


**Object-Based Video Retrieval Based on  
Motion Analysis and Description**



# Object-Based Video Retrieval Based on Motion Analysis and Description

*Aishy Amer*

 Université du Québec  
**Institut national de la recherche scientifique**  
INRS-Télécommunications  
16, place du Commerce, Verdun  
Québec, Canada, H3E 1H6

June 2, 1999

Rapport technique de l'INRS-Télécommunications no. 99-12

## Summary

Finding video of interest in a large database is a problem of increasing importance given the ever-increasing amount of available video information. Examples are the digital video archives in various domains such as arts (e.g., art history archives), environment (e.g., the Berkeley environmental digital library), and science (e.g., the digital library project of NASA). Consequently, the current highly-active research and development of video retrieval tools is an application-driven issue.

In most existing video retrieval systems, the user first either sketches or specifies an image sequence, e.g., after browsing the database, he or she is looking for. Then, the system retrieves the video query based on basic video features such as motion or color. However, browsing in large databases can be time consuming and sketching is a difficult task especially for complex scenes. In this work, an alternative query approach based on qualitative description of video content is proposed. Here, the user can give a qualitative description of a query video by specifying global video features, such as global motion, and object features, such as basic feature (e.g., color), spatial relationship features (e.g., object  $i$  is close to object  $j$ ), location features (e.g., object  $i$  is in the bottom of the image), and semantic features (e.g., object action: object  $i$  moves left and then disappears).

An advantage of such a retrieval strategy is that it allows the construction of intuitive queries based on the observation that most people's interpretation of real world domains is imprecise and that users, while viewing a video, usually memorize objects ("who" is in the scene), their action ("what" he/she is doing), and their location ("where" the action takes place). In the absence of a specific application, such a generic model allows extensibility (e.g., by introducing new definitions of object actions).

The emphasis of the proposed retrieval approach is, therefore, on video analysis and interpretation methods allowing linking of basic visual features to high-level semantics (e.g., actions and events). Since in a retrieval application, fast response is expected, fast components of the proposed object-based retrieval system are introduced.

### Keywords

Content-based video retrieval, Spatio-temporal object detection, Image segmentation, Morphological operators, Feature extraction, Motion estimation, Object tracking, Similarity measures.

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| 1.1      | Why content-based video retrieval . . . . .                | 1         |
| 1.2      | Definitions . . . . .                                      | 2         |
| 1.3      | Models for content-based video retrieval . . . . .         | 3         |
| 1.4      | Problem statement . . . . .                                | 6         |
| 1.5      | Outline of the proposal . . . . .                          | 7         |
| <b>2</b> | <b>Related Work</b>  | <b>8</b>  |
| 2.1      | Video retrieval approaches . . . . .                       | 8         |
| 2.2      | Shot boundary detection . . . . .                          | 9         |
| 2.3      | Spatio-temporal object detection . . . . .                 | 10        |
| 2.4      | Motion estimation . . . . .                                | 13        |
| 2.5      | Video interpretation . . . . .                             | 14        |
| 2.6      | Similarity measures . . . . .                              | 15        |
| <b>3</b> | <b>Proposed Object-based Retrieval</b>                     | <b>17</b> |
| 3.1      | Motivation . . . . .                                       | 17        |
| 3.2      | Retrieval strategy . . . . .                               | 17        |
| 3.3      | Spatio-temporal object detection . . . . .                 | 19        |
| 3.4      | Global motion estimation . . . . .                         | 22        |
| 3.5      | Interpretation of global motion . . . . .                  | 22        |
| 3.6      | Interpretation of spatio-temporal objects . . . . .        | 22        |
| 3.7      | Object feature vector . . . . .                            | 25        |
| 3.8      | Shot feature vector . . . . .                              | 25        |
| 3.9      | Video shot pruning . . . . .                               | 26        |
| 3.10     | Similarity test . . . . .                                  | 27        |
| <b>4</b> | <b>Progress to Date</b>                                    | <b>29</b> |
| 4.1      | Intra-image segmentation . . . . .                         | 29        |
| 4.2      | Estimation of motion using object correspondence . . . . . | 40        |
| <b>5</b> | <b>Work Plan</b>   | <b>49</b> |
| <b>6</b> | <b>Anticipated Contributions</b>                           | <b>50</b> |
|          | <b>References</b>  | <b>51</b> |

# List of Figures

|    |  |    |
|----|--|----|
| 1  | Video units . . . . .  | 2  |
| 2  | . . . . .  | 4  |
| 3  | Video analysis . . . . .   | 5  |
| 4  | Representation versus interpretation . . . . .   | 6  |
| 5  | Architecture of the proposed video retrieval . . . . .   | 19 |
| 6  | Proposed query form . . . . .  | 20 |
| 7  | Specification of object locations . . . . .  | 23 |
| 8  | Video shot representations . . . . .   | 26 |
| 9  | The proposed intra-image segmentation . . . . .  | 29 |
| 10 | . . . . .  | 30 |
| 11 | Directions of the texture analysis . . . . .   | 32 |
| 12 | Directions of the homogeneity analyzer . . . . .   | 33 |
| 13 | . . . . .  | 34 |
| 14 | . . . . .  | 35 |
| 15 | . . . . .  | 36 |
| 16 | Dilation and Erosion (Note that in this figure the operations are applied to the black pixels) . . . . . | 36 |
| 17 | Comparison of morphological edge detectors . . . . .   | 38 |
| 18 | . . . . .  | 38 |
| 19 | . . . . .  | 39 |
| 20 | . . . . .  | 41 |
| 21 | The proposed object-matching approach . . . . .  | 42 |
| 22 | . . . . .  | 43 |
| 23 | Correspondence conflict . . . . .  | 43 |
| 24 | Detection of translational motion . . . . .  | 44 |
| 25 | . . . . .  | 44 |
| 26 | . . . . .  | 45 |
| 27 | . . . . .  | 45 |
| 28 | . . . . .  | 47 |
| 29 | . . . . .  | 47 |
| 30 | . . . . .  | 48 |

# 1 Introduction

Visual information is becoming integrated into all areas of modern communication, even in such low-bandwidth areas as mobile communication. Thus, effective techniques for analysis, description, manipulation, and retrieval of visual information are important. Advanced visual data analysis, in general, and object-based techniques, in particular, have therefore become highly active fields of research. Examples are the current compression standardization processes, MPEG-4 and MPEG-7, and various video digital libraries, image and video retrieval projects.

Given the ever-increasing amount of information (text, image, video, and audio), finding items of interests in a large database is a problem of increasing importance. Considering the establishment of large digital video archives in various domains such as arts (e.g., art history archives), environment (e.g., the Berkeley environmental digital library), science (e.g., the digital library project of NASA), and politics (e.g., on-line news archives), the current highly active research and development of video retrieval tools is an *application-driven* issue. Key issues in such systems are how to query, retrieve, and display desired information from large collections, e.g., “How can a biologist retrieve a specific video of animals or plants from a large database” or “How can a journalist find a specific video clip from a large collection of video tapes, ranging from sport to history, from politics to science?”.

## 1.1 Why content-based video retrieval

Today, text-based mechanisms such as Structured Query Language (SQL) strings are the most used and accurate methods for finding “unconstrained” video [12]. They require that text annotations of videos be available. A video sequence consists of various spatial and temporal object relationships and information. Such relationships and information are perceived and interpreted differently by different people. To represent all the different interpretations by keywords is a difficult task as one cannot foresee all possible interpretations of the data during text-based indexing. Furthermore, there is a need to capture context which is usually a highly manual effort. Also, the use of context and keywords will change over time (e.g., the movie “2001 a space odyssey” was originally a science fiction film. What will its categorization be in the year 2001?). Clearly, textual annotation is time consuming and suffers from subjectivity of the human operator.

The above problems have created a great demand for automatic and effective techniques for *content-based* video retrieval systems allowing content-based retrieval for a more independent access to information, e.g., for self-education. The trend in currently developed retrieval systems is to use extracted visual content to complement the text-based approaches. Furthermore, if audio data is available, speech analysis and recognition techniques can be used. The synergy between the textual and the audio-visual features can be used to increase retrieval accuracy. The issue here is, however, how to automatically solve inconsistency between the different features of the same data.

## 1.2 Definitions

Digital video has established itself as one of the most important and useful media. It is probably the fastest-growing section of telecommunication, driven by entertainment, teleconferencing, and security applications. Digital video is also the most difficult to handle. This is because of the wealth of syntactical/semantical and spatial/temporal object information it contains.

A *video* or a *video sequence* (e.g., a movie or a news clip) is hierarchical in nature (Fig. 1). It can be seen as a set of story units (e.g., news about flooding). Each story unit contains a set of scenes. A *scene* is a set of interrelated shots which are unified by the same point of interest (e.g., a dramatic incident). A *shot* is defined as one or more images (an *image sequence*) recorded contiguously that represents a continuous action in time or space (e.g., a zoom of a homeless person walking). There is little change in the visual content of a shot, i.e., within the images of a shot there is a short-term temporal consistency.

A shot consists of multiple real-world objects and captures their *semantic* (i.e., object meaning), their *dynamic* (i.e., object movement or action), and their *syntax* (i.e., the way objects are combined to form an image sequence, such as spatial/temporal inter-object relationships, e.g., “a person is near his house”). A shot is commonly supposed to be composed of rigid objects or objects consisting of rigid parts connected together. Two shots are separated by a *cut* which is a transition at the image boundary between two successive shots. A cut can be thought of as an “edge” in time.

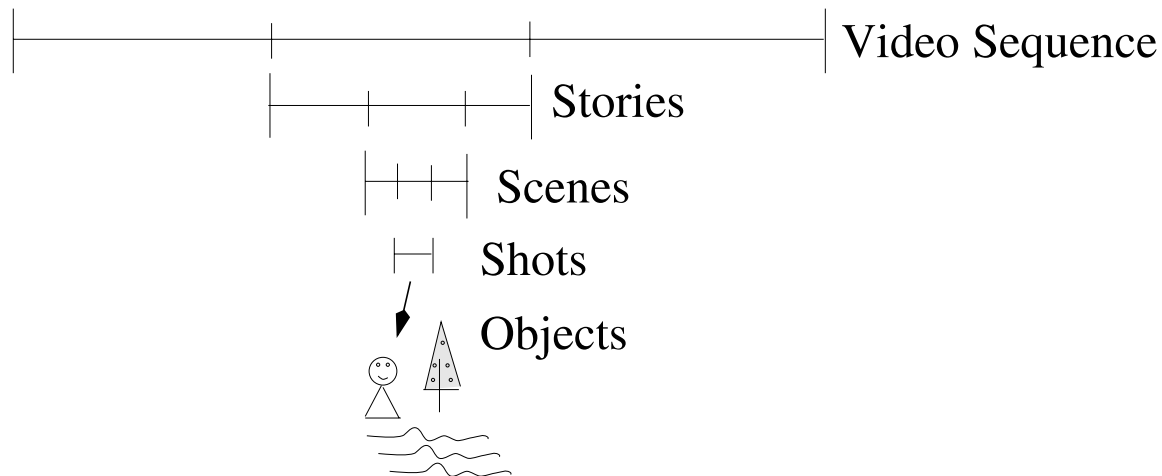


Figure 1: Video units

A *2-D object* is a projection of a 3D real-world object (e.g., a tree) onto the image plane. In the following, the term *object* will refer to a 2D planar object. Objects have basic (low-level) features such as texture or motion. In general, video analysis detects regions showing specific basic homogeneity (e.g., color, motion or texture). One region or a set of such regions may correspond to an object. Throughout a shot,



objects can be seen as entities that are both temporally and spatially coherent.

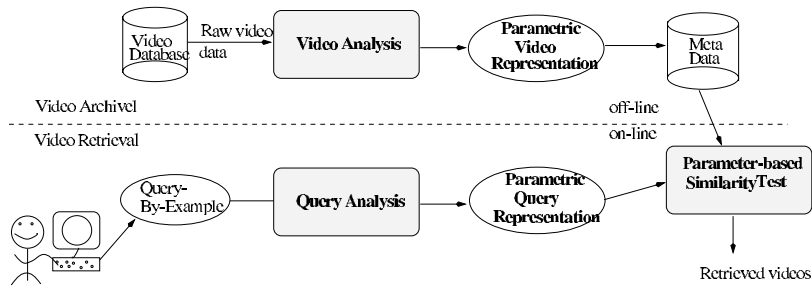
### 1.3 Models for content-based video retrieval

In a video retrieval system, a feature vector of a query shot is computed, compared to the feature vectors in the feature database, and shots most similar to the query are returned to the user. Depending on the representation of this feature vector and the similarity test used, three basic models for video retrieval can be defined (Fig. 2). In the first model (Fig. 2(a)), the user selects a video query, e.g., after browsing the video database. The video analysis then, in general, extracts a real-valued parameter vector (a representation) that summarizes the content of the video. Then, in the similarity test, this parameter vector is compared to stored parameter vectors and the video sequences most similar to the video query are selected. Comparison based on real-valued parameters can be expensive, in particular when the dimension of the parameter vector is high. Therefore, a reduced representation is needed. Although a similarity test in the parameter space is a natural choice for video queries, it is limited to the case of query by example where parameter vectors are computed from both the database video shots and the query (Fig. 2(a)). The similarity test in this case must compare the corresponding database and query parameters via a suitable norm (multiple parameters can be thought of as vectors). The norms for different sets of parameters (different physical meaning) must be combined (e.g., weighted) in order to arrive at a single decision parameter.

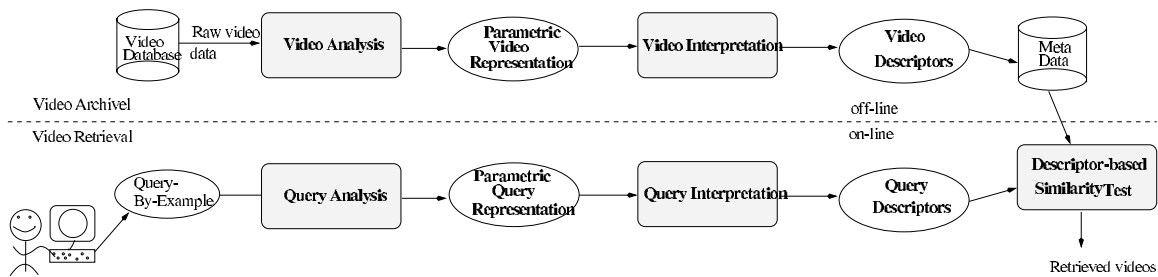
In the second model (Fig. 2(b)), the user selects a video query and the system finds a parameter vector and then interprets this parameter vector in order to obtain discrete parameters and qualitative descriptors (e.g., quantization of 6 affine motion parameters and then interpretation as a zoom). The result is a (reduced) qualitative descriptor vector. Here, the similarity test is based on the qualitative descriptors. An alternative is a similarity test in the descriptor space (Fig. 2(b)). Here, parameter vectors are transformed into descriptor vectors (interpretation step) that undergo a similarity test. Again, corresponding sets of database and query descriptors are compared. Descriptor sets with different physical interpretation are combined into a single decision parameter. Note that a similarity test in the descriptor space is applicable to the query-by-example and query-by-description cases (Fig. 2(c)).

In the third model (Fig. 2(c)), the user can specify a qualitative description of the video query and the system compares this description with the stored descriptions in the database. Such a model is useful when the user cannot specify a video but has memorized a vague description of it (e.g., global motion: zoom; a blue object entering the scene).

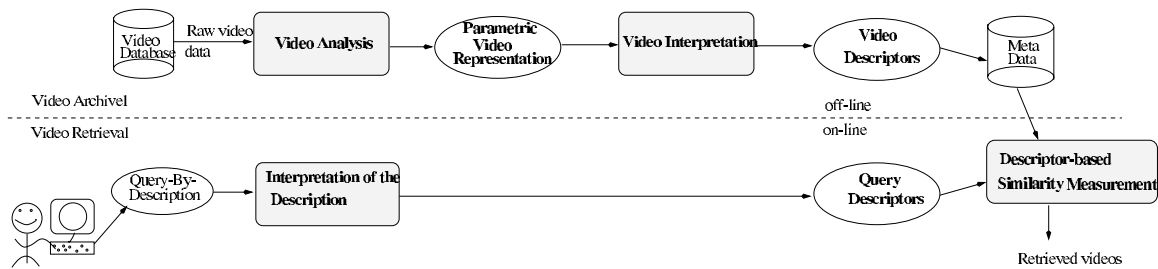
Clearly, the analysis of video sequences plays an important role, in both off-line and on-line processes. **Video analysis** aims to partition a video sequence into its natural hierarchical units (i.e., scenes, shots, and objects) and to assign attributes such as structure (syntax) and context (semantics) that can be later used to formulate specific queries on a collection of these units (Fig. 3). *Temporal segmentation* of a



(a) Query-By-Example with parameter-vector-based similarity test



(b) Query-By-Example with descriptor-vector-based similarity test



(c) Query-By-Description with descriptor-vector-based similarity test

Figure 2: Models for content-based video retrieval

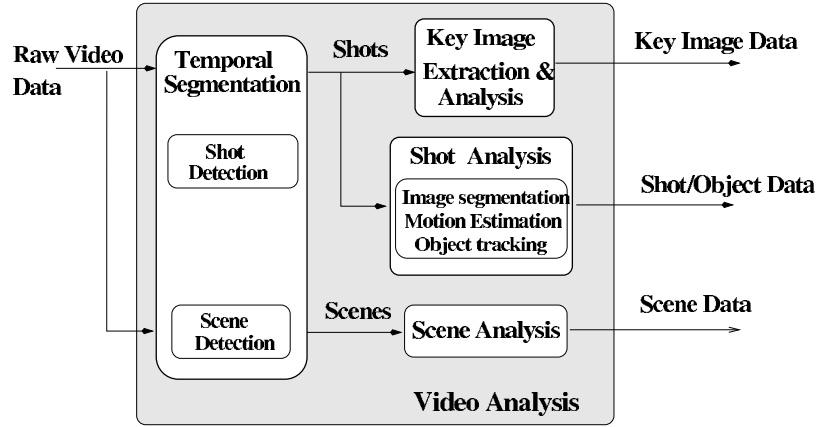


Figure 3: Video analysis

video is the partition of a video sequence into its natural temporal units: scenes and shots. Typically, a video sequence, e.g., a movie, contains thousands of shots. In order to facilitate the analysis of a video sequence, a shot boundary detection is needed. *Shot boundary detection* is a temporal segmentation technique where a video sequence is partitioned into its shots. *Scene analysis* is used to detect meaningful set of shots using automatic recognition methods. However, in movies, each scene is logically connected to previous and following ones. On the other hand, videos in surveillance applications do not have semantic flow. Since a shot database can be very large, instead of analyzing the whole shot, only key images can be used for retrieval to speed up video shot retrieval. *Key images* visually summarize the content of a shot.

