ for $i \neq j$.

(c) $P_{sc}^{k_i}$ is *congruent* to $P_{sc}^{l_i}$, for $i = 1, \ldots, n$.

As a result, one can conclude that

$$u_p^{n_l} = u_p^{n_k} = \ldots$$

and

$$P_{sc}^{k_i} = P_{sc}^{l_j} \text{ for } k \neq l \text{ and } i = j\,.$$

Since each prime component $P_{sc}^{k_i}$, $k = 1, 2, \ldots$, is a result of the decomposition of the different subsets $Is_{im}^{n_k}(x,y)$, each prime component becomes a representative of the rotationally and spatially invariant regions (subsets) $S_r(x_1, y_1) \subset W_{sc}^{n_k}$. This yields

$$\mathcal{ROT}[S_{r_1}(x_1,y_1)(\Theta_1)] = \mathcal{ROT}[S_{r_2}(x2,y2)(\Theta_2)] = \ldots = \mathcal{ROT}[S_{r_n}(xn,yn)(\Theta_n)],$$

and the sets $S_{r_k}(x_k, y_k) \subset I^n_{im}(x,y)$ have to be spheres in a metric on $I^n_{im}(x,y)$, that is,

$$S_{r_k}(x_k,y_k) = \{(x,y) : d((x,y),(x_k,y_k)) \leq r\},$$

where the function $d((x,y),(x_k,y_k))$ represents the distance between point (x,y) and point (x_k,y_k).

Two examples of decomposition of a particular subset $s^n_{im}(x,y)$ of the inner image representation $I^n_{im}(x,y)$ performed by the rotational invariant (spherical) prime components are shown in Figures 3.4 and 3.5.

The decomposition of the inner-image representation $I^n_{im}(x,y)$ with prime components P^k_{sc} can be performed in many different ways. Since every set $S_r(x_k,y_k)$ is uniquely described by its radius r and the coordinates of its center point (x_k,y_k), each individual prime component obtained from the decomposition can be represented by its center point (x_k,y_k) and the diameter d ($d = 2r$) of the set $S_{r_k}(x_k,y_k)$. One method of decomposition could be to perform the decomposition process randomly - *random decomposition* (diameters d and center point (x_k,y_k) of the S_{r_k} are taken randomly). Figure 3.4 is a particular example of such a decomposition. In another method, we could set $d = const$. Different decomposition methods would yield different performances (number of decomposed elements - i.e. prime components). In order to generate a set of prime components $\mathcal{U}_{\mathcal{A}}$ having minimum number of elements, the diameters of the prime components should be taken as large as possible. Consequently, the maximum diameter d_{max1} of every prime component that can be used as a decomposition element can be written as

$$d_{max1} = sup\{d : S_{d/2}(x,y) \in I^n_{im}\}$$

and

$$(x_{max1}, y_{max1}) = \{(x,y) : S_{d_{max1}}(x,y) \in I^n_{im}\}.$$

One can now construct a new inner-image representation I^{n2}_{im} by isolating from the set I^n_{im} the "largest" prime component $S_{d_{max1}}(x_{max1}, y_{max1})$

$$I^{n2}_{im} = I^n_{im} \setminus S_{d_{max1}}(x_{max1}, y_{max1}).$$

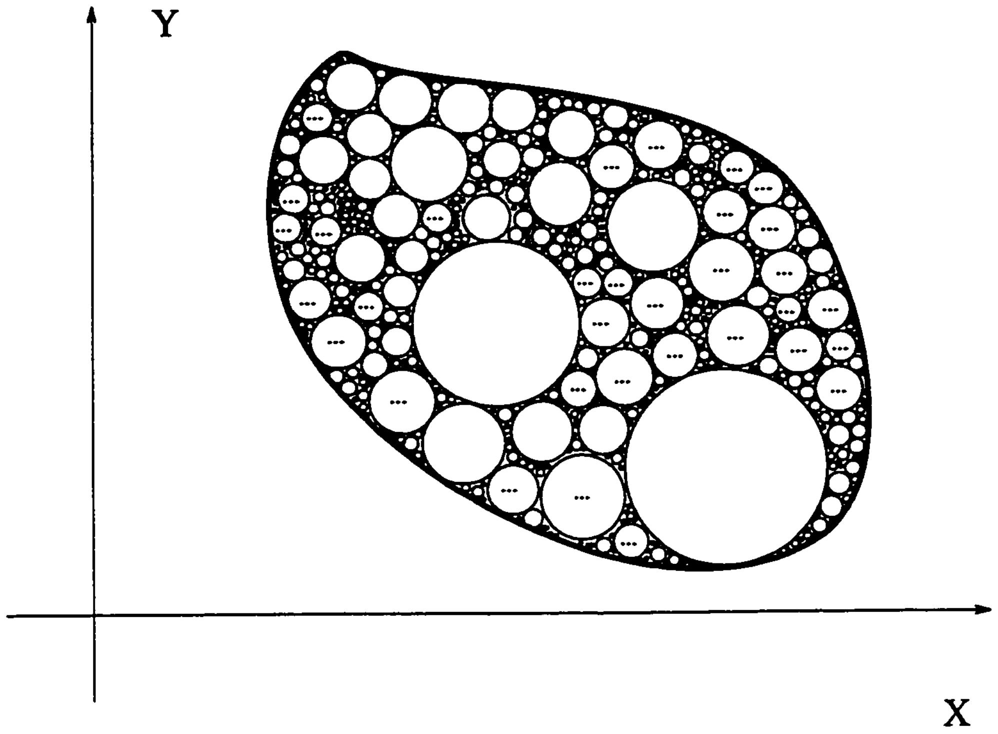


Figure 3.4: An example of random decomposition of a scene performed by rotational invariant (spherical) prime components.

By repeating this operation, the next max-diameter prime component $S_{d_{max2}}(x_{max2}, y_{max2})$ can be constructed as given below

$$d_{max2} = sup\{d : S_{d/2}(x,y) \in I_{im}^{n_2}\},$$

$$(x_{max2}, y_{max2}) = \{(x,y) : S_{d_{max2}}(x,y) \in I_{im}^{n_2}\}.$$

This process can be continued. Figure 3.5 is an example of a scene decomposition performed by the maximum size rotational invariant (spherical) prime components.

Figure 3.6 shows the decompositions of an image representation I_{im}^{n} by maximum-size prime components. The different sets of prime components shown in Figure 3.6(a) and

3.6(b) prove that in this particular case, the decomposition of the inner-image representation I^n_{im} by using the operator $\mathcal{DEC}$ is not unique. This is true for majority of the polygonal decompositions for which decomposed sets are not overlapping. That is, there could exist inner-image representations for which

$$\mathcal{DEC}[I_{im}] = \{z_1, z_2, \ldots\} = Z \,\text{and}\, z_i \cap z_j = \emptyset,\ i \neq j \Rightarrow$$

$$\exists\, Z' = \{z'_1, z'_2, \ldots\} \neq Z \text{ such that } \mathcal{DEC}[I_{im}] = Z'.$$

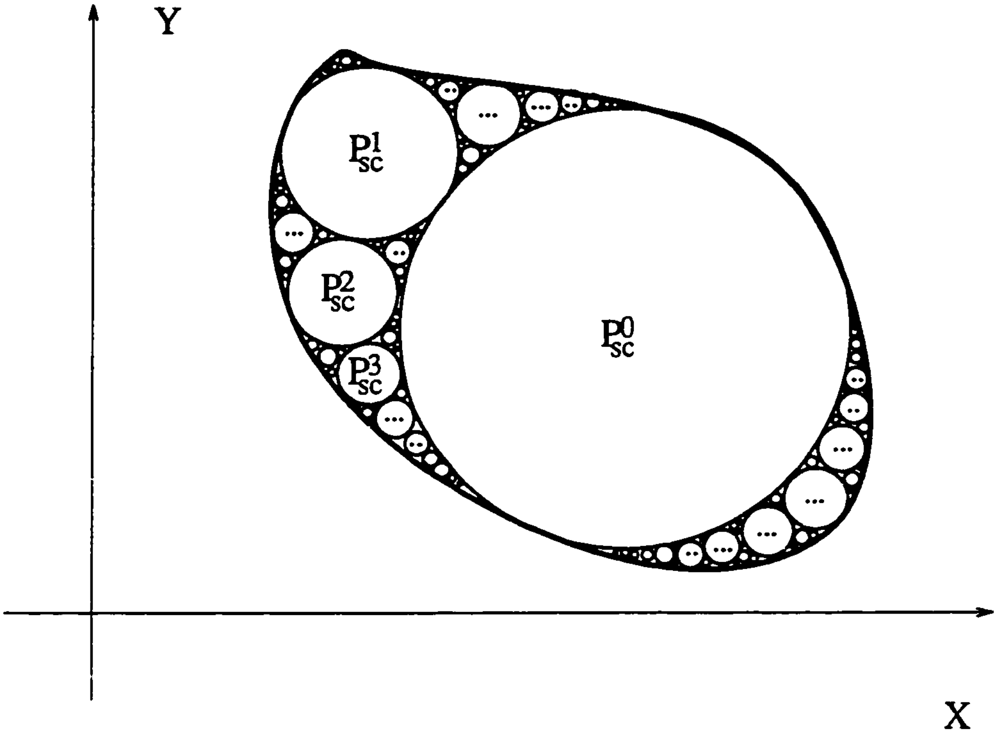
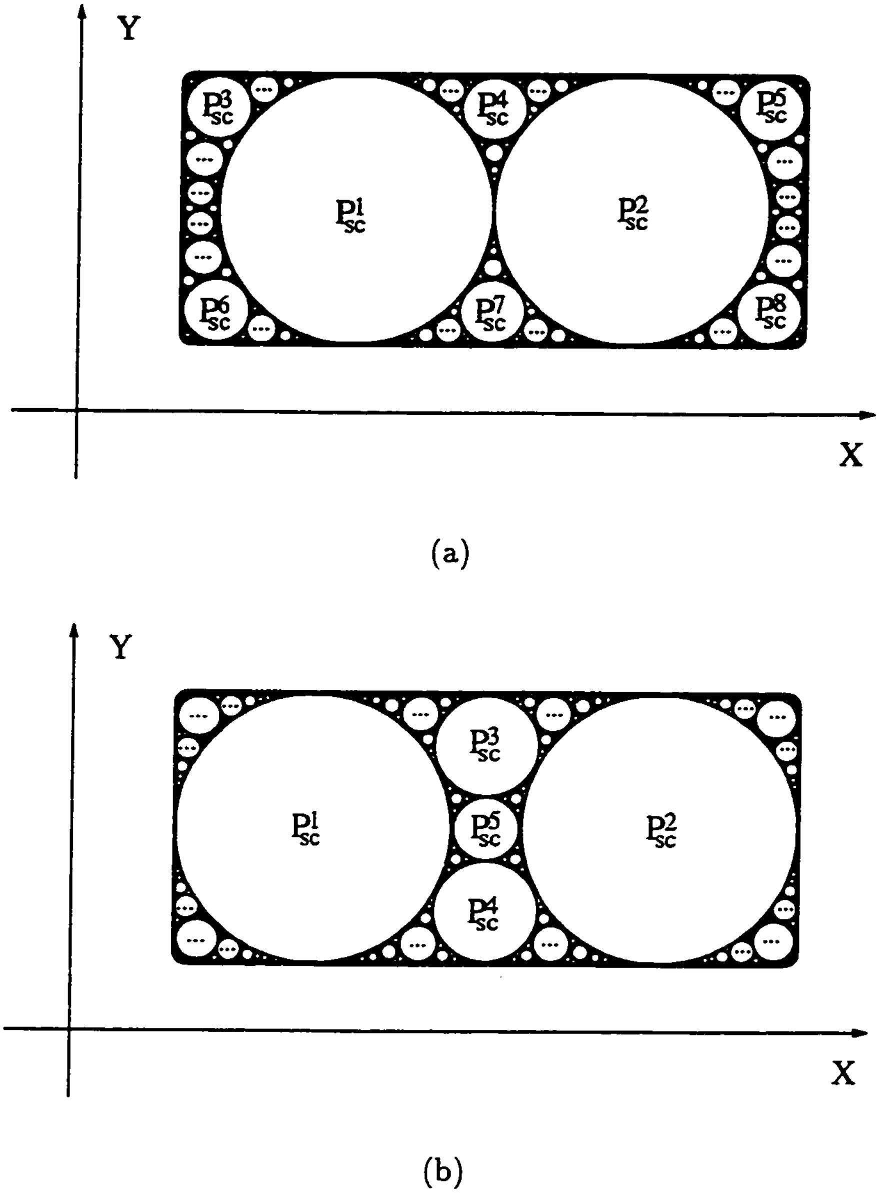


Figure 3.5: An example of scene decomposition performed by maximum-size rotational invariant (spherical) prime components.

The requirement on the decomposition operation, that the resulting prime components be disjoint, has significant consequences on the properties of the decomposition operator itself. The transformation described above is rotational invariant and shift invariant but not unique. Moreover, the resulting non-overlapping prime components are difficult to process. Thus, with an objective of achieving a unique transformation, the decomposition operator $\mathcal{DEC}$ should be modified.



(a)

(b)

Figure 3.6: Examples of non-unique decomposition of inner-image representation I^n_{im} with maximum-size rotational invariant (spherical) prime components.

Suppose that the inner image representation $I^n_{im}(x, y)$ of a world-scene $W^n_{sc}(x, y)$ can be decomposed, but not uniquely, by using the transformations $\mathcal{DEC}_1$, $\mathcal{DEC}_2$,...,$\mathcal{DEC}_m$, yielding the sets of prime components $u^n_{p_1}$, $u^n_{p_2}$,...,$u^n_{p_m}$ and the sets of associations $u^n_{a_1}$, $u^n_{a_2}$,...,$u^n_{a_m}$. That is,

$$\begin{aligned}
\mathcal{DEC}_1[I^n_{im}(x,y)] &= \{u^n_{p_1}, u^n_{a_1}\} \\
\mathcal{DEC}_2[I^n_{im}(x,y)] &= \{u^n_{p_2}, u^n_{a_2}\} \\
\cdot \quad & \cdot \quad \cdot \\
\cdot \quad & \cdot \quad \cdot \\
\cdot \quad & \cdot \quad \cdot \\
\mathcal{DEC}_m[I^n_{im}(x,y)] &= \{u^n_{p_m}, u^n_{a_m}\} .
\end{aligned}$$

By combining the sets of prime components $u^n_{p_k}$ and the sets of associations $u^n_{a_k}$, a new unique decomposition operator $\mathcal{DEC}[I^n_{im}(x,y)] = \{u'^n_p, u'^n_a\}$ can be constructed as given below

$$u'^n_p = u^n_{p_1} \cup u^n_{p_2} \cup \ldots \cup u^n_{p_m}$$

and

$$u'^n_a = u^n_{a_1} \cup u^n_{a_2} \cup \ldots \cup u^n_{a_m} .$$

Without losing the uniqueness property of the decomposition $\mathcal{DEC}$, the set u'^n_p of prime components P^k_{sc} can be reduced by eliminating such spheres $S_r(x_k, y_k)$ that are the subsets of any other prime components

$$u^n_p = u'^n_p \setminus (u^n_{p,r_1} \cup u^n_{p,r_2} \cup \ldots \cup u^n_{p,r_l}) ,$$

$$\begin{aligned}
u^n_{p,r_1} &= \{\bigcup S_r(x,y) : \exists S_{r_1}(x_1,y_1) \wedge r < r_1,\ d((x,y),(x_1,y_1)) < r_1\} \\
u^n_{p,r_2} &= \{\bigcup S_r(x,y) : \exists S_{r_2}(x_2,y_2) \wedge r < r_2,\ d((x,y),(x_2,y_2)) < r_2\} \\
\cdot \quad & \quad \cdot \\
\cdot \quad & \quad \cdot \\
\cdot \quad & \quad \cdot \\
u^n_{p,r_l} &= \{\bigcup S_r(x,y) : \exists S_{r_l}(x_l,y_l) \wedge r < r_l,\ d((x,y),(x_l,y_i)) < r_l\}
\end{aligned}$$

and

$$S_{r_i} \neq S_{r_j} \forall i \neq j .$$

The application of the de Morgan rule yields

$$u^n_p = S_{r_1} \cup S_{r_2} \cup \ldots \cup S_{r_l} , \tag{3.16}$$

where the spheres S_{r_k}'s are totally included in the inner image representation $I^n_{im}(x,y)$. Thus, u^n_p corresponds to the decomposition operator $\mathcal{DEC}$ that also generates set of associations u^n_a's and this decomposition operator can be written as

$$\begin{aligned}\mathcal{DEC}[I^n_{im}(x,y)] &= \{u^n_p\,,\,u^n_a\}\\ &= \{(\bigcup P^k_{sc})\,,\,(\bigcup A^k_{sc})\}\,.\end{aligned}$$

In this particular case of the decomposition, the set u^n_p is equivalent to the set that represents centers of the maximal-sized spheres totally included in $I^n_{im}(x,y)$. Figure 3.7 shows the decomposition of a particular inner-image representation I^n_{im} - a rectangle in $\mathbf{R}^2$.

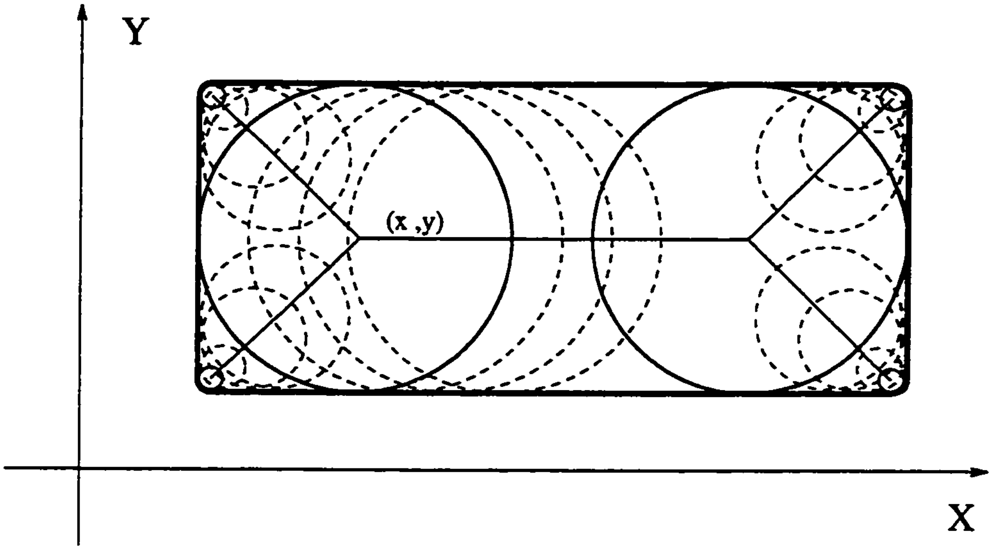


Figure 3.7: Decomposition of a particular inner-image representation I^n_{im} - a rectangle in $\mathbf{R}^2$, performed by the operator $\mathcal{DEC}$.

Let $u^n_{p_r}$, $r > 0$, denote the rth decomposition subset, i.e. the set of the centers of the maximal diameter spheres $S_r(x,y)$ with radii equal to r. This decomposition subset can also be obtained by using morphological operations: *erosion* and *opening*. Assume that the class of subsets of the inner-image representation $I^n_{im}(x,y)$ on which the decomposition is defined is the class of closed sets by considering their complements. Selecting open sets of $I^n_{im}(x,y)$ as the class of decomposable sets excludes the isolated points and lines of zero thickness, which are closed sets and their own subsets (subsets of $u^n_{p_r}$). Further, the open sets of $I^n_{im}(x,y)$ are assumed to be nonempty and to contain no half-space. Under this assumption, it can be proved that a decomposition of the $I^n_{im}(x,y)$ by erosion and opening exists ($u^n_{p_r} \neq \emptyset$), that is,

$$u^n_p = \bigcup u^n_{p_r} = \bigcup [(I^n_{im}(x,y) \ominus S_r) \setminus (I^n_{im}(x,y) \ominus S_r)_{drS_r(x,y)}], \tag{3.17}$$

where $\ominus$ is a *Minkowski subtraction* operator [83] and $drS_r(x,y)$ is a sphere $S_r(x,y)$ of infinitesimal radius *dr*. The *Minkowski subtraction* operator $\ominus$ is defined as

$$I^n_{im}(x,y) \ominus S_r(x,y) = \bigcap \{ I^n_{im}(x - x_1, y - y_1) \; : \; (x_1, y_1) \in S_r(x,y) \} . \tag{3.18}$$

Finally, based on properties of the morphological operator [83], the following properties of the decomposition $\mathcal{DEC}$ operator can be identified:

- *Property 1*

  Decomposition operator $\mathcal{DEC}$ is translation invariant.

- *Property 2*

  Decomposition operator $\mathcal{DEC}$ in the two-dimensional Euclidean space is invariant under the change of scale.

- *Property 3*

  Decomposition operator $\mathcal{DEC}$ is anti-extensive and indepotent.

- *Property 4*

  If u^n_p is a decomposition set of $I^n_{im}(x.y)$ obtained by using the *Minkowski operator* $\ominus$ and the structuring element $S_r(x,y) \in I^n_{im}(x,y)$, then the decomposition $\mathcal{DEC}$ which uses the structuring element $S_r(x - x1, y - y1)$ gives the decomposition set u^{nt}_p that is translation of the set u^n_p by the vector $(-x1, -y1)$.

- *Property 5*

  Members of the set u^n_p are disjoint.

3.5 Discrete-Case Optimization Approach

Section 3.4 the decomposition operator $\mathcal{DEC}$ has been defined as an operator on a continuous set. In order to examine the discrete case of the decomposition, let us first describe the process of image formation in the discrete domain. The names of variables and functions remain the same as in the continuous domain, keeping in mind the discrete context. The function $I^n_{im} : X \times Y \rightarrow G$ is now understood as a discrete bounded function of the inner-image formation and the variables X, Y and G take discrete finite values as given by

$$\begin{aligned} X &= \{x_{min}, x_1, x_2, x_3, \ldots, x_{max}\}, \\ Y &= \{y_{min}, y_1, y_2, y_3, \ldots, y_{max}\}, \\ G &= \{g_{min}, g_1, g_2, g_3, \ldots, g_{max}\}, \end{aligned}$$

Without loss of generality, x_{min}, y_{min} and g_{min} can be set to zero. The coordinates $X \times Y$ represent a square grid and the input transformation $F_{in} : X' \times Y' \times G' \rightarrow X \times Y \times G$ will now map continuous values of the function $W^n_{sc}(x', y')$ into the discrete domain of the function $I_{im}(x, y)$ as

$$F_{in}(x', y', g', x, y) = \{(x', y', g', x, y, g) : (x', y', g') \in W_{sc} \wedge (x, y, g) \in I_{im}\}. \quad (3.19)$$

Assuming the transformation given by (3.19) to be linear, the inner-image function can now be expressed as

$$I_{im}(x, y) = \sum_{x_1=-\infty}^{\infty} \sum_{y_1=-\infty}^{\infty} h(x, y, x_1, y_1) W_{sc}(x_1, y_1) . \quad (3.20)$$

Similarly, imposing the constrain of spatial invariance on (3.19) simplifies (3.20) to

$$I_{im}(x, y) = \sum_{x_1=-\infty}^{\infty} \sum_{y_1=-\infty}^{\infty} h(x, y) W_{sc}(x_1, y_1) . \quad (3.21)$$

The spatially invariant point-spread function $I_{im}(x, y)$ with geometrical transformation of the spatial coordinates becomes

$$I_{im}(x, y) = \sum_{x_1=-\infty}^{\infty} \sum_{y_1=-\infty}^{\infty} h(x, y) \delta(CXYX'(x, y) - x_1, CXYX'(x, y) - y_1) W_{sc}(x_1, y_1) . \quad (3.22)$$

It is to be noted, that the functions CXYX', CXYY' now describe the discrete case of transformation of the coordinate system $X \times Y$ and the rotation angle Θ can have only values that are determined by the discrete grid $X \times Y$.

The previously defined decomposition operator $\mathcal{DEC}$ was derived in a continuous metric space and the issues of accuracy of the representation which are related to the number of prime components used to represent the inner-image $I_{im}(x, y)$ was not addressed. It is well known that a complete description of any continuous function requires infinite number of the discrete values of the function, which obviously is not practical. Therefore, describing a continuous function $I_{im}(x,y)$ by a countable and finite set of prime components makes the result of the decomposition $\mathcal{DEC}$ an approximation of the real continuous function $I_{im}(x,y)$. On the other hand, the decomposition of the discrete function $I_{im}(x,y)$ by using a finite set of prime components should be done, so that the set fully represents the discrete function $I_{im}(x,y)$. However, the previous postulates concerning the rotation invariance properties of prime components $S_r(x,y)$ are not valid in the discrete case of function $I_{im}(x,y)$ and they have to be reconsidered. This difficulty is overcome by defining *semi-rotational invariant* 2-dimensional objects (functions). For this purpose, the following definition is first stated.

Definition 3.3:
A 2-dimensional object (function) $\mathcal{F}(x,y)$ is said to have a *semi rotational invariance of degree $\mathcal{K}$ in the plane $X \times Y$* iff

$$\begin{aligned} \exists\Theta_{min}, \Theta_{min} &= inf(\Theta) \in (0;2\pi), \text{ such that} \\ \text{ROT}[\mathcal{F}(x,y)(\Theta)] &= \mathcal{F}(x,y), \end{aligned} \tag{3.23}$$

where the rotation is performed in the plane $X \times Y$ and the degree of rotation $\mathcal{K}$ is equal

$$\mathcal{K} = 2\pi/\Theta_{min}$$

In addition, if $\forall\Theta, \Theta \in (0;2\pi)$ and (3.23) holds, then the object (function) $\mathcal{F}(x,y)$ is *rotational invariant* in the plane $X \times Y$.

According to this definition, a sphere in a continuous 2-dimensional metric space represents a *rotational invariant* object (function). Based on Definition 3.3, many models of prime components can be constructed. Figure 3.8 shows a few examples of prime components having different degrees of rotation, that can be used as decomposition elements. It is to be noted that in a discrete metric space with a given type of the grid $X \times Y$ (the domain of the inner image representation function $I_{im}(x,y)$), only some of the prime components shown in Figure 3.8 can be employed. For example, if the inner image representation $I^n_{im}(x,y)$ is mapped onto the rectangular grid $X \times Y$, only the cases (d) and (f) are applicable.

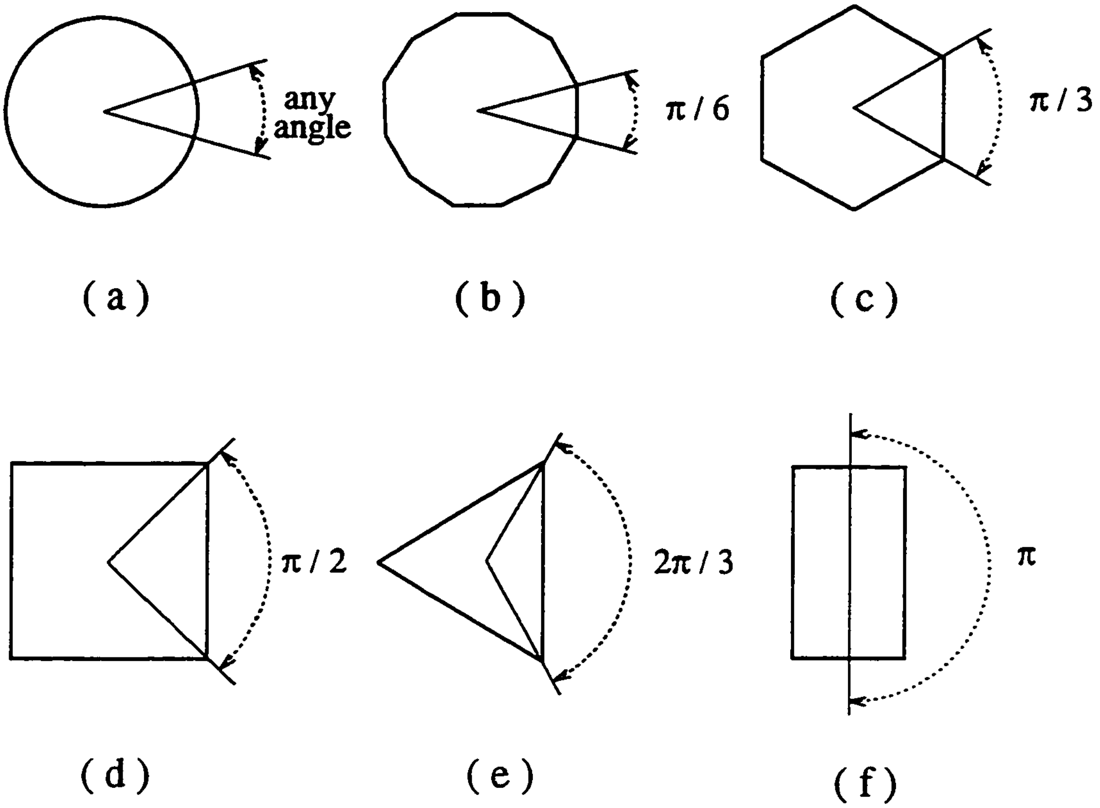


Figure 3.8: Examples of rotational invariant prime components.

Rotational invariant prime components of different degrees have different topological and statistical characteristics. Consequently, the decomposition transformation which utilizes prime components with different values of the parameter $\mathcal{K}$ will have different properties. The decomposition that uses semi rotation invariant prime components having a degree $\mathcal{K}_1$ will have statistical properties more desirable than the one having a degree $\mathcal{K}_2$, when $\mathcal{K}_2 > \mathcal{K}_1$. On the other hand, the decomposition algorithms that use rotational invariant prime components having a degree $\mathcal{K} << 1$ is computationally very expensive.

Considerations as to the choice of prime components will now be examined under the assumption that the prime component P_{sc} refers to the polygon belonging to the two-dimensional discrete space $X \times Y$, that is, the discrete function $P_{sc}(x,y) \subset X \times Y$. In this case, the set $X \times Y$ describes the rectangular discrete grid of the inner image representation. One of the main statistical criteria which determines the resulting shape of the decomposition element is the probability of its occurrence in the sequence of real world-scenes $W_{sc}(x,y)$. This criterion can be formulated as follows.

The probability of occurrence of a region $\mathcal{R}_k(x,y)$ selected as a prime component $P_{sc}(x,y)$ having a given area **A**, i.e., $\mathcal{A}(\mathcal{R}_k(x,y))$ =**A**, should be as large as possible. That is,

$$P(P_{sc}(x,y) \subseteq W^n_{sc}(x,y)) =$$

$$sup\{\, P(\mathcal{R}_k(x,y) \subseteq W^n_{sc}(x,y))\,\} \;\wedge\; \mathcal{A}(\mathcal{R}_k(x,y)) = \mathbf{A}\,. \tag{3.24}$$

It is to be noted, that the probability of occurrence of the prime component $P_{sc}(x,y)$ is a monotonically decreasing function of its area $\mathcal{A}$. Thus, the minimum-area prime component $P_{sc}(x,y)$ used as the decomposition element would result in a set $\mathcal{U}_\mathcal{P}$ having the largest number of elements, making the transformation process by the next stages of a image understanding system very difficult. Thus, the area of the desirable prime component should be a compromise between the number of decomposed elements in a given world-scene $W_{sc}(x,y)$ and the total number of elements in the set $\mathcal{U}_\mathcal{P}$.

The analytical approach of finding the shape of an optimal prime component is based on the theoretical analysis of the probability of occurrence of the given prime component in the sequence of scenes, the set $\mathcal{U}_\mathcal{W}$. A theoretical evaluation of the probability of occurrence of the given prime component P^k_{sc} in the set $\mathcal{U}_\mathcal{W}$ is very difficult, since there is no strict theoretical description of a world-scene W_{sc} and there do not exist simple explicit formulae that count the number of occurrences of a given pattern in the two-dimensional array. Therefore, all the functions that can be derived are only approximations of particular cases. In order to simplify the the analysis, the following assumptions are made.

(a) A prime component is considered to be a two-dimensional rectangle $p(x-x_0, y-y_0) \in W^n_{sc}(x,y)$ (discrete function $p(x,y) \rightarrow G$), $x = 0,1,2,\ldots,x_{m-1}$, $y = 0,1,2,\ldots,y_{n-1}$, coordinates x_0 and y_0 describing the position of the prime component in $W^n_{sc}(x,y)$

(b) A prime component can belong only to a specific subset $sy(x,y)$ (strip) of $W_{sc}^{n}(x,y)$ defined as

$$sy(x,y) = \{W_{sc}^{n}(x,y) \, : \, y \in (y_0, y_1, \ldots, y_{n-1})\}.$$

These two assumptions are schematically presented in Figure 3.9.

