CERIAS Tech Report 2000-04

AN ACCESS CONTROL MODEL FOR VIDEO DATABASE SYSTEMS

by Elisa Bertino, Ahmed K. Elmagarmid,
Moustafa A. Hammad

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47909

2000-04

An Access Control Model for Video Database Systems

Elisa Bertino[1], *Ahmed K. Elmagarmid*[2], *Moustafa A. Hammad*[2]

[1] Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41 20135 Milano, Italy
bertino@dsi.unimi.it

[2] Computer Science Department, Purdue University
1398 Computer Science Bld, West Lafayette IN 47907
{ake,mhammad}@cs.purdue.edu

Abstract

In this paper we addresses the issue of providing an access control model for video databases. Our model allows one to specify authorization subjects not only by their identifiers, but also according to their credentials. Credentials have been used in access control mechanism for textual data. Their use in access control for video data is a new application. The use of credentials for specifying subjects makes specification of authorization policies clear and easy. Our model also provides a clear definition of authorized objects that can be either specified implicitly by using concept expressions, or explicitly by providing a set of logical video elements. Thus we provide access control based on the semantics of video data and not just on its physical representation. By providing access control based on semantic contents, we exploit the richness of information in video data and enable semantically high-level specifications of authorizations. A distinct contribution of our work is to provide access control for different video granularities ranging from whole logical video stream to subframe regions. The access model also provides a categorization of privileges that are meaningful for video data. The privileges we have devised are abstract and suitable to interact with video. Because our authorization model is based on video contents, it can be easily applied to different video data models, for example MPEG-7.

1 Introduction

Handling video data needs multidisciplinary skills and leads to problems that go far beyond video storage and databases or image processing. Video possesses unique characteristics such as volume and complexity of data even for simple and short video clips. There are by the most conservative estimates over six million hours of feature films and video archived with a steady rate of growth. It is envisioned that over 1.8 million Gbytes of MPEG digital video is available [8, 23]. It is obvious to see that soon many other applications will find their way into video repositories. Medical, financial, engineering and scientific as well as entertainment are but a few examples of what is coming the way those solving video database access and retrieval problems.

There are a number of issues that are being addressed by researchers such as standards, video data modeling, video cut and scene change detection, video query, browsing and retrieval, video editing and hardware to enhance the quality and resolution of video.

A lot of work has been done in describing video data content. There is a possibility of additional inroads in the use of the new MPEG standards, MPEG-7 for example, which is a video database standard. Video content can be classified as [7]:

- Semantic content: this deals with the knowledge or information contained in a given video segment

- Audiovisual content: this includes audio signal, color intensity and distribution, texture patterns, object motions to name a few.

- Textual content: which provides important annotations that can be used in constructing the complete video metadata. Video caption information is also used to index the video and partly to aid in the retrieval of the correct video frames required in a query.

However, little work has been done in protecting against the unauthorized access of video data content. In order to deal with video in a correct way, we must incorporate all three aspects listed above in the proposed security protocols. This issue is addressed in this paper in an attempt to take advantage of what we already know about access control, video semantics and access patterns. The problem of protection for a video database (VD) system entails addressing several issues. On one hand the system must be secure against malicious use, data-corruption and illegal dissemination, thereby ensuring the quality and correctness of the data and protecting the intellectual property of information producers. On the other hand, the system should ensure that all users entitled to see the data, according to the organization security policies, are actually able to do so. Therefore, not only users must be given proper access authorizations, but also denial of service situations should be prevented. Such situations could easily arise given the intensive computation resource usage which could be required by VD queries or other operations performed by a VD server. Thus, a complete solution to VD protection should address authentication and security services to guarantee privacy, integrity, confidentiality, copyright issues, and resource usage. A first step towards such a solution is to provide a full-fledged access control model and related mechanisms allowing one to selectively give access to data according to the organization policy. The development of such a model is the contribution of this paper.

In a conventional database environment access control is usually performed against a set of authorizations stated by security administrators or users according to some security policies. In its most basic form, an authorization is specified as a triple $< s, o, m >$. Such a triple specifies that subject s is authorized to access object o (the protected object) under mode m, where the mode refers to the actions that can be executed on the protected object, such as read or write. Whereas such an approach provides the basic conceptual framework for reasoning about authorization models, it is clear that it must be properly extended in order to satisfy the additional challenging requirements characterizing VD systems.

First of all, video data are not only used today for entertainment. They are used in variety of applications environments, such as medical applications, teaching, environmental protection, manufacturing processes, scientific research, just to name a few. In such environments, it is often the case that different classes of users within the same organization must receive different authorizations for the same set of data. For example, consider a school giving access to VD to both teachers and students. Consider a set of videos illustrating firearms. Whereas teachers can be allowed to see all such videos, the students may only be allowed to see the videos not showing how to operate guns with the exception of students having age equal or greater than 18. It is clear that an organization must have a way to specify such authorizations that are then used for filtrating data before returning them to the users. There is thus the need for models and mechanisms supporting the specification of authorizations on the basis of user qualifications, characteristics, tasks, or positions within organizations rather than user identity.

Another crucial requirement is the support for content-dependent authorizations on video data objects. By content-dependent authorizations we mean that authorizations are granted or denied to a given user (or class of users) depending on the actual content of the video data objects. Consider again the policy stating that all videos showing how to operate guns must be made available only to students who are 18 or older. Supporting such a policy requires determining which videos show gun operations. Such a requirement thus calls for the integration of the access control mechanism with mechanisms able to support content-based video access and indexing.

A third important requirement is that different views of the same video data may need to be provided to different classes of users. This requirement calls for an access control model supporting varying granularity levels in authorization objects. In other words, the access control mechanism must allow one to exclude from a given video, a specific frame, a set of frames, or even an object from a frame or set of frames. The last possibility requires the ability to obscure or hide part of a frame.

In this paper we propose an access control models satisfying the above requirements. Specific features of the proposed model include: access control specification for video data objects based on their contents rather than their identifiers; flexible specification of authorization based on the notion of user credentials; varying granularity of authorization objects ranging from an entire video, to part of a video to specific portions of the frames. Our model also provides functions for resource usage controls, such as limitations on the play time or video resolution. In order to support content-based access control, we have integrated our access control model with the Logical Hypermedia Video Data Model, LHVDM [15]. Therefore, we use all the facilities provided by LHVDM in order to identify video semantic contents. Note, however, that our access control model is quite general and can be integrated with any system able to perform semantic concept extraction from VD or with any system supporting description of video data object contents through the use of meta-data, such as MPEG-7.

The following sections are structured as follows. Section 2 introduces related work and contrast it with our work. Section 3 introduces the reference video data model and the video elements used throughout this paper. The authorization model and the detailed specification of its components are described in detail in Section 4. Section 5 presents the access control mechanisms and the algorithms proposed in this paper. Section 6 and Section 7 introduce the access control architecture and some implementation issues. The paper is concluded with Section 8 where future work is also briefly discussed.

2 Related Work

Several efforts have been reported to extend conventional database access control models to deal with new data types and models and to provide new functions in authorization management. Such efforts include authorization models for object-oriented databases [9, 20], and for web pages [21], temporal authorization models [4], and extended authorization models for relational databases [5]. Such models are not however fully adequate for the protection of information in a VD system. The main reason is that authorizations are specified in terms of user, or user groups, and object identifiers rather than in terms of user profiles and object contents. In particular, content-based authorizations for VD objects is rather novel and has never been addressed before. Also, how to support varying protection granularity levels for VD objects has not been addressed before. The only approach we are aware of has been proposed by Kumar and Babu [18]. This approach is however very primitive in that it only allows one to hide some frames for specific classes of users; it neither supports content-based authorizations nor allows one to hide part of a frame. Moreover, even though such an approach considers users as partitioned into user categories, it does not support authorizations containing predicates against user profiles. However, there has been some preliminary work on specifying authorizations with different levels of granularity, on content-based authorizations, and on user credentials, which we discuss below.