**Video representation** provides a formal description (*meta-data*) of the video contents, i.e., a parameter feature vector capturing certain essential properties of the video. This representation is based on models of the content of the video (e.g., an affine motion model can be assumed and estimated parameters represent the feature “motion”). For each shot, a parameter vector is computed and stored in a meta-database to speed up the data retrieval process. Accordingly, the result of the analysis and representation of a video is a multi-dimensional real-valued *parameter vector*  $\mathbf{v}_p \in \mathcal{R}^n$  that quantitatively represents the visual features of a shot or its objects.

However, for fast retrieval, a quantization and an **interpretation** of these parameters to obtain *descriptor vectors* (Fig.4) is needed. Thus, the video interpretation  $\Upsilon$  is a function mapping the real-valued parameter vector  $\mathbf{v}_p$  onto a descriptor vector  $\mathbf{v}_d$ .

$$\Upsilon : \mathcal{R}^n \xrightarrow{Q} \mathcal{I}^n \xrightarrow{D} \mathcal{A}^m \quad (1)$$

where  $m \leq n$ ,  $Q$  denotes the quantization operator,  $D$  the interpretation operator,  $\mathcal{R}^n$  is real space,  $\mathcal{I}^n$  is the discrete (integer) space, and  $\mathcal{A}^m$  is the alpha-numerical description space. For example, if  $\mathbf{v}_p = ((a_1 \cdots a_6), (\text{contour coordinates}), (\mu, \sigma))$  then  $\mathbf{v}_d = (\text{rotation, ellipse, blue})$ . The interpretation is based on underlying models for visual features, e.g., if the parameters describing the translational motion in an affine

motion model are the only non-zero ones, then “translation” is the descriptor of the object motion. If the shape feature can be approximated by an ellipse, then “ellipse” is the descriptor of the object shape.

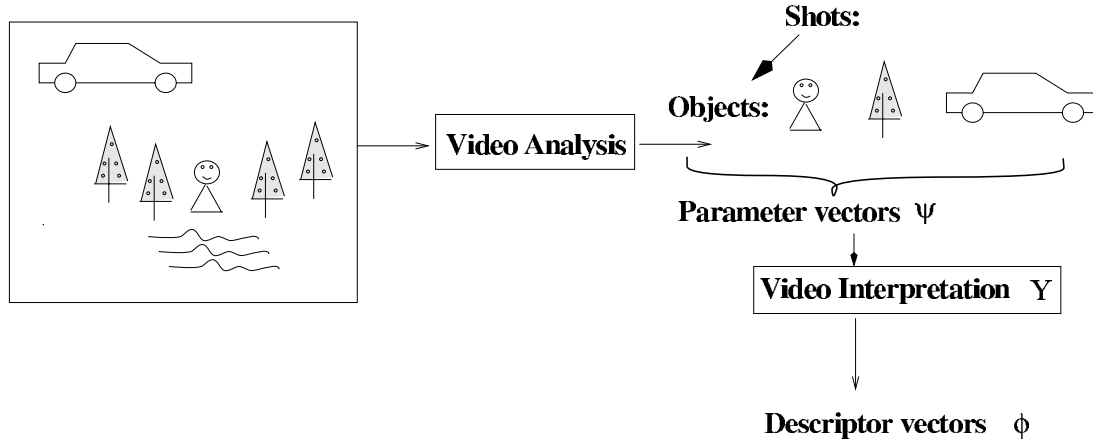


Figure 4: Representation versus interpretation

In a **query interface**, the user selects or describe the shot or the objects he or she is looking for. An alternative to querying is browsing. **Browsing** allows users who cannot specify exactly what they seek to search around in an information space to find it. A browsing space should be efficiently structured so that users can move in a useful and efficient way without getting lost.

The **similarity test** computes the difference between either two parameter vectors or two descriptor vectors, a query vector and a stored vector.

## 1.4 Problem statement

As digital video databases become more and more popular and wide-spread, providing means for efficient (speed) and effective (accuracy) video retrieval through the analysis, representation, and interpretation of visual content are important and practical topics of research.

Recently, video retrieval systems supporting content-based queries have been developed. These systems allow video search through the use of either basic video features such as global motion (e.g., zoom) or basic features of key images of the video sequence (e.g., color histogram). However, when retrieving video, users are generally interested in the objects in an image sequence. Few systems have dealt with retrieval of a video sequence using objects. Since motion is an important feature of objects, these object-based systems mainly use motion as a basic feature for the video representation and retrieval. In general, however, average users are interested in higher-level features of objects such as in the object actions (e.g., an object enters) throughout an image sequence and in the inter-object relationships.

In addition, some systems allow a quantitative specification of shot and object features (e.g., motion 10 pixel/second, distance: 50 pixel) Normally, users do not ex-

actually know what the image sequence they are looking for looks like. They do not have *exact* information (do not memorize) the motion, color, or texture (basic features). Thus, user-oriented query forms are important to allow qualitative formulation of a query. Furthermore, in most existing video retrieval tools, the user either sketches or selects an example of an image sequence, e.g., after browsing the database, he or she is looking for. However, browsing in large databases can be time-consuming and sketching is a difficult task, especially for complex scenes.

## 1.5 Outline of the proposal

The focus of this doctoral research project is the development of effective and efficient methods for video analysis and interpretation. Automatic techniques for *spatial image segmentation*, *object tracking*, *motion estimation*, and for *qualitative description* of object motion are proposed. In order to adapt the video analysis techniques to the amount of noise present, a noise estimation technique will be investigated. Furthermore, techniques for the *interpretation of spatial and temporal relationships* of multiple objects of a video shot are introduced. These techniques aim to link previously extracted basic object features, such as motion, to high-level object features such as actions. This high-level information is then used to allow the user to retrieve a video shot by giving a qualitative description of it. To speed up the retrieval process, *shot pruning* based on a qualitative description of global features, such as global motion, is also proposed.

In Section 2, pertinent literature is reviewed. Then, an overview of the proposed retrieval approach and its main components is given in Section 3. Section 4 gives details of image segmentation and motion estimation developed thus far. It also includes simulation results. The work plan is given in Section 5. Anticipated contributions are listed in Section 6.

## 2 Related Work

### 2.1 Video retrieval approaches

Video retrieval systems can be divided into *key-image-based*, *global-motion-based*, *object-based*, *pixel-based* systems. In a key-image-based system, the basic unit of a video is a shot. A user gives a query shot and the system extracts key images, computes basic features (e.g., color histogram) of these images, compares them to key-image features in meta-database, and finds similar shots. In global-motion-based retrieval, the global motion in the video sequence is measured and shots are represented by their quantitative global motion (e.g., zoom or pan). For a given query video, the parameters of the camera motion are estimated and compared to the stored global-motion parameters using similarity measures. Hybrid techniques combining both strategies to achieve more robust retrieval approaches are also used. Substantial research has been carried out on video representation using shot abstractions using key-images and global motion [26, 24, 2]. Techniques for providing formal description schemes and languages for content-based video retrieval were also introduced [27, 24].

For many real-world video sequences, however, global characteristics such as key-images and global motion alone cannot ensure satisfactory retrieval results. In many domains useful information consists of local features (such as texture or color) of local regions of a video. Recently, pixel-based video retrieval systems have been developed. Here, a sequence is characterized using basic features such as the motion of individual pixels or image texture [2, 44, 28]. A user gives a shot query, the system then finds its basic representation and compares this representation with those stored in the database.

When retrieving video, users, in general, are interested in objects. The focus is first on the the objects and their features depicted in the video and then on the specific basic features. Few systems have dealt with retrieval of a video sequence based on objects. In object-based retrieval systems, the basic unit of a video is the life span of a visual object. A video is abstracted by the basic features (motion, texture, etc.) of its objects. The main problem with object-based methods is their dependence on video analysis performance (i.e., image segmentation and motion analysis) that are usually far from being perfect.

*VideoQ* [11] is the only complete and on-line video retrieval system that relies on object- and motion-based retrieval strategy. It allows a user to sketch a motion trail represented by a sequence of 2D points over time. Here, two similarity modes are used: contour-based similarity test by projection of the sketched motion trails into the image plane so that temporal object contours over the sequence are created. Then, a  $L_2$  distance of contour points is calculated; motion-based similarity test by calculating the similarity between the entire motion trail and the stored motion trails. This *query-by-motion-trail* is especially useful when browsing and searching sports video sequences. However, in a sketch-based query, a user must prepare query sketches which may be quite difficult, especially in complex (motion) cases. According to results shown in [11], it seems that VideoQ works well when searching video shots

with few dominant objects with a simple background. In addition, the user has to select weights to combine basic features of the query object. Although this should not be an obstacle, average users do not normally know how to adjust these weights in order to get the desirable video sequences.

*Netra-V* [13] is a prototype system that uses a representation of a video based on basic object features (color, texture, and motion). Considering the results given in [13], the used tracking method is not robust to object occlusions. In addition, the designers of the system assume that a user has a shot query example. However, users usually do not know what the image sequence, they are looking for, looks like. They do not have *exact* information (do not memorize) the motion, color, or texture (basic features). Users normally memorize fuzzy descriptions of basic features of objects and their actions, e.g., when objects stop or disappear. This means that qualitative formulation of object features and their *action* in a shot query are more suitable for real-world video retrieval applications.

An object-based retrieval system using qualitative action descriptions is the *AVI* system introduced in [14]. It allows video search based on object motion classification. Moving objects are first detected and then tracked throughout the video. Based on the tracked object information, the system classifies the following events of interest: an object moves, rests, appears, and disappears. The system is developed to assist human analysis of video data and is especially useful when searching surveillance video data where few objects and a static camera are typical. However, most non-surveillance image sequences contain multiple moving objects and non-stationary camera motion. The system uses a fast motion detection method but the retrieval quality is poor in presence of camera motion. Furthermore, event analysis provides false alarms in the presence of acceleration and object occlusions. In addition, video noise can also affect the performance of the methods used ([14], p. 622).

## 2.2 Shot boundary detection

Typically, a video sequence, e.g., a movie, contains thousands of shots. In order to facilitate the analysis of a video sequence, shot boundary detection is needed. In movies, shots are normally short, while they can be long in a surveillance application. Two shots are separated by a *cut* or a *boundary* which is a transition at image boundary between two successive shots. A cut can be thought of as an “edge” in time. In a video, two general types of cuts or shot boundaries can be produced: abrupt and gradual. *Abrupt cuts* occur over a single image as a result of joining two dissimilar shots together. Abrupt changes can occur due to background change, largest moving objects, and completely different scene. *Gradual changes* are the results of special editing effects such as fade-ins, fade-outs, wipes, and dissolves. They occur over multiple images. In a fade, the luminance gradually decreases to, or increases from, zero. In a dissolve, two shots, one increasing in intensity, and the other decreasing in intensity, are mixed. Wipes are generated by translating a line across the image in some direction, where the contents on the two sides of the line belong to the two

shots separated by the edit. Many other edits exist that may not be simple linear transformations like the ones described above.

Shot boundary detection was addressed in TV applications in the 1970's [34]. Abrupt camera transitions can be detected quite easily as the difference between two consecutive images is so large that they cannot belong to the same shot. Gradual transitions are more difficult to detect because they can be confused with object motion or special camera effects.

Shot boundary detection methods can be classified into *pixel-based*, *histogram-based*, and *motion-based* techniques. They are performed in the uncompressed domain, mainly using certain global features of the video such as histograms, edges, or a set of certain local features of the image. While pixel-based techniques are extremely sensitive to motion and noise, histogram-based techniques are more robust to motion changes but suffer in the case of noise and illumination changes. Motion-based techniques depend on the quality of the motion-estimation algorithm used. Because of its high computational cost and because histogram-based techniques can achieve high quality results, motion-based techniques are not commonly used [34, 2, 9].

## 2.3 Spatio-temporal object detection

### 2.3.1 Requirements and limitations

For decades, video processing research has focused on pixel-level processing, with the goal to acquire, enhance, restore, or compress video sequences. In recent years efforts have been made towards region- and object-level processing, with the additional goal to allow content-based manipulations of video sequences.

Segmentation has been a topic in computer vision for many years. Recently it has become an active research area in video processing, e.g., for object-based compression. An image segmentation is, usually, defined as a partition of an image into homogeneous regions. Thus, a region depends on a specific definition of homogeneity (e.g., color or motion). A segmented region may correspond to a 2-D object, i.e., may have a semantic meaning. Because of the difficulties in establishing such a meaning, current segmentation techniques introduce regions rather than objects. Note that the term object is often used when region is meant. The problem of automatic (non-supervised) video segmentation is now receiving widespread interest due to the development of the content-based audiovisual coding standard MPEG-4, the Multimedia Content Description Interface MPEG-7, and content-based video retrieval.

In general, image segmentation is an ill-posed problem, e.g., no unique segmentation solution may exist and the a solution does not depend continuously on the data (e.g., a change of a few pixels may dramatically change the resulting segmentation). Furthermore, there can be no fixed algorithm which will always satisfy needs of different applications and domains. Even designers of segmentation algorithms would have difficulties adapting the segmentation parameters if the application goals change. Another problem with traditional segmentation is that it assumes one optimal partition exists. This is rarely the case. In addition, for robust object-based techniques robust



image segmentation methods are needed. In a video scene various types of image changes (e.g., noise, artifacts, illumination changes and object overlaps) can appear. In this case the reconstruction of objects from video signals should stay robust in order to avoid failures in object-based techniques. In order to adapt the video analysis techniques to the amount of noise, noise estimation techniques are needed. Due to the psychophysical nature of video and image segmentation and due to the various nature of its applications (e.g., entertainment, military, medical images) and requirements (e.g., fast responses), the use of heuristics is an unavoidable part of solution approaches [35, 10].

Because of these difficulties, semi-automatic segmentation methods and domain knowledge are integrated in most retrieval systems. Such methods perform better than automatic methods due to user interaction; users can give examples of regions of interest or examples of irrelevant regions. However, since visual data are perceived differently by different people and since the visual nature of video data is complex, manual segmentation could end up with numerous interpretations of the same data. Thus, manual segmentation, clearly, suffers from subjectivity of the human operator. In addition, manual segmentation is time-consuming.

In automatic segmentation, several homogeneity criteria have to be combined to achieve useful segmentation resulting in regions being homogeneous, e.g., in color, texture, motion and/or some semantic meaning. The different types of segmentation methods differ in how these types of information are combined and estimated. For object-based video retrieval, spatial (*intra-image*) and temporal (*inter-image*, i.e., *object tracking*) segmentation methods are used to detect spatio-temporal objects and their features. Other than in video filtering and compression, in video database applications, perfect segmentation of objects is no longer the goal. Obtaining pixel-precise edges and regions is not necessary for most content-based retrieval. Instead, the representation of objects or regions has to have some semantic meaning, e.g., recognizing the events perfumed by the objects such as entering or changing direction.

### 2.3.2 Intra-image segmentation

Intra-image segmentation methods can be classified as *local-feature-oriented*, e.g., based on edge detection and contour extraction; *region-growing-oriented*, e.g., based on texture or motion, and *global-cost-function-oriented* based on optimization of image global criteria, e.g., an energy function [45]. Local-feature-oriented methods show low computational cost but make use only of local information and thus are sensitive to degradation of image quality. Region-growing-oriented methods are more robust in complex video scenes but they generate regions with small holes and require high implementation costs. Global-feature-oriented methods use a global criteria but it is often very difficult to find their minima [45]. Further, these methods, in available implementation, demand high computational costs.

In video retrieval, intra-image segmentation is an important step where it is used as an effective means of investigating the spatial relationships and properties of the image data with respect to spatial basic features and syntactical object relationships.

It is also needed in order to obtain regions for local-feature extraction [11, 13].

### 2.3.3 Object tracking

Motion has been recognized as a viable technique for video retrieval and representation. This is mainly because motion gives information about spatio-temporal object relationships. The motion of objects in the real world is in general smooth. When an object moves, its spatial features such as color and texture do not vary significantly. Most motion estimation techniques used in video processing compute displacements between two consecutive images. However, this is not sufficient to analyze and describe spatio-temporal object behavior or actions. Because of noise and to assure coherent motion trajectories throughout time, object tracking techniques need to be introduced to obtain spatio-temporal objects for retrieval purposes. In the context of video retrieval, limited work has been done to temporally integrate or track objects or regions throughout an image sequence.