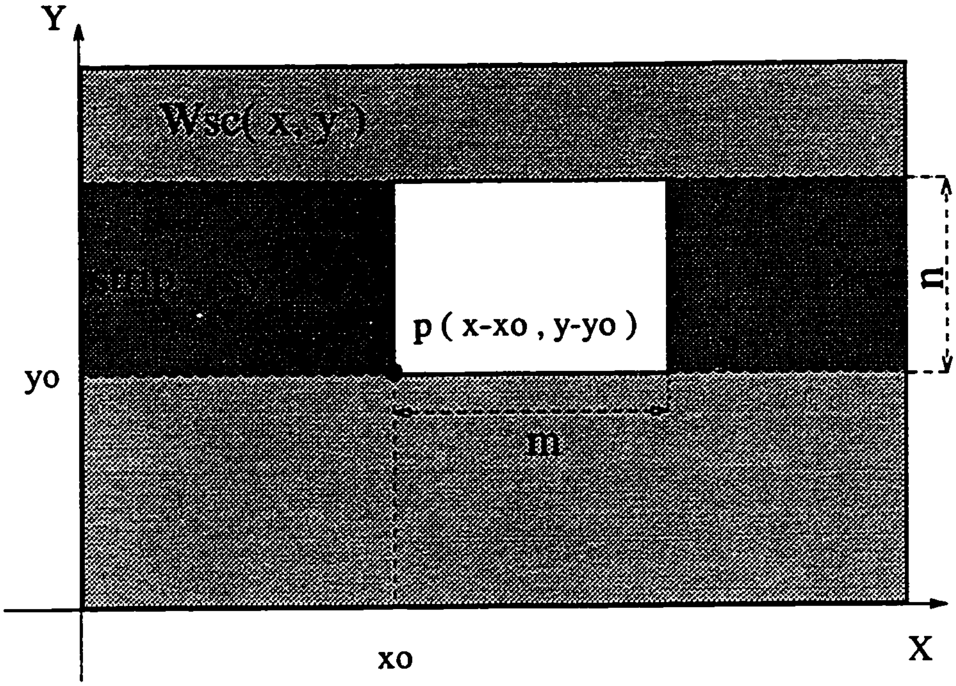


Figure 3.9: Two-dimensional prime component (white rectangle) in the strip (dark strip) along the x-axis.

The two-dimensional prime component denoted by the function $p(x,y)$ can be expressed as a matrix $PAT_{n\times m}$, whose entries are discrete values of the gray level $pg \in G$ (x represents columns, y represents rows), as given by

$$p(x,y) \leftrightarrow PAT_{n\times m} = \begin{bmatrix} pg_{11} & pg_{12} & \cdots & pg_{1m} \\ pg_{21} & pg_{22} & \cdots & pg_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ pg_{n1} & pg_{n2} & \cdots & pg_{nm} \end{bmatrix} = [P\hat{G}_1, P\hat{G}_2, \ldots, P\hat{G}_m], \qquad (3.25)$$

where $\hat{PG}_j$ ($j = 1,2,\ldots,m$) represents a vector comprising the elements of the jth column of PAT_{nxm}. Similarly, a matrix $STR_{nx\hat{m}}$ can be formulated for the strip sy(x,y), as given below

$$sy(x,y) \leftrightarrow STR_{nx\hat{m}} = \begin{bmatrix} sg_{11} & sg_{12} & \cdots & sg_{1\hat{m}} \\ sg_{21} & sg_{22} & \cdots & sg_{2\hat{m}} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{n\hat{m}} \end{bmatrix} = [\hat{SG}_1, \hat{SG}_2, \ldots, \hat{SG}_{\hat{m}}], \qquad (3.26)$$

where $\hat{SG}_j$ ($j = 1,2,\ldots,\hat{m}$) represents a vector comprising the elements of the jth column of $STR_{nx\hat{m}}$. The variable n represents the width of the strip $sy(x,y)$ along the y-axis and the variable $\hat{m}$ is its length along the x-axis. Every vector $\hat{PG}_j$ or $\hat{SG}_i$ belongs to the set $\mathcal{G}$ having a finite number of elements G_{EL}, i.e.,

$$\hat{PG}_n,\ \hat{SG}_n \in \mathcal{G} \text{ and } G_{EL} = \mathcal{POW}(\mathcal{G}), \qquad (3.27)$$

where $\mathcal{POW}(.)$ represents the number of elements in the set (.). If the maximum number of gray levels g is equal to g_{max}, then for the strip $sy(x,y)$ the number of elements G_{EL} of the set $\mathcal{G}$ can be expressed as

$$G_{EL} = \mathcal{POW}(\mathcal{G}) = (g_{max})^n . \qquad (3.28)$$

Proposition II:

The number $SY_{num}(\hat{m})$ of strips $sy(x,y) \in W^n_{sc}(x,y,)$ which do not contain the prime component $p(x,y) \in W^n_{sc}(x,y,)$ is given by $SY_{num}(\hat{m})$

$$SY_{num}(\hat{m}) = \mathcal{Z}^{-1}\{\mathcal{SY}_{num}(z)\}, \qquad (3.29)$$

where $\mathcal{SY}_{num}(z)$ can be expressed as

$$\mathcal{SY}_{num}(z) = \frac{z\,P_z}{1 + (z - G_{EL})P_z}, \qquad (3.30)$$

with

$$P_z = \sum_{i=0}^{m}(1 - U(\sum_{j=0}^{n}\sum_{k=i}^{m} \mid p(k,j) - p(k-i,j) \mid))z^{(m-j)}, \qquad (3.31)$$

and $U(.)$ represents a unit step function.

Proof:
To prove this proposition, we first define $SY_A(\hat{m})$ to be the number of different strips of the length $\hat{m}$ that contains the prime component $p(x,y)$ at the right and only at the right end of the strip.

Let us now consider any of the $SY_{num}(\hat{m})$ strips having $\hat{m}$ vector elements $\hat{SG}_i$ $(i = 1, 2, \ldots, \hat{m})$ that do not contain the prime component $p(x,y)$ as a subset. Append to such an $STR_{nx\hat{m}}$ at its right end any one of the vectors $\hat{SG}_i$ from $\mathcal{G}$. In this way we obtain $G_{EL}SY_{num}(\hat{m})$ distinct strips of length $\hat{m}+1$. Each of the appended strips may or may not contain the prime component. If it does contain the prime component, it must appear at the rightmost position of the strip Therefore, the expression

$$G_{EL}SY_{num}(\hat{m}) = SY_{num}(\hat{m}+1) + SY_A(\hat{m}+1) \tag{3.32}$$

holds for all $\hat{m} \geq 0$. Taking one-sided $\mathcal{Z}$-transform, we obtain

$$(z - G_{EL})\mathcal{SY}_{num}(z) + z\mathcal{SY}_A(z) = z. \tag{3.33}$$

Next, let us consider any one of the $sy(x,y)$ strips having $\hat{m}$ vector elements $\hat{SG}_i$ that does not contain the prime component $p(x,y)$. Now, append the prime component $p(x,y)$ at its right end. Suppose that the first (from the left) appearance of the prime component $p(x,y)$ in the strip $sy(x,y)$ has its rightmost vector $\hat{PG}_m$ at $x = \hat{m}+r$. Then,

$$0 < r \leq m.$$

Furthermore, since the prime component $p(x,y)$ will necessarily appear in the last m positions of the appended strip, the first r vectors $\hat{PG}_i$ $(i = 1, 2, \ldots, r)$ of prime component $p(x,y)$ have to be equal to the last r vectors $\hat{PG}_i$ $(i = m - n + 1, \ldots, m)$ of $p(x,y)$. The number of strips $sy(x,y)$ having $\hat{m}+r$ vector elements $\hat{SG}_i$ is then counted at $SY_A(\hat{m}+r)$. Conversely, for any given strip counted at $SY_A(\hat{m}+r)$, there is a unique way to extend it to a strip of length $\hat{m}+m$ in which the last m vectors equal to the prime component

$p(x,y)$. Consequently, for $\hat{m} > 0$, we obtain

$$SY_{num}(\hat{m}) = \sum_{r=1}^{m} P(r) SY_A(\hat{m} + r)\,, \tag{3.34}$$

where $P(r)$ is given by

$$P(r) = (1 - U(\sum_{j=0}^{n} \sum_{k=r}^{m} \mid p(k,j) - p(k-r,j) \mid)). \tag{3.35}$$

Taking one-sided $\mathcal{Z}$-transform of (3.34), we obtain

$$\mathcal{SY}_{num}(z) = z\, P_z\, \mathcal{SY}_A(z)\,. \tag{3.36}$$

Combining together (3.33) and (3.36), we can easily obtain (3.29). □

Thus, the probability of finding the prime component $p(x,y)$ in the set $sy(x,y)$, where each value $g_{xy} = sy(x,y)$ is treated as an independent variable with equal uniform distribution, can be written as

$$P(p(x,y) \in sy(x,y)) = 1 - \frac{\mathcal{Z}^{-1}\{\frac{z P_z}{1+(z-G_{EL})P_z}\}}{(g_{max})^{((m)(n))}} \tag{3.37}$$

One can note here that for a fixed m and n, the probability as given above depends only on the polynomial P_z which reflects the autocorrelation (periodicity) of the prime component $p(x,y)$. For two prime components $p_1(x,y)$ and $p_2(x,y)$, the probability of occurrence is higher for the one having more periodicity along the X axis.

Similar analysis can be carried out for the prime component belonging to the strip in the direction of the Y axis, yielding an expression similar to (3.37), with only appropriate changes in notations.

It can be concluded, that the probability of occurrence of a square prime component $p(x,y)$ (assuming independence and uniform distribution of the variable g_{xy}) strongly depends on polynomial P_z obtained from autocorrelation function performed along the diagonal of the prime component itself. Several examples of prime components $p(x,y)$ with different "strengths" of autocorrelation periodicity along the X or Y axis are presented in Figure 3.10, with the polygon i as an optimal choice for a decomposition element.

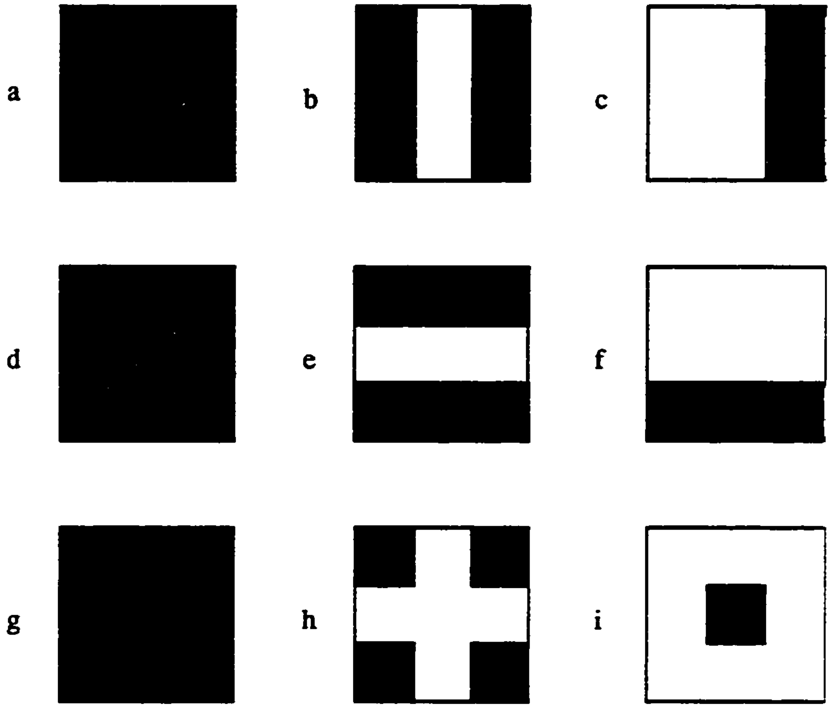


Figure 3.10: Prime components with different symmetrical properties and autocorrelation periodicity along X- or Y-axis. (a),(d),(g) Prime components without periodicity along any axis. (b) Prime component with strong periodicity along X-axis. (c) Prime component with weak periodicity along X-axis. (e) Prime component with strong periodicity along Y-axis. (f) Prime component with weak periodicity along Y-axis. (h),(i) Prime components with strong periodicity along both X- and Y-axes.

An analysis of (3.37) indicates, that probability of occurrence of a prime component is higher for the polygon with strong periodic properties along its axes. Combining this observation with the fact that most of the real world-scenes contain physical objects, one can conclude that an optimal shape of the prime component in the discrete metric space should be the rotational invariant polygon defined in Section 3.4 with the degree of rotation as small as possible. For example, considering the shapes of the prime components of Figure 3.10 in the rectangular grid $X \times Y$, one should rather chose the square polygon i as a prime component instead of the rectangular polygon f.

The algorithm used to perform square prime component decomposition of the objects contained in a world-scene is now formally presented using pseudo codes.

Square Prime Component Decomposition Algorithm

```
start
{
Clear prime_component_array
Initialize number_of_pixels to number of scene_pixels
Set current_pixel_index to zero
}
while ( "current_pixel_index smaller than number_of_pixels" )
    {
    while ( "current_pixel_index belongs to an object" )
        {
        Set size_of_prime_component to one
        Set prime_component to current_pixel_index
        while ( "boundary_area belongs to an object" )
            {
            Increase size_of_prime_component
            Fill boundary_area around prime_component
            }
        Store prime_component into prime_component_array
        }
    Increase current_pixel_index
    }
end
```

3.6 Examples and Simulation Results

In this section, the results of the previous section are evaluated based on the analysis of the simulation results applied to image sequences. In order to facilitate the experiment, special test sequences of binary images and gray scale images are generated. The total number of images used in the experiment is 1200 in the case of the binary test image sequence and 2500 in the case of the gray level sequence. A test sequence $\mathcal{U}_{\mathcal{W}}$ represents varieties of transformations that could be found in the real physical world: translation, reflection, magnification and rotation. A few examples of images from the binary training sequence are presented in Figure 3.11. Some examples of images from the gray-level test sequence are shown in Figure 3.12.

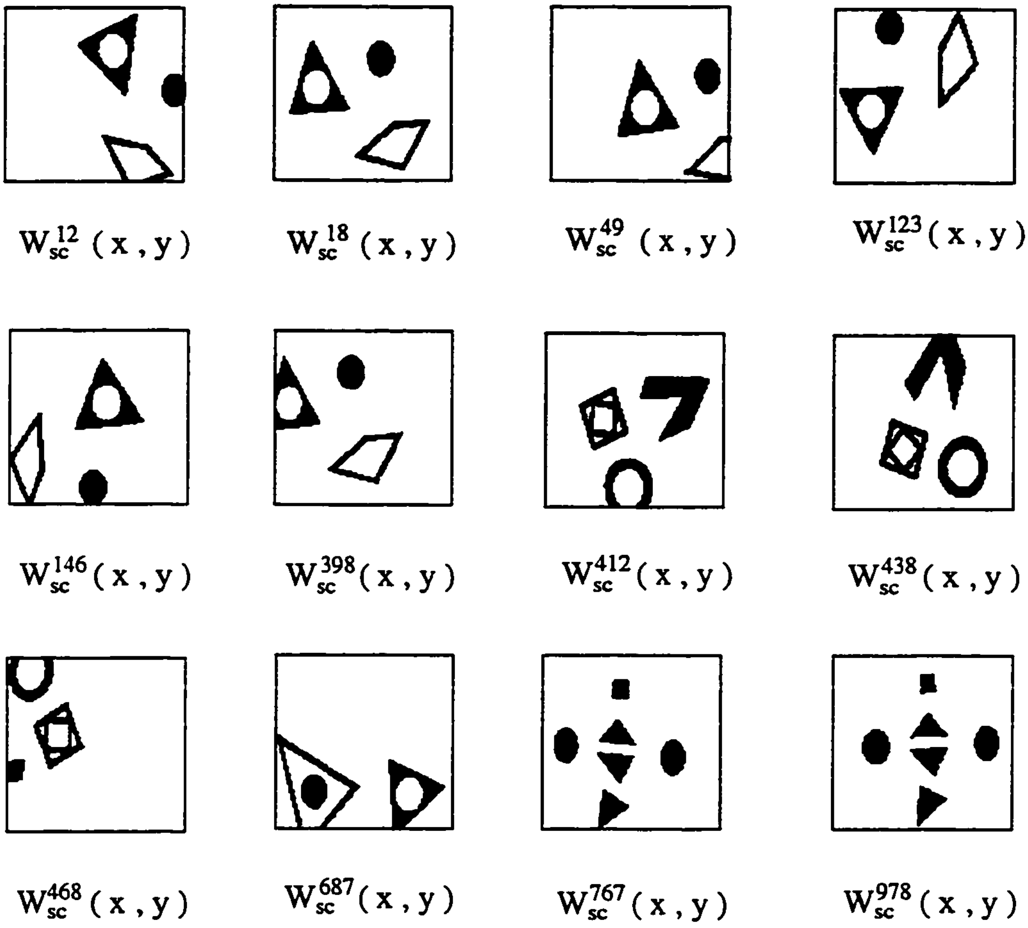
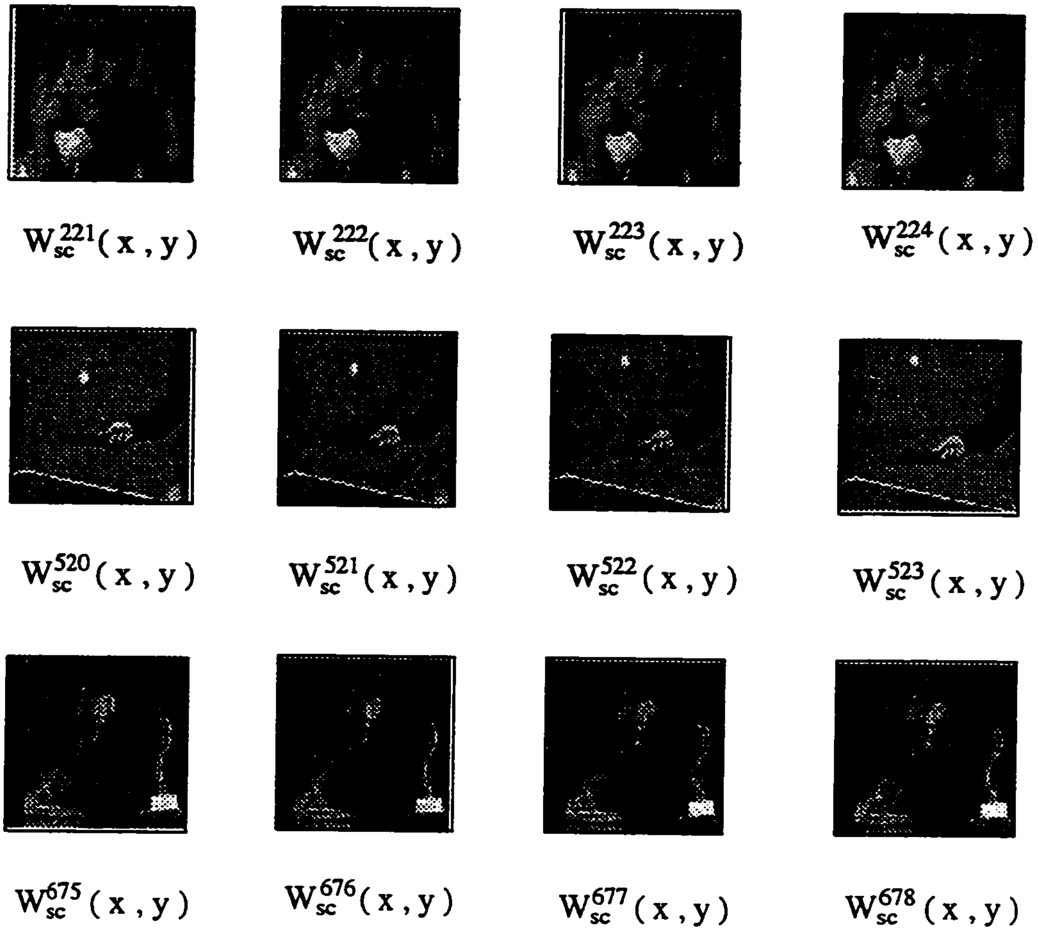


Figure 3.11: A few examples of images I_{im}^{n} from the test binary image sequence.



$W_{sc}^{221}(x,y)$ $W_{sc}^{222}(x,y)$ $W_{sc}^{223}(x,y)$ $W_{sc}^{224}(x,y)$

$W_{sc}^{520}(x,y)$ $W_{sc}^{521}(x,y)$ $W_{sc}^{522}(x,y)$ $W_{sc}^{523}(x,y)$

$W_{sc}^{675}(x,y)$ $W_{sc}^{676}(x,y)$ $W_{sc}^{677}(x,y)$ $W_{sc}^{678}(x,y)$

Figure 3.12: A few examples of images I_{im}^{n} from the test gray level image sequence.

The test sequences used in our experiments are composed of the images representing continuous motion of the physical objects. This representation simulates in a natural way the perception of the images performed by a human visual system.

The first step in the simulation process was to investigate the correctness of the hypothesis of rotation invariance of the prime components. In order to evaluate this hypothesis, the training binary and gray level sequences have been used. The method of searching for the shape of the optimal prime component $P_{sc}(x,y)$ is to generate random sets of polygons (possibly disjoint) representing the prime components and to observe the statistical results of the scene decomposition performed by these generated prime components. This procedure performed on a large set of the image sequences should give an indication about the topological properties of the prime components itself.

The direct construction of the random polygons representing prime components (sets $\mathcal{R}_k(x, y)$) is not simple, and in practice, the hierarchical construction method is used [2], [12], [25], [41], [85], [91]. The hierarchical construction of the prime components is based on the approach that the structure of polygons representing the prime components can be expressed in graph-theoretical trees such that one prime component is represented by one tree. The terminal nodes (leaves) of the tree correspond to prime components called *the zeroth level prime components* and the non-terminal nodes correspond to various clusters of elements called *the general-level prime components.* In this type of representation of the prime components, the zeroth level prime components correspond to pixels and the general-level prime component is a set of any number of zeroth level prime components or/and general level prime components. Every general-level prime component has to have a decedent prime component(s) (children) and every zeroth level prime component has to have its parent(s). Figure 3.13 presents a test frame $I_{im}(x, y)$ from the binary sequence showing examples of the zeroth and general-level (level 1 to level 6) prime components.

The algorithm analyzes sequentially the input image, creates general-level prime components by generating clusters of the tree nodes. In this stage of simulation, creating general-level prime components is arbitrary with no restriction on the shape (topological properties) of the general-level prime component. The input image is then decomposed using these general-level prime components created, and the procedure to create general-level prime components and the decomposition process is repeated. The only criterion which stops analysis of the current image is its full decomposition. The analysis process of the gray-level training sequence (see Figure 3.12) is similar to that of the binary training sequence. The only difference is that each frame from the test gray-level sequence is first partitioned into 16 binary frames corresponding each of the 16 gray levels of the input image. In this manner, the two training sequences have been processed sequentially with a total of 3700 images. The results of the analysis for the binary test sequence of images are presented in Figure 3.14.

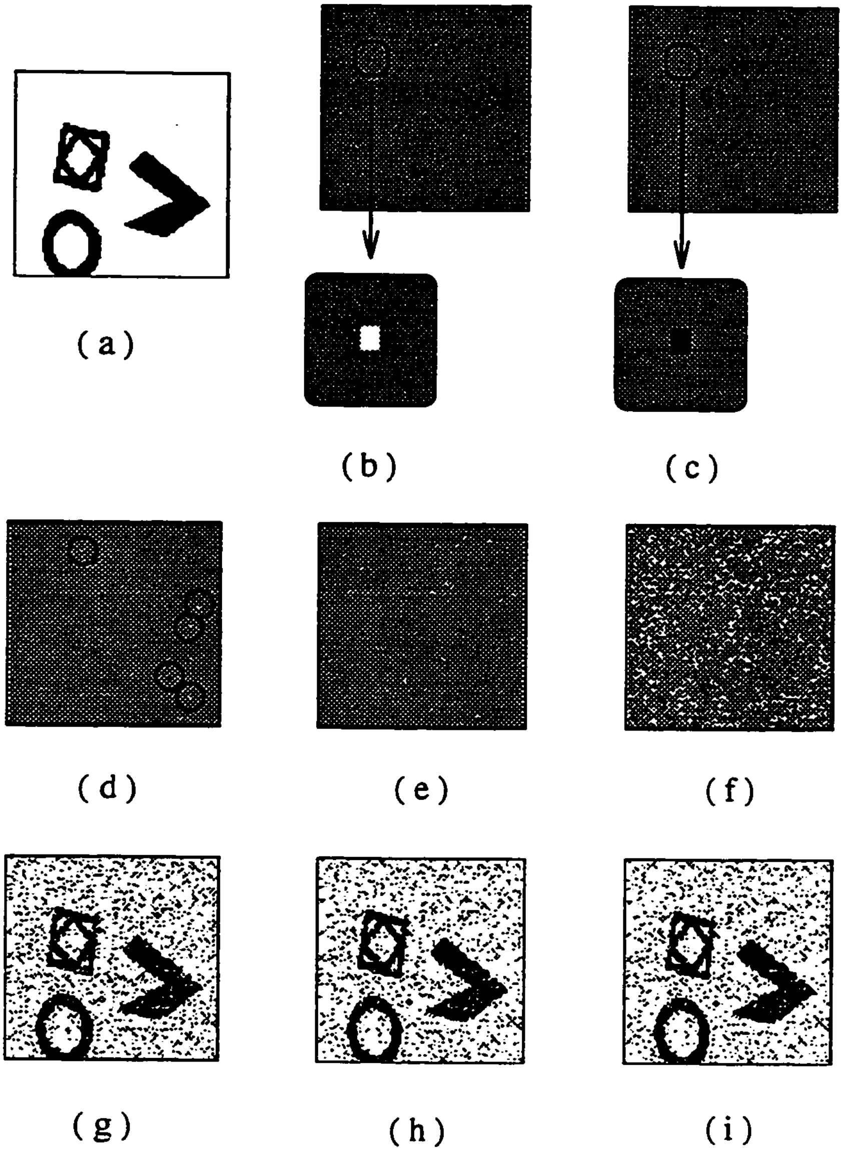


Figure 3.13: An example of a test binary image $I_{im}(x, y)$ together with its decomposition by prime components. (a) Test binary image, (b), (c) zeroth level prime components, (d)-(i) Prime components with levels 1, 2, 3, 4, 5 and 6, respectively.

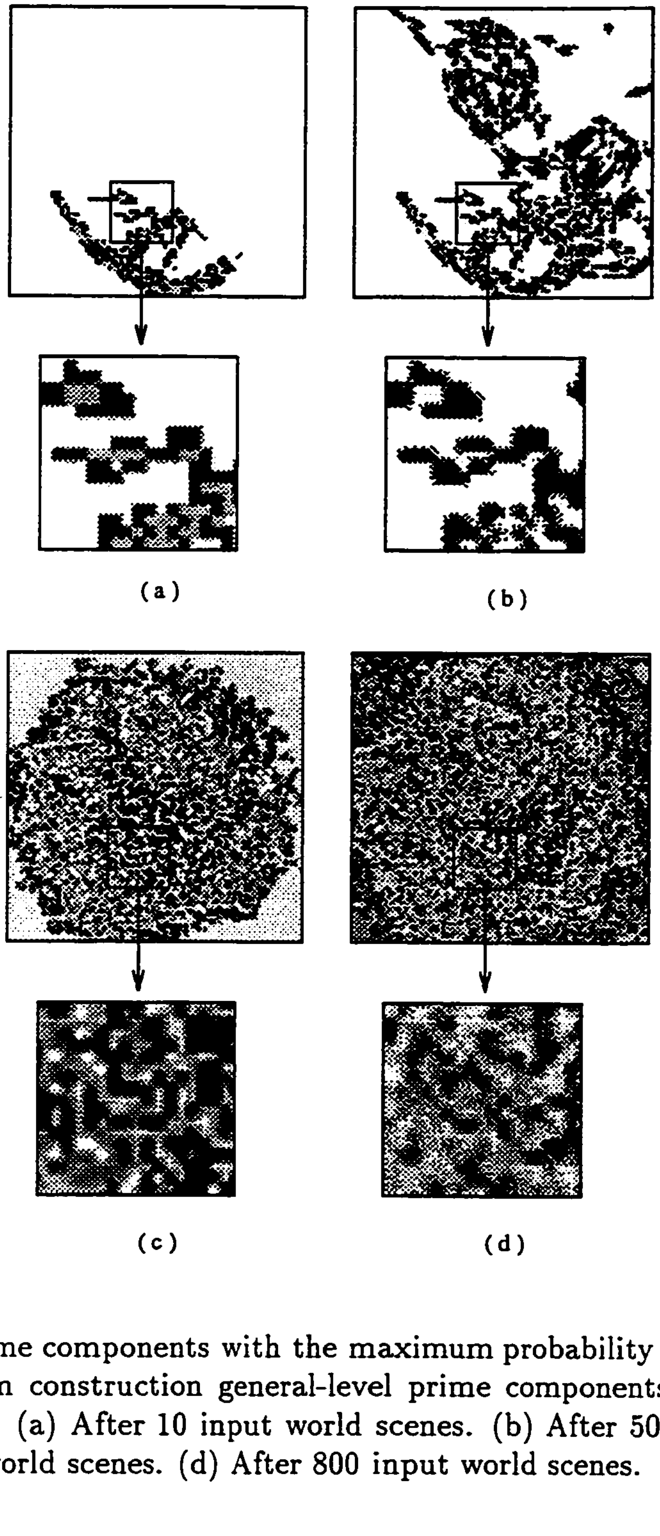


Figure 3.14: The prime components with the maximum probability of occurrence in the set $\mathcal{U}_W$ using random construction general-level prime components in the case of the binary test sequence. (a) After 10 input world scenes. (b) After 50 input world scenes. (c) After 200 input world scenes. (d) After 800 input world scenes.

Careful observation of Figure 3.14 indicates two important facts:

- *Symmetry of the prime components.*

The symmetry of the prime components means that any particular prime component P_{sc}^{k} has a tendency to form symmetric layout (shape). The symmetry of the prime components is observed by noticing the fact that as the number of images processed increases, the shape of generated prime components with the maximum probability of occurrence converges to become circular.

- *Grouping property of the prime components* - clustering of the prime components.

Grouping property of the prime components means that for an arbitrarily chosen set of prime components, the highest probability in the set $\mathcal{U}_{\mathcal{W}}$ occurs for the prime components composed of polygons whose decedents in the tree structure are the neighbours (joined prime components). The grouping property of the clusters has been noticed by many researches who have investigated biological visual systems, such as Julesz' works on clusters [45] and Beck's work on grouping properties of random patterns [13]. The practical experiments performed on prime component generation, fully conform the hypothesis that prime components with symmetrical layout (ideally of circular shape) should be used as decomposition elements.

3.7 Summary

In this chapter, a detailed theoretical analysis for the construction of the decomposition element - prime component has been presented. First, basic relations (associations) among the prime components have been studied. Second, the basic relations among the prime components have been described in terms of relations found in the physical world. Based on these relations, a set of properties which characterize directly the decomposition operator and indirectly the topological properties of the prime components itself, have been obtained. Those properties have allowed to develop an optimized decomposition operator in the continuous metric space, which utilizes symmetrical circular polygons (prime components) as decomposition elements. Further, an optimization of the decomposition elements in the discrete domain has also been performed.

A mathematical analysis for an optimal shape of the prime component has been performed for semi-rotational invariant polygons having degrees of rotation of $\pi/2$ and π. An expression that allows to identify the usefulness of the decomposition element by the analysis of its autocorrelation function performed in the direction of the principal axes of these polygons has been derived. An analysis of the autocorrelation function indicates that the semi-rotational invariant polygon having a degree of $\pi/2$ is a better decomposition element than the one having a degree of π. In addition, it has been shown, that the shape of the optimal prime components should converge to the polygon having a degree tending to zero, i.e., circles.

The mathematical analysis has also been verified by a statistical simulation which shows that the probability of the randomly generated polygons representing decomposition elements (prime components) is the highest for a polygon with a symmetrical layout (circular shape).

In the next chapter, the discrete metric space decomposition elements developed in this chapter will be used to propose an efficient shape extraction scheme and to produce an intermediate object representation for an IUS.

Chapter 4

SHAPE REPRESENTATION

The notion of shape is one of the most fundamental concepts to understand and to process visual information. Many complex processing tasks such as object identification, feature extraction or segmentation is based on the analysis of the shape of the objects in a scene. The properties of the objects found in typical world-scenes suggest highly structured methods of shape description, in which the complex objects can be described starting out from some simple shape primitives and group them into levels of relational structures, typically characterized by graphs.