A proposal for a content-based access control for textual digital libraries (DL) has been recently proposed [2]. In such an approach, authorizations are based on concepts associated with textual data objects. Concepts to be associated with a given set of documents are identified by means of a document classification system [13] based on information extraction [19]. Such an approach also supports a two-level granularity for authorization objects by which authorizations can be associated either with an entire object or with a part of it. Different parts of an object, relevant for access control, must be manually marked by the security administrator or some other users through some slot-identifiers. Such an approach also supports access control based on user characteristics through the use of user credential. A credential is a set of information, called credential attributes, concerning a given user that are relevant for security purposes. Even though our model uses the same credential approach of the DL authorization model, our work differs from this model under three respects. The first is that such a model has no provision for video access control and therefore the object granularity model supported is very limited. By contrast, our model provides a more articulated object granularity model deriving from the need of supporting video data. Such data can be organized, from the point of view of security, according to several hierarchical levels, such as entire video, logical segments, frames, parts of frames. The second is that access modes required for video data objects are different with respect to those used for textual documents. For video data objects one must provide access modes such as query, browsing, as well as the possibility of browsing for a specified duration (such as the first 10 minutes of the videos), or browsing the video data objects according to different resolutions [25]. The last function is used when one wants to use authorization to control resource usage. Also, the approach used in the DL authorization model for concept extraction cannot be used for VD objects, since it works only for textual data.

Winslett et al. [24] have first proposed the notion of user credentials as a way to effectively enforce data access control in a distributed system. The idea is that in a distributed system, access control cannot be simply based on user-ids. Rather, it should be based on some form of user profile. We adopt such an approach in our model. However, the model proposed by Winslett et al. only deals with subject specification in terms of credentials and does not deal with specific types of protection objects

Content-dependent access control has been addressed both in relational DBMS, through the use of views, and in object-oriented databases [12]. However, such approaches only deal with conventional, formatted data. In such context, content-based access control can be enforced by simply specifying some conditions against attribute values of data objects. To this purpose, authorization rules can be augmented with predicates allowing one to specify conditions on the attributes' values. By contrast, due to the nature of video data objects, content-dependents access control for a VD must be based on the semantics of the video objects, rather than on the attributes characterizing them. Video attributes often only deal with physical characteristics of the video data objects (for example, the color intensity and distribution, texture patterns, and number of frames or segments composing the video) and therefore are not significant for access control. Access control in a VD thus becomes mode difficult since one must rely on a mechanism able to provide semantics information about the contents of video objects.

Finally, we would like to remark that the notion of credential has some similarity with that of role [22]. Roles can be considered as a set of tasks or responsibilities associated with a particular activity within a given organization. Under role-based models, all authorizations needed to perform a given activity are granted to the role associated with that activity, rather than being granted directly to users. Users are then made members of roles and they selectively play such roles, thus acquiring all authorizations granted to the roles. Our notion of credential is similar to roles in that also in our model we use credential as subject authorizations, rather than only relying on user-ids. However, in our model credentials carry on a set of attributes allowing one to specify more articulated access control policies in terms of these attributes.

3 A Reference Video Data Model

In this section we introduce the reference video model used in the development of our access control model. We build our access control mechanism on top of Logical Hyper Video Data Model (LHVDM) proposed in [15]. Such model uses an annotation-based layered approach for modeling video data. Under this model a video is seen according to different abstraction levels. *The physical video level* represents the raw video data and can assume the form of a whole *video stream* or a set of frames that represent different scenes in a video according to a segmentation criteria, for example those frames with no significant inter-frame difference in terms of their visual contents. Such a set of frames is called *video segment. The logical video level*, on the other hand, represents the composer view of the video. It also supports a whole or segmented video view. A key feature of LHVDM is the notion of *hot object* which represents a subframe region in a sequence of video frames. An hot object, representing a semantically relevant object within the video, is defined over a set of frame intervals having the hot object as one of their components. Those intervals represent the life time, *LTI*, of the hot object over the video stream. All logical video elements contain annotations that describe the semantics associated with them. The annotations can be extracted from the closed caption [1] associated with the video using a closed caption decoder, by voice recognition technique of the audio signals in video, or be edited by the video composer. To represent the relations between the various video abstraction levels, a set of mapping relations are used, for example to map logical video elements into the corresponding physical ones. The main components of the model and the relations between them are graphically shown in Figure 1. A shorted description of each of them is presented in what follows.

PVS a *physical video segment* is a sequence of frames that segment the physical video based on their visual content variation.

PV a *physical video stream* represents the actual video stored in the database. The PV consists of a set of PVSs. PV and PVS together represent the physical components of the video data.

LV a *logical video stream* is a temporally ordered set of physical video segments, possibly from different physical video streams. A LV is conceptually defined as a 4-tuple:
(*pvs, t, vid, uid*), where *pvs* denotes the set of physical video segments of which the logical video stream is composed, *t* represents the associated annotations, *vid* is the logical video identifier, and *uid* is the creator identifier.

LVS a *logical video segment* represents a consecutive sequence of video frames within a given logical video stream. Each logical video segment represents a user's comment or interpretation of that sequence of frames. A LVS is represented as:
(*[start,end], t, vid, uid*), where *[start,end]* represents the video frame interval, *t* represents the associated annotations, *vid* denotes the identifier of the logical video stream from which the LVS is defined, and *uid* denotes the creator identifier.

HO a *hot object* represents a logical concept that can be defined as a temporally ordered but not necessary consecutive sequence of sub-frame regions in a logical video stream [14]. A hot object can, for example, represent a face of a person, a new weapon, a building. The geometric description of hot objects can be extracted either manually by editing the video frames, or semi-automatically by using scene change detection techniques [16]. The concept of hot object is an important feature in the video model since it enables handling video data at a finer granularity level. A hot object is represented in the model as a 5-tuple:
(*g, LTI, t, link, id*), where *g* represents the geometric representation of the hot object, *LTI*, life time interval, represents a set of video frame intervals that include this hot object, *t* represents the associated annotations, *link* is set of pointers to other LVs, LVSs, or HOs, and *id* represents a triple (*vid, oid, uid*) specifying the logical video stream identifier, the hot object identifier, and its creator identifier.

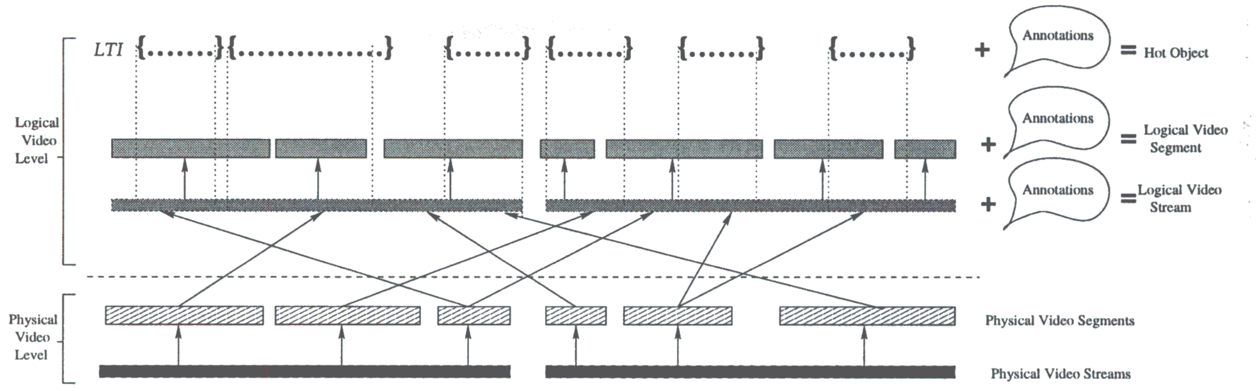


Figure 1: Main components of LHVDM

MAP a set of mapping relations are defined between various video components such as $PV \leftrightarrow PVS$, $PVS \leftrightarrow LV$ etc.

It should be clear that logical video components are introduced into the video database by users and the same video object can be interpreted differently by different users. Therefore, the *uid* is a relevant component of the identifiers of any logical video components.