In an object tracking system, objects have to be first detected. The system then follows them as they move. When an object is represented by its visual features, tracking based on one or a combination of these features can be applied. For an effective tracking, it is important to use feature representations which are robust to changes such as noise, rotation, or scale [17].

- Motion-based methods: allow the tracking of objects based on the motion. Here, Kalman-filtering methods are used to estimate the evolution of object's dynamic parameters. These parameters support any parametric motion model such as the affine model.
- contour- or shape-based methods: based on spatio-temporal edges and shape computed from two successive images for each moving object.
- region-based methods: aim at representing the object through its gray-level pattern. This feature is robust to noise and to different conditions.
- Color-based methods: allow the tracking of objects based on its color.

While the use of a Kalman filter relies on an explicit trajectory model, the three other models do not require any trajectory model. However, the definition of an explicit trajectory model may prove to be complex.

### 2.3.4 Object detection schemes currently used

In the VideoQ [11] system, an image segmentation based on the combination of color and edges is used. Region merging is performed using optical flow motion. In Netra-V [13], a semi-automatic (the user has to specify a scale factor in order to connect region boundaries) image segmentation is applied based on color, texture and motion. This is, however, not suitable for large collections of video data. Here, the shot is first divided into image groups, then the first image of a group of images is spatially segmented into regions. Then, a 2D affine model is used to track these regions throughout the image group. Motion estimation is done independently in each group

of images. The AVI retrieval tool [14] uses only motion detection information for retrieval purposes. Major drawbacks of these segmentation approaches are:

- Optical flow methods (VideoQ) do not deliver robust motion information especially in the presence of large motion and object occlusion,
- In the motion-based region merging (VideoQ) and tracking (Netra-V), regions segmented based on coherent-motion criteria may contain multiple objects and need further segmentation for object extraction,
- The main difficulties of the image-group-based segmentation (Netra-V) are the determination of the number of images in each group (here it is fixed). This introduces many artifacts, when the cut is done at images where important object activities are present. Further, objects disappearing (respectively appearing) just before (respectively after) the first image of the group are not processed in that group.
- Motion detection methods (AVI) are very sensitive to noise and artifact, and
- To assure coherent motion trajectories through time, tracking techniques have to be introduced for retrieval purposes.

## 2.4 Motion estimation

In video processing applications, motion estimation is a key technique that determines the quality of processed image sequences [16, 19]. Motion occurring in a video can be divided into camera motion and object motion. A rigid object can basically perform a translation or rotation. Camera motions are: *pan* resulting from turning the camera from right to left (and vice versa), *zoom* resulting from changing the focal length of the camera, and *tilt* resulting from moving the camera up or down. Whenever there is motion, there are regions which become visible and regions which are occluded. Object occlusions and exposures make the task of motion estimation difficult.

### 2.4.1 Object motion estimation

In video processing, the estimation of 2-D apparent motion of rigid or almost rigid objects and of pixel-based or dense motion (optical flow) are usually considered. Optical flow gives the distribution of velocity with respect to an observer over the points in an image. For example, the Horn and Schunk gradient equation  $f_x u + f_y v + f_t = 0$  relates the spatial ( $f_x, f_y$ ) and the temporal ( $f_t$ ) derivatives of the image intensity function to optical velocity,  $(u, v)$  at each point [32]. In the presence of large motion and object occlusion, such pixel-based approaches are unreliable [31]. A more reliable approach is the estimation of motion of a larger region of support such as blocks or arbitrary-shaped regions based on parametric motion models. Here, multiple motions (e.g., translation, rotation, zoom) of this region of support are detected. In practice, as a compromise between complexity and flexibility of different motions, 2-D affine motion models are used.

Motion estimation that is most frequently used and implemented in hardware is block matching [19, 6]. Three advantages of block-matching algorithms are: 1) easy implementation, 2) better quality of the resulting motion vector fields compared to previously used methods such as gradient methods or phase correlation, in particular in the presence of large motion, and 3) regular architectures. Furthermore, block-based motion estimation is stable, i.e., it never breaks down totally. This cannot be said of object- and model-based methods, that generally give better motion vectors but occasionally fail to interpret object motion correctly. This is mainly because of their dependence on object detection performance. Thus, it is likely that in many applications block-based motion estimation will be further used.

Since real objects in real scenes do not coincide with the block boundaries, block matching algorithms suffer from certain drawbacks. This is particularly true surrounding the object boundaries where these methods assume an incorrect model and result in erroneous motion vectors leading to discontinuity in the motion vector fields (causing *ripped boundaries* artifacts). Another drawback is that the resulting motion vectors inside objects or object regions with a single motion are not homogeneous (producing *ripped region* artifacts). Additionally, using block-based algorithms results in block patterns in the motion vector field (causing *block patterns* or *blocking* artifacts). These patterns often result in block artifacts in coded or interpolated images. The human visual system is very sensitive to such artifacts (especially abrupt changes). Depending on the performance of object detection methods, the integration of object information in the process of estimation object motion contributes to reduce such artifacts and enhance the motion vector fields [4, 7, 25].

#### 2.4.2 Global motion estimation

Global motion can be a result of camera motion or global illumination changes which cause a change in the overall intensity. The latter is not a true motion but rather an apparent motion. The estimation of the motion of the whole image results in more compact representation of the motion than when estimating the motion of each block or object in the image. Therefore, such estimation is suitable in global-motion-based retrieval, e.g., for the purpose of pruning. Different parametric motion models can be used to estimate global motion [18, 9] In practice, as a compromise between complexity and flexibility of different motions, 2-D affine motion models are used.

For orthographic projection combined with a 3-D affine motion model of a planar surface, the instantaneous velocity of a pixel at position  $\mathbf{x}$  in the image plane is given by a 6-parameter  $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6)$  motion model [18]:

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} a_3 & a_4 \\ a_5 & a_6 \end{pmatrix} \mathbf{x} \quad (2)$$

### 2.5 Video interpretation

The interpretation of extracted (quantitative) object or camera motion parameters is important to allow a user to search for objects or shots based on the perceived object

motion (e.g., rotation) or camera motion (e.g., zoom). Little research work in that direction has been done thus far.

### 2.5.1 Interpretation of global motion

Techniques for the estimation of camera motion either analyze a dense motion field (e.g., optical flow) or estimate a set of parameters (e.g., parametric motion model). In order to search for a shot based on the perceived motion, an interpretation (i.e., to find a qualitative description) of the camera motion is needed (e.g., for shot pruning based on camera motion, a bi-level decision (e.g., zoom or no zoom) is needed).

Little work has been done to qualitatively describe global motion. In [43] optical flow is first estimated and then motion vectors are analyzed to detect the type of camera motion. To detect zoom, the following strategy is applied: motion vectors describing zoom have a focus of expansion. The assumption is that the sum of the motion vectors around this center is zero or very small. However, due to noise and estimation errors, it is difficult to find the focus of expansion and to be able to decide whether the sum is significant or not.

In [9], a 6-parameter affine motion model is used to estimate the dominant camera motion. Then, each parameter or a linear combination of these parameters is analyzed to qualitatively characterize the type of camera motion at each time instant. For example, while  $a_1$  and  $a_2$  describe the translational motion, the linear combination  $\frac{1}{2}(a_2 + a_6)$  determines zoom [9]. Rotation is expressed as the combination  $\frac{1}{2}(a_5 - a_3)$ . For instance, if the dominant camera motion is a pure pan, the only non-zero parameter is supposed to be  $a_1$ . In the case of zoom, the linear combination  $\frac{1}{2}(a_2 + a_6)$  is assumed to be the only non-zero parameter. However, due to estimation errors, these quantities can differ from zero. Therefore, methods to determine whether a value of a parameter is significant or not have to be introduced. In [9], a statistical approach based on likelihood ratio test is used. This test uses a threshold for the final decision.

### 2.5.2 Interpretation of object motion and action

Verbal description of object motion such as translation and rotation can be achieved when analyzing estimated motion parameters. It can be shown that basic motion features can be useful for the recognition of motion and even of object activities. In [14] moving objects are first detected and then tracked throughout the video. Based on the tracked object information, the system classifies the following events of interest: an object moves, rests, appears, and disappears.

## 2.6 Similarity measures

In a content-based retrieval strategy, a video is usually described using a set of features. When retrieving, the system uses a *distance* or *similarity measure* which can be defined as a score function that rates the similarity between two video sequences.

Given two features, where a feature could be treated as a vector or a distribution of a random variable, a similarity function computes the difference of the two vectors or distributions. Thus, a similarity computation replaces matching as the core operation in video retrieval systems. Similarity search in the  $n$ -dimensional feature space thus consists of comparing the target feature vector with feature vectors stored in the database. It is, in general, desirable to reduce the dimensionality of the feature vector to allow faster similarity tests.

The most basic and simplistic search mechanism is to use a Euclidean measure. In many retrieval systems, a linear combination of the features with nonuniform weights is used. However, as the dimensionality (i.e., the total number of features) increases, the selection of relative weights among features becomes difficult. Similarity measurement is complicated by many factors, and no unique criterion of similarity, e.g., perceptual or semantic, can be appropriate for all applications. The selection of a similarity model or a combination of models is generally non-intuitive. There may not be one winning model or fixed combination of models, but these may need to vary within the database.

Furthermore, Euclidean similarity measures and linear-combination-based measures can be inadequate as they do not correspond to perceived similarity nor adapt to different applications. Therefore, relevance feedback and/or iterative refinement techniques based on the user feedback have been proposed to adjust the similarity metric [40]. A robust and selective technique that assigns weights to significant features but can ignore weak features is an open issue in video retrieval.

In video coding and processing, simple measures, such as the mean-square error, are widely used to measure the performance. This measure has been shown to poorly correlate with human perception. However, in image retrieval, simple similarity measures applied to *suitable* color and texture features can perform surprisingly well [36]. Furthermore, researchers use the simple Euclidean measure in order to illustrate the robustness of a representative feature vector of a video sequence [28]. In addition, this measure is easy to compute and thus enables real-time performance.

## 3 Proposed Object-based Retrieval

### 3.1 Motivation

When retrieving video, users, in general, are interested in objects. The focus is first on the object semantics (e.g., activities or actions) depicted in the video and then on the specific basic features (e.g., color). Therefore, object-based video retrieval is needed. An important issue here is what level of object semantics and features is most important to retrieval? For example, are high-level intentional descriptions (e.g., what a person is doing or thinking) needed? One important observation is that the subject of the majority of video sequences is an object or group of objects and basic actions [29].

In this work, an object- and action-based video retrieval approach based on qualitative description of video content is proposed. Here, the user can give a qualitative description of a query video by specifying video-global features, such as global motion, and object features, such as basic features (e.g., color), spatial relationship features (e.g., object  $i$  is closed to object  $j$ ), location features (e.g., object  $i$  is in the bottom of the image), and semantic features (e.g., object action: object  $i$  moves left and then is occluded).

An advantage of such a retrieval strategy is that it allows the construction of intuitive queries based on the observation that most people’s interpretation of real world domains is imprecise and that users, while viewing a video, usually memorize objects (“who” is in the scene), their action (“what” he/she is doing), and their location (“where” the action takes place) and not the exact (quantitative) object features [29, 24]. In the absence of a specific application, such a generic model allows extensibility (e.g., by introducing new definitions of object actions).

Another motivation for choosing this retrieval strategy is the fact that the human visual system adapts quickly when viewing a video, or a page of text, and finds a desired piece of information while ignoring unwanted information. A user is able to search for a page or an image by flipping through a book or a video sequence and to select and combine desired information. Different flipping strategies can be applied: flipping by jumping a fixed number of images (e.g., using key images) may omit the target information completely. Therefore, information of video object actions, such as exposure, with the knowledge of object features, such as color, will allow a more focused retrieval.

### 3.2 Retrieval strategy

In most existing object-based video retrieval tools, the user either sketches or selects a query example, e.g., by browsing. However, browsing a large database can be time consuming and sketching is a difficult task, especially for complex scenes. In this work, the user either selects an image sequence or gives a qualitative description of a query video shot by specifying

- shot's global feature, e.g., global motion such as zoom,
- objects' basic features (i.e., motion, size, texture, position, and color),
- object's actions (i.e., moving, entering, exposure/occlusion, changing direction), and
- inter-object relationships.

These inputs are then interpreted to form a descriptor vector. If the user selects a shot query, the system analyses the content of this query shot and provides a descriptor vector. To speed up the retrieval, the selected global feature of the query shot is used to prune (on-line) the shot database. Thus, the pruning results in candidate shots. Then, an object-based similarity test is performed on these candidates based on extracted object descriptors. The system retrieves finally a ranked list of similar video shots. An advantage of such a retrieval strategy is that it allows the construction of an intuitive query interface. In the absence of a specific application, such a framework is extensible since it allows a generic description of a video with regard to the objects (e.g., by introducing new definitions of object actions).

It is usually very difficult to anticipate all possible types of objects in which a user might be interested. On the other hand, recognizing objects entirely at query will limit the scalability of a system, due to the high expense of such computing. The proposed object-based framework allows flexible composition of queries relying on spatial object properties, their actions, and their interrelationships, aiming to alleviate both problems.

To identify object actions, a qualitative description of the object motion is an important step towards linking basic features to high-level feature retrieval. For this purpose, object detection using intra-image segmentation, motion estimation and object tracking is proposed. In this work, significant objects (e.g., large moving objects) are detected and tracked. Then, the motion behavior of these objects is analyzed in order to represent important events included in the image sequence. This means basic features can be combined in ways to produce an illusion of high-level features. Since the vision community seems to have achieved far more success in modeling basic than high-level phenomena, this appears to be the most feasible approach at this time.

Since video shot databases can be very large, pruning techniques are essential for efficient queries. Therefore, two methods for pruning are proposed: the first is based on global motion detected in the scene, and the second is based on *dominant objects*, i.e., objects that are (continuously) visible throughout the whole sequence, e.g., background or a large moving object. To do this, global motion and/or dominant objects have to be estimated and qualitatively described.

Fig. 5 shows a block diagram of the proposed video retrieval. The components under investigation in this proposal are marked with white boxes delimited by a dashed line. Fig. 6 shows a prototype of a query form. Main tasks of the proposed system can be summarized as follows:

- video analysis:



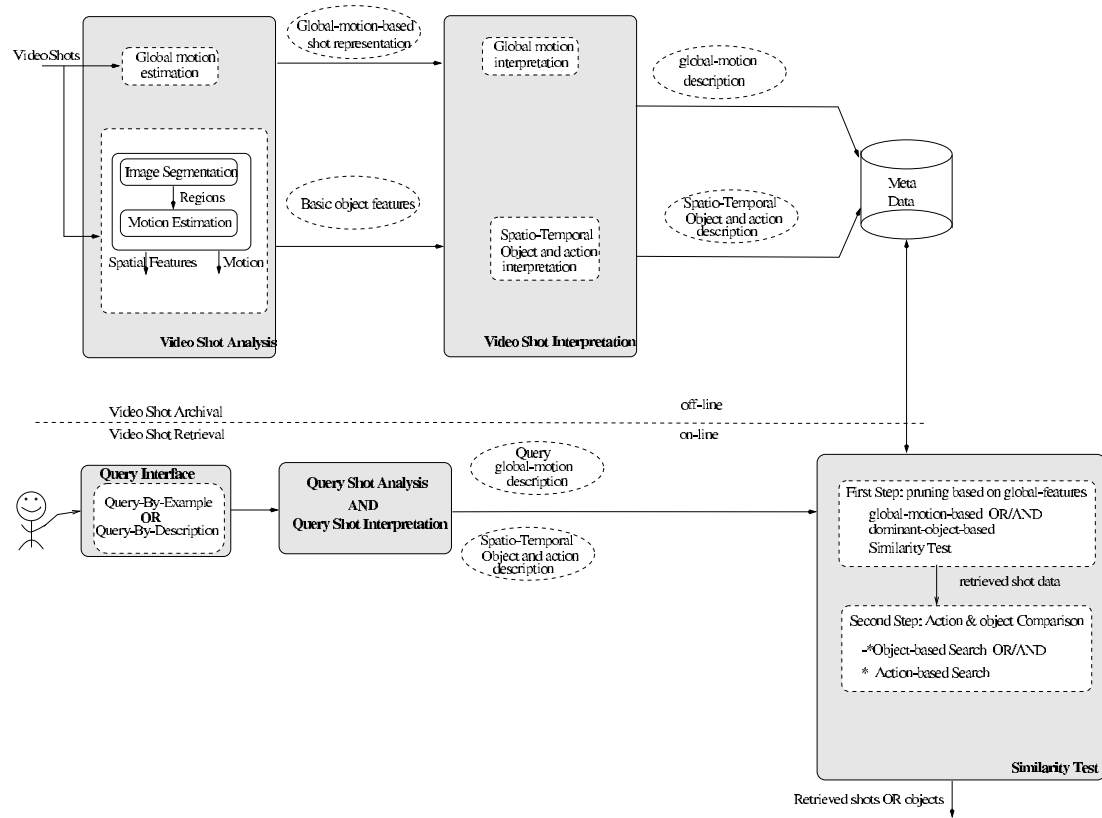


Figure 5: Architecture of the proposed video retrieval

- spatio-temporal object detection using intra-image segmentation, object motion estimation, and object tracking and
- global motion estimation.