One of the most important steps for the shape representation process is the extraction of shape elements. There are many models for the extraction of shape elements. One of the most promising approaches is based on the skeletonization concept and is commonly known as skeleton or medial axis transform [59]. However, despite the elegant original definition in the continuous domain, the extension of the skeleton concept to the discrete domain, in particular in the raster world, turns out to be surprisingly tedious. The difficulties come from the finite number of the raster points in the discrete domain and from the particular properties of the skeleton transform in which small changes of the object's boundary change the object's skeleton drastically, introducing new branches and removing old ones. In order to overcome the difficulties of the shape representation in the discrete metric space, in this chapter the model of shape extraction based on the prime component decomposition presented in Chapter 3, is proposed [54].

The proposed approach of extraction of shape elements based on the polygonal decomposition is derived using a modified definition of the skeleton in the discrete metric space. This extraction scheme combines the boundary description model with a modified skeletonization process performed by the square prime components. This modification allows to utilize efficiently the prime component decomposition technique developed in Chapter 3 and it minimizes the errors in the uncertainty of the description which might be critical in the processing performed in the subsequent stages of an IUS.

In order to further improve the shape representation model and make it more suitable for practical implementation, the concept of the shape equalization scheme based on Fourier descriptor and nonlinear interpolation is introduced [55], [56] . This scheme avoids the deficiencies of the discrete medial axis transform. These deficiencies may include noise sensitivity, the description errors of the diagonal objects and objects having complex edges that rapidly change their curvature, and the description errors caused by a small sampling frequency. The scheme also reduces the distortions and artifacts of the input transformation and maximizes the utilization of the system resources.

The scheme of the shape representation begin with a brief description in Section 4.1 on the general issues related to the shape analysis. It also explains the purpose and importance of the shape extraction and shape equalization process in an IUS. In this section, a brief review of the current major techniques used in the domain of the shape extraction such as Voronoi diagrams and morphological operators is carried out. The detailed analysis concerning shape extraction model is presented in Section 4.2. Some of the available operators for the boundary detection are considered. Next, the decomposition process of the object's interior using the prime component decomposition technique is analyzed. The shape equalization concept is described in the Section 4.3. This concept is developed by analyzing the model of the extraction of the shape elements in the context of the description errors, computational complexity, and its practical implementation. In Section 4.4, the proposed models for the extraction of the shape element and shape equalization are compared with the two main techniques of the shape extraction based on the Voronoi diagrams [3], [29] and morphological operators [83]. Simulation and comparisons are performed on two different hardware platforms for sequential and parallel versions of the algorithms, respectively.

4.1 Shape Representation

The goal of shape representation is to encode the object's 2-dimensional representation in a manner that captures the object's most salient features. Those features are then used to classify and identify the objects in an input world-scene. In the domain of image analysis in an IUS, the process of shape representation should posses the following properties.

(1) *Uniqueness* - The algorithm used to represent shape elements in an input world-scene should return a single solution.

(2) *Invariance* - It is necessary that that the shape descriptor is invariant under geometrical transformations of the shape such as rotation, translation, and scaling. That is, the different varieties of the shape instances should not create problems in the identification process.

(3) *Noise tolerance* - The descriptor should be tolerant of artifacts introduced by noise. There is little hope that noise and its effects can ever be eliminated completely at the lowest level of image processing. Commonly , it is very difficult to decide whether an artifact is actually due to noise or it represents a salient shape feature. It seems that many of these ambiguities can be resolved only at a higher level of scene analysis, possibly even at the highest level of visual semantics, during the proper interpretation of the world-scene.

(4) *Rich local support* - The shape descriptor should be insensitive to the modification of the scene that occurs apart from the area of focusing. For example, for two overlapped objects, the descriptor should enable a local match between the new compound object and each of the isolated components.

(5) *Semantic support* - Shape description should enable to facilitate a semantic of the description. That is, the way the shape descriptor combines and assimilates the new shape elements should be easily mapped into a semantic information of the shape descriptor itself.

There are many different models that have been proposed for the shape descriptors based on the description of the boundary of the object (e.g. Fourier descriptors, chain codes, moments analysis) [94]. However, the description of the object's boundary is not effective enough for carrying out most of the tasks performed in an IUS, such as object identification, segmentation or scene matching. The decomposition of the object's interior constitutes another approach of the shape description [75]. Some of the interior decomposition methods include partition of the interior of the object into smaller polygons (e.g. quad-trees, skeletons, pyramidal decompositions) [62], while others are based on the description of the deformation from a primordial shapes such as circles, ellipses, squares etc (e.g geon decompositions) [47]. Many methods are based on grammatical approach [69] and use semantic description of the decomposed elements. Unfortunately , none of the shape decomposition models devised so far sufficiently fulfills all above prerequisites. Many commonly used models cannot provide a "rich local support" and encode rather global characteristics of the shape. In addition, almost all of the existing methods require complicated algorithms, tremendous hardware demands and produce huge amount of information which is very difficult to process by the succeeding high-level processing stage of an IUS.

In order to introduce the basic concepts of the shape representation, in the next few subsections. an overview of the frequently employed methods to describe geometrical shapes are presented.

4.1.1 Moment Descriptors

Moment invariant transforms were introduced by Hu [38] as a method of image recognition, which has the desirable property of being invariant under such variations of the image content as shift, scaling, and rotation. In the continuous metric space, the moment of order $(p+q)$ is defined by the relation

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x,y) dx dy \quad p,q = 0,1,2,\ldots \tag{4.1}$$

where $f(x,y)$ represents an object in a world-scene. The central moments are expressed as

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y) dx dy \quad , \tag{4.2}$$

where the center of gravity $(\bar{x}, \bar{y})$ is given by $(m_{10}/m_{00}, m_{01}/m_{00})$, in which m_{00} is referred to as the total function mass. A set of normalized central moments can be also defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \ , \ \gamma = \frac{p+q}{2} + 1 \ . \tag{4.3}$$

The central moments are used to define a set of seven moments that are invariant to linear transformations. The set is given by

$$\begin{align}
M_1 &= \mu_{20} + \mu_{02} \tag{4.4}\\
M_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \tag{4.5}\\
M_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})_2 \tag{4.6}\\
M_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})_2 \tag{4.7}\\
M_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) \nonumber\\
&\quad +(3\mu_{21} - \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \tag{4.8}\\
M_6 &= (\mu_{21} - \mu_{01})((\mu_{30} + \mu_{21})^2 - (\mu_{21} + \mu_{03})^2) \nonumber\\
&\quad +4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{30}) \tag{4.9}\\
M_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) \nonumber\\
&\quad -(\mu_{30} - 3\mu_{12})(\mu_{12} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \ . \tag{4.10}
\end{align}$$

The normalized central moments η_{pq} are also invariant to scale changes. Assuming that α is the scale change, it follows that $x' = \alpha x$, $y' = \alpha y$, and

$$\eta'_{pq} = \frac{(\mu')^{pq}}{(\mu'_{00})^{(p+q)/2+1}} = \frac{\mu_{pq}\alpha^{p+q+2}}{(\alpha^2\mu_{00})^{(p+q)/2+1}} = \frac{\mu_{pq}}{(\mu_{00})^{(p+g)/2+1}} = \eta_{pq} \ ,$$

where

$$\mu'_{00} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x', y')dxdy = \alpha^2 \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x, y)dxdy = \alpha^2\mu_{00} \ .$$

The moments M_1 through M_6 are also invariant under reflection , while M_7 changes sign.

In the discrete metric space, the regular moments m_{pq}, and the central moments μ_{pq} are defined, respectively, as

$$\begin{align}
m_{pq} &= \sum_x \sum_y x^p y^q f(x, y) \tag{4.11}\\
\mu_{pq} &= \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \, . \tag{4.12}
\end{align}$$

It should be noted, that the moment descriptors can be applied to both binary and gray-level images. In the latter case, however, the interpretation of the results is even more delicate than for binary silhouettes. According to the definition of m_{pq} , the set of moment invariants is a global shape descriptor. Consequently, the moment descriptors are useful only for isolated objects. If this method is applied to scenes containing overlapped or abutted objects, it must be preceded by a reliable separation procedure.

4.1.2 Fourier Descriptors

Fourier description is another very popular model of shape description. Let us consider a clockwise-oriented simple closed curve γ having a parametric representation $Z(l)$, $Z(l) = x(l) + i(y(l))$, where l $(0 \leq l \leq L)$ is the arc length [63]. The angular direction $\frac{\partial x(l)}{\partial y(l)}$ of the curve γ can be normalized to be in the interval $[0,2\pi]$, yielding a periodic function $\Theta(t)$ that is invariant under translation, rotation, and changes of the perimeter L:

$$\Theta(t) = \frac{\partial x(\frac{Lt}{2\pi})}{\partial y(\frac{Lt}{2\pi})}. \tag{4.13}$$

The function $\Theta(t)$ can be expanded into a Fourier series as

$$\Theta(t) = A_0 + \sum_{k=1}^{\infty}(a_k \cos(kt) + b_k \sin(kt)) \ . \tag{4.14}$$

The coefficients A_0, a_k and b_k are known as Fourier descriptors of the curve γ having the parametric representation $Z(l)$, and they are given by

$$A_0 = \frac{1}{2\pi}\int_0^{2\pi} \Theta(t)dt \tag{4.15}$$

$$a_n = \frac{1}{\pi}\int_0^{2\pi} \Theta(t)\cos(nt)dt \tag{4.16}$$

$$b_n = \frac{1}{\pi}\int_0^{2\pi} \Theta(t)\sin(nt)dt \ . \tag{4.17}$$

The function $Z(l)$ can be easily reconstructed using the Reconstruction Theorem [79] given below,

$$Z(l) = x(l) + i(y(l)) = Z(0) + \frac{L}{2\pi}\int_0^{2\pi} \exp\{i(\Theta(t))\}dt \ . \tag{4.18}$$

The Fourier descriptors can be also derived for a polygonal curve α. A polygonal curve α is a closed curve with m vertices, $V_0, \cdots, V_{m-1}$ and the edge (V_{i-1}, V_i) with length Δl_i is a straight line. The change in the angular direction at vertex V_i is $\Delta\phi_i$ and $L = \sum_{i=1}^{m} \Delta l_i$. It can be verified that [33]

$$\Theta(l) = \sum_{i=1}^{m} \Delta\phi_i \quad for \quad \sum_{i=1}^{m} \Delta l_i \leq \ (lL/2\pi) \ \leq \sum_{i=1}^{k+1} \Delta l_i \tag{4.19}$$

and

$$\Theta(l) = 0 \quad for \ \ 0 \leq l \leq \Delta l_1 \ . \tag{4.20}$$

Expanding $\Theta(l)$, we obtain the same equation as (4.14).

Since $\Theta(t)$ is a step function, the integrals (4.15), (4.16) and (4.17) can be expressed as

$$A_0 = -\pi - \frac{1}{L}\sum_{k=1}^{m} l_k \Delta\phi_k \tag{4.21}$$

$$a_n = \frac{-1}{n\pi}\sum_{k=1}^{m} \Delta\phi_k \sin\frac{2\pi n l_k}{L} \tag{4.22}$$

$$b_n = \frac{-1}{n\pi}\sum_{k=1}^{m} \Delta\phi_k \cos\frac{2\pi n l_k}{L} \tag{4.23}$$

where

$$l_k = \sum_{i=1}^{k} \Delta l_k \ \ .$$

Shift, scale and rotational invariant descriptors $s(n)$ can be obtained as [21]

$$s(n) = (\sqrt{[|a_n|^2 + |b_n|^2]})(\sqrt{[|a_1|^2 + |b_1|^2]}) \ . \tag{4.24}$$

4.1.3 Polygonal Chain

A simple way to describe the boundary of an object is to approximate it by a set of piecewise linear fits l_i's, that consist of eight standardized line segments. The code of a contour is then the chain V of length K [49]

$$V = a_1 a_2 a_3 \cdots a_K \ \ , \tag{4.25}$$

where each link a_i is an integer between 0 and 7 oriented in the direction $(\pi/4)a_i$ (as measured counter-clockwise from the x- axis of an x-y coordinate system) and of length

1 or $\sqrt{2}$ depending on whether a_i is even or odd, respectively. The vector representation of the link a_i, using phasor notation, is given by

$$l_i = (1 - \frac{(\sqrt{2}-1)}{2}(1-(-1)^{a_i}))e^{i(\frac{\pi}{4}a_i)} \ . \tag{4.26}$$

The construction of a chain code starts out from the detection of the points that connect the segments of the straight lines for the boundary approximation. In many cases this is not very easily achievable, since majority of the objects found in a typical world-scene are characterized by smooth boundaries and slowly changing curvature. This makes the computation of the curvature difficult and complex. In addition, one has to deal with even a more challenging issue of invariance and uniqueness of a straight line approximations. Consequently , each attempt to provide a shape descriptor which is based on such an approximation must carefully analyze the consequences of missing or accidentally misplaced straight line segments. A good example of an object recognition system which is based on polygonal approximation is given in [49].

4.1.4 Pyramids, Quadtrees, Ribbons and Higher Order Curves

The *pyramid* models of shape description are based on the idea that the shape information is stored according to a hierarchy of increasing levels of resolution. In the Laplacian pyramid [60], the hierarchy is built by sampling the shape (possibly the gray-level) with Laplacian operators of many scales. This approach of representing objects makes the analysis of world-scenes very complicated and practically very difficult. An IUS would have to process many dependent planes of different resolutions representing the same physical objects, which is highly redundant and inefficient.

Quadtrees [62] are obtained by successive subdivision of the shape in quadrants, subquadrants, and so on until the process arrives at the lowest resolution level, typically single pixels. The information is then organized as a tree structure. The root of the tree corresponds to the rectangular pixel block, enclosing the entire shape. Its four 'children' denote the four quadrants of the rectangle. The modification of the quadtree techniques are binary and ternary tree decomposition methods. Analogous to the Laplacian pyramid, there are many problems in employing a quadtree technique as a reliable shape extraction method. One of the most critical problems in this type of shape representation is the construction of "fixed" division regions. First, it is very difficult to propose practical algorithms to perform the divisions of the world-scene into regions. Second, it is impossible to have the dividing lines that correspond with the real boundaries of the physical objects in the world-scene, which is one of the most fundamental requirements of an IUS.

The pyramid and quadtree concepts introduce important concepts of hierarchical rank ordering of shape information, since tree-like data structures support object recognition methods which first analyze the global shape before focusing on the details.

Ribbons and *higher order curves* [75] are the shape description methods in which the straight line has been extended to a more flexible and general set of approximating elements. The richer set of primitives, however , increases the problem of a unique representation which makes it difficult to ensure the uniqness of the representation and to have an easy semantic description. Moreover , even modest variations of the contour can dramatically affect the segmentation process, e.g., it can force a selection of two second-order polynomials instead of one straight line segment. In the case of ribbon shape representation, one of the most fundamental problems is a reliable and invariant segmentation of a compound object into simple subparts that can be described by ribbons. Moreover , even in the case of relatively simple objects, often several equivalent choices do exist, which results in non-unique transformation of the object into its shape descriptors.

4.1.5 Skeletons

The idea of transforming an image to a skeleton was mentioned by Blum [36] in 1961. Since then, a mathematical theory for the skeletons of images has been developed. The skeleton SK(X) of a continuous [18], [48] (discrete [58], [59]) object X viewed as a subset of $\mathbf{R}^2$ ($\mathbf{Z}^2$), is defined as the set of the centers of the maximal disk totally included in X. Hence, a maximal disk must touch the boundary of the object X at least at two different points. Each point of the skeleton is associated with a radius r that corresponds to the maximal disk. Therefore, a function $skf(z)$ can be defined, whose domain is SK(X) and which takes the values in **R**,

$$z \in SK(X) \;\; \rightarrow \;\; skf(z) \in \mathbf{R}\,. \tag{4.27}$$

This function is called the ***skeleton function*** or ***quench function.***

Some very important properties of skeletons and advantages of the skeletonization techniques can be summarized as follows.

(1) Skeletonization transforms the a binary two-dimensional object into a one-dimensional planar graph. Thus, all the methods of graph theory, e.g., graph traversal [26], can be readily applied to the shape analysis techniques.

(2) Each segment of the skeleton naturally represents a local symmetry axis. Since detection of symmetries is an important issue in recognition both of artificial and natural shapes, the skeleton can be used as a valuable tool in this domain. Moreover, the radius of each maximal inscribed disk indicates the contribution of the associated boundary to the entire shape.

(3) The skeleton inherently embodies proximity information; therefore, the handling of adjacency and neighborhood relations is effectively supported. Consequently , the skeleton can be employed to assist the grouping processes and to solve obstacle avoidance and path planning problems.

(4) Every segment of the skeleton represents the result of a process of self-organization, that only reflects the shape features of a limited neighborhood. Thus, even drastic changes of the global shape, such as the overlapping of two objects, usually leave most sections of the skeleton unmodified.

(5) Frequently, the characteristic branching points of the skeleton indicates a candidate for the separation of the original shape into subparts. Consequently, the skeleton can also serve as a tool for shape decomposition, and thus it can serve as a base for a semantic shape description that utilizes pure geometrical polygons.

4.2 Shape Extraction

As described in Section 4.1, there is no ideal method to describe and to extract shape elements. In order to derive a flexible and robust technique of the extraction of shape elements, we will try to combine the best features of the shape description techniques [86] already developed.

It is obvious that the shape extraction method to be developed should consider both the boundary and the interior of the object in a scene at the same time. Consequently, the proposed technique of extraction of the shape elements is based on the modification of the two shape description models: skeletonization [18] and piecewise chain coding [49]. In order to improve the traditional skeletonization technique, the skeletonization process is performed using prime component decomposition yielding a much more effective shape extraction scheme. In fact, as it will be shown in this section, the polygonal decomposition of a world-scene performed by the maximum size square prime components will be the principal part of the shape extraction process. The proposed shape decomposition scheme also simplifies the preliminary transformations in the high-level processing stage of an IUS by more explicit characterization of the shapes of the objects in a scene.

The first step in the extraction of the shape elements of the object is to determine the object's boundary (edges). Consider the image representation $I(x,y)$ of a real world-scene. The edge points can be thought of as pixel locations of abrupt gray-level change. Consequently, for the image representation $I(x,y)$, its partial derivative assumes a local maximum in the direction of the edge. Therefore, using the polar coordinate system, the expression,

$$\frac{\partial I(r,\theta)}{\partial r} = I_r(r,\theta) = I_x(x,y)\frac{\partial x}{\partial r} + I_y(x,y)\frac{\partial y}{\partial r}, \tag{4.28}$$

should have its maximum value at the edge point of the object for a given θ. The symbol $I(r,\theta)$ denotes the image representation of the world-scene expressed in the polar coordinate system and the partial derivatives $I_x(x,y)$ and $I_y(x,y)$ are given by

$$I_x(x,y) = \frac{\partial I(x,y)}{\partial x} \tag{4.29}$$

and

$$I_y(x,y) = \frac{\partial I(x,y)}{\partial y}. \tag{4.30}$$

The maximum value of (4.28) is obtained when $(\partial/\partial\theta)(\partial I(r,\theta)/\partial r) = 0$. This yields

$$-I_x(x,y)\sin\theta_e + I_y(x,y)\cos\theta_e = 0, \tag{4.31}$$

$$\theta_e = \tan^{-1}\left(\frac{I_x(x,y)}{I_y(x,y)}\right), \tag{4.32}$$

and

$$\left(\frac{\partial I_x(r,\theta)}{\partial r}\right)_{\max} = \sqrt{(I_x(x,y))^2 + (I_y(x,y))^2}. \tag{4.33}$$

The angle θ_e represents the direction of the edge. Therefore, the natural technique of edge detection is to measure two gradient fields G_1 and G_2 of the image $I(x,y)$ in two orthogonal directions ϕ_1 and ϕ_2 and to use the expression given by (4.32) for the angle of the edge point.

Let $\mathbf{M}_1$ and $\mathbf{M}_2$ denote two k×k squared mask (gradient operators). For a binary discrete image representation of $I(x,y)$, the gradient fields G_1 and G_2 in the two orthogonal directions ϕ_1 and ϕ_2 can be calculated as

$$\begin{aligned} G_1(m,n) &= \sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{M}_1(i,j)I(i+m,j+n) \\ &= I(m,n) \circledast \mathbf{M}_1(-m,-n), \end{aligned} \tag{4.34}$$

and

$$\begin{aligned} G_2(m,n) &= \sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{M}_2(i,j)I(i+m,j+n) \\ &= I(m,n) \circledast \mathbf{M}_2(-m,-n), \end{aligned} \tag{4.35}$$

where the symbol $\circledast$ denotes the discrete convolution operation between the image representation and gradient operators. After the two orthogonal gradient fields G_1 and G_2 have been constructed, the gradient vector magnitude field $g(m,n)$ and direction field $\theta_e(m,n)$ are given by

$$g(m,n) = \sqrt{(G_1(m,n))^2 + (G_2(m,n))^2} \tag{4.36}$$

and

$$\theta_e(m,n) = \tan^{-1}\left(\frac{G_2(m,n)}{G_1(m,n)}\right) . \tag{4.37}$$

By thresholding the vector magnitude field $g(m,n)$ with the threshold value T_h, the edge map E of the object for tracing its boundary can be obtained as

$$E(m,n) = \begin{cases} 1, & \text{if } g(m,n) \geq T_h \\ 0, & \text{otherwise .} \end{cases} \tag{4.38}$$

The binary masks $\mathbf{M}_1$ and $\mathbf{M}_2$ should be chosen in order to achieve the necessary sensitivity of the edge detector. There are two suitable gradient operators $\mathbf{M}_1$ and $\mathbf{M}_2$ that are very convenient for edge detector, the Prewitt operator and the Sobel operator [64]. For the Prewitt operator, the masks $\mathbf{M}_1$ and $\mathbf{M}_2$ are defined as

$$\mathbf{Mp}_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \ \mathbf{Mp}_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

and for the Sobel operator, they are given by

$$\mathbf{Ms}_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \ \mathbf{Ms}_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

The decomposition of the object's interior is the next step in the shape extraction process. After constructing the map of the objects' boundaries, we can represent the objects' interior in the image using intrinsic coordinate system. Let the object map $O_n(x,y)$ of one particular nth object in the image representation $I(x,y)$ be expressed as

$$O_n(x,y) = \begin{cases} 1, & \text{if point } (x,y) \text{ belongs to the object } O_n \\ 0, & \text{otherwise} \end{cases} \tag{4.39}$$

In the intrinsic coordinate description system, every point (x,y) which belongs to the object $O_n \in I(x,y)$ (i.e., $O_n(x,y) \neq 0$) having its own edge map E_n in the image representation $I(x,y)$ is specified by giving its distance to the nearest boundary point (edge) of the object $O_n(x,y)$. The transformation of the image using intrinsic coordinate system yields

$$Is(x,y) = \begin{cases} \inf_{(m,n)}(d((x,y),(m,n))) \ \wedge \ E_n(m,n) \neq 0 \\ 0, \ \text{otherwise} \ , \end{cases} \tag{4.40}$$

where $Is(x,y)$ is the image representation in the intrinsic coordinate system and the operator $d((x,y),(m,n))$ denotes the Euclidean distance between the point (x,y) and the edge point (m,n). We can reduce the redundancy of the intrinsic coordinate system by choosing the local maxima of the $Is(x,y)$ and constructing a skeleton map $s_n(x,y)$ of the object $O_n(x,y)$, as given by

$$s_n(x,y) = \begin{cases} Is(x,y), & \text{if } \frac{\partial Is(x,y)}{\partial x} = 0 \vee \frac{\partial Is(x,y)}{\partial y} = 0 \\ 0, & \text{otherwise} \ . \end{cases} \tag{4.41}$$

Because the skeleton map $s_n(x,y)$ preserves all of the information about the object $O_n(x,y)$, it is possible to recover the original shape of the object O_n, given its skeleton $s_n(x,y)$, by taking the union of the circular neighborhoods having radii $s_n(x,y)$ and centered at points (x,y).

The process of constructing an object's skeleton s_n (skeletonization process) is conceptually very similar to the polygonal decomposition of the object O_n with circular shape polygons. It has already been shown in the Chapter 3, that polygonal decomposition with rotational invariant circular shape polygons (prime components) gives statistically the best decomposition of the interior of the object in the scene. It has also been shown, that polygonal decomposition of an image can be carried out by other semi-rotational invariant polygons, like rectangles, triangles or squares. Because of the availability of simple and effective algorithms for implementation, especially interesting is the square prime component decomposition. By modifying the distance operator d in (4.40), the skeletonization process can be performed with square prime components as

$$Is(x,y) = \begin{cases} \sup(\mathcal{SQR}(m,n)), & \text{if } O_n(m,n) \neq 0 \\ 0, & \text{otherwise} \ , \end{cases} \tag{4.42}$$

where the symbol $\sup(\mathcal{SQR}(m,n))$ denotes the size of the maximum-sized squared prime component belonging to the object O_n and centered at (m,n). We can now construct the full skeleton $S_n(x,y)$ of the object O_n composed of the edge map E_n and the skeleton s_n

obtained from square prime component decomposition (prime skeleton) of the object O_n, as

$$S_n(x,y) = \begin{cases} 1, & \text{if } E_n(m,n) \neq 0 \text{ and } s_n(m,n) \neq 0 \\ 0, & \text{otherwise .} \end{cases} , \tag{4.43}$$

The set of points (x,y) for which $S_n(x,y) = 1$ is known as the point skeleton of the object O_n.

The boundary decomposition of the input image is performed using a modified Sobel operator. The Prewitt operator gives less discontinuities than does the Sobel operator, but fails for less detailed edges. Therefore, the masks $\mathbf{M}_1$ and $\mathbf{M}_2$ of the Sobel operator are modified as

$$\mathbf{M}_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1.5 & 0 & 1.5 \\ -1 & 0 & 1 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} -1 & -1.5 & -1 \\ 0 & 0 & 0 \\ 1 & 1.5 & 1 \end{bmatrix}.$$

Next, the skeletonization process is performed using square prime component decomposition. In order to proceed further in the extraction of the shape elements, the full skeleton of the test object is filtered by a 2-D low-pass filter and a skeletonization process is performed once more. The low-pass filtering process removes the possible discontinuities from the skeleton description and makes the final extraction easier to perform. The full skeleton point S_n is composed of the boundary description of the object E_n and its prime skeleton s_n. The full skeleton of the object is then transformed directly into $\mathcal{OS}$ (shape-descriptor), a set of extracted shape elements, as given by

$$\mathcal{OS} = \{l_1, l_2, \ldots, l_j, \ldots, l_{z-1}, l_z\}. \tag{4.44}$$

The shape-descriptor $\mathcal{OS}$ contains z shape elements (curves) l_j, $j = 1, \ldots, z$, of the form

$$l_j = \{x1_j, y1_j, x2_j, y2_j\}, \ j = 1, \ldots, z \ , \tag{4.45}$$

where $(x1_j, y1_j)$ is the coordinate of the starting point of the curve l_j and $(x2_j, y2_j)$ describes the coordinate of the end point of the curve l_j. The difficulties in the description and further interpretation of complicated curves, practically reduces the type of possible curve models to simple straight lines. Therefore, four parameters are used to describe the basic shape element. The use of straight lines as a basic shape element does not restrict the accuracy of the representation, since we can always increase the number z of the extracted shape elements. Consequently, the shape-descriptor $\mathcal{OS}$ is obtained by a 2-D interpolation of the full object skeleton $S_n(x,y)$ by straight lines. Set of parameters $(x1_j, y1_j)$ and $(x2_j, y2_j)$ of the shape element l_j, $j = 1, \ldots, z$, satisfies the equation

$$A_j x + B_j y = C_j \; . \tag{4.46}$$

Therefore, finding the unknown parameters $(x1_j, y1_j)$ and $(x2_j, y2_j)$ can be considered as finding the coefficients A_j, B_j and C_j in (4.46). Thus, the interpolation process consists of modifying the coefficients A_j, B_j and C_j adaptively in such a way that distance d between curve described by (4.46) and the set of m skeleton points SP_j ($SP_j = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, $j = 1, \ldots, z$) is less than d_{min}, i.e.,

$$\forall_{k=1,\ldots,m} \; d = \frac{Ax_k + By_k - C}{\sqrt{A^2 + b^2 + C^2}} \leq d_{min}. \tag{4.47}$$

Every set SP_j denotes one extracted shape element l_j, $j = 1, \ldots, z$. The process of interpolation is performed as long as all the skeleton points (x, y) have been transformed into the shape elements l_j. The minimum distance d_{min} is chosen appropriately to meet the required accuracy of the shape extraction process. The adaptive algorithm for the determination of the coefficients A_j, B_j and C_j combines the skeleton points SP_j into a chain given as

$$CH = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\} \tag{4.48}$$

under the constraint that the Euclidean distance d_k between the point (x_k, y_k) and the point (x_{k+1}, y_{k+1}) is less than the searching distance d_s given by

$$d_k = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} < d_s. \tag{4.49}$$

From the constrain given by (4.49), it can be concluded that the searching distance d_s determines the minimum number z_{min} of the extracted shape elements l_j, $j = 1, \ldots, z$. Each extracted shape element represents a specific case of the general level prime component linked to the object of the world-scene by an appropriate equivalence relation $\mathbf{R_L}$, the points of the world-scene which belong to one and only one shape element. The spatial relations between the extracted shape elements can be treated as associations between general-level prime components (see Figure 3.1).

4.3 Shape Equalization

As it has been already mentioned, the minimum distance d_{min} together with spatial sampling frequency determines the accuracy of the boundary description. An example of poor accuracy of the representation of an object's boundary caused by a too small spatial sampling frequency for rectangular sampling grid can be seen in Figure 4.1. This type of distortion manifests itself as a large contour distortion and an overall loss in the resolution of small details of an object in a world-scene, particularly at corners and edges of the objects which rapidly change their curvature. The poor accuracy of the representation of the object's boundary is one of the major factors resulting in errors in shape extraction process and consequently it causes segmentation errors in the subsequent stages of an image understanding system. In addition, poor accuracy of the representation together with a rectangular sampling grid, makes the detection and extraction of the diagonal shape elements, lines and rotated objects a very difficult problem. However, increasing the description accuracy by increasing the spatial sampling frequency, using a polar sampling grid or minimizing the distance d_{min} is not a good idea. The computational complexity and time required for the extraction of shape elements increases exponentially with an increase of the sampling frequency or with a decrease in the distance d_{min}. The polar sampling grid creates other types of difficulties.

Without changing the distance d_{min} and the sampling frequency, an increase in accuracy can be achieved by an equalization of the detected object's boundary (shape equalization) through a filtering operation performed on parametric representation of the objects' boundaries. The object's boundary (edges) can be obtained using the modified Sobel operator described in Section 4.2. Then, a very simple filtering operation would increase the accuracy of the description significantly. This filtering operation can be performed with a little increase in computational complexity of the algorithm.

The accuracy of description in the continuous metric space can be defined as the norm

$$\mathcal{A}_c = \|Z_o - Z_s\|_c = \int_l \mid Z_o(l) - Z_s(l) \mid \, dl \, , \tag{4.50}$$

where Z_o and Z_s designate the parametric contour representation of the original and the sampled version of the elements, respectively. For discrete representation, the accuracy of description can be defined as

$$\mathcal{A}_{d1} = \|Z_o - Z_r\|_{d1} = \sum_{n=0}^{N-1} \mid Z_o(n) - Z_r(n) \mid , \tag{4.51}$$

or as

$$\mathcal{A}_{d2} = \|Z_o - Z_r\|_{d2} = \sup_n \mid Z_o(n) - Z_r(n) \mid . \tag{4.52}$$

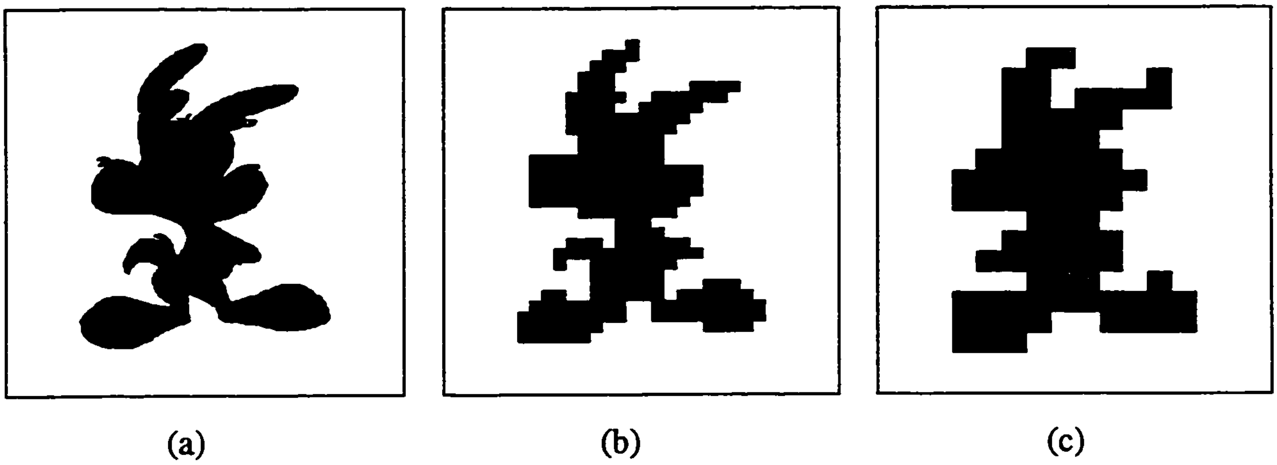


Figure 4.1: An example of the binary image. (b) and (c) Sampled and thresholded version of the image in (a) at two different sampling frequencies.