In this video model, logical video segments and hot objects always have a life time interval, that is, a single interval [Start, End] for a logical video segment and possible multiple intervals for an hot object. A set of temporal relations are defined over the life time interval of both of these components. This set consists of the thirteen Allen's relations covering all the possible temporal relations between two intervals [3] (see Figure 2).

A distinct property of any hot object is the existence of spatial properties in addition to temporal ones. A hot object has a geometric representation that defines its appearance in the sequence of frames and change over time as hot object changes its position. Different geometric representations for a hot object are used; for example a *polygon, minimum bounding box (MBB), minimum bounding circle (MBC) and centroid.* The spatial relations between hot objects are either topological relations like *overlap, disjoint, touch and inside*, directional relations like *north, south, east, west, above, below, left, right, middle, center*, or distance relations like *far, near, close.*

As hot object possesses both spatial and temporal characteristics, a spatio-temporal relations are also defined among hot objects. For example, the *approach* relation indicates an approaching hot object and is interpreted as the spatial distance between them decreases over their common life time interval.

LHVDM has several features that are relevant for security. First, it supports different levels of granularity within each abstraction level, like physical and logical levels. For example, at a logical level, one can associate access restrictions with an entire logical video stream, or with some logical video segments, or hot objects. Handling video at different granularity levels exploits the richness of video data and provides a rich set of different views for different privileged users by distributing

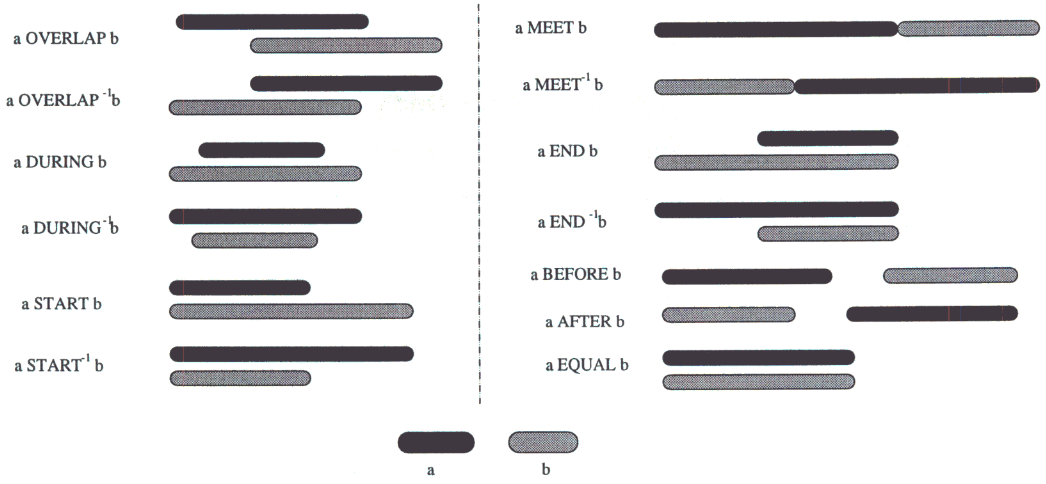


Figure 2: Different temporal relations between frame intervals

access provision over those components. Second, the introduction of hot objects and video segments enables the incorporation of spatial and temporal operation in describing access requirements. Third, the association of annotations with logical video components provides a semantic content based access to video and increases its expressive power. Furthermore, a *knowledge base* can be built based on the association between concepts contained in the video annotations. The use of such a knowledge base greatly simplifies and improves video data access [17] and enables a smart understanding of the required permissions or denials. For example, all strong violent scenes in a movie can be determined by including strong violence as a prefix in their annotations, or extracting violent-related words from the knowledge base, and be prohibited for viewers under a certain age.

4 The Authorization Model

Incorporating access control in a VD introduces a new layer of abstraction to video content. For example, the same video can be played to different users according to different views depending on the specific access privileges of each user. Take for example the case of video classification according to its contents. R-rated movies should not be viewed by children where TVG-rated movies can be seen by general audience. Also, some parts of the video can be blurred for certain audience according to their privileges or concern. For example a face of a person in a TV interview or strong violent scenes in a news broadcast can be blurred for general broadcast programs. Figure 3 represents a set of frames in a video clip where the face of the person is blurred, for the purpose of hiding identity.

Access control can also be used to provide some sort of different quality of service to various classes of users according to their privileges. For example, during network congestion, low privilege users can be given lower quality video clips to save network bandwidth. This feature adds up to the reliability of VD in terms of maintaining a certain level of service in different situations.

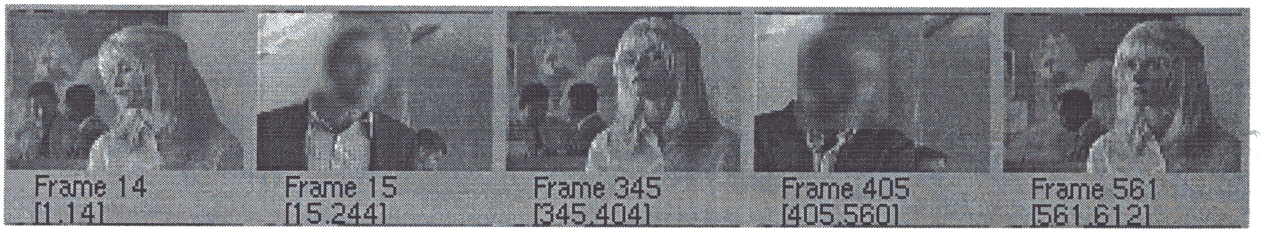


Figure 3: A video clip with blurred parts (face of a person)

Another important application is to provide access to part of the video like its beginning or a brief description of the content for the purpose of demos and advertising. Although access control introduces a layer of overhead in the processing of users request can end up in saving bandwidth and providing organized and secure access to video database.

Integrating access control in database management system is usually achieved by specifying a set of *authorization rules* and *control procedures* [6]. Authorization rules describe *who* is allowed to access *what* in the database. Control procedures deploy these rules on database transactions. For example, an authorization rule could specify that *user A* is allowed to play *movie B* and the control procedure checks this rule each time *user A* tries to access the database. If the user tries for example to change the content of the movie, his transaction will be denied by the access control component. This mode of access control is always referred to as a *closed system access control.* In closed systems access control only explicitly authorized accesses are allowed [6]. In other words the default is to deny every access to the system except otherwise specified by existing authorization rules. The general format of an authorization rule is a 3-tuple $< subject, object, mode >$. In the following sections a detailed specification of the main components of the authorization rules, e.g. subject, object, are presented.

4.1 Subject Specification

Subjects in an authorization model represent entities trying to access the database. In our model, we will assume subjects to be end-users. However, our model can be easily extended by including subjects, such as roles and groups.

Because video data is rich in terms of amount of information it contains, and incorporates different media types, it is important that the audio and visual capabilities of subjects be related with the various dimensions of video data. A suitable access control model must thus be able to incorporate information about user characteristics and profiles. To this purpose, we adopt in our model the notion of credentials [2]. A credential is a set of security-relevant information, called *credential attributes*, pertaining to a given user. Credentials extend the expressive power of security specifications and allows one to better relate subject qualification and characteristics with semantic contents of video data objects. The credential mechanisms allows one to specify subjects in authorization rules not only by using their user-ids (or other system-defined identification

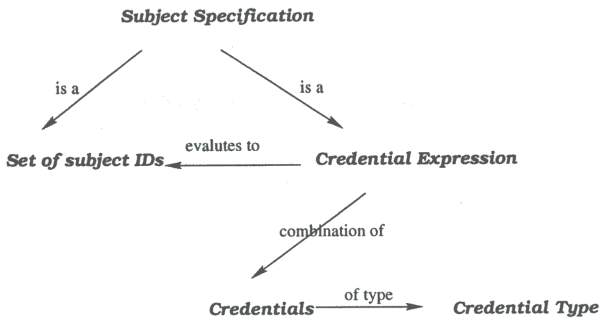


Figure 4: Possible relations between subject and other credential components

mechanism), but also implicitly by specifying the conditions users need to verify in order to access a given video, or sets of video. To this purpose, the credential mechanism provides a simple language for formulating *credential expressions*, that is, Boolean combination of predicates, against which user characteristics are matched. Evaluation of a credential expression thus results in a set of users that satisfy the expression requirements. To ease the process of credential specification, credentials with similar structures are grouped into a *credential type.*

In the following, we recall from [2] definitions of some the above-mentioned notions. The relations among those notions are also graphically illustrated in Figure 4. In the following definitions, we assume that the following sets are given: $\mathcal{AN}$ - it denotes a set of attribute names; $\mathcal{T}$ - it denotes the set of the possible types (such as `integer`, `real`, `Boolean`, `character`, and `string`) of attributes in $\mathcal{AN}$; $\mathcal{V}$ - it denotes the set of legal values for types in $\mathcal{T}$. Moreover, we denote the set of credential-type identifiers and the set of credential identifiers with $\mathcal{CT}$ and $\mathcal{CI}$, respectively. We use $\mathcal{U}$ to denote a set of account identifiers, associated with the users authorized to access the system.