The result of this step is a parameter vector that gives a real-valued quantitative representation of the shot.

- video interpretation: Parameter quantization and interpretation that provides a descriptor vector that describes qualitatively the content of the shot.
- similarity test: comparison of two descriptor vectors. The result is a ranked list of similar shots.

Accordingly, the new retrieval approach aims to introduce new functionalities that are oriented to two main requirements when developing a content-based retrieval system [29, 38]: 1) orientation to the way users, usually, describe and judge video similarity and 2) efficient analysis and interpretation methods.

### 3.3 Spatio-temporal object detection

Although today's computer vision systems cannot recognize high-level objects in unconstrained images, experiments show that basic visual features can be used to characterize video content. Furthermore, the extraction of basic object features (such

**Find video shots in which**

**Action specification**

object i the scene

appears in  
disappears from  
moves  
left in  
right in  
up in  
down in  
rests within

**Spatial object locations**

lays in the the scene

left  
right  
top  
bottom  
center

---

**Inter-object relations**

object i object j

left  
right  
above  
below  
inside  
near  
within a circle of 50 pixels  
50 pixels left/right/ ... to

**Spatial object features**

shows

texture (example)

Color

size  
(small, medium, large)

---

**global shot specifications**

global motion: zoom  
pan  
rotation  
stationary

**global objects:**

texture (example)

Color

size  
(small, medium, large)

Motion  
(slow, medium, fast)

Figure 6: Proposed query form

as color, texture, shape, motion) and their spatial/temporal relationships can be achieved. An important issue in object-based retrieval is that automated analysis of objects does not need to accurately identify real-world objects contained in the imagery. The goal is to extract large objects that have some semantic meaning and to represent video objects accurately enough without significantly impacting the user's ability to extract and understand retrieved information. In a Query-By-Example retrieval model (Fig. 2), the video analysis and interpretation has to be done on-line, therefore a fast response is important. Therefore, main components of the spatio-temporal object detection will be devised for a fast retrieval. Such a goal induces severe time and hardware constraints.

Objective of the proposed detection of spatio-temporal objects are:

- robustness to image changes, such as illumination and noise
- detection and separation of objects,

- detection of large objects which are necessary for further video processing steps. This choice is oriented towards the human visual system that tracks primarily large moving objects.
- low computational cost and regularity of main components.

### 3.3.1 Intra-image segmentation

In this proposal, the potential for an automatic analysis in the visual domain is studied. Fully automatic real-time image segmentation and feature extraction methods for flexible multi-level object descriptions are proposed. The design of the proposed segmentation is oriented to that object-based video retrieval requires fast responses and not exact region detection.

The non-supervised image segmentation method is carried out by texture-based object isolation (binarization), morphological edge detection, contour analysis, and object reconstruction. Parameters used in the proposed image segmentation are adjusted to account for the amount of noise present in the image. A noise estimation approach will be developed, therefore.

### 3.3.2 Object motion estimation

A new approach to object-based motion estimation in video sequences between two successive images is proposed. This approach consists of an explicit matching of pre-extracted arbitrarily-shaped objects in order to estimate their motion. In the current implementation, extracted object information (e.g., area (size in pixels), minimum bounding box (MBB), positions, motion direction) is used in a rule-based process with four steps: finding of object correspondence, estimation of the MBB motion based on the displacement of the sides of the MBB, i.e., the estimation process is independent of the intensity signal, detecting object motion types (zoom, translation, rotation) by analyzing these displacements of the MBB sides, and updating the object motion.

### 3.3.3 Region merging

Ideally, regions in the image showing specific homogeneity form one region. However, segmentation methods tend to divide one homogeneous region into several regions. The reasons for this behavior are twofold: 1) optical, i.e., noise, shading, illumination change, and reflection and 2) physical: a homogeneous region could consist regions with different features. When a segmentation method assume one-feature-based homogeneity, it will fail to extract such regions. Thus, region merging is an unavoidable step in segmentation methods. Region merging is based on similar characteristics.

In a static segmentation, like the one introduced here, geometrical relationships between segmented regions are used to merge regions together in order to form one region. Examples of such geometrical relationships are: i) inclusion: one region is included by another region and ii) size ratio: the size of a region is much larger

than the other. If, e.g., a region is inside another region and its size much smaller, this region could be merged in that region if they show similar characteristics such as texture or color. Furthermore, when motion information is available, it is used to merge regions together. The use of various basic features for the region merging process will be studied further.

### 3.3.4 Object tracking

In order to describe the temporal behavior of objects (e.g., when an object enters the scene), the tracking of objects throughout the image sequence is needed. The first step in the proposed object tracking is to spatially segment the images into objects and to determine object's basic features such as color and size. The second step is to track each object throughout the sequence using object lists associated with each image in the sequence. The object matching between two images is based on several pre-extracted basic object features.

The object corresponding establishing (Section 4.2) approach used in the proposed motion estimation tracks objects between two images. This approach will be expanded to track object throughout an image sequence.

## 3.4 Global motion estimation

In this proposal, affine global motion parameters (equation 2) are estimated using a fast and robust method described in [18].

## 3.5 Interpretation of global motion

For the purpose of pruning, the user is asked to specify the global motion of the shot. This is useful, since a user can better describe a shot based on its qualitative features such as global motion rather than giving a parametric description. Therefore, estimated motion parameters need to be classified. At this time, no contributions can be specified to classify the global motion. Further reading and investigation is needed.

## 3.6 Interpretation of spatio-temporal objects

In the following,  $I_t$  denotes an image at time instant  $t$ ,  $O_i$  represents the  $i^{th}$  object,  $\mathcal{M}_i$  is the object Minimum bounding box (MBB),  $C_i$  represents the centroid of  $O_i$ ,  $k_i$  denotes the column of  $C_i$ , and  $r_i$  is the row of  $C_i$ .

### 3.6.1 Object location

To permit users to specify object locations, the image is divided into 9 sectors as shown in Fig. 7.  $O_i$  is declared to be in the center (right, left, top, bottom respectively) of the image if its centroid is located in the center (right, left, top, bottom respectively) sector of the image.

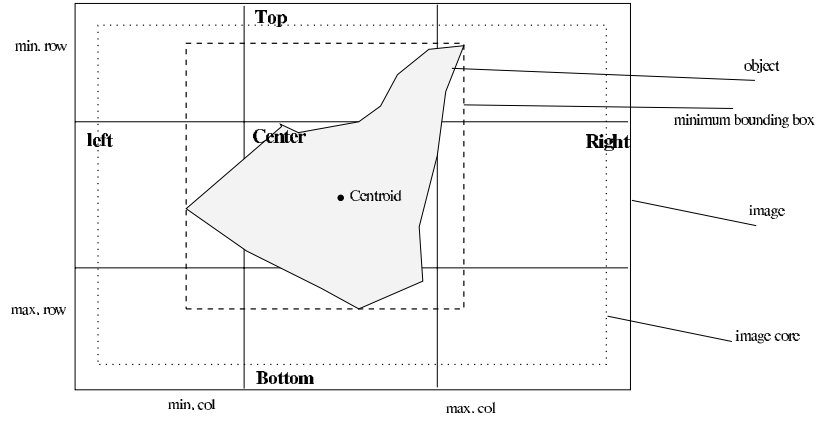


Figure 7: Specification of object locations

### 3.6.2 Basic object features

Since in video retrieval, a fast response is expected, simple measures of object features are proposed that may compromise the effectiveness of retrieval. The full potential of the proposed simple feature representations and their combination will be, therefore, further studied. For each detected object, the following basic features are computed:

- *Minimum bounding box (MBB)*: the MBB of an object is the smallest rectangle that induces all pixels of  $O_i$ , i.e., such that  $O_i \subset \mathcal{M}_i$ . It is parameterized by its top-left and bottom-right corners. A MBB is also specified by its sides, e.g., its length  $L$  and width  $W$ ,
- *Size*: it is defined as the ratio  $S_O$  of the object area in pixels and the image area  $S_I$  in pixels,
- *Color*: The mean and the variance of the chrominance (object) signal provide a simple and compact representation,
- *Shape*: Shape is described by two simple measures:  $\frac{L}{W}$ , the ratio of the sides of the MBB rectangle, and compactness, defined as the ratio of the object size to the size of the minimum bounding box,
- *Spatial homogeneity of a region*: A spatial homogeneity indicator of a region  $R$  measures the connectivity of a spatial region. A practical definition of region homogeneity can be as follows [23]:

$$\text{Homog}(R) = \frac{S_R - S_H}{S_R} \quad (3)$$

where  $S_R$  is the size of the region  $R$ , and  $S_H$  is the total size of the regions inside  $R$ . A region  $R_i$  is inside region  $R_j$  if  $R_i$  is completely surrounded by  $R_j$ . In the ideal case,  $\text{Homog}(R) = 1$ , i.e., no small regions inside a homogeneous region,

- *Spatial position*: The position of an object is quantitatively represented by the coordinates of its centroid. Qualitative position description, such as left,

right, top, bottom, top left, top right, bottom left, and bottom right, are used (Section 3.6.1),

- *Motion*: average motion vector throughout the shot, and
- *Texture*: The following texture measure, shown to be as effective as the more computationally demanding co-occurrence matrices [15], is adopted. The average grey value difference (AGVD) for each pixel  $\mathbf{p} \in R_i$  is defined as

$$AGVD(\mathbf{p}) = \frac{1}{L} \sum_{l=1}^L |I(\mathbf{p}) - I(\mathbf{q}_{\delta l})| \quad (4)$$

$$AGVD(R_i) = \frac{1}{S_i} \sum_{l=1}^{S_i} (AGVD(\mathbf{p}_l)) \quad (5)$$

where  $R_i$  is the  $i^{th}$  region,  $N_\delta = \{\mathbf{q}_{\delta 1} \cdots \mathbf{q}_{\delta L}\}$  is a set of points neighboring  $\mathbf{p}$  at a distance of  $\delta$  pixels,  $|x|$  represents the absolute value of  $x$ , and  $S_i$  is the size of  $R_i$  in pixels. Optimally, the size of  $\delta$  should depend on the coarseness of texture within the image. This quantity varies over the image;  $\delta$  was set equal to one pixel as an estimate, and the neighborhood size  $L$  was set to four, including the pixels adjoining to the north, south, east and west of  $\mathbf{p}$ .

### 3.6.3 Object action

The goal is to define a set of dynamic primitives to describe basic object actions. Other composite actions can be then compiled from basic actions to allow more flexible action-based search.

**Basic object actions** Assume a video shot contains  $P$  images  $I_1, \cdots, I_t, \cdots, I_P$ . The following basic object actions are defined:

- *moving* (towards right, left, top, bottom): the object is visible and its estimated velocity in a given direction is significant (i.e., above a certain threshold),
- *stopping*: the object is visible, was moving and its velocity is not significant (below a threshold),
- *exposed*: the object is not visible in image  $I_{t-1}$ , but becomes visible (its centroid) in the subsequent images  $I_t \rightarrow I_P$ ,
- *occluded*: the object is visible in image  $I_{t-1}$ , but is not visible (its centroid) in subsequent images  $I_t \rightarrow I_P$ , and
- *reversing direction*: the object was moving in one direction (right, left, top, or bottom) and then starts to move in the opposite direction.

**Composite object actions** Based on the basic action, the following composite object actions can be compiled:

- $O_i$  is visible in the image and moves (towards right, left, top, bottom), then it stops, is occluded, or reverses directions,

- $O_i$  is exposed (from right, left, top, bottom) the image and moves (towards right, left, top, bottom), then it stops, is occluded, **or** reverses directions, and
- $O_i$  is not visible in the image and is exposed (from right, left, top, bottom), and moves (right, left, top, bottom), then it is occluded, **or** reverses directions.

### 3.6.4 Inter-object relationships

The following inter-object relationships are defined:

- $O_i$  is to the left of  $O_j$  if  $k_i < k_j$ ,
- $O_i$  is to the right of  $O_j$  if  $k_i > k_j$ ,
- $O_i$  is below  $O_j$   $r_i < r_j$ ,
- $O_i$  is above  $O_j$   $r_i > r_j$ ,
- $O_i$  is inside  $O_j$  if  $\mathcal{M}_i \subset \mathcal{M}_j$ , and
- $O_i$  is near  $O_j$  if  $C_i$  is within a circle of radius  $r$  around  $C_j$ .

## 3.7 Object feature vector

As a result of the video analysis and interpretation, a list for each detected object containing the following items is provided:

- *basic feature vector*: MBB, size, shape, texture, location, and motion (Section 3.6.2),
- *life span* of an object represents the time interval when the object is (continuously) visible,
- *event* descriptions: exposure, occlusion, moving, stopping, changing direction (Section 3.6.3),
- *relationship* to other objects: left, right, above, below, inside, near (Section 3.6.4), and
- information whether this object is a dominant object.

## 3.8 Shot feature vector

Studies [29, 24] show that while viewing and searching a video users usually focus and memorize: 1) events, i.e., “what happened”, 2) objects, i.e., “who is in the scene”, 3) location, i.e., “where did it happen”, and 4) time, i.e., “when did it happen”. Therefore, three shot representations are proposed (Fig. 8):

- *event-based* representation aims to abstract important events occurring within a shot. This can be used to retrieve shot events that include objects and their activities,
- *object-based* representation aims to specify significant objects in the shot, and
- *global-motion-based* representation aims to represent each shot by a qualitative description of its global motion, e.g., for pruning.

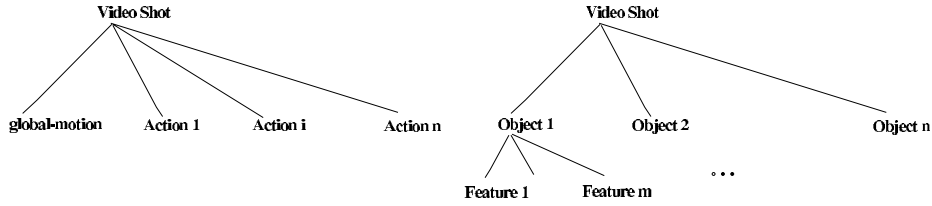


Figure 8: Video shot representations

**An example** Assume the analysis of a shot results in the following parameter vector  $\psi \in \mathcal{R}^7$ :

$$\psi = [\mathbf{a}, \mu_{\mathbf{v}_{O_d}}, t_s, t_e, S_{O_s}, Z_{O_s}, \mu_{\mathbf{v}_{O_s}}]^T \quad (6)$$

where

- $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6)$  represents the global affine motion,
- $O_d$  denotes the dominant object,  $O_s$  denotes the secondary object,
- $\mu_{\mathbf{v}_{O_d}}$  denotes the dominant object's average motion vector,
- $t_s$  ( $t_e$ ) is the dominant object's exposure (occlusion) time,
- $S_{O_s}$  is the secondary object's shape (in pixels),
- $Z_{O_s}$  is the secondary object's size (in pixels), and
- $\mu_{\mathbf{v}_{O_s}}$  is the secondary object's average motion vector.

Then, an interpretation of such a parameter vector  $\Upsilon(\psi)$  (equation 1) is a descriptor vector  $\phi \in \mathcal{R}^6$ :

$$\phi = [m_g, m_{O_d}, a_{O_d}, s_{O_s}, z_{O_s}, m_{O_s}]^T \quad (7)$$

where

- $m_g$  the global motion (zoom-in, zoom-out, pan-left, pan-right, slant-up, slant-down, stationary),
- $m_{O_d}$  dominant object's motion (translation, rotation, stationary),
- $a_{O_d}$  dominant object's action (moving-left, moving-right, moving-up, moving-down, stationary),
- $s_{O_s}$  secondary object's shape (round, elliptic-hor., elliptic-ver., rectangular-hor., rectangular-ver.),
- $z_{O_s}$  secondary object's size (large, medium, small),
- $m_{O_s}$  secondary object's motion (translation, rotation, stationary).

### 3.9 Video shot pruning

A typical video sequence of a movie contains from 500 to 1000 shots per hour [2]. In a video shot retrieval system, feature vectors of query objects or query images are compared to all those stored in a database. The search time, therefore, increases linearly with the size of the database. Efficient feature representations have been



used to speed up the search process. Even if the time required to compare two shots is short, the cumulative time in an exhaustive shot comparison is long.

In image retrieval systems, image pruning based on a global feature such as color to avoid an exhaustive comparison [30] is used. Surprisingly, this problem has received little attention in available object-based video retrieval systems. One approach to avoid comparison of a large number of shots is to extract a shot-global-feature vector (preferably short). When searching for a video shot this global feature can be used to filter those shots most similar to the query shot by comparing their global feature. This is a filtering process called *pruning*. The pruning must be performed in such a way that the retrieval accuracy is not sacrificed in this process. Retrieval accuracy here can be measured in terms of the retrieval obtained with exhaustive search.

In pruning, a subset of shots is selected from a large set of shots using shot-specific global features such as global motion. Therefore, one or more global features of each shot are first extracted and then shots showing similar feature vectors are provided for further shot retrieval steps.