Let

$$\begin{aligned} Z(n) &= x(n) + jy(n) = r(n)e^{j\alpha(n)} , \\ & n = 0, 1, ..., N-1 \text{ and } 0 < \alpha \leq 2\pi , \end{aligned} \tag{4.53}$$

represent the discrete parametric contour representation of the curve Z (space domain representation). It is obvious that for a polygon, the curve Z is closed, and the sum of the differences $\Delta\alpha(n)$ defined as

$$\Delta\alpha(n) = \alpha(n) - \alpha(n-1) , \, n = 0, 1, ..., N-1 \tag{4.54}$$

is equal to 2π, that is,

$$\sum_{n=0}^{N-1} \Delta\alpha(n) = \sum_{n=0}^{N-1} \alpha(n) - \alpha(n-1) = 2\pi . \tag{4.55}$$

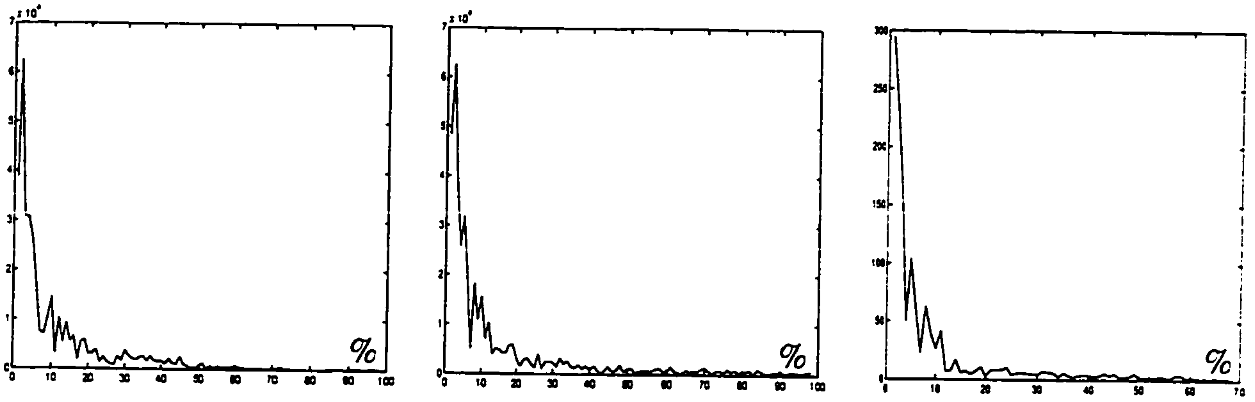
For a closed boundary, the function given by (4.53) is periodic with a period N. Therefore, the DFT of a parametrically represented closed contour $Z(n)$ (Eqn. (4.53)) can be expressed as

$$fd(k) = \sum_{n=0}^{N-1} Z(n)e^{-j\frac{2\pi kn}{N}}. \tag{4.56}$$

The complex coefficients $fd(k)$ known as *Fourier descriptors* (*spectrum*) of the boundary $Z(n)$ have already been described in Section 4.1.2. The boundary $Z(n)$ can be recovered using the inverse DFT as given by

$$Z(n) = \frac{1}{N}\sum_{k=0}^{N-1} fd(k)e^{j\frac{2\pi kn}{N}}. \tag{4.57}$$

The Fourier descriptor representation $fd(k)$ directly reflects the changes in the curvature of the boundary $Z(n)$. This means, that the spectrum $fd(k)$ of the boundary $Z(n)$ contains higher frequencies for complex shaped objects for which the boundary curve changes its direction (angle) rapidly. Conversely, the spectrum of simple and smooth curves will mostly contain low-frequency components. Typical spectrums of some object boundary are shown in Figure 4.2.



(a) (b) (c)

Figure 4.2: The typical Fourier spectrums of parametric curve representations. (a) Spectrum of the boundary curve of the shape shown in Figure 4.1(b). (b) Spectrum of the boundary curve of the shape shown in Figure 4.1(c). (c) Spectrum of the extracted contour from the gray-level image shown in Figure 4.5.

The spectrum $fd(k)$ of the boundary $Z(n)$ can be transformed using a 1-D filter as

$$fd_y(k) = \mathcal{F}\{fd(k)\} \ , \ k = 0, 1, ..., N-1 \, , \tag{4.58}$$

where the transformation $\mathcal{F}\{.\}$ performs a shape filtering of the boundary $Z(n)$ by transforming its spectrum fd into fd_y. For linear filtering, the transformation $\mathcal{F}\{.\}$ reduces itself to a multiplication in the frequency domain, as given by

$$fd_y(k) = Hs(k)fd(k) \ , \ k = 0, 1, ..., N-1 \, , \tag{4.59}$$

where $Hs(k)$ is the sampled frequency characteristic of the filter. In the space domain representation, the transformation given by (4.58) can be understood as a circular convolution of the boundary $Z(n)$ with the impulse response of the filter $hs(n)$. Thus, the equalization process consists of filtering the Fourier spectrum $fd(k)$ of the parametric curve representation. The key point in this procedure is to transform the parametrically represented contour into the Fourier description domain, and then to recover the shape of the object by applying the traditional filtering. The amplitude characteristic of the filter transfer function is determined in a learning phase using a sequence of real world-scenes and their undersampled versions. For every object Z_o^j and the filtered output of its undersampled version Z_r^j, the algorithm computes the norm $\mathcal{A}_d^j$, using (4.51) or (4.52). After all norms (for every pairs Z_o and Z_r from the training sequence) have been computed, the average norm $\overline{\mathcal{A}_d}$ is determined as

$$\overline{\mathcal{A}_d} = \frac{1}{M}\sum_{j=1}^{M} \mathcal{A}_d^j. \tag{4.60}$$

The upper limit M denotes the total number of word-scenes in the training phase. The amplitude characteristic of the filter is modified and the average norm $\overline{\mathcal{A}_d}$ is calculated repeatedly until there is an appreciable improvement in $\overline{\mathcal{A}_d}$.

Generally, a binary image contains a set of polygons each having its own boundary representation Z_{r_i} $(i = 1, 2, \ldots)$ and the difference function $\Delta\alpha_r(n)$ for its sampled version. For the set of difference function $\Delta\alpha_r(n)$, we can consider their statistical properties as

$$m_{kq} = E\{(\boldsymbol{\Delta}\alpha_{\mathrm{r}})^q\} = \sum(\Delta\alpha_r)^q f_k(\Delta\alpha_r) \, , \tag{4.61}$$

and

$$\mu_{kq} = E\{(\boldsymbol{\Delta}\alpha_{\mathrm{r}} - \eta)^q\} = \sum(\Delta\alpha_r - \eta)^q f_k(\Delta\alpha_r) \, , \tag{4.62}$$

where $f_k(\Delta\alpha_r)$ is the probability density function of the k-th contour, and m_{kq} and μ_{kq} are statistical q-th order moments of the k-th contour. The symbol $E\{.\}$, denotes the expectation operator of the random variable. The polygons in a typical binary image have typical statistical properties (moments as defined above), which allows the use of this information to predict the phase characteristic of the filter. The coefficients' angles in the transfer function of the filter are determined in a similar adaptive manner as described above. The only difference being that instead of minimizing the average norm $\overline{\mathcal{A}_d}$, the differences between the moments of the same order are taken into account. The moment values m_{kq} and μ_{kq} are determined by estimating of the probability density function $f_k(\Delta\alpha_r)$ from a scene sequence. Consequently, the filter's coefficients are determined during the initial learning phase using a set of real-world scenes. After the amplitude and phase characteristics of the filter are determined for the training sequence, the algorithm is capable of modifying the filter characteristics on-line as a result of processing of real world-scenes. However, the changes at this stage are only minor and they are in the frequency domain.

The shape equalization operation (filtering operation) that consists of 1-D multiplication in frequency domain, is applied right after combining the skeleton points SP_j into a chain given by

$$CH = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\} \tag{4.63}$$

and before finding the coefficients A_j, B_j and C_j for every extracted shape element l_j, $j = 1, \ldots, z$.

As discussed above, the shape equalization process has two phases - the learning phase and filtering. The algorithms corresponding to these two phases are now formally presented using pseudo codes.

Shape Equalization Algorithm I (Learning Phase)

```
start
{
Initialize maximum_trial_number
Clear trial_counter
Clear accuracy_array
Initialize filter_coefficient_array
Initialise temporary_threshold_delta
Initialize threshold_delta
Clear amplitude_direction_vector
Clear phase_direction_vector
while ( “temporary_threshold_delta greater than threshold_delta” )
    {
    Modify filter_coefficient_array according to amplitude_direction_vector
    Modify filter_coefficient_array according to phase_direction_vector
    while ( “scene from training sequence” )
        {
        Compute accuracy $\mathcal{A}_d$
        Store accuracy $\mathcal{A}_d$ into accuracy_array
        }
    Compute temporary_average_norm $\overline{\mathcal{A}_d}$
    Compute threshold_delta
    Compute temporary_threshold_delta
    Compute amplitude_direction_vector of filter coefficients
    Compute phase_direction_vector of filter coefficients
    }
    Increment trial_counter
    if ( “trial_counter greater than maximum_trial_number” )
        {
        break
        }
}
end
```

Shape Equalization Algorithm II (Filtering)

```
start
{
Compute Fourier descriptors using parametric representation of object's boundary
Perform filtering operation in Fourier domain
Compute Inverse Fourier Transform of the result of filtering operation
}
end
```

The shape equalization technique described above can also be applied to gray-level image sequences. In this case, the areas of the images having the same gray-level (un-equalized contours) are first extracted by a quantization process of the gray-level. The quantization process produces a sequence of extracted regions which can be treated as a binary scene sequence. Next, the filter coefficients are determined proceeding as in the case of binary images presented earlier. Finally, the equalization process and the subsequent construction of the shape elements can be performed.

The algorithm for the shape extraction that employs the square prime component decomposition and the shape equalization algorithms can now be formally presented using pseudo codes.

Shape Extraction Algorithm

```
start
{
Clear shape_element_array
Detect objects' boundaries
Perform shape equalization procedure of objects' boundaries
Compute prime_skeleton $s_n$ using square prime component decomposition
Compute full_skeleton $S_n$
Compute chain_of_points representing shape elements: $CH = \{\ldots\}$
Match B_spline curves to the chain_of_points
Store B_spline curves into shape_element_array
}
end
```

4.4 Intermediate Object Representation

In the Section 4.2, the method of the extraction of shape elements has been presented. The set of extracted shape elements given by

$$\mathcal{OS} = \{l_1, l_2, \ldots, l_j, \ldots, l_{z-1}, l_z\}. \tag{4.64}$$

is an intermediate object representation used in the proposed IUS. The quality and efficiency of the intermediate object representation is improved significantly by applying the shape equalization procedure as introduced in Section 4.3. The equalization scheme reduces the number of shape elements in the intermediate object representation without decreasing the description accuracy. It also reduces the errors caused by the description of diagonal and rotated objects, the use of low sampling frequency and quantization errors.

Since the intermediate object representation composed of the extracted shape elements includes the description of an object's interior, the high-level processing tasks and interaction with the knowledge base becomes simpler and more efficient. For example, the internal description of an object included in the intermediate object representation and based on the prime component decomposition greatly simplifies the process of separation and isolation of the objects in a scene. In addition, the prime component-based intermediate object representation is very suitable to perform one of the most computationally expensive algorithms within an IUS, namely, the searching and matching of the elements of the intermediate object representation within the knowledge base of an IUS.

4.5 Examples and Simulation Results

In this section, the model of shape extraction together with the shape equalization technique is experimentally tested. The proposed square decomposition is compared to two skeletonization algorithms, the Voronoi skeletonization algorithm [3], [29] and the morphological skeletonization algorithm [83]. We examine the decomposition process and generation of the intermediate object description performed on a test sequence of binary images. We also examine the efficiency of the skeletonization and the extraction processes.

4.5.1 Shape Equalization

The simulation of the shape equalization technique is performed on binary images having sizes ranging from 96x96 to 512x512 pixels. A Fast Fourier Transform algorithm has been employed in order to perform the filtering operation in the Fourier domain. The analysis of the shape equalization process is performed based on the comparison between the original and the reconstructed binary image at two different spatial sampling frequencies.

The results of the reconstruction process of the shape of the binary image is presented in Figure 4.3. Figure 4.3(a) shows the original binary image used in the experiment. The original binary image is sampled giving the internal image representation a size of 96x96. The sampling process simulates the image formation function in the low-level stage. The total number of points of the contour is 236. In order to increase the image resolution, the contour of inner image formation is equalized with a second-order filter. The filter coefficients are obtained through a training phase using a set of binary images similar to the original one. The result of the filtering operation is presented in Figure 4.3(c). As it can be observed from this figure, the filtering operation has significantly increased the spatial resolution of the inner image representation. The spatial resolution of the inner image representation presented in Figure 4.3(c) is equivalent the inner image representation having a size of 260x260, i.e., the spatial resolution has been increased by a factor of 2.5. It is important to note that the increase in the spatial resolution does not increase the number of points that are required to represent the contour of the object. The number of points that represent the inner image representation shown in Figure 4.3(c) is the same as in Figure 4.3(b) and it is equal to 236.

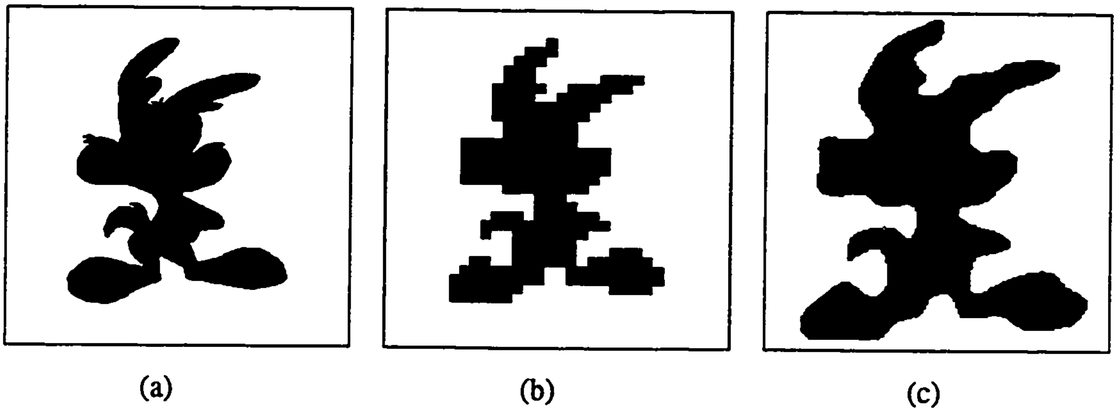


Figure 4.3: Shape reconstruction of a binary image by filtering the contour. (a) Original image. (b) Sampled and thresholded as a 96x96 image. (c) Reconstructed shape image.

A similar equalization operation performed on an image having a size of 192x192 is presented in Figure 4.4. In this case the number of points of the contour of the inner image representation is equal to 874. The resulting image of the shape equalization presented in Figure 4.4(c) is equivalent to the inner image representation having a resolution of 510x510. Consequently, in this case the equalization process has increased the spatial resolution of the inner image representation by a factor of 2.

An example of the equalization process of a particular region of a 96x96 gray-level image Lenna is presented in Figure 4.5. Figure 4.5(a) shows a 4 gray-level 96x96 Lenna image with one extracted binary region obtained through a quantization process of the gray scale. The spectrum of the extracted region in the Fourier domain has been already shown in Figure 4.2(c). The shape equalization process of the extracted region using the filter coefficients obtained from the different stages of the training phase are shown in Figure 4.5(b)-(d). The resulting inner image representation of the extracted region has a resolution equivalent to a 190x190 image, i.e., the equalization process has increased the overall resolution to more than twice without increasing the number of contour points.

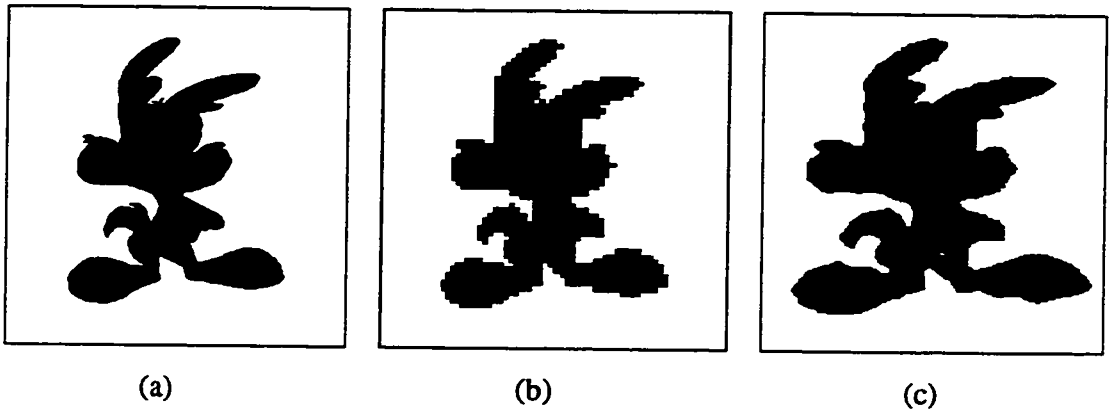


Figure 4.4: Shape reconstruction of a binary image by filtering the contour. (a) Original image. (b) Sampled and thresholded as a 192x192 image. (c) Reconstructed shape image.

4.5.2 Extraction of Shape Elements

In this subsection, the application of the square prime component decomposition to shape extraction is considered. The results of the shape element extraction using the proposed square decomposition is compared to that using two skeletonization algorithms, the Voronoi skeletonization algorithm [3], [29] and the morphological skeletonization algorithm [83]. We examine the decomposition process and generation of an intermediate object description performed on a test sequence of binary images. We will consider the practical aspects of the extraction of shape elements such as quality of the extraction of the shape elements, and the computational complexity of the algorithms in sequential and parallel implementations.

The first step in the simulation is to perform the edge detection, and independently, the square prime component decomposition of the scene. The prime component

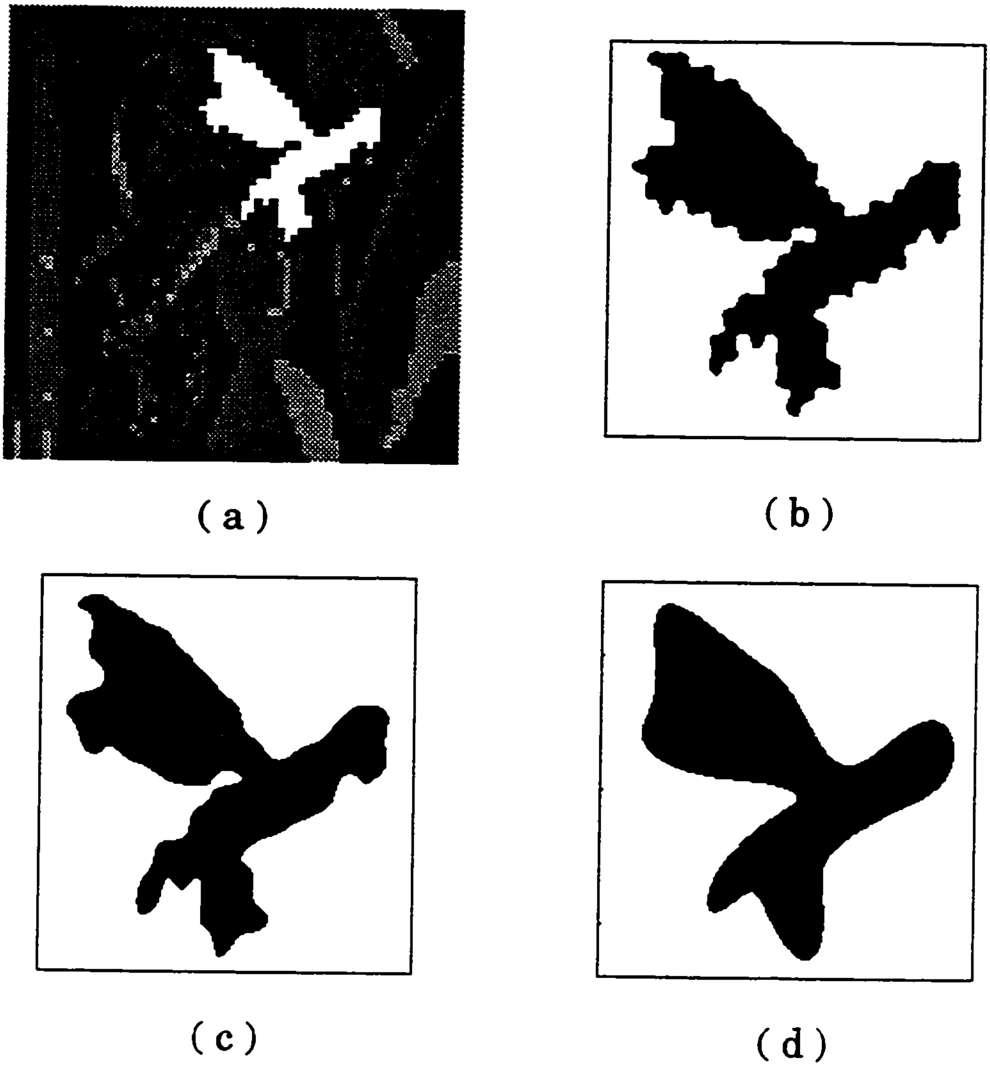


Figure 4.5: Shape equalization applied to a gray-level image. (a) A 4 gray-level Lenna image with an extracted region. Equalized shape using a second-order 1-D filter obtained with (b) 30%, (c) 60 %, and (d) 100% training.

decomposition allows the construction of the prime skeleton of the object(s). After the skeletonization process is completed, the full skeleton of the object is constructed from the prime skeleton and the edge map. The images using the skeletonization algorithm performed by employing the square prime component decomposition together with the detected object's boundary are shown in Figure 4.6.

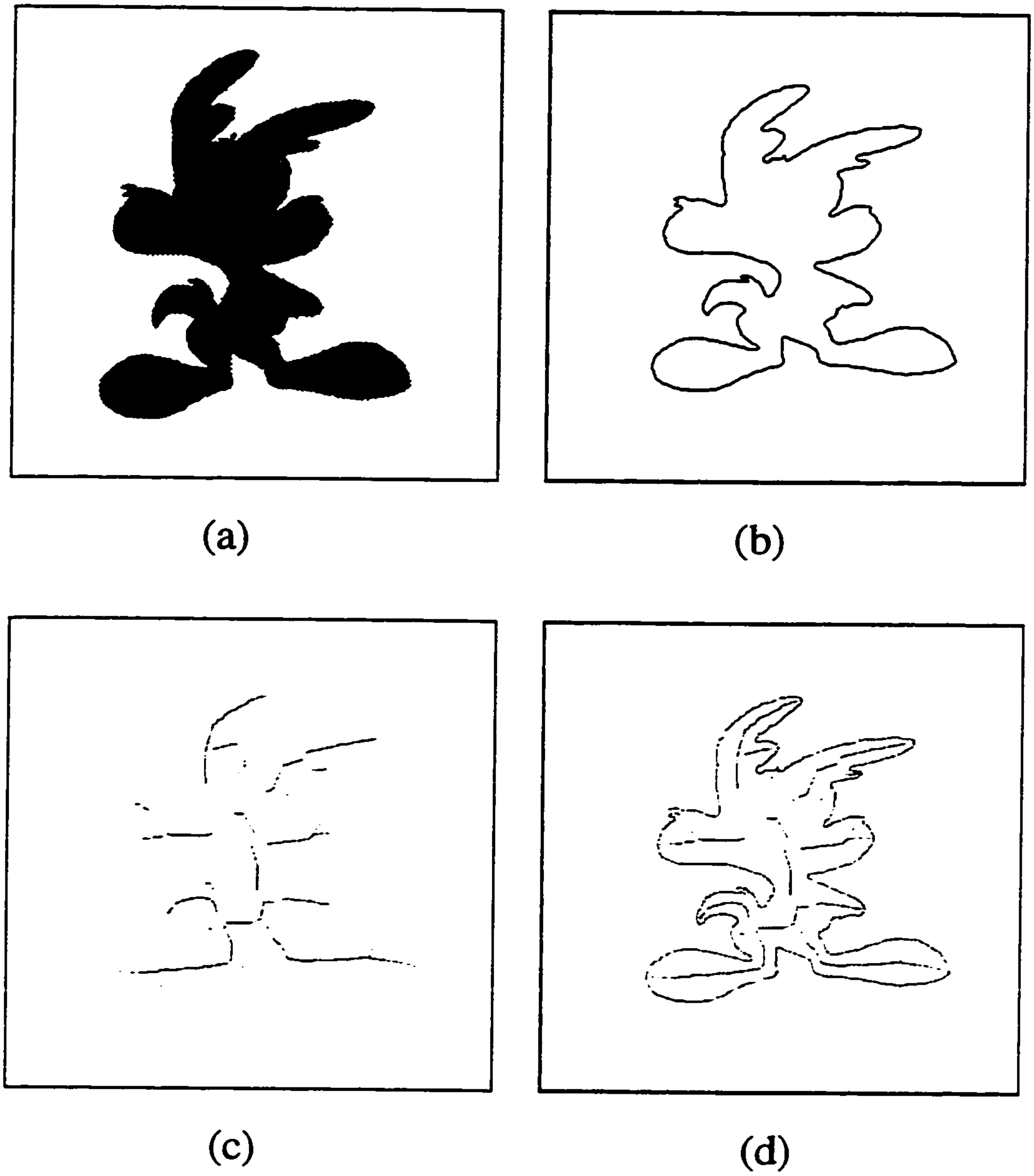


Figure 4.6: The full skeleton of the object composed of the boundary description of the object and its prime skeleton. (a) Original scene. (b) Detected contour. (c) Prime skeleton. (d) Full skeleton.

The full skeleton of the object is also obtained by using an optimized skeletonization algorithm based on Voronoi skeletons and morphological operators. In order to compare the effectiveness of the proposed square decomposition algorithms with those based on the Voronoi skeletons and the skeletons obtained from the morphological operators, the skeletonization process is performed on the same 1200 binary test image sequence. The comparison of the algorithms is performed using the same hardware platform and the OS environment. The comparison results for the three skeletonization methods are presented in Table 4.1. This table clearly shows that the prime component decomposition algorithms outperform those based on the optimized Voronoi skeletons and on the morphological operators. The performance of the proposed algorithm gets even better for the parallel implementations.

Time	Implementation	Type of algorithm		
		Prime component decomposition	Voronoi Optimized skeleton	Morpholo operat
Average decomposition time in ms. of 1200 96 × 96 image sequence	Sequential	3.784	8.701	10.92
	Parallel	0.1577	0.5801	0.840
Decomposition time in ms. of a particular 512 × 512 scene	Sequential	98.4	236.2	277.4
	Parallel	4.260	16.85	20.46

Table 4.1: Comparison of real-time performance of the skeletonization algorithms.

The last step in our experimentation is to extract the shape elements of the objects of a scene. As mentioned earlier, the process of the extraction of shape elements is flexible. The extraction of the shape elements can be controlled by optimizing of the parameters such as d_s and d_{min}. Figure 4.7 shows the shape-descriptor (extracted shape elements) of the test object. The search distance and the minimum distance have been chosen as d_s=6 and d_{min}=3 (Figure 4.7(a)) and d_s=10 and d_{min}=8 (Figure 4.7(b)). In real applications, the controlling parameters should be adjusted automatically in an adaptive manner.

In Table 4.2, the numerical results of the real-time performance of the proposed shape extraction algorithm are compared to those obtained by using the algorithm based on the Voronoi diagrams [3], [29]. The algorithms were executed on a SPARC Station-2.

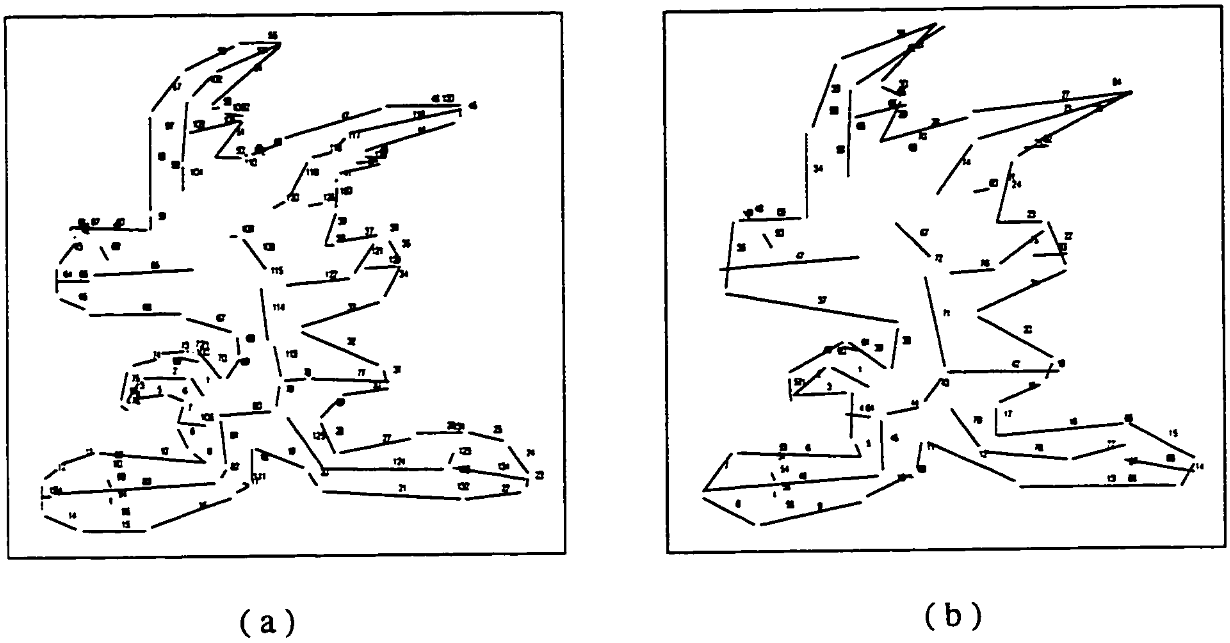


Figure 4.7: The results of extraction of the shape elements. (a) Extraction of z = 134 shape elements ($d_s = 6$ and $d_{min} = 3$). (b) Extraction of z = 88 shape elements ($d_s = 10$ and $d_{min} = 8$).

Shape extraction algorithm	Prime components	Voronoi diagrams
Data complexity	11104 points	11104 points
Computation time of raster points in seconds	0.13	0.96
Computation time of edges in seconds	0.32	2.89

Table 4.2: Comparison of real-time performances of the shape extraction algorithms with sequential implementations applied to a 512×512 scene.

As seen from Table 4.2, the proposed shape extraction algorithm is about 8 times faster than the one based on the Voronoi diagrams. Storing the information regarding the raster points and the edges of the shape elements constitute the intermediate object description.

The total real-time performance for processing a single 512×512 world-scene is shown in Table 4.3. The total processing time indicates the processing time of one complete scene. Integration timings of the intermediate objects description into the knowledge base tree structure have not been included in this comparison.

Approach	Prime component decomposition	Voronoi diagrams	Morphological operator
Total processing time of a 512×512 scene.	0.54 s	3.61 s	4.2[1]s

Table 4.3: Comparison of the real-time performances of three different approaches with sequential implementations for a 512×512 world-scene.

4.6 Summary

In this chapter the techniques for the extraction shape elements and shape equalization have been developed. In the proposed approach of shape extraction, the polygonal decomposition of the interior of the object and the description of its boundary (edges) have been used. The decomposition of the object's interior is based on a skeletonization concept and it is performed using square polygons (prime components). The boundary of the object are determined using a modified Sobel operator.

In order to improve the accuracy of the extraction of the shape elements, an equalization scheme has been employed. The shape equalization operation increases the accuracy of the shape extraction process without increasing the complexity of the input information. The method is based on data interpolation using Fourier descriptors taking advantage of

[1] An estimated real-time performance of the transformation

the statistical properties of the shape boundary. After the transformation of the shape boundary into the Fourier descriptor domain, the traditional 1-D filtering techniques has been applied.