Definition 4.1 (Credential Type) [2] A credential type is a pair (*ct_id, attr*), where *ct_id* $\in \mathcal{CT}$ is the credential type identifier; and *attr* is a set containing an item for each attribute of the credential type. *attr* in turn is a triple (*name, dom, a_type*), where *name* $\in \mathcal{AN}$ is the attribute name, *dom* $\in \mathcal{T}$ is the attribute domain, and *a_type* $\in$ {`opt`, `mand`} specifies whether the attribute is optional (`opt`) or mandatory (`mand`) . □

We use notation A(*ct_id*) to denote the set of the names of attributes in credential *ct_id.*

Example 4.1 The following is an example of credential type:
`(Student,{(address,string,mand), (GPA,number,mand), (status,string,opt), (registered, Boolean,mand)})`
This credential type specifies that for users that are students the information relevant for security purposes are the address, the GPA, the status, and whether the student is registered or not. Also, the credential type specifies that the status attribute is not mandatory, whereas all the others are. △

A credential is an instance of a credential type and provides the corresponding values to the specified attributes, as specified by the following definition.

Definition 4.2 (Credential) [2] A credential c is a 4-tuple *(c_id, user_id, state, ct_id)*, where *c_id* $\in \mathcal{CI}$ is the credential identifier, *user_id* $\in \mathcal{U}$ is the identifier of the user with whom the credential is associated; *state* $= (a_1 : v_1, \ldots, a_n : v_n)$, where $a_1, \ldots, a_n \in A(ct\_id)$ are the names of the attributes of c, $v_1, \ldots, v_n \in \mathcal{V}$ are their values; and *ct_id* $\in \mathcal{CT}$ is the identifier of the credential type of which c is an instance. □

Note that the same subject can have different credentials. For example, a student may have a restricted access to his school video library but a full access in a video database of a cable company. In order to identify the credential, both user and credential identifiers are required.

Example 4.2 The following is an example of credential:
`(c`$_1$`, John,(address:Waldron street, GPA:3.5, Status:Graduate, Registered:Yes),Student)`. The credential is associated with the user, whose user-id is John, and is an instance of the credential type `Student`. △

Credential expressions provide a high-level mechanism to specify subjects according to the values of their attribute credentials. Credential expressions are specified by a simple language which consists of the following components [2]:

- a set of variables Var_U, ranging over the set $\mathcal{U}$ of user identifiers;
- a set of predicate symbols *Pred* of arity one, with type Var_U. For each credential type $ct \in \mathcal{CT}$, a corresponding predicate symbol $ct()$ is defined in *Pred*. Such predicates capture the associations of users with credential types.

Expressions that can be specified in our language are formally defined as follows.

Definition 4.3 (Credential Expression) Let $\Theta = \{=, \neq, >, <, \geq, \leq, \in, \notin, \subseteq, \not\subseteq, \subset, \not\subset, \supset, \supseteq, \not\supset, \not\supseteq$ $\}$ be a set of relational comparison operators. The set $\mathcal{CE}$ of credential expressions is built from atoms and Θ as follows.
Atoms can be of the following types:

- $P(x)$, where $P \in$ *Pred* and $x \in Var_U$;
- $x.a$ *op* v, where $x \in Var_U$, $a \in \mathcal{AN}$; $v \in \mathcal{V}$, and $op \in \Theta$.

Then the set $\mathcal{CE}$ of credential expressions is recursively defined as follows:

- Every atom is a credential expression.
- If CE_1 and CE_2 are credential expressions, then $CE_1 \wedge CE_2$, $CE_1 \vee CE_2$, $\neg CE_1$, (CE_1) are also credential expressions.

□

Example 4.3 The following are examples of credential expressions:

- `Student(x)`: it is a credential expression representing all subjects that are students;
- `x.age` $\geq$ `18`: it is a credential expression denoting the subjects having age ≥ 18. △

The evaluation of a given credential against a credential expression consists of replacing the attribute names, in the credential expression, with the corresponding values in the credential and determining the truth value of the resulting credential expression. This process is very much the same of evaluating a query predicate against a given tuple. The result of a credential expression evaluation against a given credential thus depends on the values of attributes in the credential. A problem arises if this attribute has a null value. Such a situation may arise for credential attributes that are optional. To deal with such a case, we adopt the same approach of the SQL language and we use a three-valued logic. Therefore, if the evaluation of a given credential against a credential expression returns the truth value *unknown*, we say that the credential does not verify the expression. Therefore, the corresponding user will not belong to the set of users denoted by the credential expression. Then any authorization rule, where such a credential type appears, will not apply to such user. In other words, the users will not be given any authorization by such rules.

The set of subjects to which authorizations apply can thus be specified either explicitly through the use of user-ids or implicitly by means of some credential expression, as stated by the following definition.

Definition 4.4 (Subject Set Specification) A subject set is either a set of subject identifiers in 2^U, or a credential expression in $\mathcal{CE}$. □

Example 4.4 In what follows we present some examples of the different forms that can be used to introduce subjects in our model.

- `(Viewer(x) AND (x.age` $\geq$ `18))`: this expression denotes all viewers who are ≥ 18 years old.
- `(Student(x) AND (x.registered = OK))`: this expression denotes students who are already registered.
- $\{\mathtt{uid}_1, \mathtt{uid}_2, \mathtt{uid}_3 ...\}$: this is an explicitly specified list of authorized subjects. △

The use of credentials to specify subjects increases the expressive power of the video access control model. More specifically, the use of credential types is crucial in categorizing subjects in different classes according to specific characteristics. This classification can be used to provide certain levels of quality of service to different classes in case of resource degradation. This technique is already deployed in many service companies. For example in telecommunication companies different levels of quality of service are provided for different rates of subscription. An immediate extension would be to provide a reduced resolution movie to low rate subscription in a pay-by-view cable company. Also, credential types can be extended to include audio and visual characteristics of subjects that affect the way they access video contents. For example, in a deaf school, students may be allowed to view video with the associated audio content replaced by a superimposed sign language (introduced by an illustrator) and hence save the bandwidth and speed up the display process.

4.2 Object Specification

Whereas subject specification is independent of the underlying data model, object specification is tightly coupled with it. As discussed in Section 3, video data have a physical and a logical representation. We can think of the logical level as virtual video elements that are composed of the underlying physical video components. Physical video elements can itself be easily seen as logical video components that map directly to themselves.

The specification of an *authorization video object* in our model is based on the logical video elements, that is, logical video stream, logical video segment, and hot object (cfr. Section 3). In our access control model, these video elements can be specified either directly, by providing their identifiers, or through a set of *concepts*. Concepts are extracted from the annotations associated with the logical video elements. In their simplest form, concepts are just keywords present in the video annotations. Concepts can be combined in *concept expressions*. A concept expression involves one or more concepts combined according to some Boolean, spatial, temporal or spatio-temporal operators. The use of concepts is the key to support content-based access control in our model. Concepts specify semantic information about the actual contents of a set of video data objects and can be considered a way to identify those video elements. By using concepts we can thus restrict the access to videos dealing or not dealing with a specified content.