Humans have an excellent ability to describe the camera motion. Although these descriptions are seldom quantitative, there is a lot of information available. An example of such descriptions is “The scene is a long shot with a pan to the right followed by a zoom”. For shot pruning, a bi-level decision about global features (e.g., zoom or no zoom) are needed rather than an exact measure. Two pruning methods are proposed:

- a global-motion-based method
  - 1 selects the global motion descriptor from the shot descriptor vector
  - 2 selects candidate shots based on the qualitative global motion description, e.g., pan or no-pan
- a dominant-object-based method
  - 1 selects the first and the last images of each shot,
  - 2 extracts the objects that exist in the first and last image and their feature description vector (e.g., motion, color)
  - 3 selects candidate shots based on these object features

### 3.10 Similarity test

Although a similarity test in the parameter space is a natural choice for video queries, it is limited to the case of Query-By-Example (Fig. 2(a)); parameter vectors are computed from both the database video shots and the query. The similarity test in this case must compare the corresponding database and query parameters via a suitable norm (multiple parameters can be thought of as vectors). The norms for different sets of parameters (different physical meaning) must be combined (e.g., weighted) in order to arrive at a single decision parameter.

An alternative is a similarity test in the descriptor space (Fig. 2(b) and 2(c)). Here, parameter vectors are transformed into descriptor vectors (interpretation step)

that undergo a similarity test. Again, corresponding sets of database and query descriptors are compared. Descriptor sets with different physical interpretation are combined into a single decision parameter. Note that a similarity test in the descriptor space is applicable to the query-by-example and query-by-description cases (Fig. 2(b) and 2(c)).

The example below shows what such a descriptor-based similarity test may look like. Assume the following query descriptor vector (either after interpretation in query-by-example or after specification by the user)

- global motion: zoom,
- dominant object: moves to right and then is occluded,
- query object: “shape: ellipse, size: large, motion: translation to left, action: reversing the direction”

Then, a descriptor-based similarity test with pruning can be performed as follows:

1. extract candidate shots with zoom by pruning the database by using a descriptor-space similarity test with respect to the zoom/no zoom parameter,
2. from the remaining shots extract candidate shots with the dominant object moving and then occluded by using another descriptor-space similarity test with respect to dominant-object action parameter,
3. examine all remaining shots applying a similarity test to the shape, size, motion and action parameters of the descriptor vector; identify the closest shot with respect to a suitable norm.

## 4 Progress to Date

### 4.1 Intra-image segmentation

Within this work, *image segmentation* denotes the technique for extraction of image entities or structures (regions or objects) so that the outlines of these structures coincide as accurately as possible with the physical 2-D object outlines in the recorded 3-D real scene. A *2-D object* is a projection of a 3D real-world object (e.g., a tree) onto the image plane. A *region* (e.g., the leaves of a tree) is a set of image pixels which are similar with respect to a homogeneity criterion such as color or texture. In the current implementation of the proposed non-supervised intra-image segmentation, two regions  $R_i$  and  $R_j$  with means  $\mu_i$ ,  $\mu_j$  and standard deviations  $\sigma_i$  and  $\sigma_j$  are similar with respect to their luminance variations (texture) when  $|\mu_i - \mu_j| < T_\mu$  and  $|\sigma_i - \sigma_j| < T_\sigma$ . In future work, this texture-homogeneity notion will be extended to chrominance variations.

The intra-image segmentation consists of four steps (Fig. 9 and 10): texture-based binarization (separation of objects), morphological edge detection, contour analysis and tracking, and object reconstruction. The binarization provides the morphological edge detector with a binary image in which regions are separated by one or more black pixels (Fig. 10).

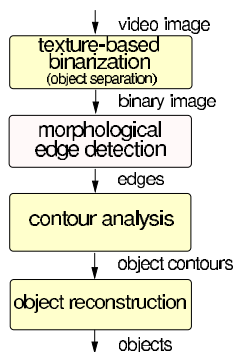


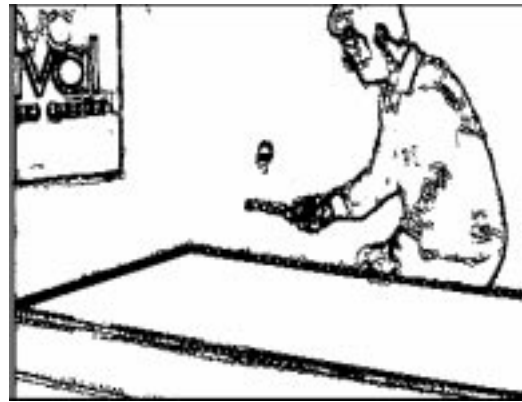
Figure 9: The proposed intra-image segmentation

One objective of this image segmentation is to detect large objects which are necessary for further video processing steps. This choice is oriented towards the human visual system that tracks primarily large moving objects.

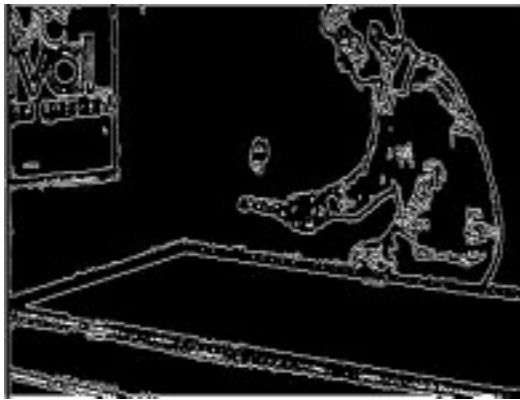
As a result of the segmentation process, a *list of objects* with their features such as area (size in pixels), minimum bounding box (MBB), texture, shape, color, and position is provided for further object-based processing. To reduce the storage space, object and contour points are compressed using a differential run-length code. The proposed segmentation method is robust with respect to noise, has low computational cost, and has regular structure of main components (e.g., morphological operators and block-based binarization).



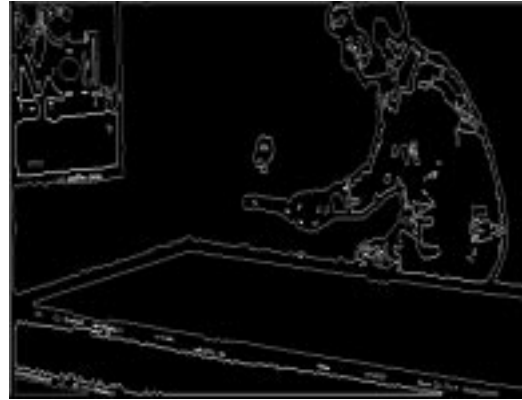
(a) Original



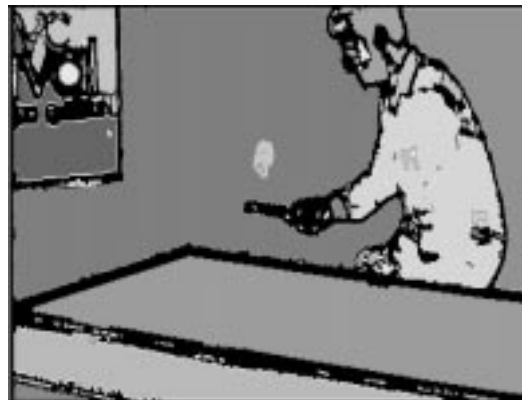
(b) Binarization: note the object separation (black pixels separate white regions)



(c) Morph. edge detection: note the one-pixel wide edges



(d) Contour analysis: note the gap-free contours



(e) Object reconstruction: each object is labeled with a unique gray level

Figure 10: Four step segmentation

#### 4.1.1 Texture-based object separation (binarization)

The objectives of the first step of the proposed segmentation is to provide the subsequent binary morphological edge detection with a binary image in which regions are already separated by black pixels. The goal is to differently label pixels of the input gray-level image  $I(x, y, t)$  which are located in the interior of a texture-homogeneous region (add as a white pixel to the binary image  $B(x, y, t)$ ) and pixels on the boundary (add as black pixels to the binary image) of these regions. Doing this a binary image is created in which smooth white regions are separated by black pixels.

The pixel labeling consists of four stages: detecting the interior of texture-homogeneous regions using large detection masks; adding pixels located at the boundaries; adding small regions which cannot be detected using these large detection masks; filling of holes (these are small black regions which are inside a white region) which were not labeled as white regions because of detection error or noise.

For complex intensity variation, however, the mean and standard variation alone are not sufficiently discriminatory. In future work, other discrimination factors will be introduced, e.g., the correlation function of the block pixels.

**1. stage: detection of the interior of texture-homogeneous regions** Let  $I(x, y, t)$  represent the input (gray-level) image at time instant  $t$  with  $X$  columns and  $Y$  rows and  $B(x, y, t)$  denote the binary image of the same size as  $I(x, y, t)$ . Let  $B_c(x, y)$  (or  $B_c$ ) be the block size  $L \times L$  centered at position  $(x, y)$  of  $I(x, y, t)$ . Let the eight neighboring blocks of  $B_c$  corresponding to the eight directions ( $2 \times$  horizontal,  $2 \times$  vertical,  $4 \times$  diagonal) of the same size be  $B_0, \dots, B_7$  (Fig. 11). Let the distance (in pixels) between the central pixel of block  $B_c$  and the centers of the 8 blocks be  $\delta$ . In this processing stage  $\delta = L$  (Fig. 11).

A pixel  $\mathbf{p} = (x, y)$  of  $I(x, y, t)$  is labeled white, i.e.,  $B(x, y, t)$  is set to 1, if  $B_c$  is located in a *texture-homogeneous sector*  $S_* \in S$ , with  $S = \{S_{to}, S_{bo}, S_{ri}, S_{le}, S_{ab}, S_{be}, S_{re}\}$  as defined in table 1 and shown in Fig. 11.

|          | Sector              | consists of                                   |
|----------|---------------------|---|
| $S_{to}$ | “horizontal top”    | $B_c, B_0, B_1, B_2, B_3, B_4$                |
| $S_{bo}$ | “horizontal bottom” | $B_c, B_0, B_4, B_5, B_6, B_7$                |
| $S_{ri}$ | “vertical right”    | $B_c, B_0, B_1, B_2, B_6, B_7$                |
| $S_{le}$ | “vertical left”     | $B_c, B_2, B_3, B_4, B_5, B_6$                |
| $S_{ab}$ | “diagonal above”    | $B_c, B_1, B_2, B_3, B_4, B_5$                |
| $S_{be}$ | “diagonal below”    | $B_c, B_0, B_1, B_5, B_6, B_7$                |
| $S_{re}$ | “rectangular”       | $B_c, B_0, B_1, B_2, B_3, B_4, B_5, B_6, B_7$ |

Table 1: Sectors in which the texture analysis is performed

$S_*$  is texture-homogeneous if

$$|\mu_{B_c} - \mu_{B_i}| < T_\mu \quad \text{and} \quad |\sigma_{B_c} - \sigma_{B_i}| < T_\sigma, \quad \forall B_i \in S_* \quad (8)$$

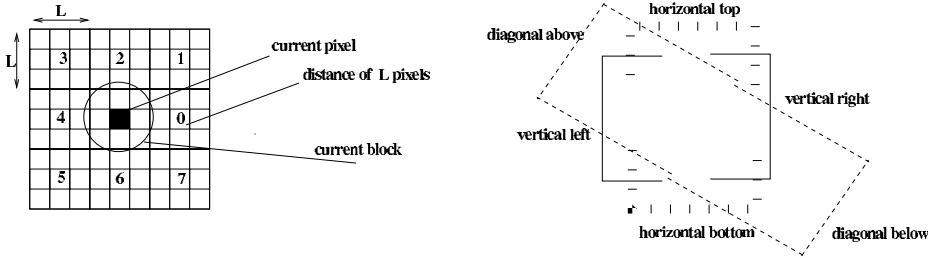


Figure 11: Directions of the texture analysis

where  $\mu_i$  and  $\sigma_i$  denote the mean and standard deviation of  $B_i$ .  $T_\mu = \psi(\sigma_{noise})$ , where  $\psi(x)$  is a strictly monotonic function.

This detection is done hierarchically (Fig. 13), i.e., the labeling starts with a block size  $L \times L$  ( $L = 2n - 1$ ,  $n \in \mathcal{N}$ ,  $n > 2$ ) and then the block size is set to  $(L - 2) \times (L - 2)$ . This is repeated until the block size is 3 (Fig. 13). In each step, non-labeled pixels are examined and eventually set to white. Since the smallest block is 3 its smallest surrounding area is 9 (this is the size of the eight neighboring blocks). Note that the distance in pixels  $\delta$  between the centers of the blocks is set to  $L$  in this stage.

As can be seen in Fig. 13, the interior of each region is detected and set to white. However, pixels located in regions smaller than  $9 \times 9$  and pixels located near the boundary (due to large  $\delta = L$ ) are not labeled.

**2. stage: adding boundary pixels** In this stage, pixels located at object boundaries that were not labeled in the previous stage are detected and labeled using a similar strategy as above but using a fixed block size  $L = 3$  and a variable  $\delta = 3, 2, 1$ .

**3. stage: adding intensity-homogeneity regions** Pixels inside regions smaller than  $9 \times 9$  will not be labeled using the above texture-based strategies. For these regions, an *intensity-homogeneity* test is used for labeling, i.e., a non-labeled pixel is set white if it is located in an intensity-homogeneous block. A block of size  $L \times L$  is intensity homogeneous if the result of **all** the 8 low-pass filters (masks) ( $1 \times$  horizontal,  $1 \times$  vertical,  $2 \times$  diagonal, and  $4 \times$  corners; Fig. 12) are below a given threshold  $T_{inthomog}$ . For example, for  $L = 3$  intensity homogeneity is detected if  $|2 \cdot I(x, y) - I(x, y + 1) - I(x, y - 1)| < T_{inthomog}$ , where  $T_{inthomog}$  is a function of the estimated noise present in the image.

Within this work, the demand was for components that are easy to implement. Therefore, the detection of intensity-homogeneity is done by using fast low-pass masks that are able to detect 8 different directions of homogeneity. In Fig. 12, the directions of the analysis are depicted. As can be seen, special masks for corners are considered. Thus, non homogeneities in object-corners are detected which stabilizes the binarization process in all directions. In this local image analyzer, low pass filters with coefficients  $\{-1, 2, -1\}$  are applied along given directions for each pixel to be added. The result is an effective homogeneity detection, which allows robust binarization.

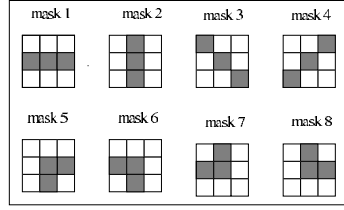


Figure 12: Directions of the homogeneity analyzer

**4. stage: filling of holes** Due to noise and illumination changes, regions created by the above processes may contain small holes of one or more black pixels. These holes need to be identified and filled. A pixel in a hole is labeled white if it lays in i) a texture-homogeneous region as described above with  $\delta = 1$ , ii) a  $L \times L$  intensity-homogeneous block, or iii) a  $P \times Q$  intensity-homogeneous block.

**Adaptation of the binarization to image changes** Because of noise and illumination changes homogeneities could be lost, therefore, the thresholds  $T_\mu$ ,  $T_\sigma$ , and  $T_{inthomog}$  are adapted to the amount of noise present in the image and to the standard deviation of the whole image.

**Adaptation to noise** A method for noise estimation will be developed to allow image segmentation adapted to noise. The threshold for matching similarly textured regions is adapted to the amount of noise in the image as follows: a “clean” region  $R_c$  is a set of pixels  $x_i$  with the mean  $\mu_{R_c}$  and variance  $\sigma_{R_c}^2$ . If noise (with estimated variance  $\sigma_{noise}^2$ ) is added to that “clean” region and if we assume that there is no correlation between the noise and the image signal, then the new mean and variance of the region are:  $\mu_{R_n} = \mu_{R_c} + \mu_{noise}$  and  $\sigma_{R_n}^2 = \sigma_{R_c}^2 + \sigma_{noise}^2$ .

When in the binarization process the mean and the standard deviation of two regions are compared, the estimated standard deviation of the noise signal is used as the threshold  $T_\sigma$ . In the current implementation the following function is used:

$$T_\sigma = \alpha + \beta\sigma_{noise}^2 \quad (9)$$

with  $\alpha, \beta \in \mathcal{R}$ .

**Adaptation to the image structure** The underlying assumption is that images with low standard deviations contain intensity smooth regions, where the intensity variations between these regions may not be significant. Thus the thresholds for identifying homogeneity have to be adapted to the structure of the image (e.g., the image sequence “miss america” shows close homogeneous regions (e.g., background and T-shirt). The threshold in such an image should be set low, in order to detect the fine boundaries between these similar regions.