The final stage of the extraction is to transform the equalized discrete skeleton of the object into its shape description (shape descriptor). This process has been carried out by matching the B-spline curves to the whole discrete skeleton of the object. The resulting shape descriptor of the object is composed of a set of the extracted shape elements (set of B-spline curves) and inter-relations among them.

As the skeletonization process uses a square decomposition approach, the speed of the shape extraction has been improved over that of the most of the morphological-based skeletonization algorithms. The set of parameters d_{min} and d_s enables a full control of the accuracy of the extraction process. The simple geometrical structures such as circles, squares, rectangles etc. have been successfully extracted using as few as 5 to 6 extracted shape elements. The description of more complex objects has required about 150 to 300 shape elements.

The shape elements of an object arising from shape equalization constitute the intermediate object representation for an IUS. The equalization operation has reduced the number of extracted shape elements in the intermediate object representation. This process can also simplify the computational complexity of the algorithms used in low-level and high-level processing tasks.

Chapter 5

AN IMAGE UNDERSTANDING SYSTEM

As mentioned in Chapter 1, a typical image understanding system (IUS) which performs the knowledge transformation is composed of two major processing stages known as low-level processing stage and high-level processing stage. These two stages are significant components in modeling a complex IUS. For a knowledge-based IUS, another very important component that is included in the model is the knowledge base itself.

The task of a low-level processing stage is concerned with the extraction of an efficient intermediate knowledge representation from an input world-scene. This topic has been treated in Chapters 3 and 4. The high-level processing stage, on the other hand, is responsible for the analysis, interpretation and understanding of an input world-scene. The knowledge base of an IUS helps to perform the high-level processing tasks by storing and retrieving the necessary information required for analysis and interpretation of input world-scenes. In this chapter, a new knowledge-based image understanding system is presented. The proposed model as shown in Figure 5.1 differs from the traditional knowledge-based image understanding systems in that that low-level processing stage also interacts with the knowledge base and the system utilizes a modified architecture for its black-board approach in high-level processing.

In this chapter, the models of the low-level and high-level processing stages of the proposed IUS shown in Figure 5.1 are presented. The model of the low-level processing stage includes processing paradigms that have already been presented, i.e., those involving prime component decomposition, shape equalization and extraction. The proposed model of the high-level processing stage is based on the black-board (BB) architecture with an agenda type scheduling mechanism [28]. This model allows to perform more effectively the goal-driven and data-driven reasoning by using incremental problem solving strategies and by employing a selective focusing mechanism. The knowledge base of the IUS is designed in terms of knowledge representation models and their data structures. The interactions between the low-level processing stage and the knowledge base are presented as a bottom-up reasoning process of updating the knowledge base. The proposed scheme of integrating a new data structure within the knowledge base simplifies the implementation and minimizes the hardware requirements of the low-level processing stage. The interactions between the high-level processing stage and the knowledge base are presented as a process of modifying the knowledge base by active knowledge sources of the BB architecture.

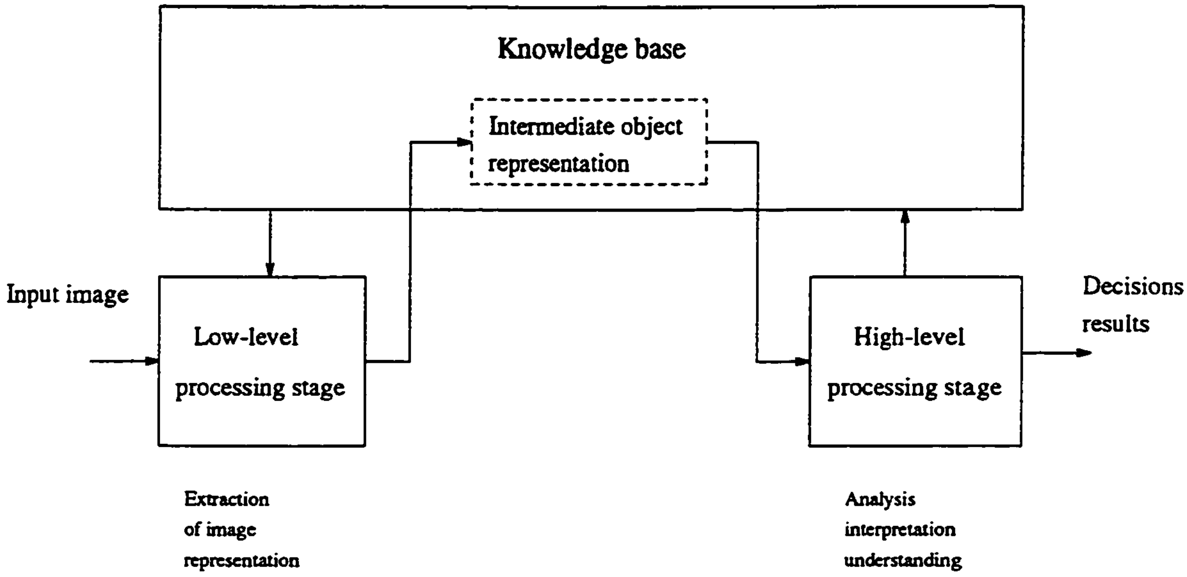

Knowledge base
Intermediate object representation
Input image
Low-level processing stage
High-level processing stage
Decisions results
Extraction of image representation
Analysis interpretation understanding

Figure 5.1: The proposed model of an IUS.

5.1 Knowledge Base of the Image Understanding System

Before proceeding with a detailed description of the low-level and high-level processing stages, one should consider one of the most important aspects of a knowledge-based IUS, namely, its knowledge base model. Informally, the knowledge base of a system is a set of representations of facts about world-scenes. It is a data of the system that helps to perform knowledge transformation within the IUS and simplifies the tasks of internal computations. Moreover, in most cases, the interaction by a preprocessing stage with the knowledge base is the only way to process a typical context-dependent and ambiguous information that is present in real world-scenes.

5.1.1 Knowledge Representation Models

In order to propose an efficient and practically implementable model of the knowledge base, the following issues should be addressed.

- The type of knowledge to be represented within the knowledge base of an IUS.
- The type of knowledge representation scheme to be used to represent the knowledge within the knowledge base.

From the implementation point of view, the knowledge representation scheme should be able to store and process effectively a huge amount of information from the physical world through a knowledge compilation process. This should include the compilation of a semantical (knowledge content [20]) as well as a procedural knowledge within the knowledge base of the IUS. Other important issues that needs to be addressed during the development of an effective model of the knowledge base are the initial creation and further run time transformations of the knowledge within the IUS - modeling of the learning process.

Considering the IUS as a two-stage knowledge compilation system, one can infer some preliminary characteristics of the knowledge representation model based on the nature of the transformations within the IUS. For example, there is a need to represent the processing algorithms as a procedural knowledge in the high-level processing stage. However, within the low-level processing stage this type of knowledge representation will have limited use. By analyzing particular aspects of the knowledge representation, one can find the requirements of a knowledge representation scheme in the low-level processing stage of the IUS as follows.

- *Inhibition of grouping principles* - The knowledge base of the IUS should allow to effectively represent (incorporate and extract) the relevant features of the object related to each other by following grouping principles:
  - (i) *Proximity* - objects with spatially similar features,
  - (ii) *Continuity* - features enclosed within a single contour,
  - (iii) *Symmetry* - features of symmetrical objects, and
  - (iv) *Common fate* - features of moving objects.
- *Inhibition of abstraction hierarchy* - The knowledge base of the IUS should effectively represent a hierarchy of feature abstractions.
- *Inhibition of feature aggregation* - The knowledge base of the IUS should effectively combine subsets of object's features under the given context into feature aggregates.

The above requirements for the knowledge representation suggest a combination of two suitable models to represent the knowledge within the IUS. This combination should include the models based on the relational and hierarchical data structures.

A typical world-scene contains many complex objects composed of recognizable parts. A full description of a complex object will then consist of its global features, global features of each of its parts, and the relations among the parts. The features of an object are simply tuple of measurements that have numeric or symbolic values. The relations among the parts of the objects should be understood similar to the relations among the prime components described in Chapter 3. In terms of the relational formalism presented in Chapter 3, the relational structure $\mathcal{D}_o$ of an object O is a pair $(\mathbf{P}, \mathbf{R})$, where $\mathbf{P}$ is a set of an object's parts or primitives expressed using prime components, and $\mathbf{R}$ is a set of named (numbered by index) relations among the prime components. For each relation $\mathbf{R}_i \in \mathbf{R}$, a non-negative integer n_i indicates the number of prime components involved in the tuple of the relation $\mathbf{R}_i$ and a non-negative integer m_i indicates the number of features of the attributes involved. Denoting by $\mathbf{A}$ the set of all possible attributes (attribute values), one can write:

$$\mathbf{R}_i \in \mathbf{P} \times \mathbf{A}. \tag{5.1}$$

That is, each tuple of $\mathbf{R}_i$ represents an attribute relationship among the n_i prime components and characterized by the m_i attributes. When $n_i = 1$, and $m_i > 0$, the relation $\mathbf{R}_i$ is just a feature vector representing the global attributes of the object O. When $n_i = 2$, and $m_i \geq 0$, the relation $\mathbf{R}_i$ is the list of attributes of the prime components of the object O. When $n_i > 2$ and $m_i \geq 0$, the relation $\mathbf{R}_i$ describes the attribute relationship among the prime components.

Since the utilization of the knowledge base of an IUS by the low-level processing stage does not require retrieving and processing complex relationships, the main feature vector and attributes should mostly include spatial, gray level and time-dependent relations. Let us consider the following example. Suppose it is required to represent an intermediate object representation (extracted shape elements) by a relational description. Further, suppose that the global attributes one wishes to represent are the total number of extracted shape elements, average number of extracted shape elements per unit area, and the numbers of rows and columns of the world-scene from which the shape elements have been extracted. These global attributes can then be represented by a single-tuple relation Scene_Properties having four components: Total_shape_elements, Density, Rows and Columns. If one wants to keep track of coordinates of the starting and the end-points, the length, and the angle of each extracted shape element, one could encode this information as an attribute relation Shape_element_segments, whose tuple contains one extracted_shape_element and the values of its six attributes: Row_start, Column_start, Row_end, Column_end, Length, and Angle.

The perceptual relation between the extracted shape elements can also be described using three grouping categories: proximity, parallelism and collinearity. In this case, each relation is represented as an attribute of binary relation. The tuple of the proximity relation contains a pair of line segments and a real number indicating the minimum distance between the line segments. Each of the parallel and collinear relation contains a tuple with a pair of extracted shape elements and a measure of their parallelism or collinearity. To illustrate these concepts, in Figure 5.2, Tables 5.1, 5.2 and 5.3, a decomposed scene and the relational description employing the Scene_Properties, Shape_element_segments, Proximity, Parallelism and Collinearity relations are presented.

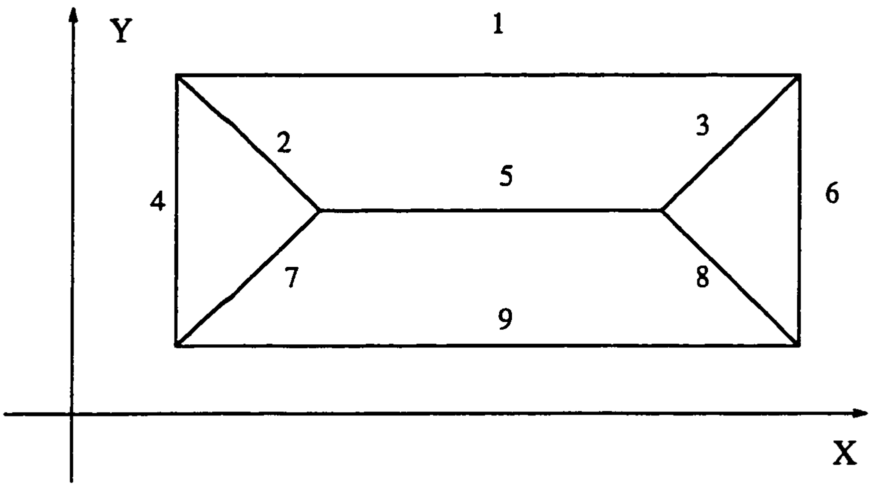


Figure 5.2: The extracted shape elements of an object within a 512x512 world-scene. The numbers indicate the extracted shape elements.

Total_shape_elements	Density	Rows	Columns
9	0.027	120	360

Table 5.1: The attributes of Scene_Properties relation of the object in Figure 5.2.

Extracted_shape_element	Row_start	Column_start	Row_end	Column_end	Length	Ang in Deg
1	200	40	200	400	360	0
2	200	40	120	120	226	31
3	120	320	200	400	226	45
4	200	40	40	40	160	27
5	120	120	120	320	200	0
6	40	400	200	400	160	90
7	40	40	120	120	226	45
8	120	320	40	400	226	31
9	40	40	40	400	360	0

Table 5.2: The attributes of Shape_element_segments relation of the object in Figure 5.2.

First shape element	Second shape element	Parallelism
1	5	YES
1	6	NO
1	9	YES
2	8	YES
2	3	NO
3	7	YES
4	8	NO
4	2	NO
4	6	YES
5	9	YES
5	8	NO

Table 5.3: The attributes Proximity, Parallelism and Collinearity relations between selected pairs of shape elements of the object in Figure 5.2.

The attributes of Scene_Properties relation presented in Table 5.1 give the global attribute values of the world-scene. The attributes of Shape_element_segments relation presented in Table 5.2 list each shape element along with its length and angle. The attributes of binary Proximity, Parallelism and Collinearity relations for certain selected pairs of shape elements are presented in Table 5.3. In this example, the the real-valued quantities of density and length have been rounded off. Length is defined as the Euclidean distance between starting- and end-point of an extracted shape element, and angle is defined as arctan((Row_end -Row_start)/(Column_end - Column_start)).

The relational structures are sufficient to describe even very complex objects within a world-scene. However, as the number of prime components (extracted shape elements) increases, the relational description becomes cumbersome and oversized. In order to maintain the benefits of the relational model without increasing the description complexity, the object to be described can be broken into its major parts, the major parts into subparts and so on, yielding a hierarchical representation structure. Since the hierarchical structure can effectively exploit the re-usability of its components, the whole description results in a more compact (better) representation model. Formally, the hierarchical relational description $\mathcal{D}_o$ of the object O is a pair $(\mathbf{P}, \mathbf{R})$, where $\mathbf{P}$ is a set of shape elements and/or prime components and $\mathbf{R}$ is a set of relatíons among them. The relations work in the same way as in a single-level non-hierarchical structure, in which the shape elements (general-level prime components) can be broken down until they reach to the zeroth level prime components. Since the zeroth level prime components (pixels) cannot be further decomposed, they are called the *atomic elements* of the world-scene. An example of a hierarchical relational structure was presented in Figure 1.1. In Figure 1.1(a), there are eight elements, scene, objects, table, legs, board, book, bottle and glass and the relations, part, on and left as shown in Figure 1.1(b), define spatial relationships among the elements of the scene.

5.1.2 Data Structure of the Knowledge Base

The definition of the data structures within the knowledge base is one of the most important steps in developing a practical IUS. Since the majority of algorithms within an IUS deal with the searching and matching problems, an appropriate construction of data structures can speed-up the real-time performance of the system significantly. The preliminary considerations related to the knowledge representation model within an IUS, discussed in Section 5.1.1, restrict the data structure to be hierarchical, and relational.

Let us consider two clearly distinct abstract data objects: LENGTH and WIDTH representing the length and the width of a typical two-dimensional object within a scene. It is quite clear, that these two data objects can be characterized using the same measurement attribute - a real number. Since these two data objects are characterized by the same measurement attribute, the classical data type definition would not distinguish these two objects as two different data types. Consequently, this restriction of the classical model would limit the flexibility of the tasks within an IUS and a new data type definition need to be proposed. In the proposed definition, the data type is defined, determined and distinguished based on a concept of labels and attributes of the hierarchical relational data structures presented in Section 5.1.1. In terms of the new data type definition, the two data objects: LENGTH and WIDTH are viewed as two distinct data types derived from the base data type representing the value - a real number. The data objects contracted in this manner will be called *clones* of the base data type. Figure 5.3 shows the inference diagram of deriving the new data types - the clones from the base data type.

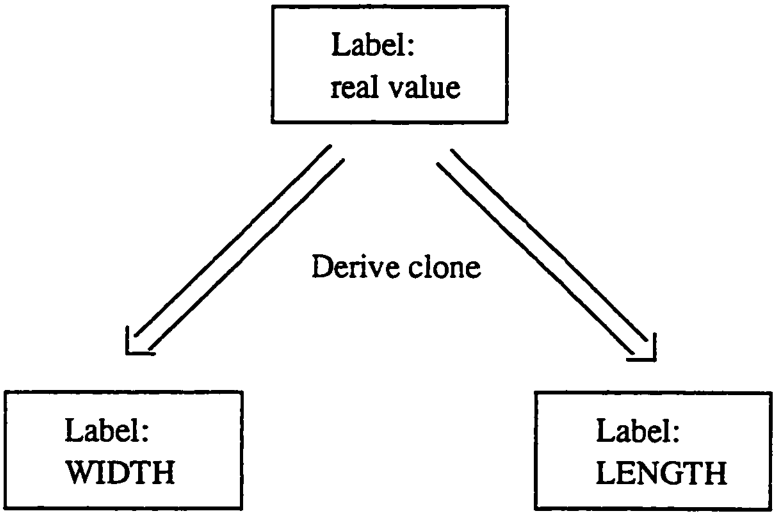
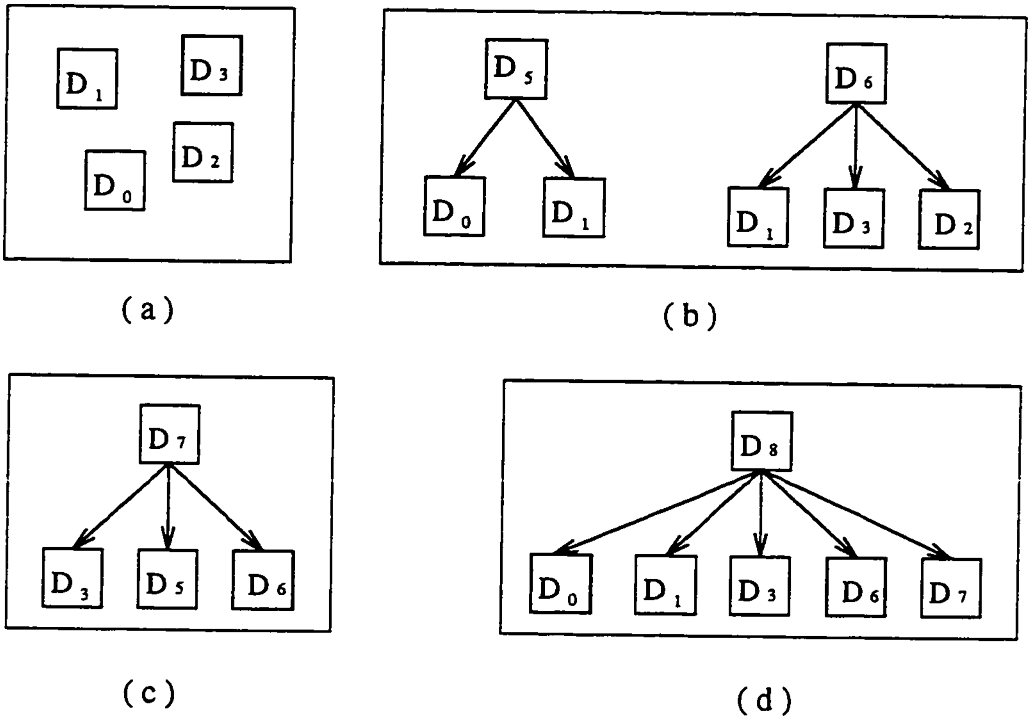


Figure 5.3: Derivation of a new data type within the IUS.

The derivation of the data types does not have to be restricted to clone data types. For instance, a new data object (new data type) can be created by performing the clustering operation on any number of data types. This operation is analogous to the construction of the general-level prime components described in Chapter 3. In this case, the zeroth level prime components correspond to the base data types and their clones, while the general-level prime components correspond to the derived data types that are subsets of any number of base and/or derived data types. Consequently, any data object within the knowledge base of an IUS can be grouped into one of two data types:

- Base data type
- Derived data type

The graphical examples that explain the idea of the data type organization and the construction of the derived data types are presented in Figure 5.4.



(a)

(b)

(c)

(d)

Figure 5.4: Examples of constructing the derived data types within the knowledge base of the IUS. (a) Base data types. (b) First-order derived data types. (c) Second-order derived data type. (d) Third-order derived data type.

In Figure 5.4(b) the derived data type D_5 is constructed from the base data types D_0 and D_1, while the derived data type D_6 is constructed from the base data types D_1, D_2 and

D_3. The derived data types D_7 and D_8 presented in Figure 5.4(c) and (d) are obtained from data types D_3, D_5 and D_6 in the case of the derived type D_7, and from D_0, D_1, D_3, D_6 and D_7, in the case of the derived data type D_8. The arrows shown in Figure 5.4 describe membership relations among the data type and its members.

The clustering operation that produces a derived data type is carried out in such a way that the resulting structure is a tree graph that can have one and only one parent. The type of the parent described by its label determines the type of the derived data. Consequently, the whole organization of the knowledge base is implemented as data objects containing only pointers to the allocated memory of its members. An example of partition of the hierarchical relational knowledge base into the data structures of the knowledge base is presented in Figure 5.5.

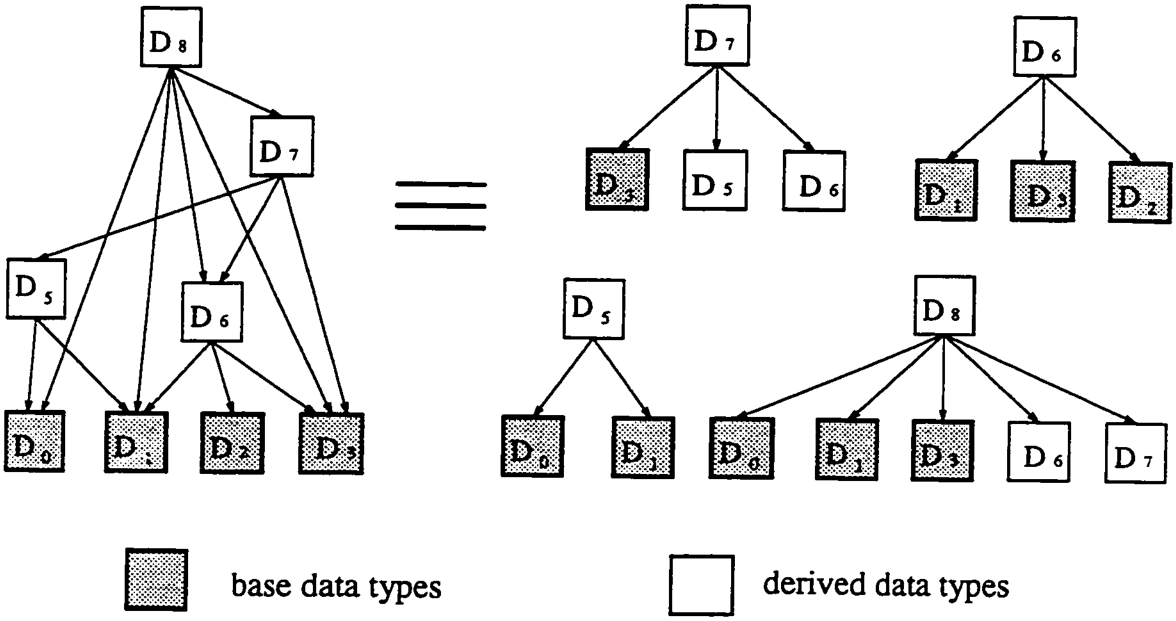


Figure 5.5: The partition of the knowledge base into the data structures containing only parents and pointers to their members.

One of the most difficult problems in an IUS is to process a context-dependent information. The notion and importance of context appear in many cases during conceptually-driven or top-down interpretation methodologies in knowledge-based processing. The analysis of context-dependent and context-driven processing determines the family of relations that can be mapped into the knowledge base of an IUS. For example, during a scene analysis and interpretation process, one cannot neglect the dynamically changing relations between the objects within a scene, which depend on the environment and perceptual and cognitive aspects of the analysis. One of the most important requirements of implementing the necessary interpretation methodologies is the global and broad understanding of the relational links between data structures. As a result, in order to manipulate complex entities and concepts from the physical world, one has to extend the traditional meaning of the relations by combining the semantical links with the operational and time-varying knowledge of the IUS. In this extended meaning, the relational links are understood very broadly. For example, they can characterize the production and inference rules, they can determine more complex time dependencies, or they can describe the procedural knowledge of the IUS.

The utilization of the knowledge base by the two stages of an IUS that transform the knowledge on two different abstraction levels, results in a specific partition of the relational links between low-level and high-level processing stages. As already mentioned, the most essential relation from the viewpoint of the low-level processing stage describes the spatial data dependencies among the elements (e.g. prime components) within a scene. We will denote this relation as an inclusion data-dependent relation. A few examples of relations between various data types are shown in Figure 5.6. The relation R_1 shown in Figure 5.6(a) represents the inclusion data-dependent relation between the parent D_7 and its members D_3, D_5 and D_6. Figure 5.6(b) shows the complement of the inclusion data-dependent relation, the relation "is subset" that describes supersets D_5 and D_8 of the parentD_0. The relation R_3 ("can produce"), shown in Figure 5.6(c), describes the procedural knowledge of an IUS such as processing algorithms, methods and inference rules. In the examples presented in Figure 5.6, the low-level processing stage will utilize the relations R_1 and R_2 that describe the clustering operation, while the high-level processing stage will utilize the relations R_1, R_2 and R_3 that describe both the clustering operation and the procedural knowledge of an IUS. In the case where the number of relational types is insufficient, the new relational links can be created dynamically. From this

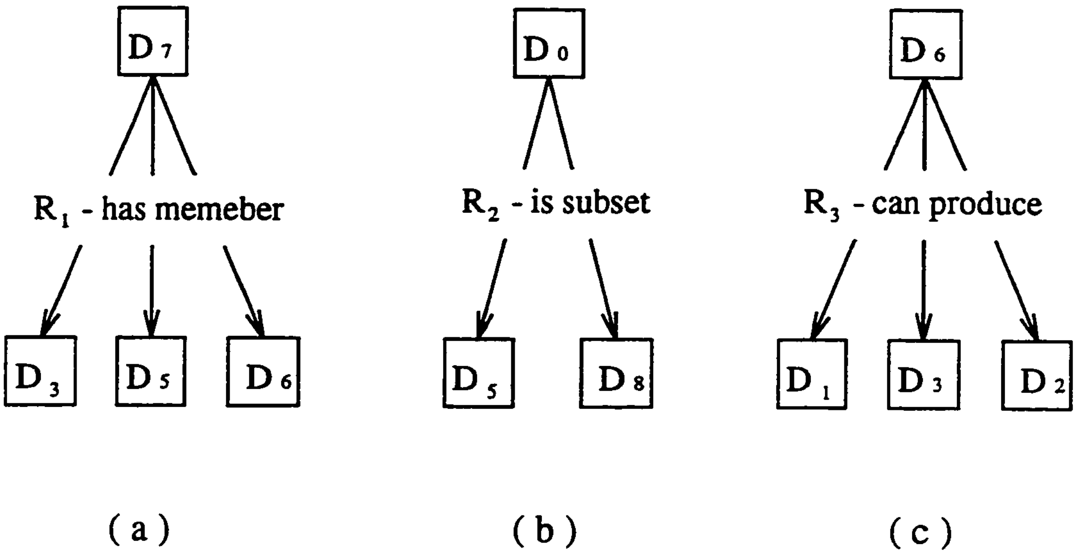


Figure 5.6: Examples of relations among various data types. (a) The example of the inclusion data dependent relation R_1 - "has member". (b) The example of the superset relation R_2 - "is subset". (c) The example of the procedural relation R_3 - "can produce".

point of view the relations can be grouped similar to the data structures into one of the two categories:

- Base relations,
- Derived relations.

The base relations are the ones that allow an IUS to begin its execution. While the derived relations are created during the run-time execution in the IUS.

One of the most computationally expensive tasks within an IUS is to execute searching and matching algorithms that operate on the knowledge base. The efficiency of these algorithms can be drastically improved by including several additional parameters to the data structures. These additional parameters describe the internal structure and the statistical properties of the data objects within the knowledge base. They also synchronize the access to the structures of the knowledge base and allow the new data types to be effectively incorporated into the existing ones. The following major parameters of the data structures in a knowledge base can be distinguished.

- *Statistical parameters*

  Life time statistic vector - $\overline{LT_{Stat}}$

  Links statistic vector - $\overline{LI_{Stat}}$

  Access statistic vector - $\overline{A_{Stat}}$

- *Relational parameters*

  Connection vector - $\overline{C_{Rel}}$

  Similarity vector - $\overline{S_{Rel}}$

  Link vector - $\overline{L_{Rel}}$

- *Structural parameters*

  Descendent vector - $\overline{D_{Str}}$

  Parent vector - $\overline{PA_{Str}}$

  Partial trees vector - $\overline{PT_{Str}}$

- *Access parameters*

  Access vector - $\overline{A_{Acc}}$

  Dimension vector - $\overline{D_{Acc}}$

  Coordinate vector - $\overline{C_{Acc}}$

Among the parameters describing the statistical properties of the data objects, the parameters, life time statistic vector $\overline{LT_{Stat}}$ and access statistic vector $\overline{A_{Stat}}$, are the most important ones. These parameters describe "how old is" and "how often has been used" a given data structure of the knowledge base. The relational parameters describe the inclusion relation, i.e. the spatial dependencies between the data structure and the knowledge base. These parameters also encode the information describing the similarity of data objects to other data structures from the knowledge base. The structural parameters specify the inner organization of the data structure such as the number of hierarchical levels, the number of parents or the number of children. Finally, the access parameters help prevent locking and hazards during multiple accessing of a given data structure by different processes of the IUS.

5.2 Proposed Model of the Low-Level Processing Stage

It can be seen from the proposed model of an IUS (Figure 5.1) that the communication process between the low-level and high-level processing stages is carried out through the knowledge base of the system. This communication link between the low-level processing stage and the knowledge base facilitates the merging of the general-level prime components within the knowledge base, obtained as a result of the extraction of shape elements from the low-level processing stage.

As it has been pointed out in Section 5.1, the hierarchical relational knowledge has the ability to represent very complex relational dependencies required by the high-level processing stage of the IUS. However, from the point of view of the low-level processing stage, the most important relations are not the complex ones describing the sophisticated spatial and time dependencies or procedural knowledge, but the ones that are needed to represent simple spatial relations among the elements of a single scene. Therefore, in order to clarify and simplify our discussion. we will focus our attention on those aspects that are essential from the point of view of the low-level processing stage. The additional explanation will be provided during a brief description of the high-level processing stage in Section 5.3. Consequently, the knowledge base of the IUS will be viewed as a set of data structures representing only the prime components and extracted shape elements discussed in Chapters 3 and 4. Furthermore, since the extracted shape elements are composed of (constructed from) the zeroth- and/or general-level prime components, we will refer to the data structures of the knowledge base simply as prime components of appropriate levels.

The knowledge base of the system is seen by the low-level processing stage as a stochastic tree graph structure containing all the information on the history of the prime components and the actions performed on them. Every time, the general-level prime component is created and merged, the structure of the knowledge base is changed. This also affects all the subsets of the prime components that belong to the newly created and/or merged general-level prime components - prime components connected by the inclusion relation. The new general-level prime component is incorporated into a temporary tree structure as a leaf node, and then, the temporary tree structure is incorporated into the whole knowledge base of the system. Every temporary tree graph generated by the low-level processing stage represents one object within the world-scene. During the merging of the temporary tree structures into the stochastic tree graph of the knowledge base, the set of statistical parameters and spatial relations of other general-level prime components are examined. A set of prime components comprising the candidates for merging is generated. If a "strong" equivalence relation among the merging candidates and the appropriate structures of the knowledge base exists, the set of candidates is merged into a new general-level prime component within the knowledge base of the system. This newly created tree has all the properties of general-level prime components and it can be used in the next merging cycle.

The concept of the merging procedure is equivalent to the bottom-up strategy of data fusion, where during the backward chaining process, simple data objects are combined in order to infer more complex structures. Processing of one world scene can generate many stochastic tree graphs. The task of interpretation and analysis of the merged and active regions of the knowledge base belongs to the next high-level processing stage of the IUS.

In Figure 5.7 a complete model of a low-level processing stage is proposed. As it can be observed from this figure the proposed model of the low-level processing stage comprises data flow channels and several processing blocks as described below.