The formal definition of concept expression is presented in what follows. In the definition, we assume that the following sets are given: $\mathcal{CP}$ - it denotes a set of concepts; $\mathcal{VOP}$ - it denotes a set of spatial, temporal, and spatio-temporal operators as described in Section 3.

Definition 4.5 (Concept Expression) The set $\mathcal{CPE}$ of concept expressions is built from atoms and $\mathcal{VOP}$ as follows.
Atoms can be of the following types:

- c, where $c \in \mathcal{CP}$.

Then the set $\mathcal{CPE}$ of concept expressions is recursively defined as follows:

- Every atom is a concept expression.
- If CpE_1 and CpE_2 are concept expressions, and $op \in \mathcal{VOP}$, then $CpE_1 \wedge CpE_2$, $CpE_1 \vee CpE_2$, CpE_1 op CpE_2, $\neg CpE_1$, (CpE_1) are also concept expressions.

□

The evaluation of a concept expression against a LV, LVS or a hot object results in *a set of frame intervals*. Such frame intervals are those containing the concepts satisfying the given concept expression.

Example 4.5 The concept expression "`World War II` $\wedge$ `Charles De Gaulle`" will result in all logical video elements that include both the phrases *World War II* and *Charles De Gaulle* in their annotations. △

Example 4.6 Whereas, the concept expression "`Charles De Gaulle` *DURING* `World War II`" will result in all frame intervals of General *De Gaulle* that temporally fall during the period of frame intervals that contain the concept "*World War II*" . △

Example 4.7 The concept expression "`Discovery` *CLOSE* `Spying Satellite`". The close operator is applied only to hot objects (which has spatial characteristic), and the expression will result in all frame intervals that have Discovery space shuttle spatially close to spying satellite in NASA video library. △

An important contribution of this paper is to provide access control on different levels of video granularity. Such feature is crucial in access control since dealing with whole video stream as an atomic entity may be too strict in specifying protection requirements. For example, instead of restricting the subject from seeing the whole video stream, he may be allowed to view it but with the restricted parts are intentionally obscecured. Also, structural comositin of video data, video stream consisting of video segments that contains frames, and the variety of media objects (audio, visual and text) included in the video requires to have a mechanism to select among them. For example, it may be necessary to present the same video to different subjects according to different views for purposes of security or moral issues.

The theoritcal level to introduce access control on video data is to specify the authority on pixel level, as the pixel is the finest granular part in video data. Video stream can be thought of as a long sequence of pixels over time, see Figure 5. Consecutive group of pixels constitute frames and consecutive frames constitute segments and consecutive segments compose the whole video stream. Specifying access on pixel/time domain provides a uniform framework to deal with video. Also, authorization requirements like hiding parts of the frame can be easily interpreted at this level, e.g as restricting access to the sequence of pixels that represent that part (restriction can be in the form of setting those pixels to a unique color). But at the same time, at pixel/time domain, it is too difficult to give a clear expression to specify authorization by end users. In our access control model we provide a compromised specification of authorized video objects that can be easily expressed by end users and at the same time provide mechanisms to express authorization on subframe level.

In order to achieve the above goal, objects in our authorization model are introduced by specifying two components. The first component identifies a logical video object, or set of logical video objects (that is, an LV, an LVS, a hot object or sets of them) the user wants to access. We refer to this component as *protected object set.* But video is more informative in its contents, and even though the user may have access to video, he may be restricted to access part(s) of it. So, the second component, in our specification specifies *censored* parts , where the user should be denied access to. Those censored parts represent logical video components that should not be accessed by the user. We refer to those censored parts as *restricted object set.* The set of video objects finally obtained after excluding the censored parts is referred to as the *authorized object set.* Those

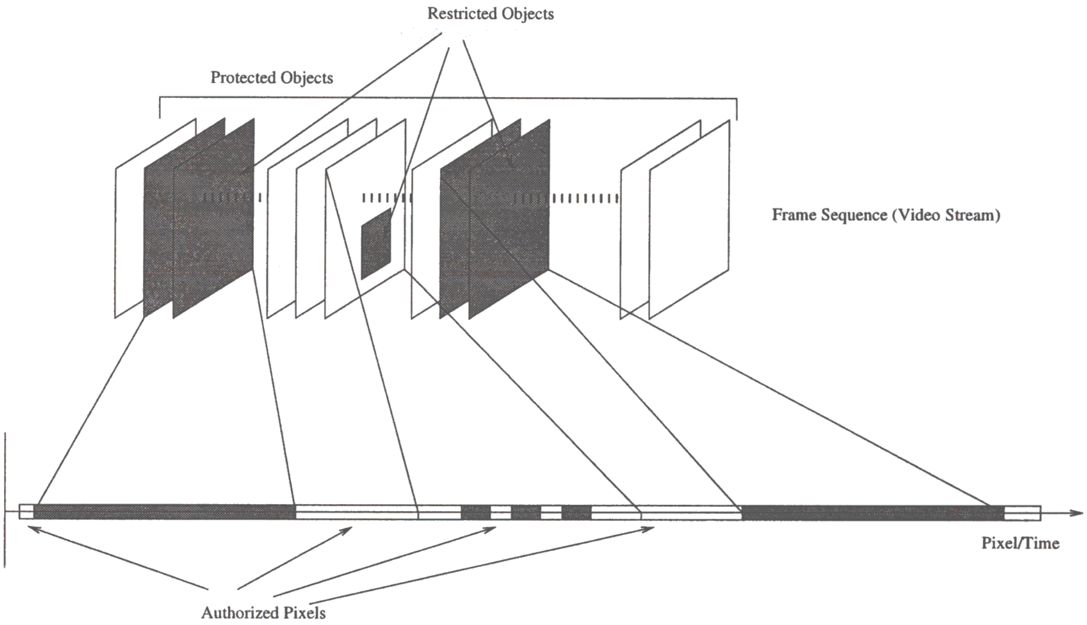


Figure 5: The authorized video objects.

are the actual objects to which the authorization applies. Figure 5 (a) shows the relation between protected, restricted, and authorized video objects with the mapping on Pixel/time domain. A key feature of our model is that both the protected and the restricted objects sets are specified according to the same language. Therefore, a seamless integration between the specification of the two components is achieved in the specification of the authorized objects set. We refer to the specification of a protected or restricted objects set as *objects set specification.* Figure 6 depicts the different relations between the two ways of specifying objects.

More specifically, in our model, protected and restricted video objects are specified either implicitly by giving a concept expression, or explicitly by directly referring to some logical video elements. A formal definition of object specification, protected or restricted, is given in what follows.

Definition 4.6 (Object Set Specification) The set $\mathcal{O}$ of object set specifications is defined according to the following grammar:
$\mathcal{O} ::= CpE \mid LVSet$
$LVSet ::= \{LVList\} \mid \star \mid \emptyset$
$LVList ::= LVList, LVElm \mid LVElm$
$LVElm ::= LvID \mid LvsID \mid HoID$
$LvID ::= (vid, uid)$
$LvsID ::= (vid, uid, [StartFrame, EndFrame])$
$HoID ::= (vid, uid, oid)$
$vid, uid, oid ::= [0 - -9]^{+}$
$StartFrame, EndFrame = [0 - -9]^{+}$

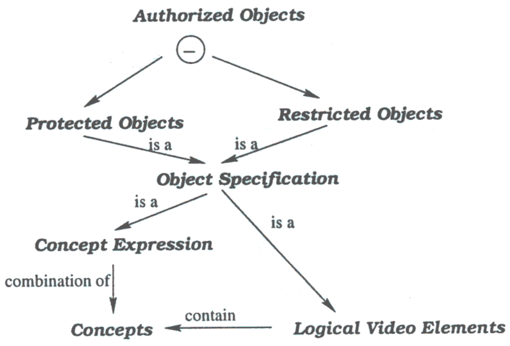


Figure 6: The relations between different object specification terms

where $CpE \in \mathcal{CPE}$. □

Note that the object specifications we support can be quite heterogeneous, ranging from specifications given just in terms of an explicit list of LV to specifications given by imposing conditions on the video contents, according to the concept expression language. However, the actual objects denoted by the specification are of the same type. More specifically:

- If the objects are explicitly specified by a list of logical video elements, such as logical video streams, logical video segments or hot objects, then these logical video elements represent the object set specification.