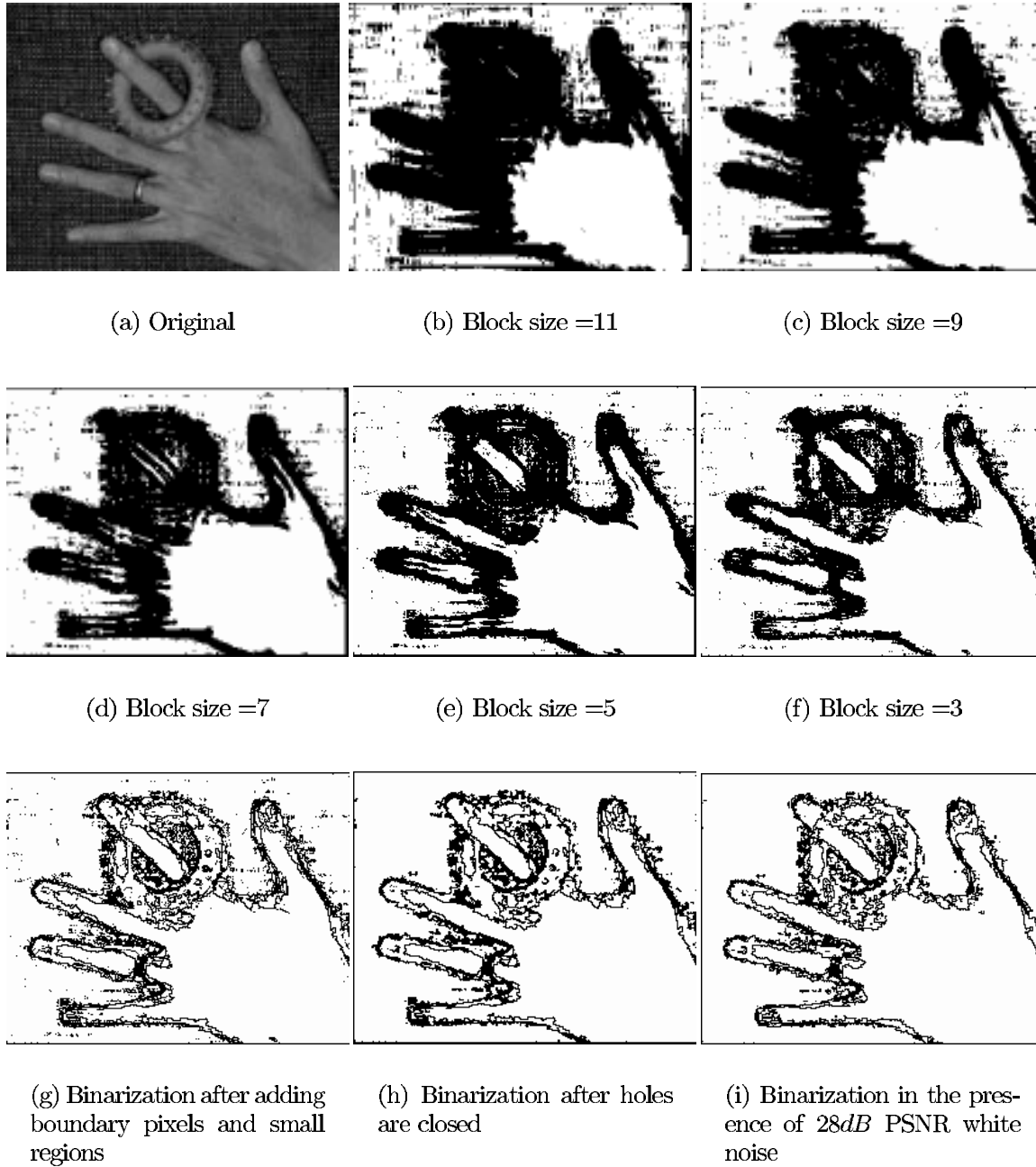


Figure 13: Hierarchical Binarization



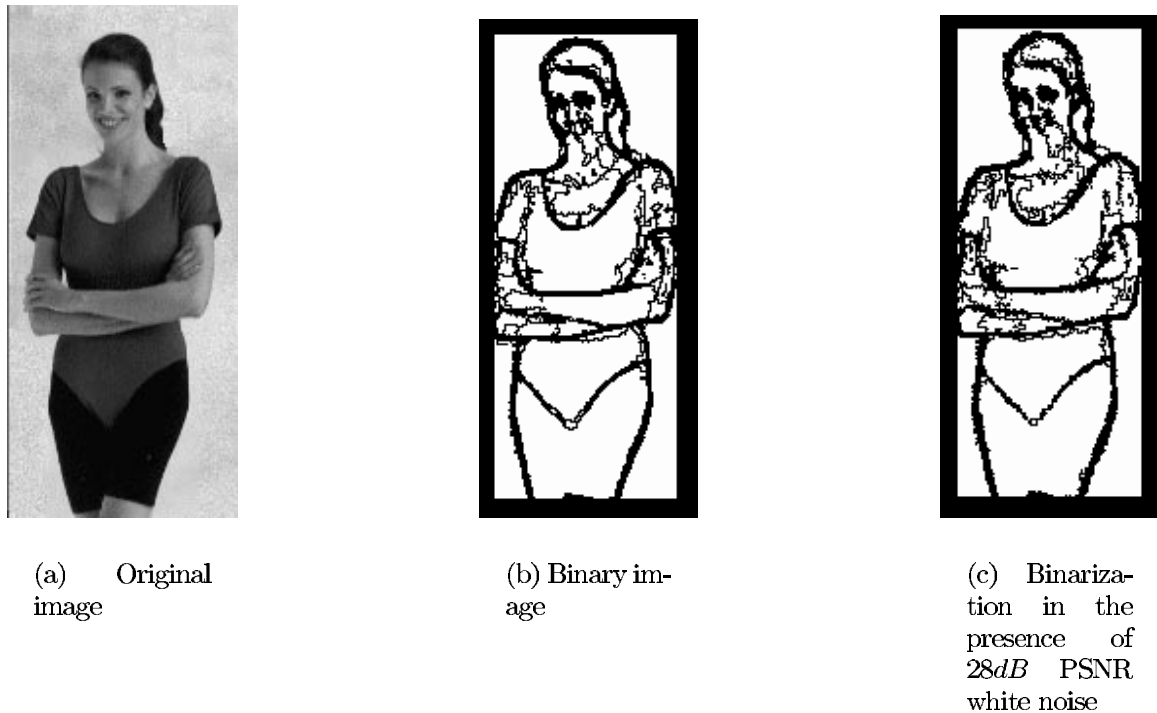


Figure 14: Binarization of a natural image (see text)

**Consideration of illumination changes** Because of illumination changes homogeneity can be lost and holes are a results of such losing. Therefore, by detection and processing of holes in regions illumination changes artifacts are enhanced.

**Simulation results** The binarization was shown to be noise-robust by various sequence simulations. Fig. 13 shows a segmentation of a hand on a non-Gaussian background (red and white fabric). Although this image is overlaid with noise, the hand and the ring are separated from the background which facilitate further processing steps. Observe that in Fig. 14, small regions such as the eyes and teeth are not merged with other regions. Fig. 15 shows a squirrel sitting on grass displayed as gray image. The grass has very variable intensity. Note that the segmentation of the grass is acceptable, although it is based only on the luminance processing. Note that the intensity variations of the grass and the squirrel are at some places very similar. This causes overlapping between the two regions. However, since the color of the squirrel and the grass are different, an introduction of color information in the segmentation process may enhance its performance.

The main distinguishing aspects of the whole binarization process are 1) the separation of regions, 2) the noise robustness which simplifies further segmentation steps such as binary contour point detection, and 3) regularity and low computational costs.



Figure 15: Binarization of a natural image (very variable intensity) (see text)

#### 4.1.2 Morphological edge detection

The field of mathematical morphology contributes a wide range of operators to image processing, based on a few simple mathematical concepts from set theory [22]. The operators are particularly useful for the analysis of binary images, edge detection, noise removal, image enhancement and image segmentation.

The two basic operations in mathematical morphology are erosion and dilation (Fig. 16). These operations are expressed by a kernel operating on an input image. Erosion and dilation work conceptually by translating the structuring element to various points in the input image, and examining the intersection between the translated kernel coordinates and the input image coordinates. For instance, in the case of erosion, the output coordinate set consists of just those points to which the origin of the kernel can be translated, while the element still remains entirely ‘within’ the input image. Erosion is the dual of dilation, i.e., eroding white pixels is equivalent to dilating black pixels. Usually, the origin of a kernel is assumed to be in the center. The

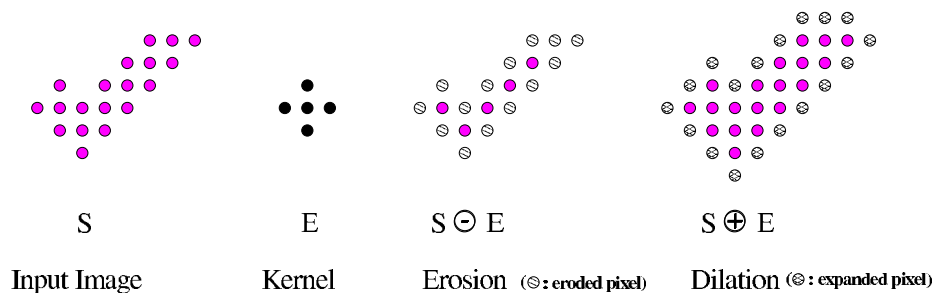


Figure 16: Dilation and Erosion (Note that in this figure the operations are applied to the black pixels)

basic effect of erosion on a binary image is to erode away the boundaries of regions of white pixels. Thus regions of white pixels shrink in size, and holes within those regions become larger. The shape and size of the kernel used determines the precise effect of the erosion on the input image.

Morphological contour point detectors are very effective for binary images  $B(x, y, t)$ , where white pixels denote uniform regions and black pixels denote region boundaries [22]. In this work the following detector is used

$$\mathcal{E}_B = B(x, y, t) \text{ XOR } (B(x, y, t) \ominus K_{(2 \times 2)}), \quad (10)$$

where  $B(x, y, t)$  denotes the binary image,  $\mathcal{E}_B$  is the edge image of  $B(x, y, t)$ ,  $\ominus$  denotes erosion operator, and  $K_{2 \times 2}$  is the erosion kernel used. Below, a new erosion rule is proposed that aims at a precise detection of edges, especially at corners.

**Standard erosion** A mathematical definition of erosion for binary images is as follows: Suppose that  $B$  is the set of Euclidean coordinates corresponding to the input binary image, and that  $K$  is the set of coordinates for the kernel, and  $\mathbf{p} \in B$  which represents a point in the coordinate system. Let  $K\mathbf{p}$  denote the translation of  $K$  so that its origin is at  $\mathbf{p} \in B$ . Then, the erosion of  $B$  by  $K$  is the set of all points  $\mathbf{p}$  such that  $K\mathbf{p}$  is a subset of  $B$ .

A frequently used kernel is a  $3 \times 3$  cross kernel with the origin at its center (Fig. 16). To compute the erosion of a binary input image by this kernel, each of the white pixels in the input image is considered. For each white pixel the kernel is superimposed on top of the input image so that the origin of the kernel coincides with the input pixel coordinates. If for every pixel in the kernel corresponding pixel in the image underneath is a black pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are black, however, the input pixel is also set to black.

For the  $3 \times 3$  kernel, this results in the removal of white pixels that are not completely surrounded by other white pixels (assuming 8-connectedness). Such pixels must lie at the edges of white regions. Clearly, white regions shrink, and holes inside a region grow.

**Proposed erosion** Standard erosion is defined for kernels around an origin (Fig. 17). To achieve precisely positioned edges with single-pixel widths the standard erosion requires  $3 \times 3$  kernel. Then, an incomplete corner detection results (Figs. 17, 18). To avoid this, a new decision rule is used with a  $2 \times 2$  kernel with no origin. In the new rule, if all the four binary-image pixels inside the  $2 \times 2$  kernel are white, then all four pixels in the output image are set to white, if they were not eroded in a previous step. If at least one of the four binary-image pixels inside the kernel is black, then all the four pixels in the output image are set to black.

Using this newly-defined rule for morphological erosion, the binary image is first eroded. Then, the result is XOR-combined with the binary image, thus producing an edge image. To achieve high performance of the subsequent image segmentation steps, the accuracy of edge position, the robustness against noise and the detection of edges with single pixel width are necessary. One main feature of the new erosion is its accuracy of detection of corner edges (Fig. 17,18). In addition, due to the small

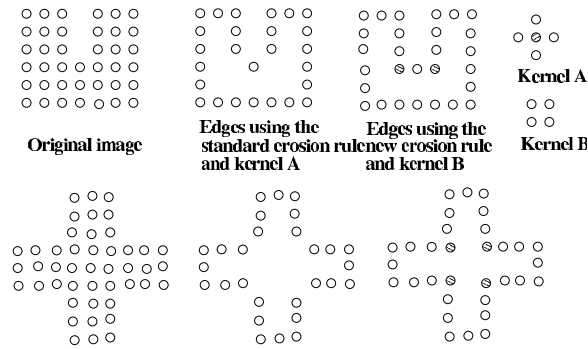


Figure 17: Comparison of morphological edge detectors

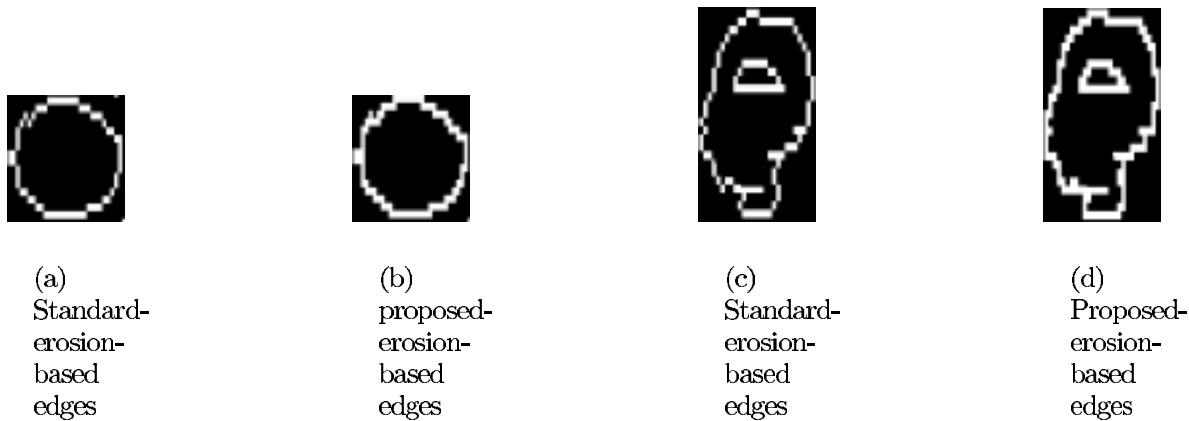


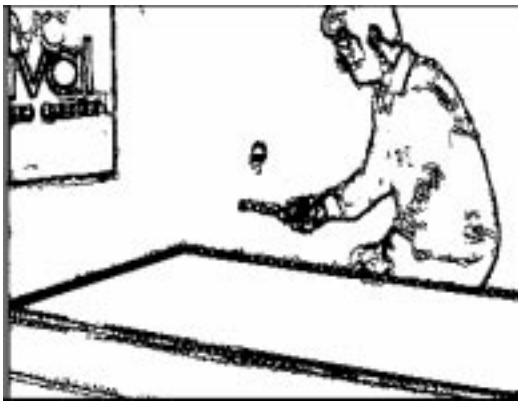
Figure 18: Comparison: proposed and standard-erosion-based morph. edge detection

kernel used, the new erosion requires less line memory and fewer calculations than the standard morphological erosion using  $3 \times 3$  kernels.

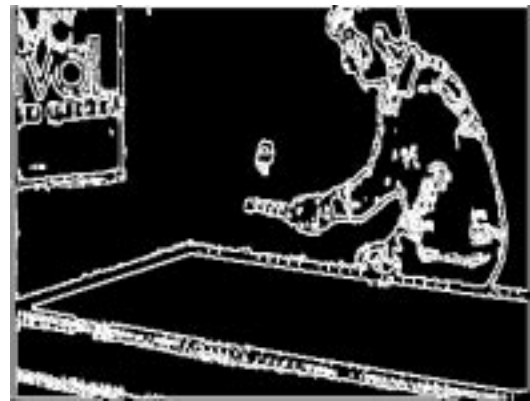
The proposed edge detection has been compared to morphological edge detection based on standard erosion. As can be seen in figure 18 the proposed edge detection preserves the shape of the objects.

The proposed morphological edge detector has been also compared to some gradient-based Sobel and Laplace [22] edge detectors (Fig. 19). It has shown low calculation, noise robustness and detection accuracy especially at corners.

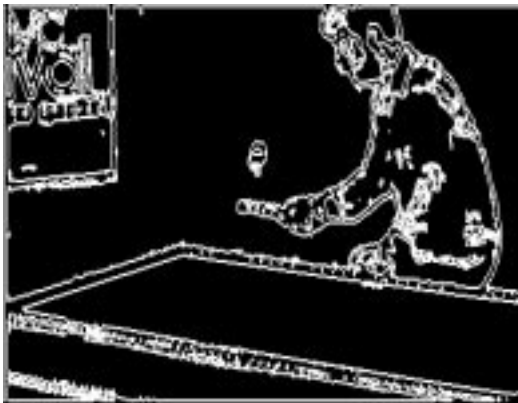
- Sobel with the kernels  $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$  in x-direction and  $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$  in y-direction.
- Laplace with the kernel  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$



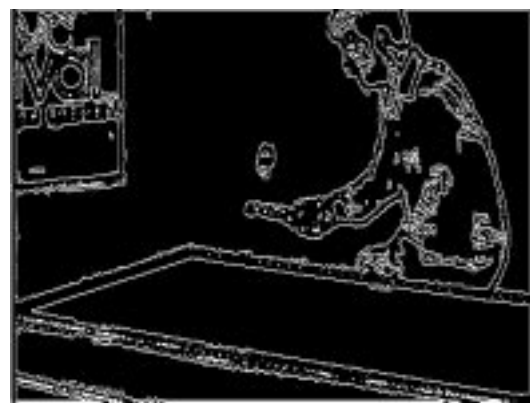
(a) Binary image



(b) Sobel edge detection: connects regions, yields multiple-pixel wide edges



(c) Laplace edge detection: better than Sobel detector but it also connects regions and yield multiple-pixel wide edges



(d) Morph. edge detection: preserve region separation and yield one-pixel wide edges

Figure 19: Comparison: gradient edge detection with the proposed edge detection

### 4.1.3 Contour analysis

The morphological edge detector delivers independent edges which are not spatially related. To become global object information (object contour), discrete chains of the contour points are generated using a contour point tracking method described in [3]. Within this processing step, contours that are small or not significant are eliminated. This is due to the fact that small objects are not important for the perception of an image and could be assumed to be the result of noise or failure in previous processing steps.