The block **Input Devices** represents input media such as camera, VCR, scanner or any other source that produces world-scene sequences. The major task of this block is to

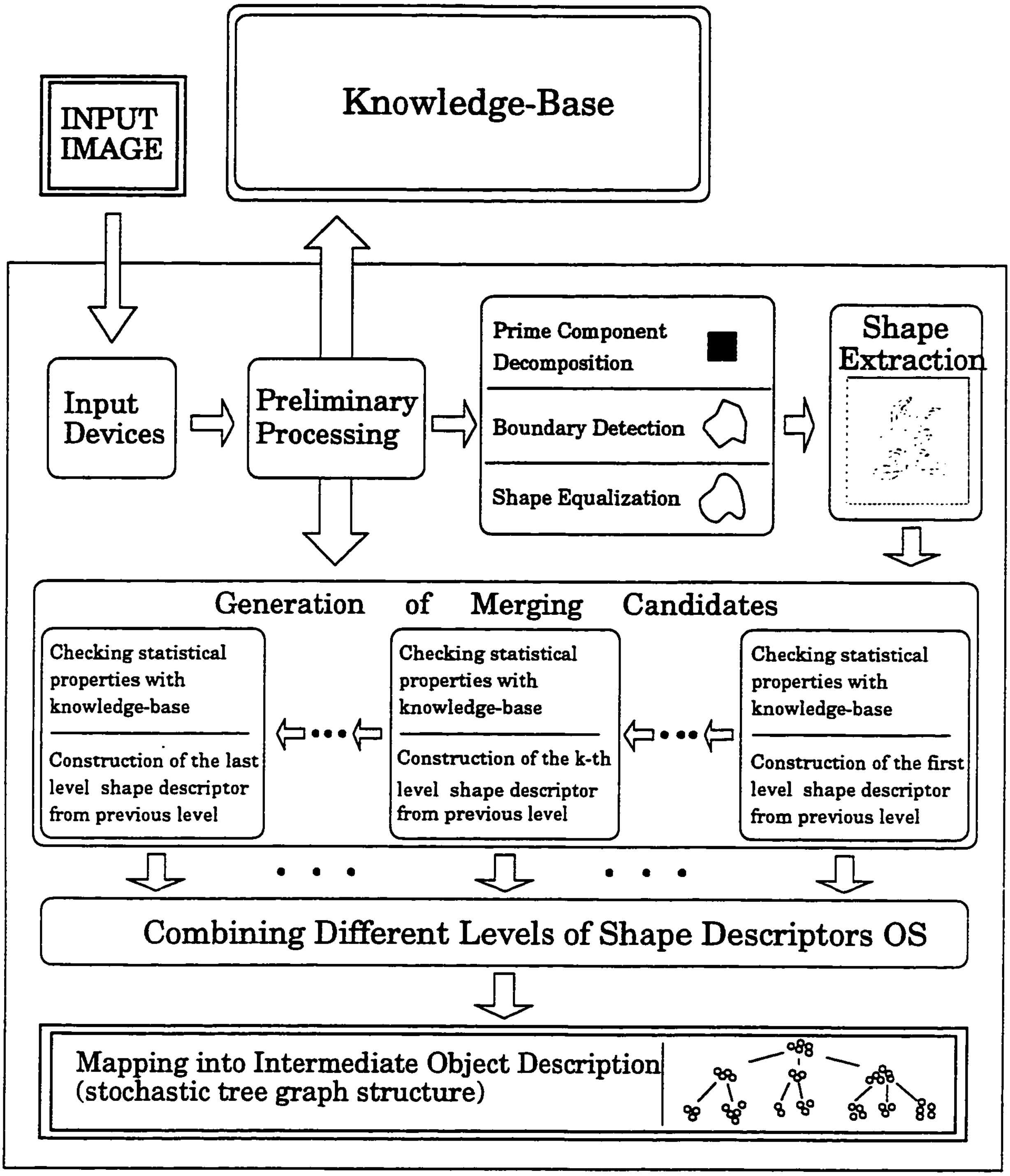

INPUT IMAGE
Knowledge-Base
Input Devices
Preliminary Processing
Prime Component Decomposition
Boundary Detection
Shape Equalization
Shape Extraction
Generation of Merging Candidates
Checking statistical properties with knowledge-base
Construction of the last level shape descriptor from previous level
Checking statistical properties with knowledge-base
Construction of the k-th level shape descriptor from previous level
Checking statistical properties with knowledge-base
Construction of the first level shape descriptor from previous level
Combining Different Levels of Shape Descriptors OS
Mapping into Intermediate Object Description (stochastic tree graph structure)

Figure 5.7: A complete proposed model of the low-level processing stage of an IUS.

transform the input world scenes $W_{sc}(x, y)$ into its inner image representations $I_{im}(x, y)$ inside the IUS with minimum distortion. This block also provides the communication services with the storage media, performs the functions of network interfacing and synchronizes the processes and the time events on various platforms of the distributed IUS.

The block **Preliminary Processing** prepares an input scene for low level processing. This block performs most of the classical image processing operations such as low-pass and high-pass filtering, contrast enhancement, histogram equalization, and simple non-linear filtering (e.g. median filtering). The operations in this block are mostly point and memoryless transformations. After preliminary processing, the inner image representation $I_{im}(x, y)$ is ready for the next level of processing - the interior and boundary decompositions of the objects within the scene.

The block **Prime Component Decomposition** is one of the most important and time consuming tasks in the whole low-level processing. This block transforms a given inner image representation $I_{im}(x, y)$ into a set of prime components u_p^n and a set of associations u_a^n as given by

$$\mathcal{DEC}[I_{im}(x, y)] = \{u_p^n\,, u_a^n\}\,. \tag{5.2}$$

The decomposition by maximum size square prime components of the objects in the inner image representation $I_im(x, y)$ is performed in parallel. The idea of the parallel decomposition is to partition the inner image representation into a set of binary sub-images, and then to perform the square decomposition process by filling the non-overlapping square areas of the objects of a particular binary sub-image. The decomposition operation is performed on all of the binary sub-images. The resulting set of prime components u_p^n and association u_a^n is then transformed in the same block into a skeleton map of the objects as given by

$$s_n(x, y) = \begin{cases} \sup(\mathcal{SQR}(m, n)), & \text{if } O_n(m, n) \neq 0 \\ 0, & \text{otherwise}\,. \end{cases} \tag{5.3}$$

The block **Boundary Detection** performs a detection of the boundaries of the objects in the inner image representation $I_{im}(x, y)$ and produces its edge map $E_n(m, n)$ as

$$E_n(x, y) = \begin{cases} 1, & \text{if } (x, y) \text{ belongs to the edge of the object} \\ 0, & \text{otherwise .} \end{cases} \tag{5.4}$$

As it described in Chapter 4, the edge detector algorithm uses a modified Sobel operator. The convolution operations between the modified Sobel operator and the inner image representation $I_{im}(x, y)$ is performed in the Fourier domain utilizing a 2-D FFT algorithm. The boundary detection and the prime component decomposition operations are carried out in parallel and independent of one another.

The block **Shape Equalization** increases the description accuracy of the edges $E(x, y)$ and the skeleton map $s_n(x, y)$ of the objects within the inner image representation $I_{im}(x, y)$. Its working principle is based on filtering the parametrically represented boundary curves of the object and its skeleton map. As it has already been described in Chapter 4, the shape equalization is performed by a 1-D convolution operation between the parametric representation of the object's boundaries and the coefficients of the 1-D filter. The filter coefficients used to equalize the object's boundaries are obtained during a learning phase with a training sequence of the real world-scenes. In order to increase the real-time performance of the block, the shape equalization process is performed in the Fourier domain utilizing a 1-D FFT algorithm.

The block **Shape Extraction** first combines the skeleton map s_n and the edge map $E_n(x, y)$ of the object into its full skeleton $S_n(x, y)$ as

$$S_n(x, y) = \begin{cases} 1, & \text{if } E_n(x, y) \neq 0 \text{ and } s_n(x, y) \neq 0 \\ 0, & \text{otherwise .} \end{cases} \tag{5.5}$$

At this point the whole inner image representation $I_{im}(x, y)$ has been transformed into its full skeleton. The full skeleton can now be further transformed, as described in Section 4.2, into its shape descriptor $\mathcal{OS}$ as given by

$$\mathcal{OS} = \{l_1, l_2, \ldots, l_j, \ldots, l_{z-1}, l_z\}, \tag{5.6}$$

where l_j ($j = 1, 2, \ldots, z$) are the extracted shape elements. Obtaining a shape descriptor $\mathcal{OS}$ is next to the prime component decomposition, the most computationally expensive operation. In order to maximize the real-time performance of the operation, the full skeleton of the object $S_n(x,y)$ is partitioned into overlapped sub areas depending on the density (nonzero values of $S_n(x,y)$ per unit area) of the skeleton itself. The partitioned overlapped sub-areas are then redistributed over the network of processors. Every processor transforms part of the full skeleton (one sub-area) into a partial shape descriptor. At the end of the operation, all of the partial shape descriptors are combined into a set of extracted shape elements - the shape descriptor $\mathcal{OS}$.

The block **Generation of Merging Candidates** generates sets of merging candidates - prime components from the set of extracted shape elements $\mathcal{OS}$. The generated prime components are of levels $1,2,...,k-1$ and k. Every extracted shape element l_j, $j = 1,2,...,z$ is considered as a general-level prime component from which, the next level of merging candidates can be constructed. The n-th level prime components can only be constructed from the (n-1)th level prime components. The process of generating the merging candidates is controlled in such a way that for every generated merging candidate P_{sc}^M, the probability of its being found in the knowledge base $\mathcal{U_P}$ is the maximum, that is

$$\forall m: \; P_{sc}^m \in \mathcal{U_P}, \;\; \mathcal{P}(P_{sc}^M) \;\geq\; \mathcal{P}(P_{sc}^m) \tag{5.7}$$

From a practical implementation point of view, finding an optimal set of merging candidates is a very complex and time consuming process, requiring random generation procedures and very efficient searching algorithms. Every generated merged candidate has to be confronted with the knowledge base $\mathcal{U_P}$ and the probability of its occurrence has to be verified. In order to minimize the searching time, the knowledge base is organized as an ordered multidimensional tree. Every dimension of the tree represents one searching context - probability, spatial organization or time correlation. Every node of the tree changes its position within the tree structure depending on the context (dimension). Using this method of a representation, the searching area can be very easily narrowed down, while the searching procedure becomes similar to a binary search algorithm. After comparison against the knowledge base, the merging candidate can:

- join the family of merging candidates, if the probability of its being found in the knowledge base is high enough,
- be discarded from the family of merging candidates if the probability of its being found in the knowledge base is insufficient.

The process of generation of the merging candidates stops if within the predefined period of time, a new merging candidate cannot be generated. In this manner the whole family of merging candidates of level k is generated. In the case when the set of generated merging candidates of k-th level is not empty, the new set of merging candidates of level (k+1) can be generated by repeating the generation procedure.

The block **Combining Different Levels of Shape Descriptors** has the ability to merge the families of prime components of different levels into a single prime component representation of the scene. Conceptually, this operation is similar to the one performed by the block "generation of merging candidates" as presented above. In this case, however, a set of families of prime components that represent the merging candidates can be clustered into a new general-level prime component regardless of the representation levels of its members. For example, the family of level k can be constructed from families of levels $(k-1), (k-2), ..., 1$ in a sequence that is reverse to the process of the constructing the merging candidates allowing only the level $(k-1)$. All the merging families of prime components are combined into one and only one world scene representation Z^n_{sc} which is the final result of the scene decomposition by the $\mathcal{DEC}[.]$ operator as given by

$$\mathcal{DEC}[I^n_{im}] = \{I^n_{im}, Z^n_{sc}) : I^n_{im} \in \mathcal{U}_\mathcal{I} \wedge Z^n_{sc} \in \mathcal{U}^n_\mathcal{P} \times \mathcal{U}^n_\mathcal{A}, n \in \mathcal{N}\}, \tag{5.8}$$

where the set $\mathcal{U}^n_\mathcal{P}$ consists of all of the shape elements and the set $\mathcal{U}^n_\mathcal{A}$ represents the spatial relations among the extracted shape elements.

The final block of the low-level processing stage **Mapping into Intermediate Object Description** maps the newly generated prime component, into the knowledge base $\mathcal{U}_\mathcal{P}$ by performing the following three steps:

(i) Determination of the inclusion coordinates in all dimensions of the knowledge base for the prime component and for its decedents.

(ii) Inclusion of the prime component and its descendent into the knowledge base.

(iii) Updating the statistical parameters.

Determination of the inclusion coordinates is based on comparison between the partial tree structures of the knowledge base with the structure of the newly generated prime component and its descendents for all possible dimensions. In the case in which there exists a similarity in the structural level, the root of the partial tree structure from the knowledge base becomes inclusion coordinate for the given prime component and/or its descendents. Structure similarity between the given prime component and the partial tree from the knowledge base means that the structure of the given prime component or its descendents and structure of the partial tree from the knowledge base are the same. Figure 5.8 explains the concepts of the structure similarities among prime components.

Searching for the inclusion coordinates begins from zeroth-level prime components. In the case, when the inclusion point cannot be found for a prime component of level k but it can be for the level $(k-1)$ (descendent of a prime component of level k), a new node of the knowledge base is created. An example of a situation in which there is a structure similarity between a decendent of the generated prime component P_n and a partial tree structure R_n from within the knowledge base U_p is shown in Figure 5.9. In this example, the similarity between the descendent of the prime component P_n and the partial tree structure R_n will result in the creation a new node in the knowledge base U_p. After the inclusion process has been performed, all of the statistical parameters of the similar prime components in the entire knowledge base are updated. The statistical parameters of the links of the prime components similar to other elements of the knowledge base (other prime components) are also updated.

The process of inclusion of general-level prime components into the knowledge base of the IUS is the last step in order to generate the intermediate object description. The newly included prime component becomes an active part of the knowledge base. The included active prime component represents the intermediate object description of the scene and at this time the flow of control is transferred to the high-level processing of the IUS.

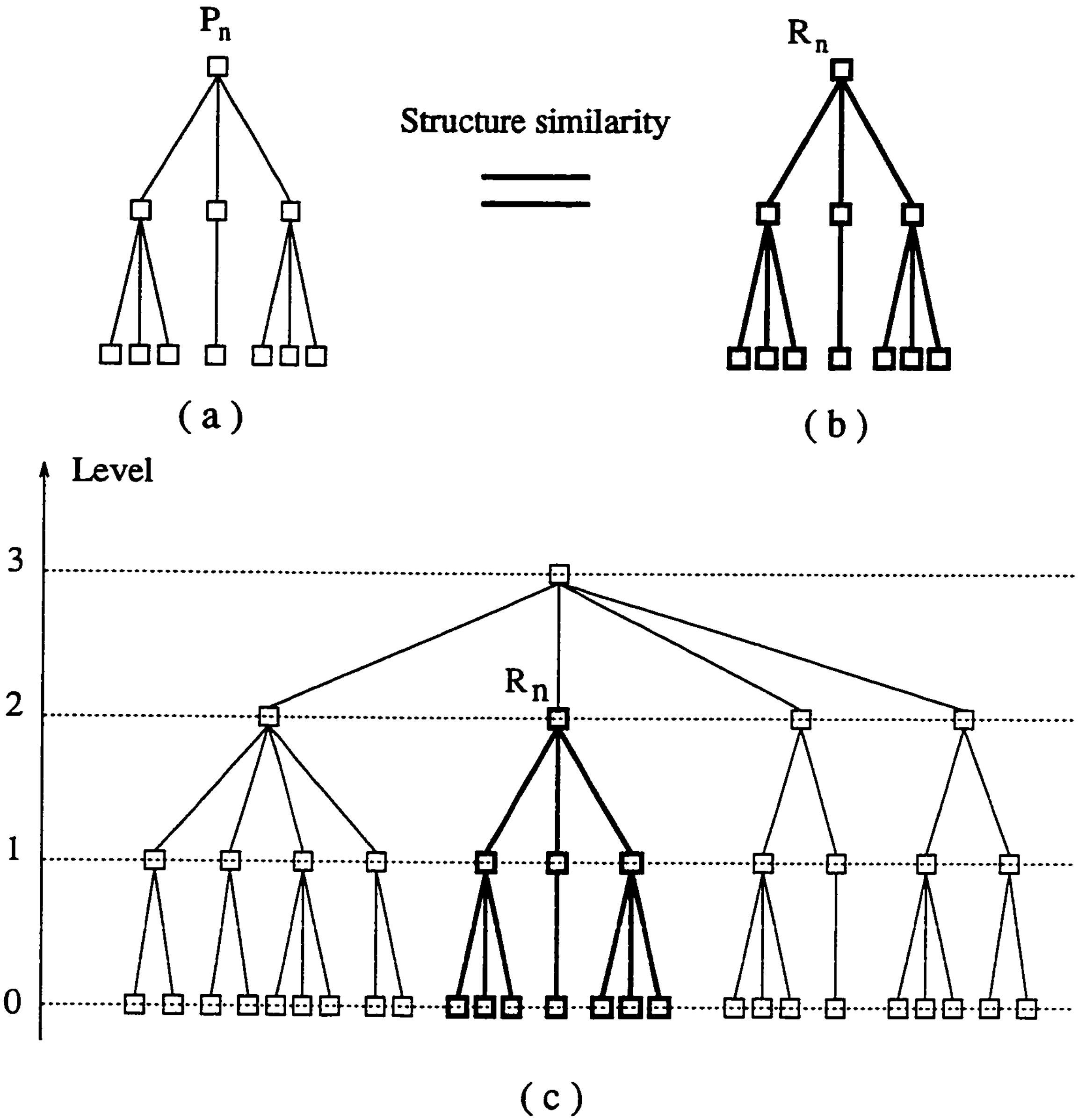


Figure 5.8: Structure similarity between a generated prime component P_n and a partial tree structure R_n from the knowledge base U_p. (a) The generated prime component P_n. (b) A partial tree structure R_n. (c) The knowledge base U_p.

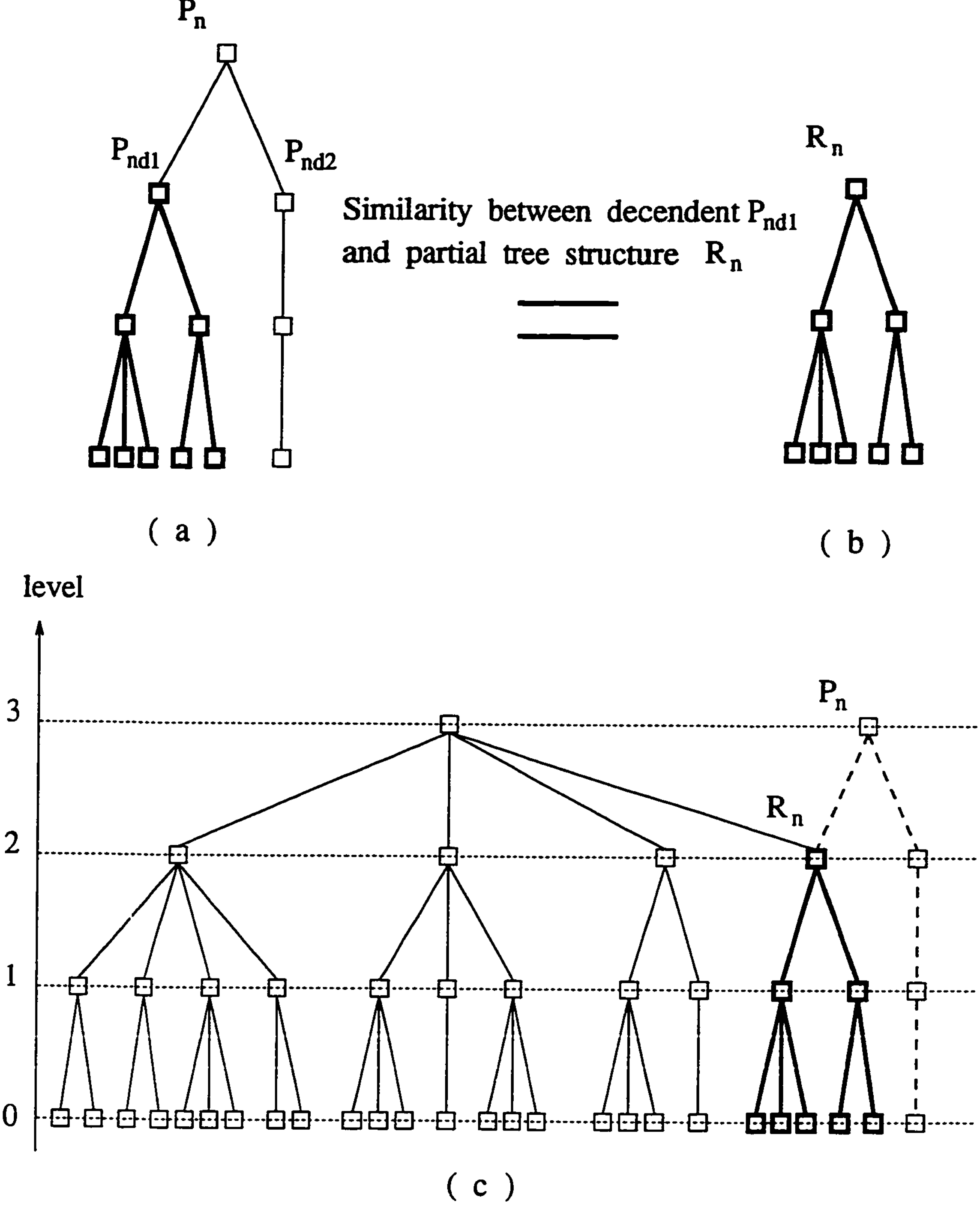

P_n
P_nd1
P_nd2
Similarity between decendent P_nd1
and partial tree structure R_n
R_n
(a)
(b)
level
3
2
1
0
P_n
R_n
(c)

Figure 5.9: Process of an inclusion a prime component P_n into the knowledge base U_p. (a) The generated prime component P_n. (b) A partial tree structure R_n from the knowledge base U_p which has a structure similarity with the descendent P_{nd1} of the prime component P_n. (c) The knowledge base U_p together with the included prime component P_n with the creation of a new node P_n in the knowledge base structure.

5.3 Proposed Model of the High-Level Processing Stage

The high-level processing stage of the IUS is based on the distributed problem solving black-board (BB) architectural model [28] developed in the field of artificial intelligence (AI). The BB model has its roots in the Hearsay speech understanding systems [28] to deal with the difficult characteristics of the speech recognition problems such as very large search space, incomplete data, and imprecise and/or incomplete problem-solving knowledge. These characteristics require a problem solving model that can support the incremental development of solution, can apply diverse types of knowledge, and can adapt itself with strategies to specific problem situations. Since these characteristics are needed to solve problems found in both speech and image understanding systems, the BB architectural model is ideally suited to perform the tasks of the high-level processing stage of an IUS [28]. In the proposed model of the high-level processing stage, the control mechanism of the BB architecture is modified to increase its efficiency.

5.3.1 The Black-Board Architecture of the High-Level Processing Stage

The conceptual model of the high-level processing stage based on the BB architecture is presented in Figure 5.10. It consists a set of knowledge sources (agents) KS_i $(i = 1, \ldots, n)$ and a set communication channels between the knowledge base and the knowledge sources. The knowledge sources KS_i's of the BB permanently monitor the knowledge base of the system for a condition that will satisfy the requirements for their activation. The conditions for the activation of the knowledge sources are described using the rule-base oriented approaches and they are implemented by using a look-up table strategy. The activation of a particular knowledge source may change the state of the knowledge base by generating new or by changing the existing data structures (general-level prime components). It can also change or generate the procedural knowledge (hypothesis and algorithms) within the knowledge base of the system. This could provide the potential stimulus for the activation of another knowledge source, and so on. The ability to self activate the knowledge sources, gives the black-board system strong data-directed characteristics that are necessary for supporting opportunistic problem solving.

The knowledge sources of the BB architecture act as specialized, separate and independent of each other execution threads, i.e., the execution of the knowledge sources do

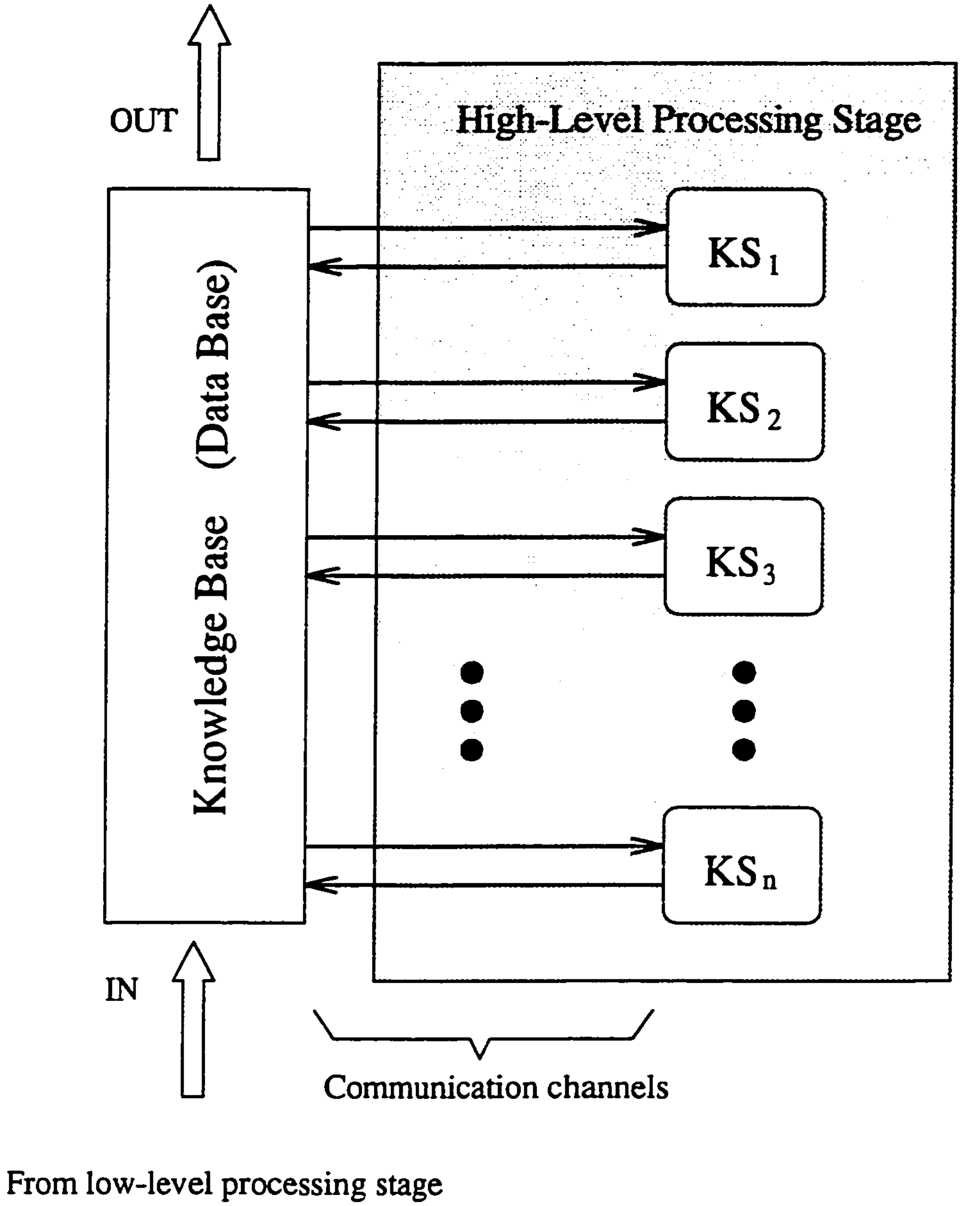

OUT
High-Level Processing Stage
Knowledge Base (Data Base)
KS 1
KS 2
KS 3
KS n
IN
Communication channels
From low-level processing stage

Figure 5.10: The conceptual model of the high-level processing stage using the black-board architecture.

not explicitly depend on the execution of other knowledge sources and any communication of information between the knowledge sources occurs only via the creation and modification of the data structures within the knowledge base of the system. In the present implementation of the high-level processing stage, a total of 24 knowledge sources and 73 condition-action pairs (rules) are used. The knowledge sources are implemented as separate functions that are executed concurrently within the IUS. Each knowledge source is executed in parallel on a separate transputer.

5.3.2 Control Mechanism of the High-Level Processing Stage

Since the knowledge sources of the high-level processing stage are both independent and self-activating, in a sense there is no need for any (additional) control mechanism. Despite the appeal of a high-level processing stage of not requiring any control components, this approach has two serious problems. First, since the number of executed instances of the knowledge sources could be larger than the number of available processors, their executions have to be synchronized with each other and sequentialized. This means that the knowledge sources cannot be executed as soon as the activation requirements are satisfied. It also means that checking of these requirements must compete with the execution of the knowledge sources for processor resources. The second problem with the "no control" approach is that the typical problems for a high-level processing stage are very complex and combinatorially highly explosive. Such problems become intractable if the IUS attempts to execute all the applicable knowledge sources. When "unpromising" actions are executed, they can "block" the system by triggering further applicable but not useful actions that compete with the useful actions for the limited computational resources. This greatly decreases the real-time performance of the system. Because of these problems the high-level processing stage incorporates the control mechanism similar to the one found in the Hearsay-II black-board architectural model - the agenda-based control mechanism [28].

The major elements of the high-level processing stage with the agenda-based control mechanism is shown in Figure 5.11. In this model the Knowledge Base Monitor receives a set of events from the Data Base that describes the status of the Knowledge Base. The Knowledge Base Monitor identifies which knowledge source should be activated by these events and it invokes the precondition rules of the activated knowledge sources. Successful preconditions return the stimulus and response information that the Knowledge Base Monitor uses to create a list of "passive" knowledge sources placed onto the Agenda. The Knowledge Base Monitor also updates the Focus of Control based on the new events. The Scheduler evaluates the

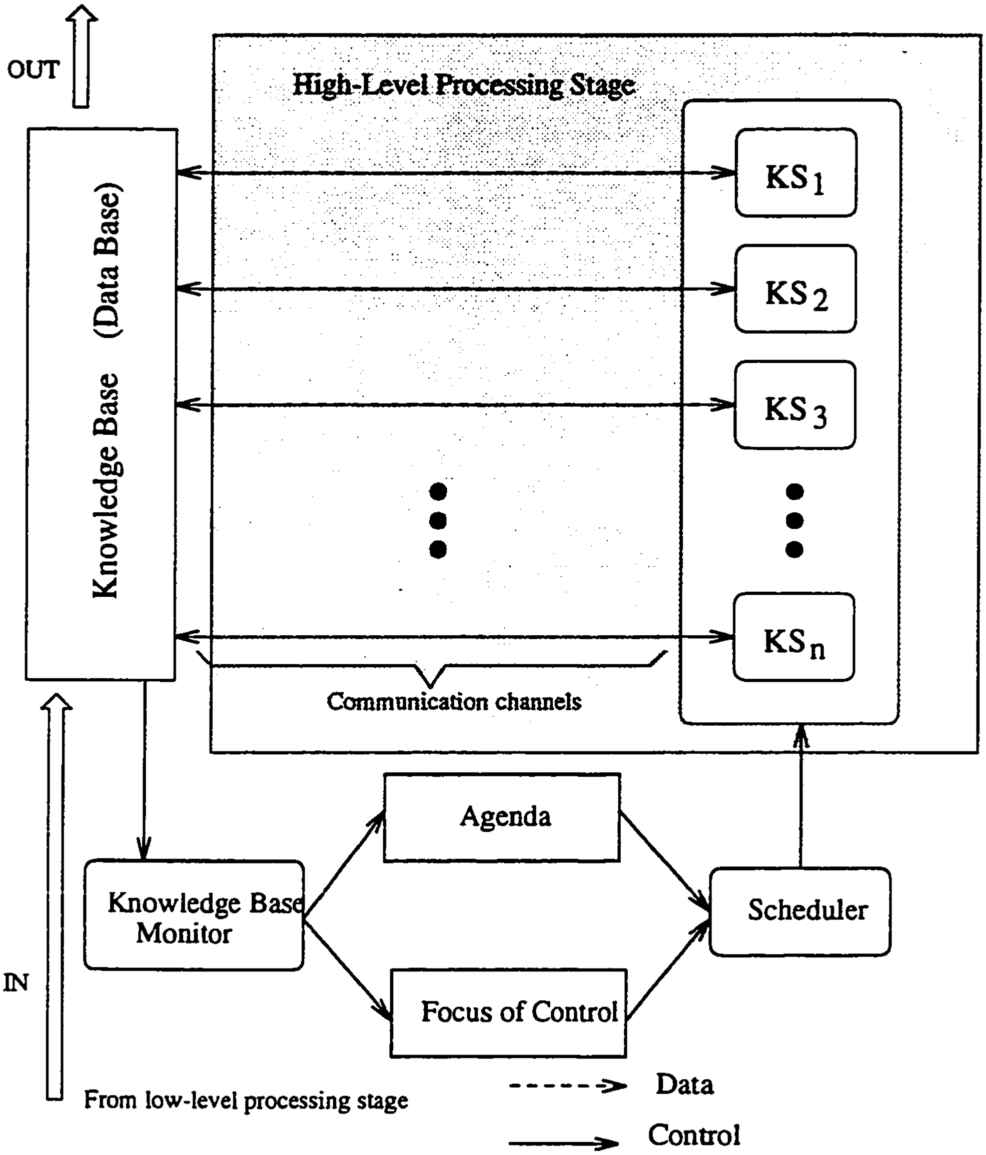


Figure 5.11: The high-level processing stage with agenda-based control mechanism.

priorities of "passive" knowledge sources that have been placed onto the agenda, selects (and removes) the highest rated one, and invokes the action component of an appropriate knowledge source.