- If the objects are specified through a concept expression, then the frame intervals that contain concepts satisfying the expression represent the object set specification.

In both cases the result is a set of frame intervals, obtained either as result of evaluating the concept expression or as those frames that contain the explicitly specified logical elements.

The following definition states our notion of authorized object set.

Definition 4.7 (Authorized Object Set Specification) Let po and ro be object set specifications defined according to Definition 4.6. An authorized object set specification,aos is a two-component expression $po.ro$, where the first component is the protected object set component, and the second component is the restricted object set component. □

Note that the restricted object set component, ro, is optional, i.e., a missing ro part indicates an empty set, $\emptyset$, of restricted video objects.

To fully define our approach, we need to state what is the actual semantics of an authorized set specification. The semantics formally states how we determine the actual objects denoted by a given authorized object set specification.

Definition 4.8 (Semantics of Authorized Object Set Specification) Let $aos = po.ro$ be an authorized object set specification defined according to Definition 4.7. The semantics of aos is defined by the following expression:

Table 1: Definition of $\ominus$ operator on logical video elements

$\ominus$	LV_{ro}	LVS_{ro}	HO_{ro}
LV_{po}	The set of frame intervals $\in LV_{po}$ and $\notin LV_{ro}$	The set of frame intervals $\in LV_{po}$ and $\notin LVS_{ro}$	LV_{po} with its frames that have HO_{ro} are displayed with blurred HO_{ro}
LVS_{po}	The set of frame intervals $\in LVS_{po}$ and $\notin LV_{ro}$	The set of frame intervals $\in LVS_{po}$ and $\notin LVS_{ro}$	LVS_{po} with its frames that have HO_{ro} are displayed with blurred HO_{ro}
HO_{po}	The set of frame intervals that includes HO_{po} and $\notin LV_{ro}$	The set of frame intervals that includes HO_{po} and $\notin LV_{ro}$	The set of frame intervals that includes HO_{po} with its frames that has also HO_{ro} are displayed with blurred HO_{ro}

LV_{po}, LVS_{po}, HO_{po}; represent protected video objects. LV_{ro}, LVS_{ro}, HO_{ro}; represent restricted video objects.

$$aos = po \ominus ro$$

where the $\ominus$ operator is defined in Table 1. □

Example 4.8 The following are some examples of authorized object set specifications.

- `Drug addiction interview.PersonID`: this specification denotes all videos reporting interviews about drug addiction without, however, showing the addicted person. `PersonID` in the specification is the HotObjectId of the addicted person and denotes the restricted object in the video.

- `(Elizabeth Taylor ADJ Richard Burton) AND movies`: this specification denotes all videos that are movies and that contain both Elizabeth Taylor and Richard Burton. This specification is equivalent to "`(Elizabeth Taylor ADJ Richard Burton) AND movies.`$\emptyset$".

- $\mathtt{vid_a}.\{(\mathtt{vid_a}, [250, 267]), (\mathtt{vid_a}, [490, 515]), (\mathtt{vid_a}, [560, 600])\}$: this specification denotes a certain documentary video, vid_a, with a restrict viewing of some frames that present classified information.

- `Firearms.(gun ADJ operate)`: this specification denotes all videos showing firearms, by restricting the video portions that show how actually guns are operated.

△

The language we provide for denoting both protected and restricted object sets can be easily extended to support other video element types like audio and visual contents. For example, strong language (audio object) can be restricted in TVG rated movies. In the current video model we adopt only logical video streams, logical video segments and hot objects as for restricting video elements.

4.3 Mode Specification

Mode specification in VD access control is strictly related to operations a subject can perform on the accessed objects. Mode specification strictly depends on the object type. Low level operations, such as physical read and write operations, are not semantically meaningful for access control purpose. Therefore, in our model we introduce a set of abstract operations that are relevant to the way users actually access video data objects.

In a typical video access scenario, a user can interact with a VD server according to what follows. The user first submits a query to search for a given video. The VD server then executes the query, and returns to the user either the annotations associated with the video, or a list of representative frames, RFrames. The user can then request to play the video or submits a more detailed query based on the previous result. Suitable access modes should be devised corresponding to operations performed in such a scenario. Another group of operations for which access control should also be provided include operations for editing the annotations of the logical video elements, for introducing new logical video elements, or for modifying existing ones. In general, authorizations to perform such operations should be given to few, selected users. The highest privilege operation is to add, delete or modify the existing physical video elements since such an operation may greatly impact the overall system.

The different modes of operation, *video privileges*, that are provided as part of our model are described in Table 2.

Table 2: Video privileges, VP, provided by the access control model

Class	Privilege	Meaning
View	`annotation`	To display the query results as the associated annotations only. It speeds up the query response time.
	`RFrames`	To display the query results as a set of representative frames. Displaying only representative frames allows one to return relevant information about the result and at the same time to save the bandwidth by not returning the whole video.
Play	`(period, quality)`	To play the result video query. *period* specifies the permission to play the result for a specified period of time, say for 10 minutes; $\star$ indicates playing the entire video. *Quality* specifies the desired displaying quality of video (*low* | *high*). Play without parameters indicate unlimited period and High quality.
Edit	`annotation`	To edit the annotations associated with the logical video elements.
	`logical-video`	To modify, delete or add logical video elements to a video.
	`physical-video`	To modify, delete or add physical video elements to a video.

The privileges in the table can be ordered, in terms of increasing power, according to user preference. In other words the model permits user to indicate which operation is subsumed by the other, for example viewing the annotation may be considered more serious than viewing the RFrames, if the text will reveal more information than the RFrames do. The $\prec_p$ is used to represent a total order relation between video privilege, for example one possible order of privileges is: View(annotations)

$\prec_p$ View (RFrames) $\prec_p$ Play (period, quality) $\prec_p$ Edit(annotation) $\prec_p$ Edit(logical-video) $\prec_p$ Edit (physical-video).

4.4 Authorization Rule Specification

In the previous sections, we have introduced subject, object and mode specifications that are relevant for a VD access control model. Those are the main components in the specification of authorization rules. A formal definition of authorization rules is given below. In the definition we assume that the following sets are given: $\mathcal{P}$ - it denotes a set of time intervals expressed according to some time unit; $\mathcal{Q}$ - it denotes a set of quality levels.

Definition 4.9 (Authorization Rule) Let s be a subject set specification defined according to Definition 4.4. Let aos be an authorized object set specification defined according to Definition 4.7. Let m be an access mode in the set {`view(annotation)`, `view(RFrames)`, `edit(annotations)`, `edit(logical-video)`, `edit(physical-video)`} $\bigcup$ {`play` $(\mathtt{p_i}, \mathtt{q_i}) \mid \mathtt{p_i} \in \mathcal{P}, \mathtt{q_i} \in \mathcal{Q}$}. An authorization rule is defined as the tuple (s, aos, m). □

According to the above definition, an authorization rule has the following components:

- s: it is a set of authorized subjects. If the subject specification is provided as a set of user-identifiers, then s represents this set. On the other-hand, if subjects are introduced through credential expression, s includes all subjects that satisfy this credential expression.

- aos: it is a set of authorized video objects representing the difference between two sets calculated according to the $\ominus$ operator. The elements of both sets can be specified either implicitly, by concept expression, or explicitly as stated in Definition 4.6.

- m: it is the video operation allowed for the subjects on the specified objects. If m is the `play` mode, the authorization may optionally contain a time duration and a quality level.

4.4.1 Examples of Authorization Rules

In what follows, we present several examples illustrating all the features of our authorization model.

Example 4.9 Let $\{\mathtt{vid_1}, \mathtt{vid_2}, ...\}$ be a set of video streams with violent contents. Then, authorization
`AR1 = (Viewer(X) AND (X.age` $\geq$ `18),` $\{\mathtt{vid_1}, \mathtt{vid_2}, ...\}$`, Play)`
gives the play authorization on a set of video materials containing violent scenes to viewers of legal age, that is, with age greater or equal than 18.