### 4.1.4 Object reconstruction

In general, the contours, only characterized by contour points and their spatial relationship, are not sufficient for object-based video processing (e.g., object manipulation and description), which is based on the data of the position of the object points. Therefore, these contours are filled so that objects are reconstructed [3].

The robustness of the whole segmentation method can also be demonstrated when segmenting noisy (even by heavy noise such as  $25dB$ ) images (Fig. 20(c), 13(i), and 14(c)). Furthermore, simulations show that the segmentation is fast. In particular, the edge detector and the contour analysis have very low calculation costs.

## 4.2 Estimation of motion using object correspondence

In general, objects can be seen as entities that are both temporally and spatially coherent over multiple images throughout an image sequence. This is so because the motion of objects in the real world is usually smooth. When an object moves, its spatial features, such as color and texture, do not vary significantly. Various segmentation-based motion estimation methods have been introduced. Most of them show high computational cost and non-regular architecture. In the context of video retrieval, a qualitative description of motion is needed rather than precise estimation. Motion estimation and tracking have been widely studied in the field of image sequence analysis and computer vision. However, video retrieval implies very different goals. Here, the main requirement is not precision of the motion vector fields, but a flexible representation that is more searchable and amenable to manipulation than a block- or pixel-based representation.

Below, an object-based motion estimation algorithm **between two images** is proposed (Fig. 21).

### 4.2.1 Principles

The extracted object information (here area, minimum bounding box, position, motion direction) is used in a four step process: finding object correspondence using extracted object features (Fig. 22(a)), measurement of the displacement based on minimum-bounding-boxes (Fig. 22(b)), motion analysis (Fig. 25(a), Fig. 26(a)), and motion update.

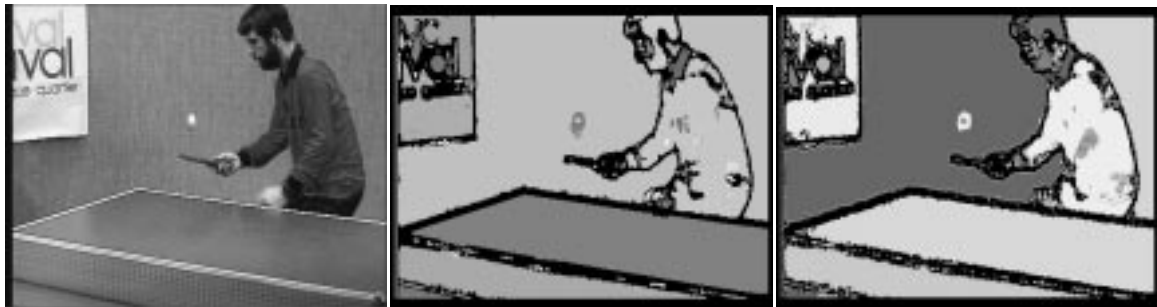
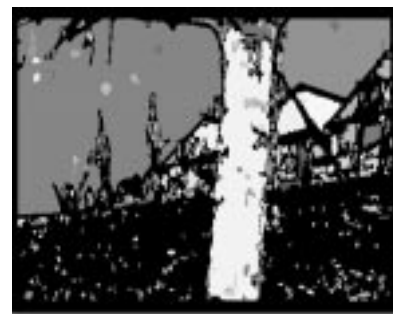
(a) Original image  $I(x, y, 1)$ (b) Segments of  $I(x, y, 1)$ (c) Segments of noisy  $I(x, y, 1)$ (d) Noisy image  $I(x, y, 1)$   
(25dB PSNR white noise)(e) Segments of  $I(x, y, 19)$ (f) Original image  $I(x, y, 1)$ (g) Segments of  $I(x, y, 1)$ 

Figure 20: Segmentation of natural images (each object is labeled by a unique gray level)

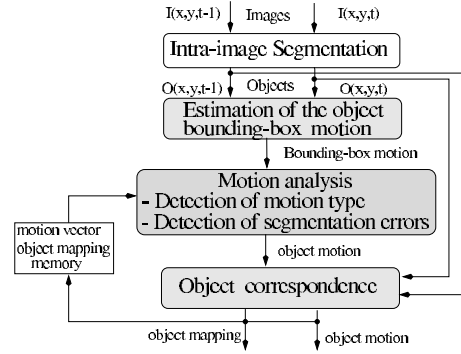


Figure 21: The proposed object-matching approach

- First, a correspondence between objects of image  $I(x, y, t - 1)$  (reference objects) and objects in the subsequent image  $I(x, y, t)$  is found (Fig. 22(a)). The object correspondence is based on the following features: shape (object bounding-box height, object bounding-box width), size in pixels (here object area), location or position (here distance between the centroids of two objects), and motion direction,  $\mathbf{v}_{current} - \mathbf{v}_{new}$ , which is defined as the difference of the current (i.e., between  $I_{t-2}$  and  $I_{t-1}$ ) and the new (i.e., between  $I_{t-1}$  and  $I_t$ ) object motion of the reference object.
- then, motion is measured using the displacement between the corresponding MBBs (Fig. 22(b)). This is also applied when two objects are matched and their MBBs are not of the same size. In this case it is assumed that the object undergoes a deformation.
- in case of a non-translational motion or object, this measured displacement will not reflect such changes. Consequently, based on the specific displacement of the sides of the MBBs, the estimated motion is analyzed and different motion types (translation, rotation, zoom, and acceleration) as well as image segmentation errors (*object-fusion* or *object-separation*) are detected.
- Finally, motion vectors of the second step are adapted to the detected motion types. As a result, depending on the motion type detection, one or more motion vectors for each detected object are estimated (Sec. 4.2.2).

Because of possible multiple object matches (i.e., an object in image  $I_{t-1}$  may be matched to several objects in image  $I_t$ ) and because of zero matches in case of occlusion, object matching is an ill-posed problem, e.g., no unique solution exists. To partially regularize this problem, the search is performed within a finite area around the centroid of the object to be matched. Furthermore, if the object to be matched and the reference object have the same feature (e.g, color), this feature is excluded from the matching process. In addition, if during the matching process a new, better correspondence than a previous one is found, the matching is revised, i.e., the previous correspondence is removed and the new one is established (Fig. 23).



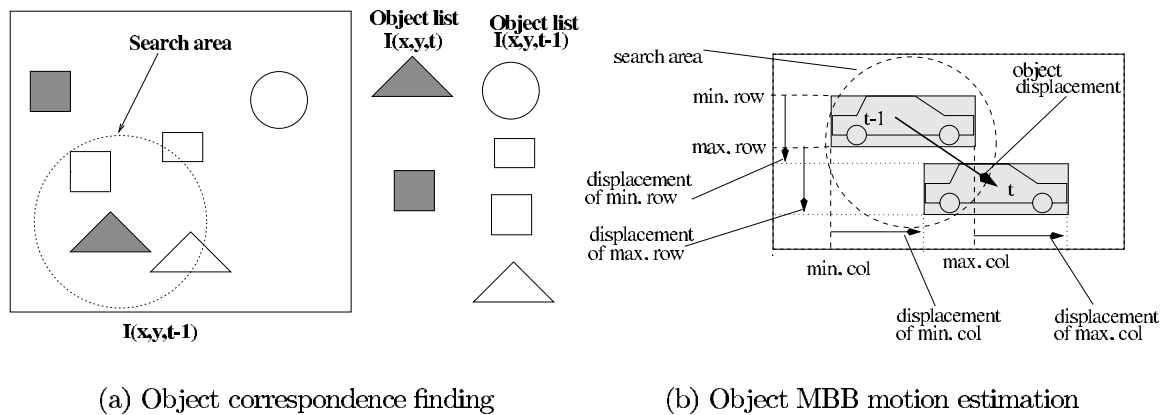


Figure 22: Principles of object-based motion estimation

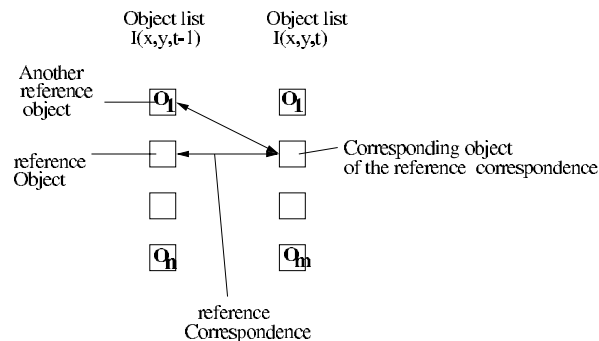


Figure 23: Correspondence conflict

The presented motion estimation scheme can be applied where the following assumptions are valid: 1) the shapes of moving objects between successive images do not change drastically, 2) the actual motion of the moving objects is within the search range, and 3) the luminance intensities of moving objects remain almost unchanged along spatio-temporal directions. The first assumption is for the accuracy of the prediction of segmentation the second and the third assumptions are for the accuracy of the motion vectors.

#### 4.2.2 Detection of multiple-object motion

Objects correspond, often, to a large area of the image. Thus, a simple translational model used in block matching is no longer applicable and a more complex motion model has to be introduced. These motion types inside segmented objects need to be detected. Depending on the detected object motion types either 'one object/one motion-vector' relations or 'one object/several motion-vectors' relations are established. In the case of zoom and rotation, e.g., objects are divided into different regions and a 'one region/one motion-vector' relation is achieved by interpolation of the motion vector found in the object-bounding-box motion estimation step.

Translational object motion is characterized by an identical displacement of the parallel boundaries of the object MBB, i.e., the object has a unique motion vector (Fig. 24). If the displacement of the parallel boundaries of the object MBB are

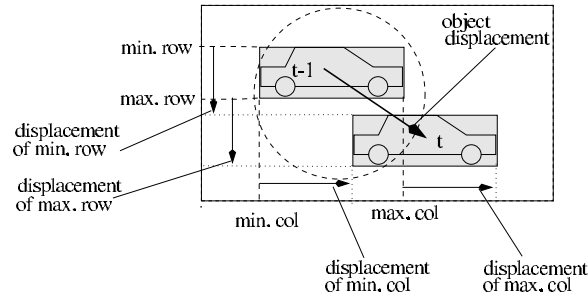
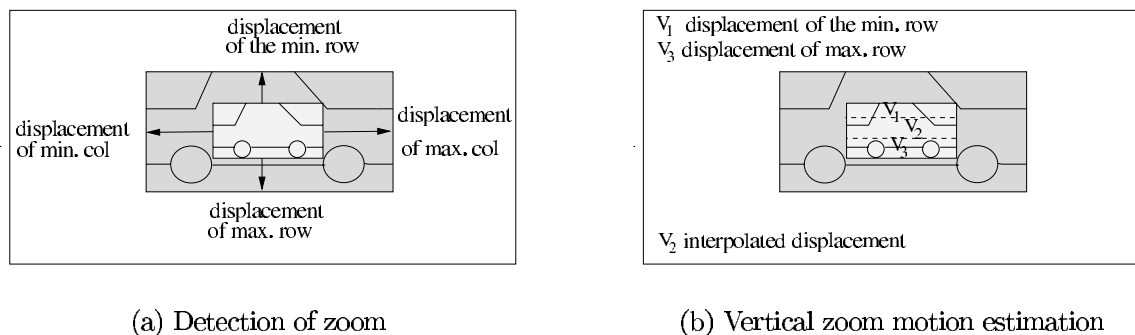


Figure 24: Detection of translational motion

symmetrical and identical (e.g., if one boundary is displaced to the right by 3 pixels, the parallel boundary is displaced by 3 pixels to the left) then the object motion will be characterized as zoom (Fig. 25(a)). In this case, each object may have several motion vectors (Fig. 25(b)). When the difference of the displacements of the parallel boundaries of the object MBB is small, then a rotation of the object is detected (Fig. 26(a)). Also here, objects are divided into different regions and a “one region/one motion-vector” relation is achieved by interpolation of motion vector found in the object-bounding-box motion estimation step (Fig. 26(b)). This detection concept was developed independent of an application. Its robustness in an object-based retrieval will be tested.

#### 4.2.3 Detection and correction of faulty segmented objects

Because of the difficulties of spatial-based image segmentation, the matching-based motion estimation technique detects image segmentation errors such as the fusion (i.e., objects are aligned) or separation (i.e., objects are apart) of objects (Fig. 27(a)



(a) Detection of zoom

(b) Vertical zoom motion estimation

Figure 25: Detection of zoom

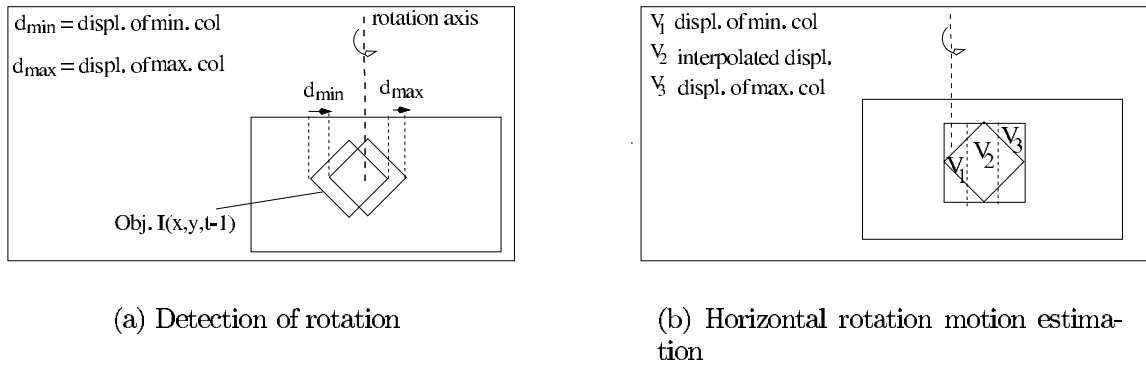


Figure 26: Detection of rotation

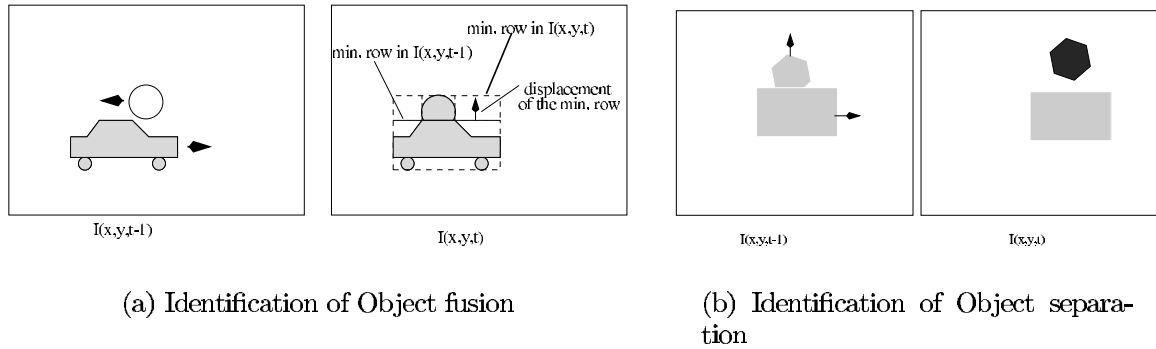


Figure 27: Detection of segmentation error

and 27(b)) and corrects the segmentation results and adapt itself to the new object segmentation. These errors are detected when a large difference between the displacement of the parallel boundaries of the MBB is given.

Common to segmentation-based motion estimation methods is a huge amount of computations. This is mainly because of underlying object and motion models. Many segmentation-based motion estimation use region growing strategies or try to minimize a global energy function but it is often very difficult to find its minima [45] for image segmentation. Furthermore, such methods include several refining and hierarchical steps. Because of the demands of the retrieval process, within the proposed estimation concept, non-regular-structured and complex operations are avoided, particularly in the image segmentation.

#### 4.2.4 Experimental results

**Computation costs:** Simulation results show that the computational cost for the object-based motion estimation is about 1/38 of the computation cost of a fast block matching described in [8]. This method has a complexity of about forty times lower than that of a Full-search algorithm [7]. Furthermore, regular operations (regular

binarization, binary morphological operator, regular contour analysis) are applied. Thus, this method seems to be suitable for real-time video applications such as video retrieval.

**Robustness:** In this stage of the project, motion compensation techniques were used in order to evaluate the performance of the estimation algorithm. Motion compensation is a non-linear prediction technique where the current image  $I(x, y, t)$  is predicted from  $I(x, y, t-1)$  using the motion estimated between these images. Object predictions using block matching and object matching are compared. The following figures show the performance of the proposed method.

- Fig. 28: (a) An original image from a synthetic sequence is shown. (b) Prediction of two objects that move in opposite directions (above the zone-plate) using object-based motion. (c) The same objects when their motion is compensated using block-based motion. (d) Two objects are merged together due to segmentation error. (e) The motion vectors (coded as gray-levels) inside these objects are shown using object-based method. (f) Block-based motion vectors of these overlapped objects.
- Fig. 29: Some of the faulty block-based predictions are compensated using the object-based motion vectors. However, other artifacts are introduced. This is mainly due to the segmentation errors at object boundaries. Note that for video retrieval, precise object reconstruction is not the goal.
- Fig. 30: Note the enhancement in the wheels and inside the object.