Based on the above discussion, the algorithm used to control the high-level processing stage is now formally presented using pseudo codes.

Control Algorithm for High-Level Processing Stage

```
start
while ( "TRUE" )
    {
    Check active prime components within knowledge base
    Check preconditions for active prime components using rules of knowledge sources
    Create list of passive knowledge sources
    Create list of selected knowledge sources
    while ( "passive knowledge source" )
        {
        if ( "priority of a knowledge source is sufficient" )
            {
            Include passive knowledge source in execution que
            }
        }
    Prune execution que using list of selected knowledge sources
    Execute knowledge source from execution que
    }
end
```

The agenda-based control mechanism used in the high-level processing stage is inherently opportunistic, since the knowledge sources are activated in a data-directed manner and all possible actions are considered during each control cycle. This allows for a rapid refocusing (at every control cycle) between different lines of reasoning, different levels of abstraction, and so on.

5.4 Summary

In this chapter a new model of a knowledge-based image understanding system along with its three major components - the low- and high-level processing stages and the knowledge base has been presented.

The model of the knowledge base has been proposed as a relational hierarchical multi-dimensional structure, composed of the data objects. The structure of the knowledge base has been optimized in order to maximize the utilization of the hardware and software resources of the IUS. In order to enhance the flexibility of the knowledge representation, the classical definitions of data types of the objects within the knowledge base has been extended to include two new data categories: base data types and derived data types. In the proposed knowledge base, the relational links between data structures have made it possible to incorporate in it the semantical, operational and time-varying knowledge. This approach thus avoids duplication of data structures and allows the knowledge base to be accessed efficiently, without conflicts from both the high-level and low-level processing stages of the IUS.

The low-level processing stage has been presented in a context of processing paradigms developed in Chapters 3 and 4, focusing on prime component data representation within the knowledge base. It has been shown that all of the transformations within the low-level processing stage can be divided into two main processing phases:

1 Extraction of the shape elements viewed as general-level prime components that describe an input world-scene,

2 Merging of the extracted shape elements into the knowledge base of the IUS.

The extraction of the shape elements has been presented as independent, self contained blocks of processing transformations. Since most of these transformations have already been described in earlier chapters, in this chapter only a brief and relevant information has been provided. The presentation of the merging process has contained a brief description of the interactions between the low-level processing stage and the knowledge base of the IUS. The concept of similarity between the general-level prime components of the IUS has been introduced. The similarity concept has been then used in the merging process to determine the inclusion coordinates within the knowledge base of the IUS.

The high-level processing stage has been proposed as a black-board (BB) architectural model having an agenda-based control mechanism. The proposed BB model comprises independent knowledge sources (agents) implemented as a subprocesses (execution threads) that are activated based on the state of the knowledge base. The agenda based control mechanism eliminates the problems related to a purely data-driven execution mode. This is achieved by a separation of the control and data flow of the system into independent processes. Similar separation has been applied between the knowledge sources and their communication mechanism. An efficient incremental and opportunistic problem solving strategies have been employed by the high-level stage to obtain the final results of its processing tasks.

Chapter 6

APPLICATIONS OF THE PRIME COMPONENT DECOMPOSITION

In Chapter 3, a new scheme of prime component decomposition has been presented. In Chapter 4, this prime component decomposition has been combined with a shape equalization technique to extract shape elements in the low-level stage of an Image Understanding System (IUS). The complete model of an IUS comprising the low-level and high-level processing stages along with a knowledge base component was presented in Chapter 5. In this chapter, the proposed model of the IUS, in the context of the prime component decomposition technique developed, will now be applied in carrying out several practical experiments. In these experiments, we are concerned with several object identification paradigms such as isolation, extraction, recognition and detection of the stationary and moving objects within an input world-scene. The applications of the prime component decomposition considered in this chapter, can be divided into two categories - the low-level processing applications and the high-level processing applications.

Figure 6.1 shows the hardware configuration of the proposed IUS together with the supporting I/O devices giving an overview of the system architecture used in our experimentation.

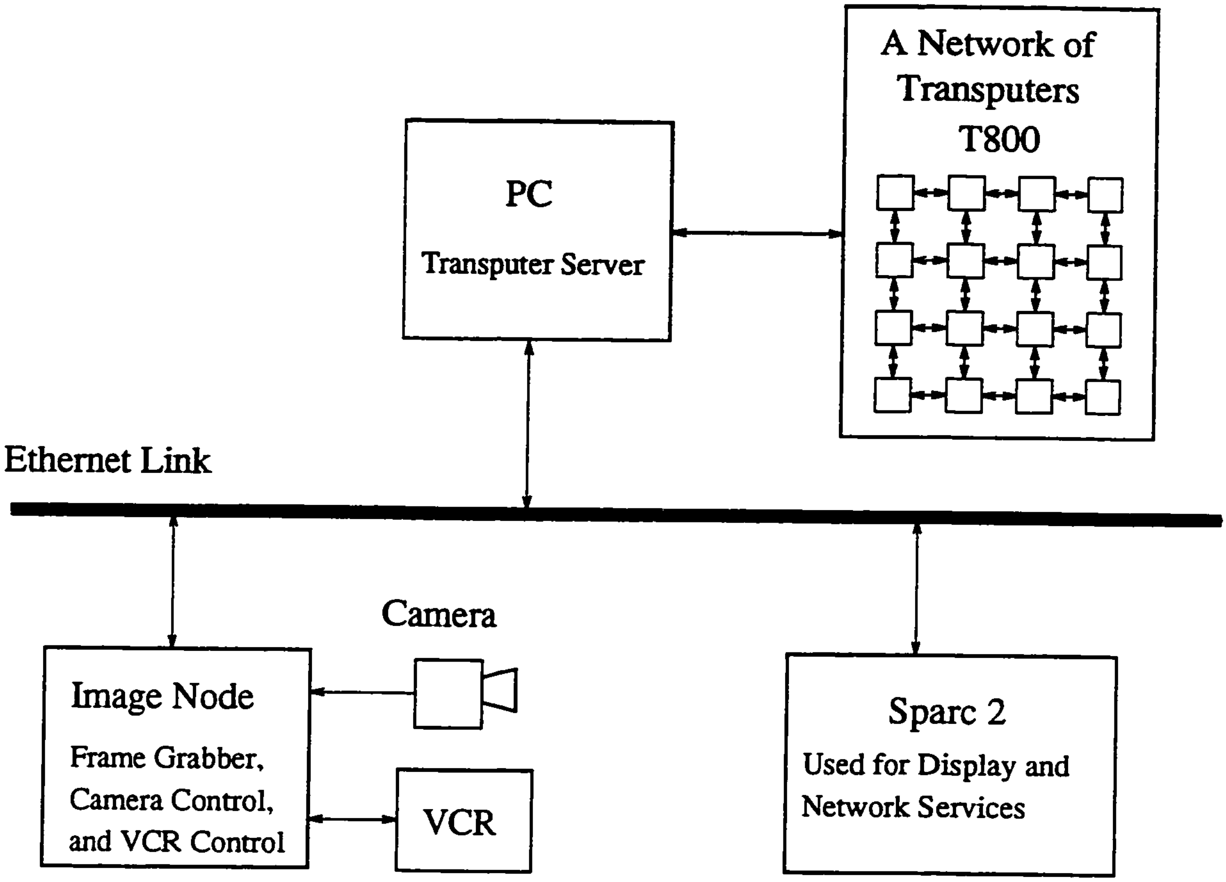

A Network of Transputers T800
PC
Transputer Server
Ethernet Link
Camera
Image Node
Frame Grabber, Camera Control, and VCR Control
VCR
Sparc 2
Used for Display and Network Services

Figure 6.1: The hardware configuration of the IUS and supporting I/O devices.

The hardware configuration of the IUS is composed of several processing blocks. I/O devices and interconnections among them. The camera and the VCR are used as sources of input world-scenes. The sequences of world-scenes are first recorded using the VCR and then, sent through the Ethernet link to the Display Unit (Sparc 2) and to the Transputer Server (PC). The process of digitizing the world-scenes is carried out by the Frame Grabber (Image Node) and is controlled by the transputer system. The low-level and high-level processing stages of the IUS are implemented by using a network of 32 T800 Tansputers linked to the Transputer Server (PC). The result of the processing by the high-level and low-level processing stages are then sent to Sparc 2.

The software configuration of the system is based on two operating systems: SunOS4 (Sparc 2) and Linux-1.6 (PC). Three high-level programming languages: C++, parallel C and OCCAM-2 have been used. Parts of the interfaces between the Transputer network and the PC have been written and implemented using assembly language.

6.1 Low-Level Processing Applications

In the low-level processing experimentation, the proposed technique of prime component decomposition and shape extraction are employed for the tasks of extraction of line graphs and path planning, isolation of occluding and abutting objects, and for the identification of stationary objects.

6.1.1 Extraction of Line Graphs and Path Planning

One of the most straightforward applications of the prime component decomposition is the extraction of line graphs from binary images. The process of the extraction of a line graph should result in a graph structure that is simple and can represent the object efficiently without redundant connections among the nodes of the graph. That is, the extraction process should preserve the graph topology and capture the essential symmetry properties of the graph structure which are important attributes to identify and map the object to the relational data structures.

The process of extraction of line graphs using the prime component decomposition performed by the low-level processing stage of the IUS is shown in Figure 6.2. Figure 6.2(a) shows a binary object. Figure 6.2(b) gives the first phase of a traditional extraction technique (without post processing). The extracted line graph performed by the traditional post processing techniques is shown in Figure 6.2(c). Figure 6.2(d) shows the result of the prime component decomposition of the object. Figure 6.2(e) displays the final result of the extraction of line graphs performed by the proposed IUS. It is important to notice the difference of the extracted line graphs obtained by using the traditional methods that destroy the object's symmetry (Figure 6.2(c)) and by using the prime component decomposition (Figure 6.2(e)) that preserves this symmetry. The simplicity of the line graph and preservation of the symmetry of the original object is essentially due to the use of the techniques of shape equalization and shape extraction developed in Chapter 4 using the prime components.

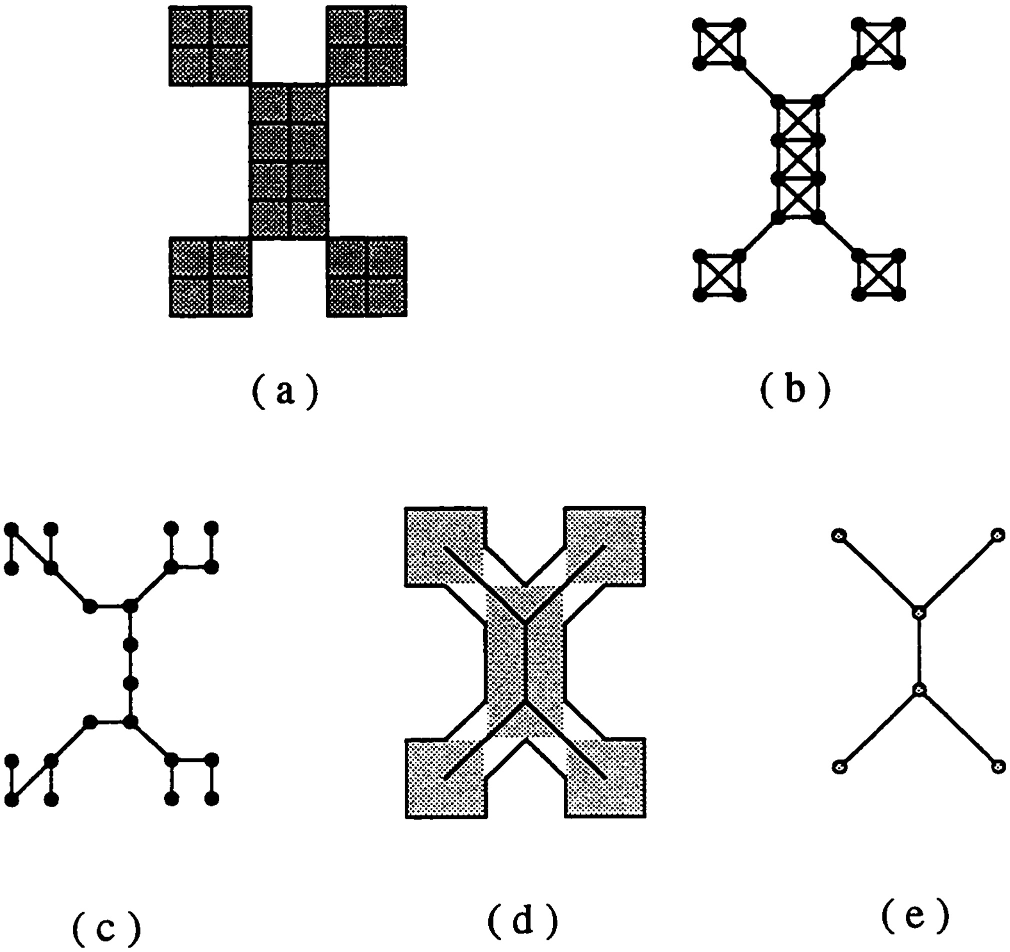


Figure 6.2: The process of extraction of line graphs. (a) The object containing non-deletable pixels. (b) The traditional graph representation of (a) in which the graph nodes correspond to the pixels. (c) The final result of extraction of line graphs performed by using the traditional methods based on deleting the graph's nodes. (d) The prime component decomposition of (a). (e) The result of extraction of line graphs by using the shape extraction technique.

The process of extraction of line graphs supports the tasks dealing with the adjacency, proximity and neighborhood relations among the prime components.

The path planning problem was originated in connection with building automatic mobile devices and constructing layouts of VLSI circuits. In order to demonstrate the main concepts, consider the square area containing some simple geometrical shapes as shown in Figure 6.3(a). The goal of the path planning problem is to determine the possible paths among the geometrical object in such a way that the distance from every point of the path to the closest object is maximized. Figure 6.3 shows the process of constructing the solution path using the prime component decomposition approach.

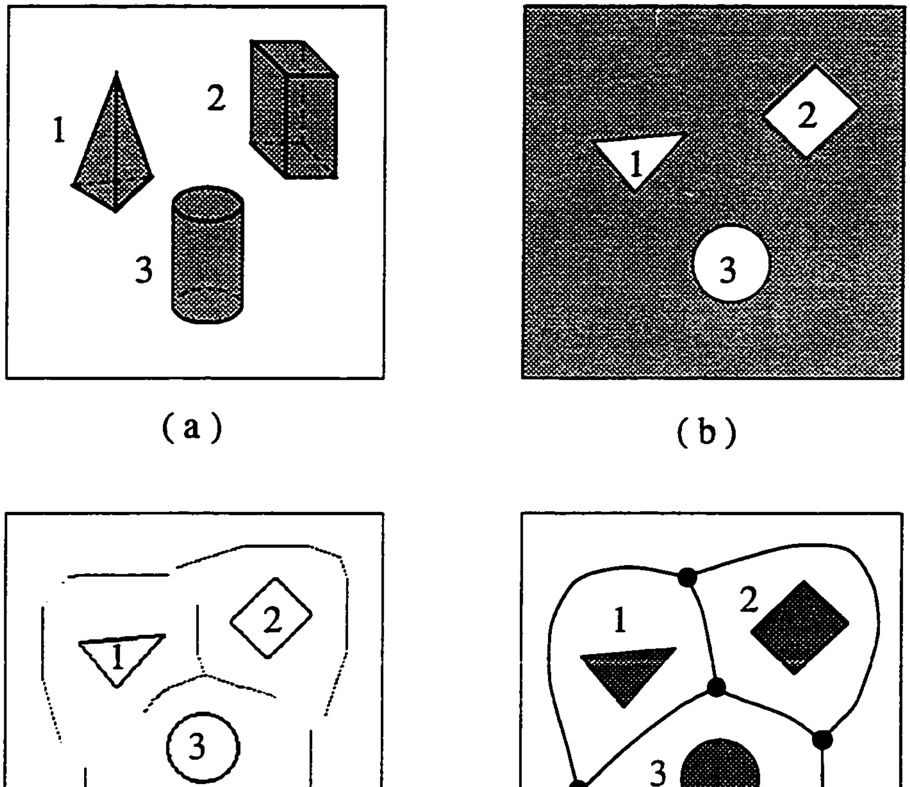


Figure 6.3: The process of finding paths among the objects. (a) A square area containing geometrical objects. (b) The binary image representing the top view. (c) Prime component decomposition of (b). (d) The resulting paths after the shape element extraction and equalization processes.

As it can be observed from Figure 6.3, the process of constructing the solution paths is composed of several steps. In the first step as shown in Figure 6.3(b), the binary image of the top view is constructed. This binary image contains the information of all possible regions where paths are allowed (gray area) and those where paths are forbidden (white regions). In the next step, the binary image is decomposed using the prime component decomposition (Figure 6.3(c)). The final solution of the path planning problem is obtained by utilizing the shape extraction and shape equalization algorithms as shown in Figure 6.3(d). The black dots shown in Figure 6.3(d) indicate connections among the paths, i.e, the connections among the equalized shape elements

6.1.2 Isolation of Occluding and Abutting Objects

In a typical world-scene, the isolation of the occluding and abutting objects is done together with their identification in high-level processing stage of an IUS. However, in specific cases, it is possible to perform a simple analysis of basic shape features within the low-level processing stage. For example, this type of analysis can indicate that certain subsets of the clusters of shape elements represents one single entity, i.e., one single object. Considering simple geometrical shapes, occlusion or abutting often results in areas with characteristic narrowings and bottlenecks. These areas can be identified as local minima along the prime skeleton of the occluding objects. Figure 6.4 shows an example of the process of isolating abutting geometrical objects by the low-level processing stage using the prime component decomposition approach. Figure 6.4(a) shows the binary scene containing abutting and occluding objects to be isolated. Figure 6.4(b) shows the prime component decomposition of the scene shown in Figure 6.4(a) with characteristic connections among prime skeletons of abutting and occluding objects. The isolation points (dark dots) among the abutting and occluding objects presented in Figure 6.4(c) have been found as local minima along these prime skeletons. It can be observed from Figure 6.4(c) that the isolation process has successfully isolated all the abutting and occluding objects within the scene.

6.1.3 Identification of Stationary Objects

The identification of stationary objects is the last application example of the prime component decomposition within the low-level processing stage of the

IUS, that is considered in this section. The approach of an object identification is based on the structure similarity between the general-level prime

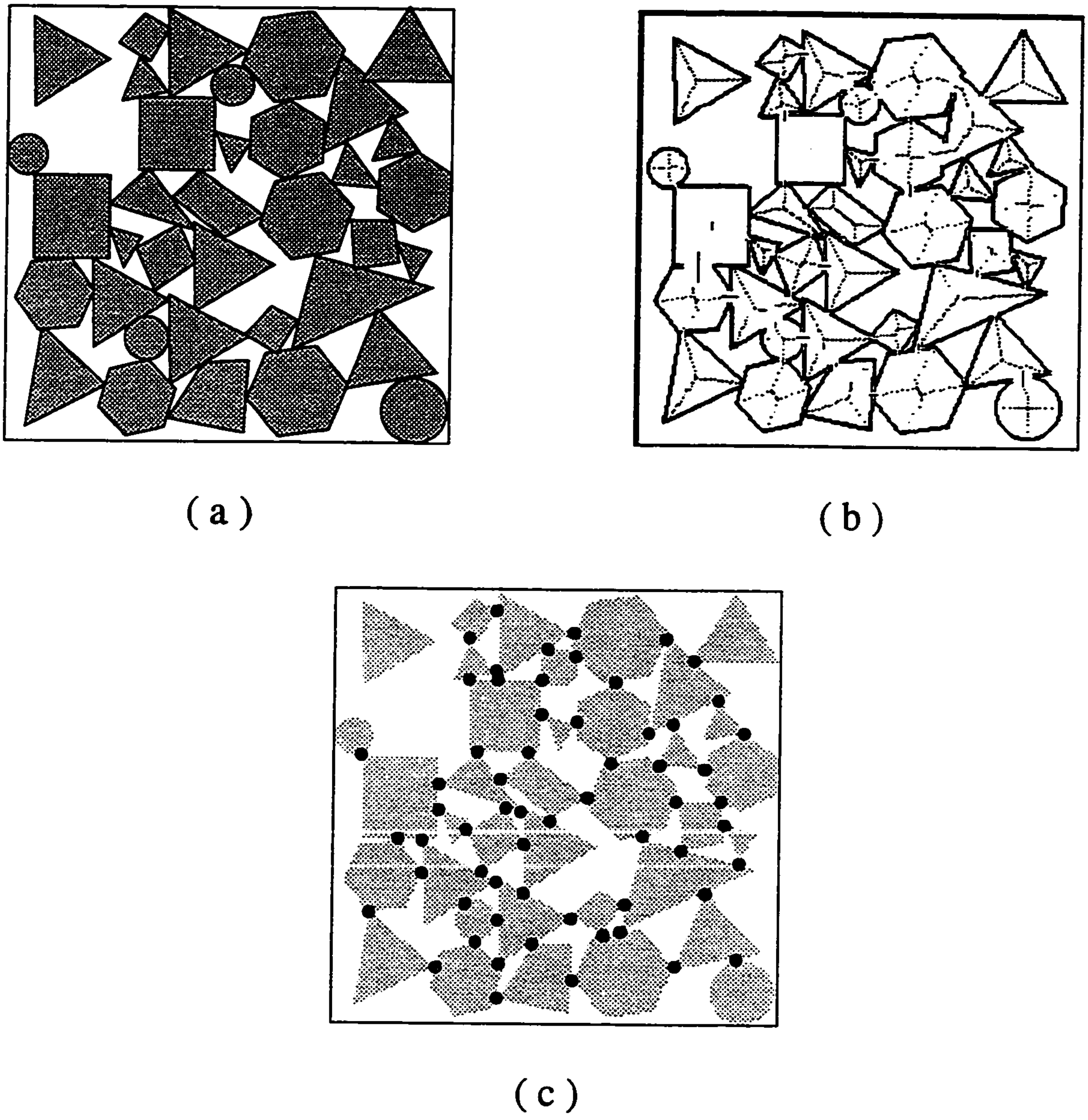


Figure 6.4: An example of the application of the prime component decomposition scheme to isolate abutting objects. (a) An input world-scene containing geometrical abutting objects. (b) Prime component decomposition of the input scene from (a). (c) Isolation of abutting objects. The dots indicate isolation of the abutting objects.

components (shape elements) and the partial reference tree structures within the knowledge base. We will utilize the merging algorithm in such a way, that the existence of structure similarity between the prime component and the partial tree structure triggers the additional action (algorithm) "display" within the low-level processing stage. This additional algorithm simply displays and highlights the corresponding general-level prime component within the input world-scene. In order for the identification process to be possible, the partial reference tree structures within the knowledge base of the system have to be created before the identification process starts. This is achieved by a learning process with a test sequence of input world-scenes that contain only the objects to be identified later. Figure 6.5 shows the complete process and the result of the identification of a stationary object (plane) within an input scene.

Figure 6.5(a) shows the gray-level input world-scene with the object plane to be identified. In order to perform the prime component decomposition, the input scene is divided into sixteen separate binary images each composed of one particular gray level, i.e, one quantization level. The number of quantization levels has been chosen based on the experience from earlier practical experiments. It has been noticed from these experiments that higher number of quantization levels does not generally improve the identification process, but it significantly increases the computational time and the required hardware resources. The sixteen binary images are decomposed using the prime component decomposition scheme. The result of the prime component decomposition for all the sixteen binary images is shown in Figure 6.5(b). In this example, the total number of prime components is equal to 9658. Each binary image is then transformed into shape elements and compared with the partial tree structures from the knowledge base and merged. Figure 6.5(c) shows fourth-level prime component (the shape elements). In the final identification phase, the identification procedure monitors the active prime components corresponding to the partial reference tree structure and displays them. Figure 6.5(d) shows the result of this identification, i.e., the active general-level prime component corresponding to the identified object.

6.2 High-Level Processing Applications

The applications of the prime component decomposition considered so far have been restricted to those that do not utilize the knowledge base at all or in which the knowledge

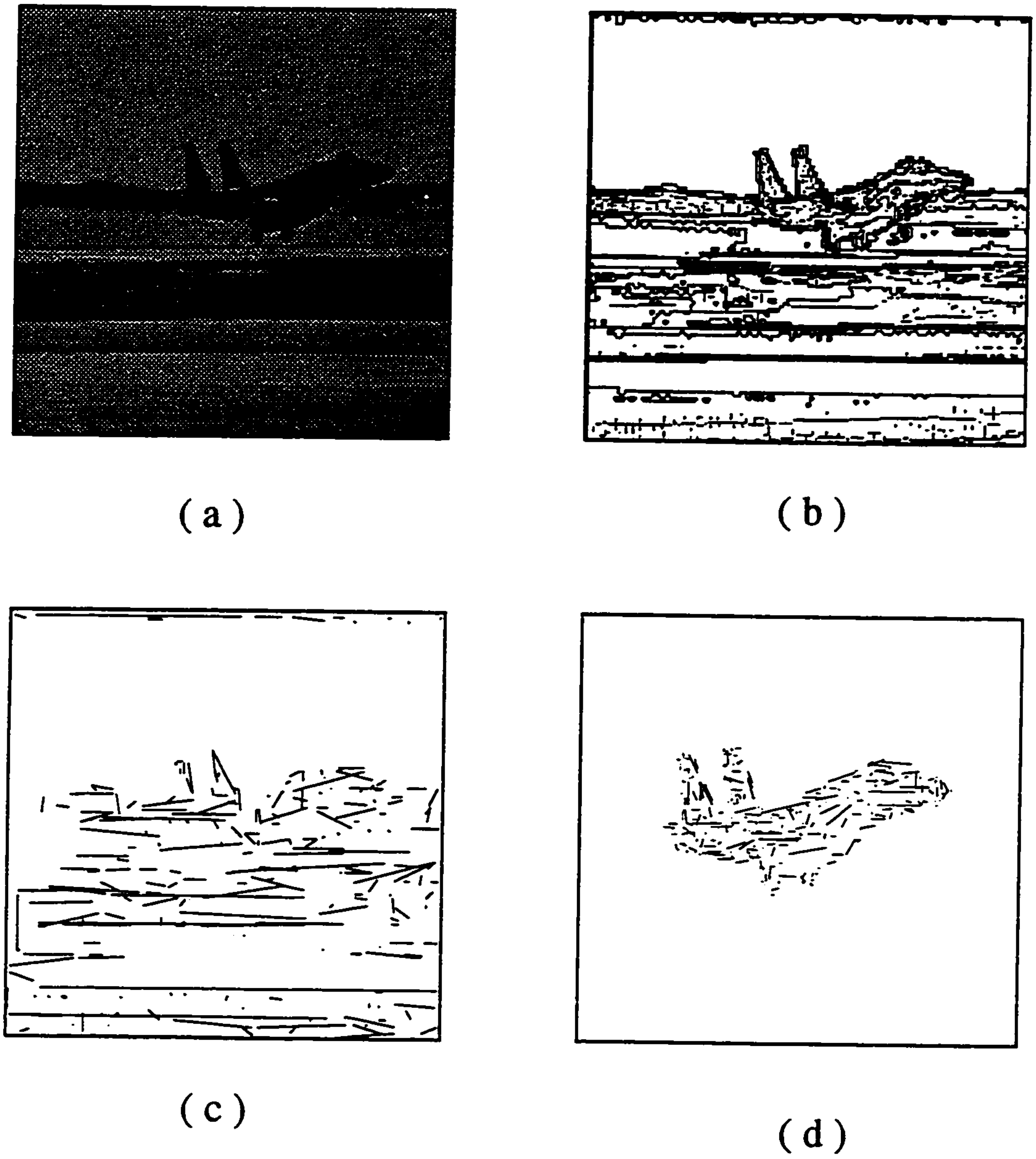


Figure 6.5: Results of the identification process of a stationary object (plane) within an input scene. (a) The input scene with the object plane to be identified. (b) The prime component decomposition of using sixteen binary frames obtained from the gray-level scene of (a). (c) The fourth-level prime component (shape element) obtained using the results of (b). (d) The final result of the object identification, i.e, active prime component of the identified object.

base is used mainly to compare the general-level prime components.

In this section, we present the examples of applications of the prime component decomposition involving more complex analysis and requiring a full interaction with the knowledge base by the high-level processing stage of the IUS. Since the high-level processing stage performs its computational tasks (access and modification of the knowledge base) at a higher level of abstraction, the domain of possible applications of the IUS increases significantly, including those that need on-line learning or those that require analysis of the relational links among the objects within the input scenes.

6.2.1 Identification of Moving Objects

In Section 6.1.3, the identification of stationary objects was carried out by utilizing a comparison procedure between the partial tree structures of the prime components obtained from the decomposition of the input scene and the reference partial tree structures (reference prime components) constructed during the initial learning phase. This initial learning phase is necessary to establish the static spatial reference relationships among the elements of the object to be identified. In this section, we are concerned with the application of the prime component decomposition with the identification of moving objects involving the high-level processing stage.

The goal of this section is to demonstrate an application of the prime component decomposition that does not require the initial learning phase. This is achieved by replacing the spatial reference relationships (identification criteria) with the time-varying and time-dependent relations. In this scheme, the necessary identification criteria used to identify the objects within a scene are constructed on-line during the identification process itself. Since the time-dependent relations within a scene can be constructed only by the analysis of the time-related activities, the identification process performed by the IUS has to include the analysis of a sequence of world-scenes. Figure 6.6 shows an example of the identification process of a moving object car within the sequence of input scenes. However, in this case, the model of the reference object does not exist and the high-level processing stage constructs the necessary relations on-line during the analysis of the sequence.

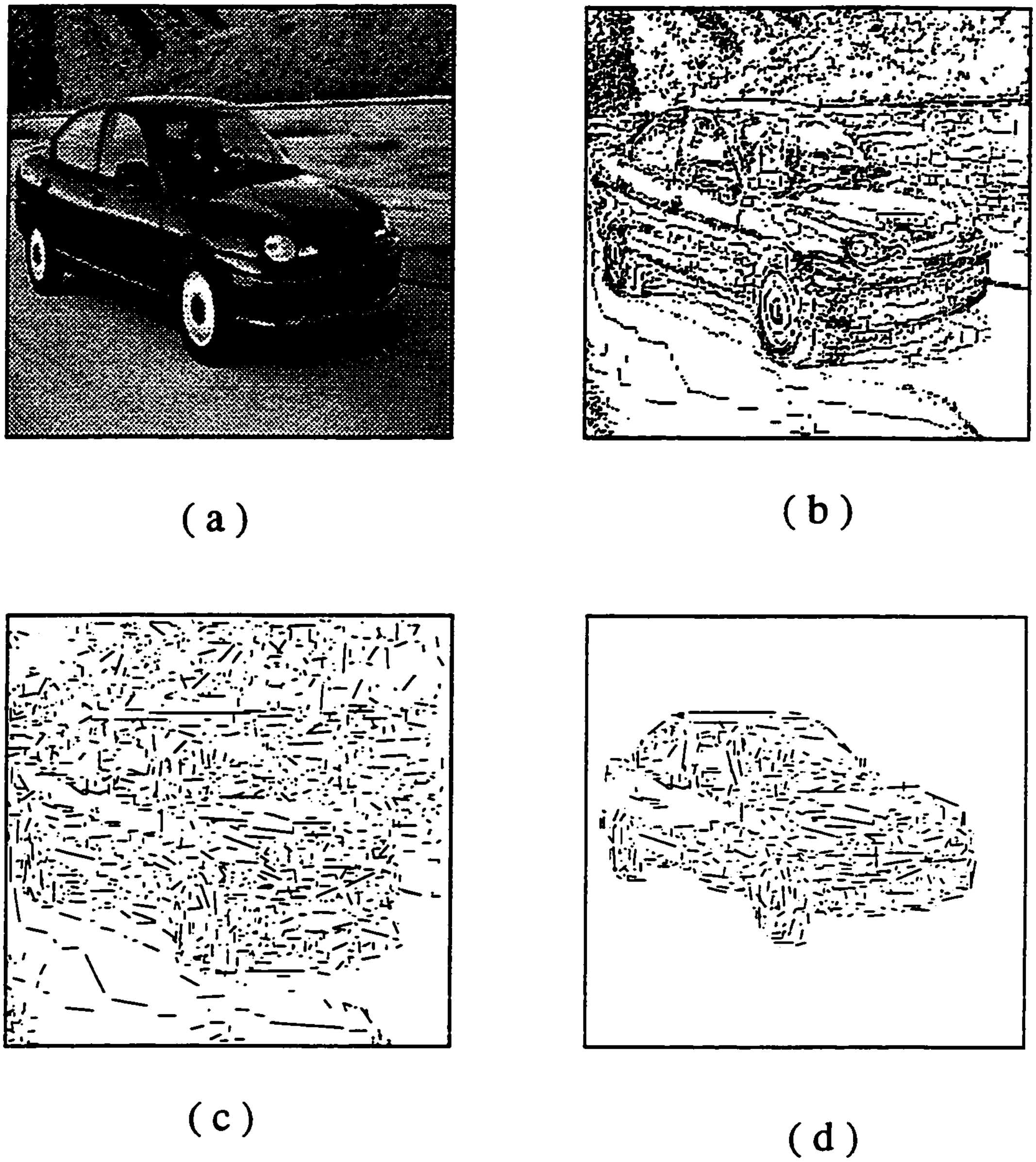


Figure 6.6: Results of the identification process of a moving object car within a sequence of scenes. (a) Frame number 80 from the sequence with the object car to be identified. (b) The prime component decomposition using sixteen binary planes obtained from the gray-level scene of (a). Every binary plane corresponds to one gray level. (c) The sixth-level prime component (shape element) obtained using the result of (b). (d) The final result of the object identification, i.e, active prime component of the identified object.