The above authorization can be also be specified as follows
`AR1 = (Viewer(X) AND (X.age` $\geq$ `18), Violence.`$\emptyset$`, Play).` △

Although both authorizations generate equivalent results, they can be used in different situations. Obviously the second form is more clear and compact.

Algorithm 4.1 Access Control Algorithm

INPUT: [1] An access request (uid, vo, p), [2] The authorization rules set $\mathcal{AR}$
OUTPUT: [1] ACCEPT and return vo', [2] REJECT, otherwise
METHOD:

```
relevant_ar_set:=$\emptyset$
For each $ar(s, aos, m) \in \mathcal{AR}$ do
   If ( IsSubject($uid, s$) $\wedge$ IsPrivilege( $p, m$) ) Then
      relevant_ar_set := relevant_ar_set $\cup\{ar\}$
   EndIf
EndFor
If ( relevant_ar_set $\neq \emptyset$ )Then
   $vo' \leftarrow vo$
   For each $ar \in relevant\_ar\_set$ do
      $vo' \leftarrow vo' \ominus \overline{aos}$
   EndFor
   If ($vo' \neq \emptyset$ )Then
      return(ACCEPT, $vo'$)
   Else
      return(REJECT)
   EndIf
Else
   return(REJECT)
EndIf
```

Figure 7: Access Control Algorithm

Example 4.10 Let *vid* be the identifier of the interview video stream, and HotObjectID be the hot object representing the face of the interviewed person. Then, authorization
`AR2 = ( Viewer(X) AND (X.Class=General Audience ), vid.HotObjectID, Play)`
gives the play authorization on the TV-interview, while hiding the face of the interviewed person, to all users whose class is general audience. △

Example 4.11 Authorization
`AR3 =( Student(X) AND (X.major = History), (World War II) AND (documentary movies), Play )`
gives the play authorization for browsing the World War II video library to all college students whose major is history. △

Example 4.12 Authorization
`AR4 =(Employee(X) AND (X.position = Director), Charles De Gaulle DURING World War II, Edit(logical-video))`

gives the authorizations to directors for composing a new documentary program in the World War II video library. △

Example 4.13 Authorization
`AR5 =(Customer(X) AND (X.subscribe = NO), Soccer World Cup, Play(5))`
grants access to all subscribers of a sport channel to watch first 5 minutes of a special event (World Cup for soccer), as a way to advertise watching the event. △

5 Access Control

The main components of an authorization system are authorization rules and control procedures that verify those rules against user transactions. In the previous section we have defined the authorization rule language and the specification of each of its components, namely; subject, object and privilege. Access control procedures are presented in this section.

The main role of the access control mechanism is to verify that user *uid*, trying to access video object *vo*, using a privilege *p*, is authorized to do so. The authorization rule repository must then be searched to verify whether appropriate authorization rule(s) exist. This access control scenario is general and common to all database subjects. However, our video access control has the following distinct features. First, subjects can be introduced not only with their id's but also by using credentials and even credential expressions. Second, objects in our model can be specified by either explicit identifiers or through concepts in a concept expression. If the object in the authorization rule is specified as concept expression, then failure of the concepts in the *checked object* to satisfy concept expression can't guarantee that the object is restricted. For example the object could still represent part of the authorized frames interval but unfourtunately is annotated with different concepts. This point is important in deciding the way to check object authorization. Though introducing authorized objects by concepts is easy and more natural, checking of objects authorization should be done at the frame level not the concept level. Obviously, this is due to the possibility of building different logical views on the same frames interval. Moreover, not all video objects are atomic in our model, but different video granularities can be used. For example, providing access to a whole video may include restricted access to some frames or even subframes. Finally, an access request to video objects is not simply accepted or rejected, but may also be accepted after applying some filter effects based on the restriction part, such as clipping restricted frames, or obscuring or blurring restricted subframes.

The implications of the above features on the access control mechanisms are obvious in the design of a control algorithm as listed below:

- The subject verification function should consider both existence of user identifier or satisfaction of user credential to the provided credential expression.

- The theoretical way to check for object authorization is to map the object to its pixel/time domain and check against authorized interval. Though this is the obvious way to check for object authorization, one can use different heuristics to bypass the tedious/unrealistic mapping process. For example if object's concepts satisfies concept expression, then this object

Algorithm 5.1 Detailed description of IsSubject and IsPrivilege

```
Function IsSubject($uid$, $s$)
If($s$ is provided as $\mathcal{CE}$ ) Then
    $\mathcal{C} \leftarrow$ all user credentials of user $uid$
    For each $c \in \mathcal{C}$ do
        If ( $c$ satisfies $\mathcal{CE}$ ) Then
            return( TRUE )
        EndIf
    EndFor
    return( FALSE )
Else
    If ( $uid \in$ s ) Then
        return( TRUE )
    Else
        return( FALSE )
    EndIf
EndIf
EndFunction

Function IsPrivilege($p$, $m$)
If ($p \prec_p m$ ) Then
    return( TRUE )
Else
    return( FALSE )
EndIf
EndFunction
```

Figure 8: Supplementary Access Control Functions (a)

is surely authorized, or mapping the objects to its frame/time level and apply any necessary interval intersection operation. working with subframe restriction can also be performed at this level, without the necessity to go deep into the pixel level.

- It's clear from the previous point that, the access control function should not only provide accept or reject answers but also process, and apply effect filters on the requested video, if necessary.

The access control algorithm is specified in Figure 7. The algorithm works as follows: it checks each authorization rule in $\mathcal{AR}$ set that has *uid* as one of its subjects and such that p is less than or equal in power to its mode. Functions *IsSubject and IsPrivilege* perform those checks, the detailed description of both of them is listed in Figure 8. This step is important in evaluation of access control since many request can be denied at this point without the overhead of object checking or query evaluation. In our model we allow the subject to have many authorization rules and