**Conclusion** The introduced object-based motion estimation is a prototype that will be adapted and improved in order to be used for video retrieval, e.g.,

- in the current implementation, feedback for the resulting motion is integrated to detect segmentation error and to correct it. Later, the integration of resulting motion vectors will be examined for the enhancement of the image segmentation steps.
- At this stage of the work, the minimum bounding box, object area, and motion direction are used for finding object correspondences. Other basic feature will be also introduced to support the object correspondence establishing.
- For the purpose of object action interpretation, this approach will be extended and modified to matching and tracking throughout the sequence.

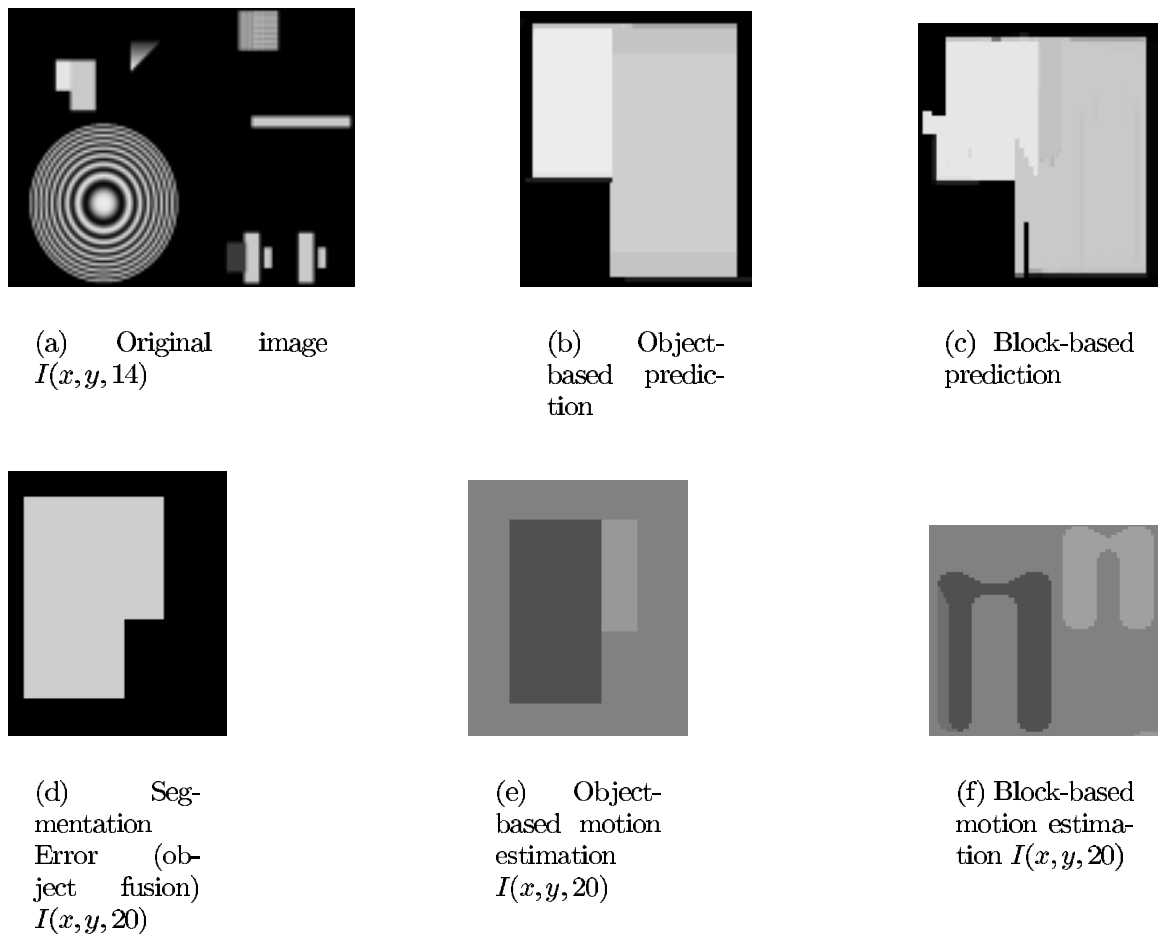


Figure 28: Behavior of the algorithm by object overlapping (see text)

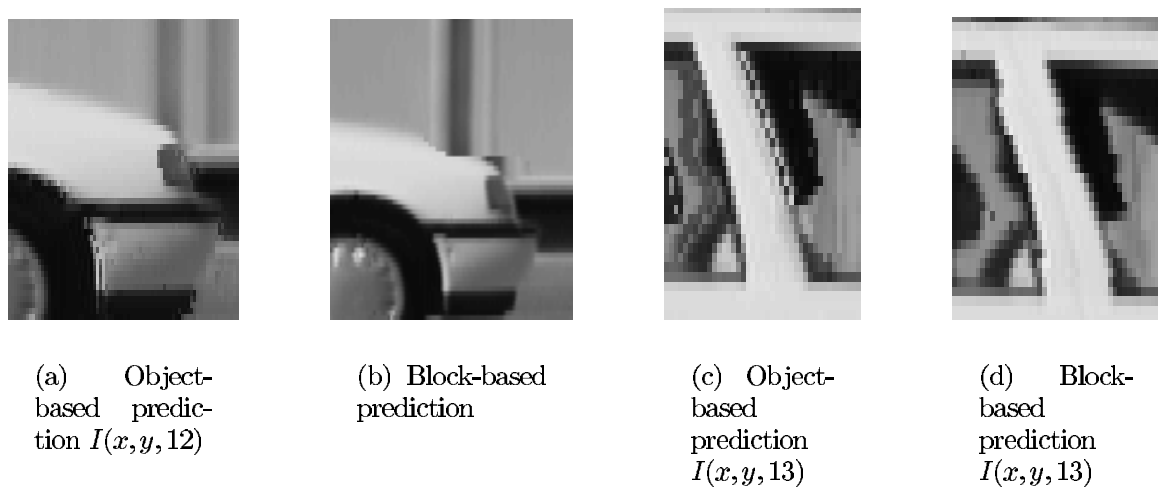
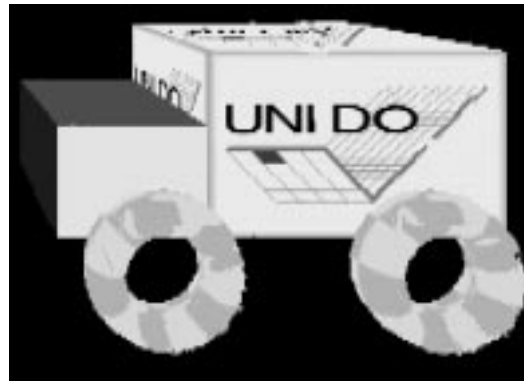


Figure 29: Comparison of prediction of objects in a natural sequence (see text)



(a) Original image  $I(x, y, 1)$



(b) Block-based prediction



(c) Object-based prediction

Figure 30: Comparison of prediction of objects in a synthetic sequence (see text)

## 5 Work Plan

The following components will be implemented, tested and their simulation results studied:

- creating a suitable video database (2 weeks).
- performing shot boundary detection manually (1 week),
- extensively testing the image segmentation and motion estimation (4 months, see below),
- developing a method to estimate the noise in an image sequence (3 weeks),
- developing an object tracking method (1 month),
- implementation of the global motion estimation method (1 week),
- developing a method for interpretation of global motion, object basic features, object actions (2 months),
- building a suitable query interface (1 week), and
- testing the whole system (1 month).

For the proposed object-based motion estimation, robust features such as shape and texture will be used for object correspondence. In the current implementation of the object matching method, no feedback of the resulting motion is used to enhance the segmentation steps. The advantages of such an iterative step will be also studied (2 months). Furthermore, the following improvements of the intra-image segmentation are planned (1.5 months):

- Object isolation:
  - in addition to the mean and the standard deviation, other discrimination factors (e.g., the correlation function) of the luminance variation in a region will be introduced.
  - instead of working with the raw image data, the mean and the standard deviation of each block centered on each pixel will be calculated and the described object isolation will be applied on that data.
  - Several studies show that color-based image processing yields more stable results. Therefore, color information will be integrated.
- Edge Detection: A mathematical foundation of the new morphological operations erosion and dilation (a new dilation rule will be defined using a similar principle used for the proposed erosion) will be introduced.

---

## 6 Anticipated Contributions

Upon completion of this research the following contributions are anticipated:

- new approach to object-based video retrieval with new functionalities,
- definition of qualitative query features of video sequence that may be an alternative to query by example,
- new interpretation techniques to qualitatively describe spatio-temporal objects, their basic features, and their actions,
- feature-based similarity test based on description vector rather than parameter vector,
- novel video analysis approaches,
  - fast intra-image segmentation,
  - fast object-based motion estimation, and
  - object tracking.
- a new approach to estimate white noise in an image, and
- pruning in the description space rather than in the parameter space



## References

- [1] E. Adelson and J. Bergen, *The Plenoptic Function and the Elements of Early Vision*, Computational Models of Visual Processing, ch. 1, (M. Landy and J. Movshon, eds.), The MIT Press, Cambridge, 1991
- [2] P. Aigrain, H. Zhong, and D. Petkovic, *Content-based Representation and Retrieval of Visual Media: A state-of-the-Art Review*, Multimedia tools and applications 3, 179-192, 1996
- [3] A. Amer, *Motion Estimation Using Object Segmentation Methods*, Master Thesis, Dortmund University, Dept. for Computer Science, Dec. 1994 (in German)
- [4] A. Amer and E. Dubois, *Segmentation-based Motion Estimation for Video Processing using Object-based Detection of Motion Types*, IS&T/SPIE's Conf. in Visual Communication and Image Processing, San Jose (USA), Jan. 23-29, 1999
- [5] A. Amer and H. Schröder, *A New Video Noise Reduction Algorithm Using Spatial Sub-bands*, IEEE Int. Conf. on Electronics, Circuits, and Systems, Rodos (Greece), October 1996
- [6] H. Blume, A. Amer, and H. Schröder, *Vector-based postprocessing of MPEG-2 signals for digital TV-receivers*, Proc. IS&T SPIE's Conf. in Visual Communication and Image Processing, vol. 3024, pp. 1176-1187, San Jose (USA), Feb. 1997
- [7] H. Blume and A. Amer, *Parallel Predictive Motion Estimation Using Object Recognition Methods*, European Workshop & Exhib. on Image Format Conversion and Transcoding, Berlin (Germany), March 1995
- [8] H. Blume, *Vector-Based Nonlinear Upconversion Applying Center Weighted Medians*, IS&T/SPIE Symposium on Electronic Imaging, Nonlinear Image Processing VII, San Jose (USA), February 1996.
- [9] P. Bouthemy, M. Gelgon, and F. Ganancia, *A unified approach to shot change detection and camera motion characterization*, IRISA, Publication interne n°1148, Nov. 1997
- [10] P. Correia and F. Pereira, *The role of analysis in content-based video coding and indexing*, Signal Processing, vol. 66, pp. 125-142, 1998
- [11] S. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, *A Fully Automatic Content-Based Video Search Engine Supporting Multi-Object Spatio-temporal Queries*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 5, pp. 602-615, Sept. 1998

- 
- [12] S. Chang, J. Smith, H. Meng, H. Wang, and D. Zhong, *Finding Images/Video in Large Archives*, Columbia's Content-Based Visual Query Project, D-Lib Magazine, February 1997
- [13] Y. Deng and B. Manjunath, *NeTra-V: Towards an object-based Video Representation*, IEEE Trans. on Circuits and Systems for Video Technology, Sept. 1998
- [14] J. Courtney, *Automatic Video Indexing Via Object Motion Analysis*, Pattern Recognition, vol. 30, no. 4, pp. 607-625, 1997
- [15] A. Cross, D. Mason, and S. Dury, *Segmentation of remotely-sensed images by a split-and-merge process*, International Journal of Remote Sensing, 9(8), 1329-1345, 1988
- [16] E. Dubois and J. Konrad, *Estimation of 2-D motion fields from image sequences with application to motion compensated processing*, in Motion Analysis and Image Sequence Processing (M. Sezan and R. Lagendijk, eds.), ch. 3, pp. 53-87, Kluwer Academic Publisher, 1993
- [17] F. Dufaux and F. Moscheni, *Segmentation-based motion estimation for second generation video coding techniques*, Video Coding: the Second Generation Approach, L. Torres and M. Kunt Eds., Kluwer Academic Publishers, pp. 219-263, 1995.
- [18] F. Dufaux and J. Konrad, *Efficient, Robust and fast Global Motion Estimation for Video Coding*, IEEE Trans. on Image Processing, June 1998
- [19] G. de Haan et al, *IC for Motion Compensated 100 Hz TV with smooth Movie Motion Mode*, Proc. IEEE ICCE, Chicago (USA), June 1995
- [20] G. de Haan, O. Ojo, and T. Kwaaitaal-Spassova, *Automatic 2-D and 3-D noise filtering for high-quality television receivers*, Phillips Research Laboratories, Television System Group, 1996
- [21] R. Haralick, K. Shanmugam, and I. Dinstein, *Textural features for image classification*, IEEE Transactions on Systems Machines and Cybernetics, 3, 610-622, 1973.
- [22] R. Haralick, L. Shapiro, *Computer and Robot Vision*, vol I,II, Reading, Addison-Wesley 1992
- [23] J. Flack, *On the Interpretation of Remotely Sensed Data Using Guided Techniques for Land Cover Analysis*, PhD thesis, eeuwin Center for Remote Sensing Technologies, Feb. 1996

- [24] M. Ferman, M. Tekalp, and R. Mehrotra, *Effective Content Representation for Video*, Proc. IEEE ICIP'98, Chicago, III., Oct. 1998
- [25] A. Gasch, *Object-based vector analysis for video restoration*, Master's Thesis, Dept. of Elect. Eng., Univ. of Dortmund, July 1997 (in German)
- [26] B. Günsel and M. Tekalp, *Content-based Video Abstraction*, Proc. IEEE Int. Conf. Image Processing, Chicago, IL, Oct. 1998
- [27] F. Golshani and N. Dimitrova, *A language for content-based video retrieval*, Multimedia tools and applications 6, 289-312, 1998
- [28] G. Iyengar and A. Lippman, *VideoBook: an experiment in characterization of video*, Proc. IEEE ICIP '96, Lausanne, Sept. 16-19, 1996
- [29] R. Jain, A. Pentland, and D. Petkovic, *Workshop report*, NSF-ARPA Workshop on Visual Information Management Systems, Cambridge (USA), June 1995
- [30] S. Krishnamachari and M. Abdel-Mottaleb, *Hierarchical clustering algorithm for fast image retrieval*, Proc. IS&T/SPIE's Conf. in Storage and Retrieval of Image and Video Databases VII, San Jose (USA), Jan. 24-29 1999
- [31] A. Mitiche and P. Bouthemy, *Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods*, Int. Journal of Computer Vision 19(1), 29-55, 1996
- [32] A. Mitiche, *Computational Analysis of Visual Motion*, Plenum Press, New York, 1994
- [33] B. Manjunath, *Image Browsing in the Alexandria Digital Library (ADL) Project*, D-Lib Magazine, Sept. 1995
- [34] M. Naphade, R. Mehrotra, A. Ferman, J. Warnick, T. Huang, and A. Tekalp, *A high performance algorithm for shot boundary detection using multiple cues*, Proc. IEEE Int. Conf. Image Processing, Chicago (USA), Oct. 1998
- [35] T. Pavlidis, *Structural Pattern Recognition*, Springer, Berlin 1977
- [36] R. Picard, *Light-years from Lena: Video and image libraries of the future*, Proc. IEEE Int. Conf. on Image Processing, Oct. 1995
- [37] R. Picard, *A Society of Models for Video and Image Libraries*, MIT MediaLab., TR no. 360, IBM Systems Journal, 1996

- 
- [38] A. Pentland, R.W. Picard, and S. Sclaroff, *Photobook: Content-Based Manipulation of Image Databases*, International Journal of Computer Vision, vol. 18, no. 3, pp. 233-254, 1996
- [39] Y. Rui, T. Huang and S. Chang, *Digital Image/Video Library and MPEG-7: Standardization and Research Issues*, IEEE ICASSP'98 , pp. 3785-3788, Seattle (USA), May 12-15, 1998
- [40] Yong Rui, Thomas S. Huang, and Sharad Mehrotra, *Relevance feedback techniques in interactive content based image retrieval*, Proc. IS&T SPIE Conf. in Storage and Retrieval of Images/Video Databases VI, San Jose (USA), 1998
- [41] H. Sundaram and S. Chang, *Efficient Video Sequence Retrieval in Large Repositories*, Proc. IS&T SPIE Conf. in Storage and Retrieval of Image and Video Databases VII, San Jose (USA), Jan. 24-29 1999
- [42] S. Santini and R. Jain, *Do Images Mean Anything?*, Proc. Int. Conf. on Image Processing, vol. I, pp. 564-xx, Santa Barbara (USA), Oct. 1997,
- [43] H. Zhang, A. Kankanhalli, and W. Smoliar, *Automatic Partitioning of Full-Motion Video*, ACM Multimedia Systems, vol. 1, no. 1, pp. 10-28, 1993
- [44] H. J. Zhang, C. Y. Low, S. W. Smoliar and J. H. Wu, *Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution*, Proc. ACM Multimedia'95, San Francisco, pp. 15-24, Nov. 1995
- [45] S. Zhu and A. Yuille, *Region Competition: Unifying Snake/balloon, Region Growing and Bayes/MDL/Energy for multi-band Image Segmentation*, IEEE Trans. on PAMI, vol.18, no. 9, Sept. 1996