During the identification process, 100 frames comprising the input sequence of moving car are transformed and processed by the low-level processing stage, as in the case of stationary objects discussed in Section 6.1.3. Figure 6.6(a) shows the 80th frame from the sequence. Figure 6.6(b) shows the prime component decomposition of this frame while Figure 6.6(c) depicts its sixth general-level prime component. After the general-level prime components representing the sequence are merged into the knowledge base as active entities, the knowledge sources of the high-level processing stage perform the analysis of the general-level prime components within the knowledge base with respect to the variations in time. This analysis is carried out by specifying the rules within the high-level processing stage that control the activation (execution) of appropriate knowledge sources. One of the most important rules used in dealing with time-varying events describes the {condition <=> action} pairs among the prime components having different time-dependent characteristics. For example, in the present application this rule activates the fifth knowledge source of the high-level processing stage when there is the structure similarity between the general-level prime components of consecutive frames. The fifth knowledge source modifies the knowledge base by activating those similar prime components. This in turn activates the tenth knowledge source that displays the active general-level prime components (Figure 6.6(d)). Thus, the high-level processing stage of the IUS has identified the moving object car.

6.2.2 Identification of Time-Varying Objects

The identification of time-varying objects is one of the most difficult tasks faced by the high-level processing stage of the IUS. A time-varying object is understood as an object within an input sequence that does not have the time-independent spatial relations ,i.e, its form, configuration and relations are time-varying. One of the most typical examples of a time-varying object is a moving person within an input sequence.

There are several problems encountered during the identification process of time-varying objects. One of the most difficult ones is the lack of static, spatial reference relations that can characterize the time-varying objects. In the case of time-varying objects, the utilization of simple rules, i.e., the assumptions based on structure similarities between the same object belonging to two consecutive frames, would not work in the present situation of time-varying objects.

The way to overcome the difficulties of constructing reference spatial relations needed for the identification procedure is to update all potential reference prime components within the knowledge base. The selection of the potential prime components within the IUS is performed by utilizing the rule that activates the fifth knowledge source (see Section 6.2.1) and by monitoring the spatial relations between frames of the sequence. Those selected prime components are then updated "frame by frame" in order to construct a temporary prime component of the identified object - reference prime component. Every consecutive frame updates the reference prime component. The updating of the reference prime components is conceptually equivalent to the learning procedure in which the system "learns" to identify the object. As soon as the updating of the reference prime component is insignificant or can be anticipated, the high-level processing stage triggers the action (algorithm) "display". Figure 6.7 shows the results from the selected phases of the identification process of the time-varying object a moving person within the sequence.

The total number of frames of the input sequence used in the experiment is 100. Figure 6.7(a) shows the 95th frame from the input sequence. Figure 6.7(b) shows the prime component decomposition of this frame. The updated fifth-level prime component of the currently processed input frame depicted in Figure 6.7(c) shows the potential reference prime components considered. The final result of the identification is shown in Figure 6.7(d) in which the area of the reference prime component instead of the reference prime component itself is shown for better visualization.

6.3 Summary

In this chapter several applications of the prime component decomposition scheme used in the low-level and high-level processing stages of the proposed IUS have been presented.

The low-level processing applications involve extraction of line graphs, path planning, isolation of occluding and abutting objects, and identification of stationary objects. The process of the extraction of the line graphs preserves the graphs topology and captures the essential symmetry properties of the objects, which makes it very suitable to perform the tasks dealing with adjacency, proximity and neighborhood relations among the prime

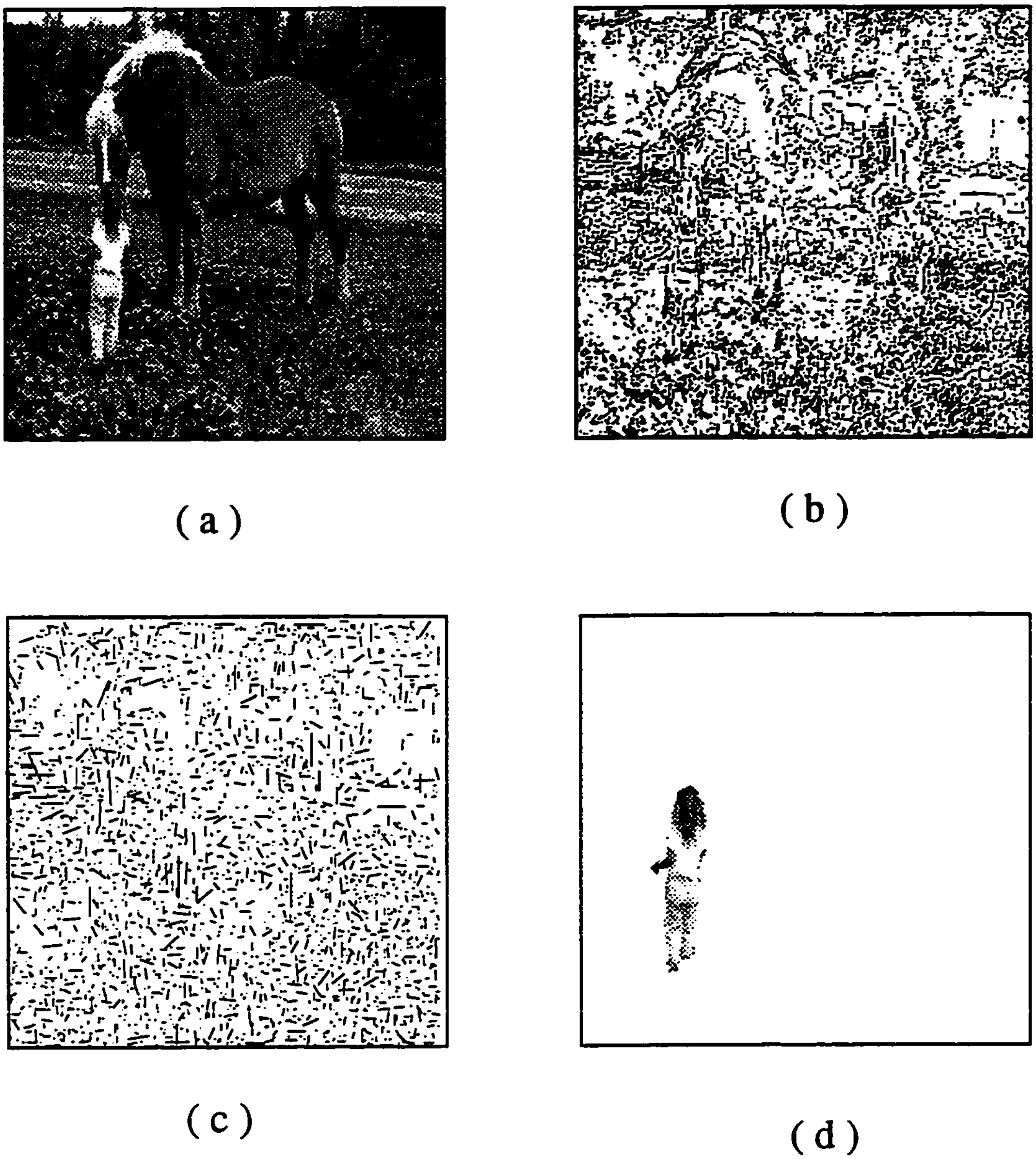


Figure 6.7: Results of the identification process of a time-varying object person within a sequence. (a) Frame number 95 from the sequence with the object person to be identified. (b) The prime component decomposition using sixteen binary planes obtained from the gray-level scene of (a). Each binary plane corresponds to one gray level. (c) The fifth-level prime component (shape element) obtained using the result of (b). (d) The final result of the object identification, i.e, the area of the currently processed input frame that corresponds to the active prime component of the identified object.

components. The application of the prime component decomposition to the path planning problem has been illustrated through an example of a rectangular area containing geometrical shapes. In this case, the low-level processing stage of the IUS has determined all possible paths among the objects within the room. The isolation of occluding and/or abutting objects is performed based on the analysis of the object's full skeleton and its local minima. The low-level processing stage has successfully isolated the abbuting objects within input world-scenes. The approach of the identification of the stationary object is based on the structure similarity between the general-level prime component representing the input scene and the reference prime component representing the object to be identified. The proposed identification method used in the low-level processing stage has identified stationary objects within input world-scenes.

Two applications, identification of moving objects and identification of time-varying objects, of the high-level processing stage have been considered. The identification of moving objects is achieved by the analysis of time-related activities within the sequence of input scenes. In the proposed approach, the analysis with respect to variations in time have been performed using specific rules defined in the high-level processing stage that control the execution of appropriate knowledge sources. Once the condition of the rule is satisfied, the activated knowledge source triggers the comparison process between the reference prime component and the prime component obtained from the decomposition of the currently processed scene. In the example considered, the IUS has identified the moving object within the sequence of the 100 input frames. The identification of time varying objects is based on updating the reference prime component representing the identified object. The updating process has been carried out by the knowledge sources that monitor spatial relations among the prime components of consecutive frames in a way similar to that of identifying moving objects. In the example considered, the IUS has identified the moving time-varying object within 100 input frames of the input sequence.

Chapter 7

CONCLUSION

7.1 Concluding Remarks

Processing of complex context-dependent multidimensional signals require constructing sophisticated expert systems capable of performing difficult tasks of knowledge acquisition, compilation, and transformation. In the filed of image processing these sophisticated systems known as image understanding systems are usually composed of a knowledge base and simpler subsystems performing tasks grouped into levels of hierarchical processing stages. Typically, one can distinct these processing stages as low-level and high-level processing stages. Traditionally, the low-level processing involves classical image processing tasks such as contrast enhancement, edge detection and mapping the image pixel values into an intermediate object description, while the tasks of the high-level processing is to generate description of complex image events and relations among them.

The main objective of this work has been to present a new prime component decomposition technique that reduces the redundant input information contained in input world-scenes by an effective extraction of maximum-size geometrical polygons (prime components), and to use this technique in developing an efficient model of an image understanding system (IUS). The characteristics sought in the IUS model have been in its ability to create a reliable knowledge representation, to produce a simple intermediate object description, to process time-varying information, and to avoid the use of computationally expensive algorithms.

The investigation of prime component decomposition has started by an analysis of generic relations among the polygons. These generic relations have been then examined in the context of real world-scenes in order to derive a new type of polygonal decomposition operator based on prime components. The mathematical analysis performed in the continuous metric space has indicated that the semi-rotational invariant polygons having a degree of rotation of $\pi/2$ radians are better decomposition elements than those having a degree of rotation of π, i.e, the optimal decomposition element in the continuous metric space should converge to a polygon having a degree of rotation tending to zero (circle). The decomposition operator has been then optimized in the discrete metric space to increase its implementation efficiency, yielding square decomposition elements.

As an intermediate object description generally relies on shape extraction, the proposed prime component decomposition has been used for shape extraction and equalization. The proposed approach of shape extraction has combined the description of an object's boundaries (edges) obtained by using a modified Sobel operator with a skeletonization process of the object's interior performed by square prime components in the discrete metric space. This approach has reduced the errors in the uncertainty of the description which might be critical in the high-level processing stage of an IUS. Further improvements in the shape extraction scheme have been achieved by introducing the concept of the shape equalization based on Fourier descriptor and nonlinear interpolation. This has helped in reducing the typical deficiencies of discrete medial axis transforms, such as noise sensitivity, the description errors of the diagonal objects and of objects having complex edges that rapidly change their curvature, and the description errors caused by a small sampling frequency. The shape equalization scheme has also reduced distortions and artifacts caused by a digitization process of input scenes and improved the utilization of the system resources. The proposed shape extraction technique based on prime component decomposition has been compared to two skeletonization algorithms, the Voronoi skeletonization algorithm and the morphological skeletonization algorithm. The simulation results have shown that the decomposition algorithm based on the proposed prime component decomposition scheme outperforms those based on the optimized Voronoi skeletons or morphological operators.

The process of developing an IUS has been divided into several overlapping phases. The model of the low-level processing stage essentially combines the processing techniques of prime component decomposition, shape extraction and shape equalization. A feedback between the low-level processing stage and the knowledge base of the IUS has also been introduced to decrease the memory and processing requirements of the system.

The high-level processing stage of the IUS has been implement based on the distributed problem solving black-board architecture suitable to deal with problems of very large search space, incomplete data, and imprecise and/or incomplete problem-solving knowledge. The undesirable computational actions of the high-level processing stage have been reduced by including a specialized control mechanism - the agenda based control. The opportunistic control mechanism has allowed to perform more effectively the goal-driven and data-driven reasoning by using incremental problem solving strategies and by employing a selective focusing mechanism. The rapid refocusing (at every control cycle) of the control mechanism has also reduced the number of computational steps of the high-level processing stage in arriving a solution.

The knowledge base of the IUS has been developed by optimizing the existing relational knowledge representation models and their data structures. In order to enhance the real-time performance of the IUS, these models have been carefully examined in the context of their flexibility, consistency, facility and inference schemes. The model of the knowledge base has been developed as a relational, hierarchical multidimensional tree structure composed of the data objects. The traditional data type definition has been extended to include the concepts such as labels, attributes, base data types and derived data types. In the proposed approach, the relational links between the data structures have incorporated within themselves the semantical, operational and time-varying knowledge. The proposed model of the knowledge base has avoided duplication of the internal data structures and allowed the data object of the knowledge base to be accessed efficiently, without conflicts by both high-level and low-level processing stages of the IUS.

The proposed prime component decomposition scheme and the IUS have been experimented with several low-level and high-level processing applications. In the low-level processing applications, examples of extraction of line graphs, path planning, isolation of occluding and abutting objects and identification of stationary objects have been considered. In the high-level processing applications, examples of identification of moving objects and identification of time-varying objects have been experimented. In all the low-level and high-level application examples, the proposed IUS has successfully performed the desired computational tasks and achieved the efficient solutions.

7.2 Suggestions for Future Investigations

During the development of the proposed IUS, many important issues have been addressed. However, there are still many areas that could be the subject for further investigation. For example, due to the focus on the main functions of the IUS itself, the implementation architecture has not been fully optimized. Consequently, one of the areas of future investigation should include the practical implementation issues of the IUS in the context of distributed network architectures. There are many topics in this area to be investigated, such as distribution of knowledge sources among the network of processors, partition of the processing algorithms, load balancing, inter-process communication, data transfer, synchronization, and memory coherence.

The elements of the low-level and high-level processing stages and of the knowledge base should be another topic of future investigation. It seems possible to merge the general-level prime components within the low-level processing stage on higher level of abstraction, by utilizing the context-dependent information obtained from the high-level processing stage. This should possibly further improve the efficiency of intermediate object representation and maximize the utilization of system resources. Another important element of the IUS that could improve its efficiency is the control system of the high-level processing stage. Several important issues related to the time-varying knowledge representation models such as knowledge acquisition of time-varying events, optimal searching or optimal comparison of partial tree structures would also need further attention. Finally, yet another open area for future research is the modeling of the IUS itself, as it concerns to self learning and inductive and non-inductive reasonings.

References

[1] G. M. AdelsonVelsky, V. L. Arlazarov, and M. V. Donskoy. Some methods of controlling the tree search in chess programs. *Artificial Intelligence*, 6(4):361–371, 1975.

[2] Narendra Ahuja. On approaches to polygonal decomposition for hierarchical image representation. *Computer Graphics and Image Processing*, (24):200–214, 1983.

[3] J. W. Brand V. R. Algazi. Continuous skeleton computation by voronoi diagrams. *Readings in Computer Vision: Issues, Problems, Principles and Paradigms Morgan Kaufmann Publisher, Inc. Los Altos CA*, pages 360–370, 1987.

[4] James F. Allen. Time and time again: the many ways to represent time. *International Journal of Intelligent Systems*, 6:341–355, 1991.

[5] James F. Allen. *Natural Language Understanding.* Benjamin/Cummings, Redwood City, California, 1995.

[6] Mark Allmen and Charles R. Dyer. Computing spatiotemporal relations for dynamic perceptual organization. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 58(3):338–351, November 1993.

[7] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. HUGIN—a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, volume 2, pages 1080–1085, Detroit, Michigan, August 1989. Morgan Kaufmann.

[8] M. Anthony and N. Biggs. *Computational Learning Theory.* Cambridge Tracts in Theoretical Computer Science (30). Cambridge University Press, 1992.

[9] Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge.* MIT Press, Cambridge, Massachusetts, 1990.

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

[11] H. Barrow and J. Tenenbaum. Computational vision. *Proc. IEEE*, 69(5):572–595, May 1981.

[12] Jean-Marie Beaulieu. Hierarchy in picture segmentation: A stepwise optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):150–163, February 1989.

[13] J. Beck. Similarity grouping and peripheral discriminability under uncertainty. *Amer. J. Psychol*, 85:1–19, 1972.

[14] Martin D. Bergevin, Robert; Levine. Part decomposition of objects from single view line drawings. *CVGIP Image Understanding*, 55(1):73–83, January 1992.

[15] Jonathan W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):116–124, January 1994.

[16] V. Ralph Brandt, Jonathan W.; Algazi. Continuous skeleton computation by voronoi diagram. *CVGIP Image Understanding*, 55(3):329–338, May 1992.

[17] Bruce G. Buchanan and Edward H. Shortliffe, editors. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, Massachusetts, 1984.

[18] L. Calabi. A study of the skeleton of plane figures. *Res. Park Mathematical Labs., Mass.*, SR20-60429, 1965.

[19] Zhixiang Chen and Wolfgang Maass. On-line learning of rectangles and unions of rectangles. *Machine Learning*, 17(2/3):23–50, 1994.

[20] L. Console and P. Toraso. An approach to the compilation of operational knowledge from causal models. *IEEE Transactions on Systems, Man, And Cybernetics*, 22:773–789, 1992.

[21] T. R. Crimmins. A complete set of fourier descriptors for two-dimensional shapes. *IEEE Transactions on Systems, Man, And Cybernetics*, 6:848–855, 1982.

[22] W. Bruce Croft. Knowledge-based and statistical approaches to text retrieval. *IEEE Expert*, 8(2):8–12, 1993.

[23] Michael F. Dacey. A two dimensional language for a class of polygons. *Pattern Recognition*, 3:197–208, 1971.

[24] Ernest Davis. *Representations of Commonsense Knowledge.* Morgan Kaufmann, San Mateo, California, 1990.

[25] Larry S. Davis and Thomas C. Hendereson. Hierarchical constrains for shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):265–277, May 1981.

[26] J. Doran and D. Michie. Experiments with the graph traverser program. *Proceedings of the Royal Society of London*, 294, Series A:235–259, 1966.

[27] J. C. (Editor). *Machine Learning: Paradigms and Methods.* MIT Press, 1990.

[28] L.D. Erman. The hearssay-ii speech understanding system: integrating knowledge to resolve uncertainty. *CS*, 2:213–253, 1980.

[29] S. Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

[30] John Freeman. The modelling of spatial relations. *Computer Graphics and Image Processing*, (4):156–171, 1975.

[31] M. Fulk, S. Jain, and D. Osherson. Open problems in systems that learn. *J. of Comput. Syst. Sci.*, 1994. To appear.

[32] K. Ghosh, Pijush. An algebra of polygons through the notion of negative shapes. *CVGIP: Image Understanding*, 54(1):119–144, July 1991.

[33] G. H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Trans. Comput.*, 3:195–281, 1972.

[34] Eitan M. Gurari and Harry Wechsler. On the difficulties involved in the segmentation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(3):304–306, May 1982.

[35] H. Altay Güvenir and Izzet Sirin. Classification by feature partitioning. *Machine Learning*, 23(1):47–67, 1996.

[36] Blum H. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form. MIT Press*, 1961.

[37] Edwin R. Hancock and Josef Kittler. Edge-labeling using dictionary-based relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(2):165–181, February 1990.

[38] M. Hu. Visual pattern recognition by moment invariants. *IRE. Inform. Theory*. IT-8:178–187, 1962.

[39] Gregory M. Hunter and Kenneth Steiglitz. Operations on images using quad trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):145–153, April 1979.

[40] Sprague A. P.; Donahue M. J.; Rokhlin S. I. A method for automatic inspection of printed circuit boards. *CVGIP Image Understanding*, 54(3):401–415, November 1991.

[41] Tadao Ichikawa. A pyramide representation of images and its feature extraction facility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 3(3):257–265, May 1981.

[42] A. Jolion, J. M.; Montanvert. The adaptive pyramid: A framework for 2d image analysis. *CVGIP Image Understanding*, 55(3):339–349, May 1992.

[43] Jean-Michel Jolion. Computer vision methodologies. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):53–71, January 1994.

[44] Peter; Jolion, Jean-Michel; Meer and Samira Bataouche. Robust clustering with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(8):791–802, August 1991.

[45] B. Julesz. Visual pattern discrimination. *IRE Trans. Inform. Theory*, IT-8:84–92, 1962.

[46] Andersen S. K., Olesen K. G., Jensen F. V., and Jensen F. HUGIN—a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, volume 2, pages 1080–1085, Detroit, Michigan, August 1989. Morgan Kaufmann.

[47] G. C. Stockman L. N. Kanal. Problem reduction representation for the linguistic analysis of waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:287–299, 1983.

[48] J.C Kotelly. A mathematical model of blum's theory of pattern recognition. *Airforce Cambridge Res. Labs. Mass.*, 1963.

[49] Frank P. Kuhl. Elliptic fourier features os a closed contur. *Computer Graphics and Image Processing*, 18:236–258, 1982.

[50] Chia-Hoang Lee. Time-varying images: The effect of finite resolution on uniqueness. *CVGIP Image Understanding*, 54(3):325–332, November 1991.

[51] Soo-Young Lee and J. K. Aggarwal. A system design/scheduling strategy for parallel image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(2):194–204, February 1990.

[52] M. Lindenbaum and J. Koplowitz. A new parameterization of digital straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(8):847–852, August 1991.

[53] Y. Lu and R. C. Jain. Reasoning about edges in scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):450–468, April 1992.

[54] M. Macieszczak and M. Omair Ahmad. On the polygonal decomposition in low level processing for image understanding. *in Proc Canadian Conference in Electrical and Computer Engineering*, 2:345–349, 1994.

[55] M. Macieszczak and M. Omair Ahmad. Extraction of shape elements in low-level processing for image understanding systems. *in Proc Canadian Conference in Electrical and Computer Engineering*, 2:411–416, 1995.

[56] M. Macieszczak and M. Omair Ahmad. Shape equalization of binary images using fourier descriptor interpolation in low-level processing for image understanding. *in Proc Canadian Conference in Electrical and Computer Engineering*, 2:457–460, 1995.

[57] M. Minsly. A framework for representing knowledge. *Mind Design The MIT Press*, 1981.

[58] U. Montanari. A method for obtaining skeletons using quasi-eculidean distance. *J. Assoc. Comput. Mach.*, JACM-15, 1968.

[59] J. C. Mott-Smith. Medial axis transformations in picture processing and psychopictorics. *B. S. Lipkin and A. Rosenfeld editors, Academic, Press*, 1970.

[60] V. V. Alexandrov N. D. Gorsky S. N. Mysko. Recursive pyramids and their use for image coding. *Pattern Recognition*, 2:301–310, 1984.

[61] A. Nowell. The knowledge level. *Artifical Intelligence*, 18:87, 1982.

[62] Y. Cohen M. S. Landy M. Pavel. Hierarchical coding of binary images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:284–298, 1985.

[63] E. Person. Shape discrimination using fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:388–397, 1986.

[64] Maria Petrou and Josef Kittler. Optimal edge detectors for ramp edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(5):483–491, May 1991.

[65] D. Poole. A methodology for using a default and abductive reasoning system. *International Journal of Intelligent Systems*, 5(5):521–548, December 1990.

[66] D. Poole. Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities. In *Proc. 13th International Joint Conf. on Artificial Intelligence*, pages 606–612, Chambery, France, August 1993.

[67] D. Poole. Representing diagnosis knowledge. *Annals of Mathematics and Artificial Intelligence*, 11:33–50, 1994.

[68] Da. Poole. Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks. *Artificial Intelligence*, 88:69–100, 1996.

[69] O. D. Faugeras K. E. Price. Semantic descriptor of aerial images using stochastic labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:633–642, 1981.

[70] Bernd Radig. Image sequence analysis using relational structures. *Pattern Recognition*, 17(1):161–167, 1984.

[71] A. Rattarangsi and R. T. Chin. Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):430–449, April 1992.

[72] T. H. Reiss. The revised fundamental theorem of moment invarients. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(8):830–834, August 1991.

[73] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):94–115, January 1994.

[74] Azriel Rosenfeld. Axial representation of shape. *Computer Graphics and Image Processing*, 33:156–173, 1986.

[75] Azriel Rosenfeld. Axial representation of shape. *Computer Vision Graphics and Image Processing*, 33:156–173, 1986.

[76] Cinque L.; Guerra C.; Levialdi S. Computing shape description transforms on a multiresolution architecture. *CVGIP Image Understanding*, 55(3):287–295, May 1992.

[77] Foresti Gianluca; Murino Vittorio; Regazzoni Carlo S.; and Vernazza Gianni. Grouping of rectilinear segments by the labeled hough transform. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):22–42, January 1994.

[78] Eric Saund. Identifying salient circular arcs on curves. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 58(3):327–337, November 1993.

[79] A. V. Oppenheim R. W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, 1989.

[80] R. B. Scherl and H. J. Levesque. The frame problem and knowledge-producing actions. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 689–695, Washington, D.C., July 1993. AAAI Press.

[81] Gunther Schrack. Finding neighbors of equal size in linear quadtrees and octrees in constant time. *CVGIP Image Understanding*, 55(3):221–230, May 1992.

[82] M.; Seibert and A. M. Waxman. Adaptive 3-d object recognition from multiple views. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 14(2):107–124, February 1992.

[83] J. Serra. Image analysis and mathematical morphology. *Academic Press, New York*, 1982.

[84] L. Shapiro and A. Rosenfeld. *Computer Vision and Image Processing.* L. Shapiro, A. Rosenfeld, 1992.

[85] Linda G. Shapiro. Data structure for picture processing: A survey. *Computer Graphics and Image Processing*, (11):162–184, 1979.

[86] Linda G. Shapiro. A structural model of shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(2):111–126, March 1980.

[87] Linda G. Shapiro and Robert M. Haralick. Organization of relational models for scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6):595- 602, November 1982.

[88] Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(2):131–147, February 1990.

[89] Louise Stark and Kevin Bowyer. Function-based generic recognition for multiple object categories. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):1–21, January 1994.

[90] Hideyuki Tamura and Naokazu Yokoya. Image database systems: A survay. *Pattern Recognition*, 17(1):29–43, January 1984.

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

[92] Robert E. Tarjan. *Data Structures and Network Algorithms.* CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM (Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1983.

[93] Mihran Tuceryan and Anil K. Jain. Texture segmentation using voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(2):211–216, February 1990.

[94] A. J. Vayda and A. C. Kak. A robot vision system for recognition of generic shaped objects. *CVGIP: Image Understanding*, 54(1):1–46, July 1991.

[95] Deborah Walters. Selection of image primitives for general-purpose visual processing. *Computer Vision Graphics and Image Processing*, (37):261–298, 1987.

[96] Li Wang and Theo Pavlidis. Detection of curved and straight segments from gray scale topography. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 58(3):352–365, November 1993.

[97] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.

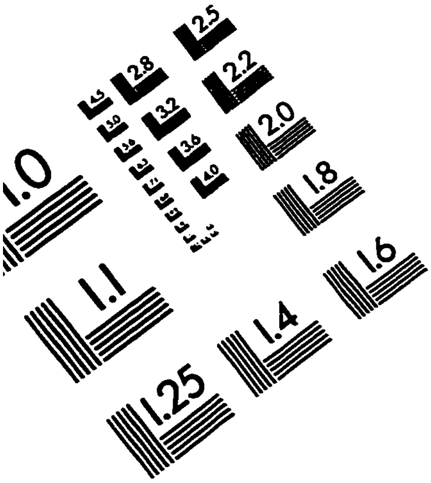
[98] Kenneth L. Williams. A multidimensional approach to syntactic pattern recognition. *Pattern Recognition*, 7:125–137, 1975.

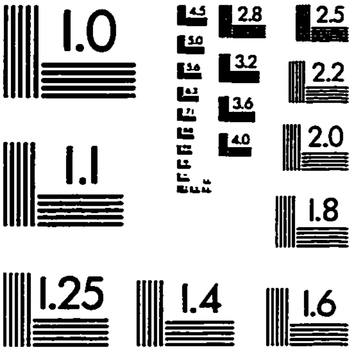
[99] T. Winograd. Computer models of thought and language. *Morgan Kaufmann, Inc.*, 1977.

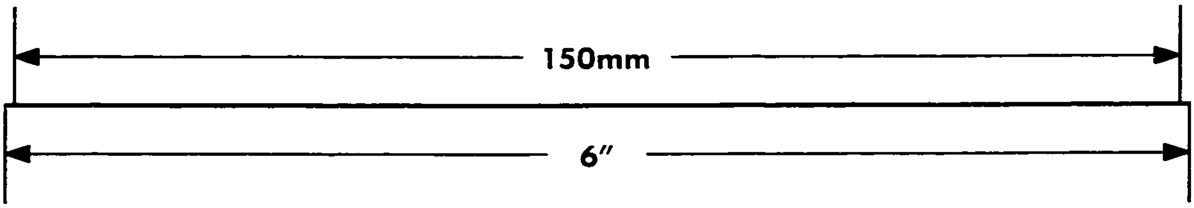
[100] Jian; Wohn, Kwang Yoen; Wu and Roger W. Brockett. A contour-based recovery of image flow: Iterative transformation method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(8):746–760, August 1991.

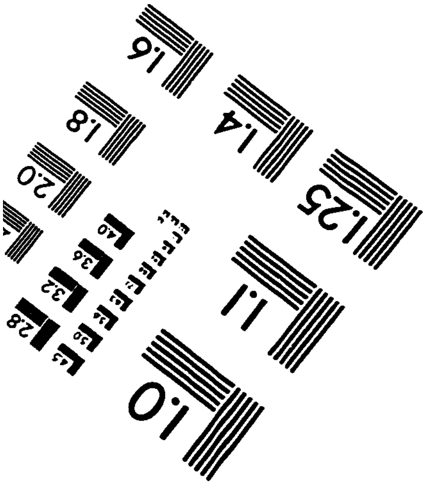
[101] Naokazu Yokoya and Martin D. Levine. Volumetric shapes of solids of revolution from a single-view range image. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):43–52, January 1994.

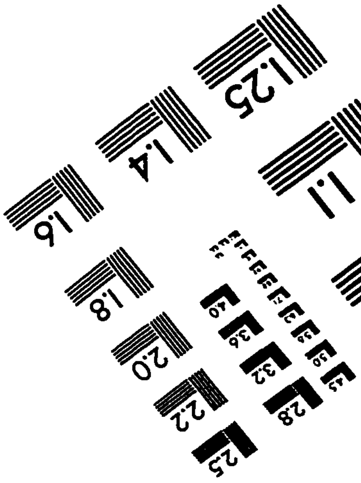
IMAGE EVALUATION TEST TARGET (QA-3)











APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989