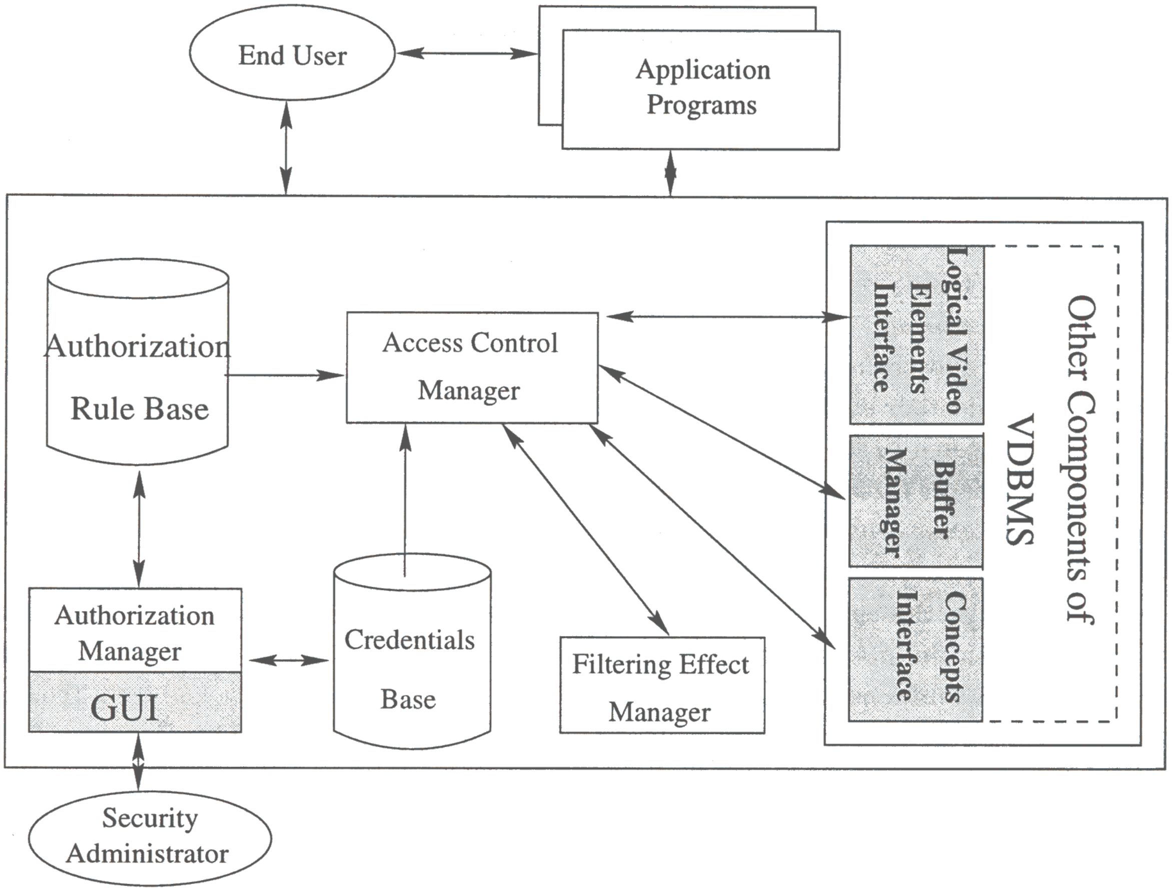


Figure 9: System architecture for a secure video database

we always follow the most conservative one (or combination). Hence, several authorization rules can satisfy those conditions and all of them are collected in *relevant_ar_set*. If after searching the whole $\mathcal{AR}$ set, *relevant_ar_set* is empty then the user request is rejected and the check is complete. Otherwise, for each authorization rule in *relevant_ar_set* the requested video objet is restricted by the authorized objects set. This can be mathematically formed as $vo' \ominus \overline{aos}$, where the $\ominus$ operator is as defined in Table 1 and $\overline{aos}$ represents the restricted domain (complement of the authorized one, aos). After processing all the relevant authorization set, vo' will contain the authorized video objects. If vo' is empty, this means user request is rejected, otherwise it is accepted and return vo'.

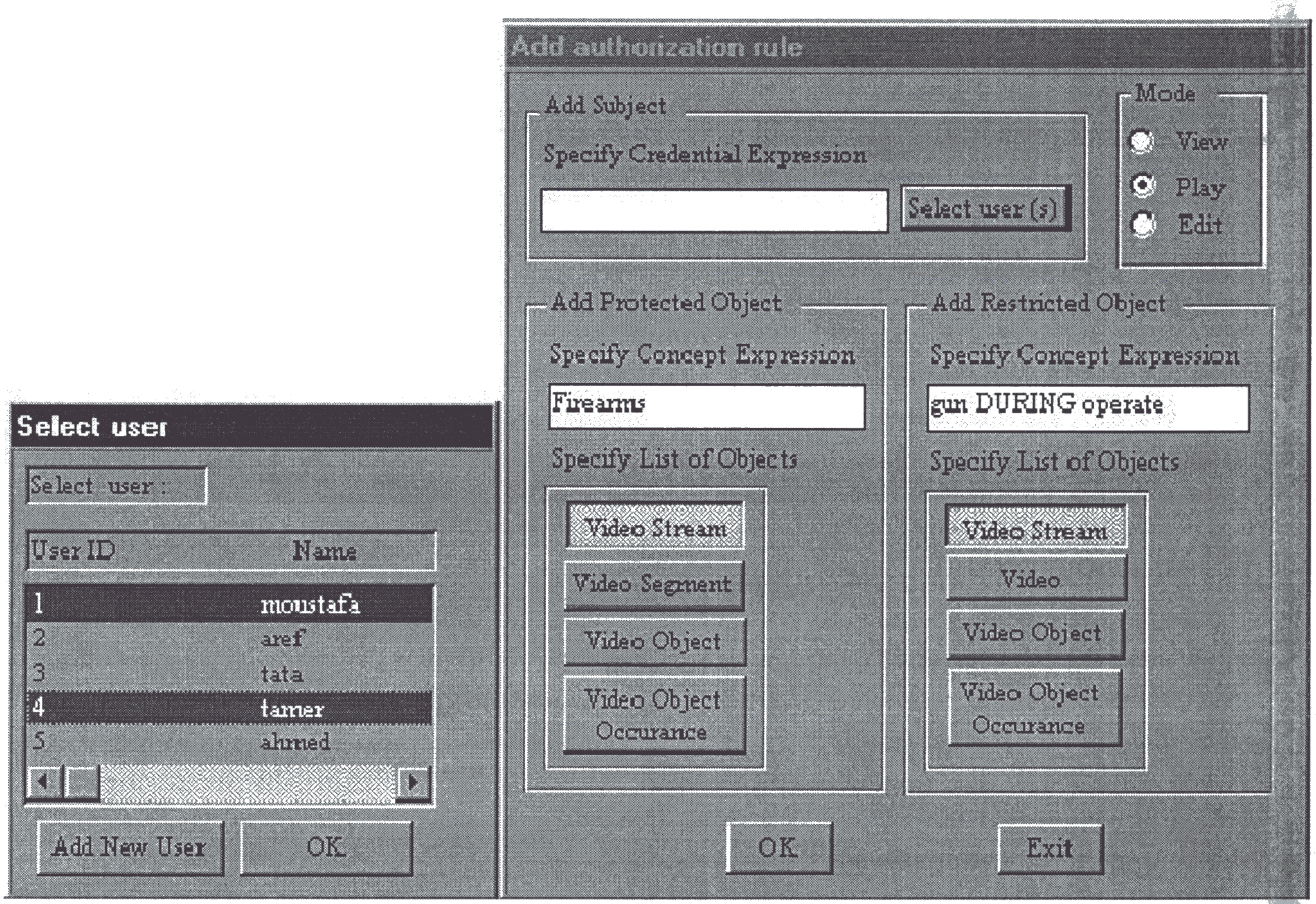
6 System Architecture

In this section we present the system architecture incorporating both access control components and other video database management system components. Our architecture has been developed with the following goals. First, the architecture has to be modular in terms of the main components of the access control system. Modularity is crucial in adapting the system to both distributed

and central environments. Second, access control components should be distinguished from other video database components and interfaces to video components should be clearly identified. The main purpose of this feature is to make our access control as portable as possible so that it can be integrated with other video databases. The system architecture is depicted in Figure 9.
In our access control system, we distinguish between two classes of users: *security administrators* and *end users*. Security administrators have the highest privileges and can access all video objects. End users can access the secure video database either directly through the user interface tools provided by the VDBMS or through application programs that interact with the video database. The *authorization manager* component is responsible for the full management of both the *authorization rule base* and *credential base*. Through the authorization manager, one can add, modify, or delete user credentials stored in the credential base. Also, one can add, modify, or delete authorization rules stored in authorization rules base. The security administrators interacts with the authorization manager through a graphical user interface *GUI* interface that facilitates the mutual interaction. *The access control manager* implements the access control algorithms specified in Section 5. Also, it interacts with the video database through the concept interface, to evaluate concept expressions, and also through the logical video elements interface. The access control manager also communicates with the *filtering effects manager* to perform any necessary exclusion of frames or blurring of certain subframes. The *buffer manager* holds temporary results to and from the access control manager from one side and to and from the VDB from the other side. For example, the buffer provides the access control manager with the protected objects and receives the final authorized objects after applying any filtering effects to be presented to the user by other database components.

7 Implementation

We have implemented our access control model and tested most the functionalities specified in this paper. We use video data sources in MPEG fromat and realized the video model using relational database engine. We used a host language to implement our access control techniques and the filtering effect manager. In our system, the security administrator has the exclusive rights to introduce new authorization rules to the system. Figure 10 shows the GUI interface for adding new authorization rule. The administrator should specify the mode of operation, the subject (either as a list of users or as a credential expression) and both the protected and restricted objects sets (also either as a list of logical video objects or as a concept expression). When a new authorization rule is added to the system, the system prevaluates the authorized objects for this authorization rule. The evaluation of authorized objects follows the definition 4.8. So, after evaluation of authorized objects, we get a *new* set of frame intervals. The term new is used because some of the frames may have parts of them blurred (restricted subframe). Evaluation of authorized objects is important in order to map them to the frame level as discussed in Section 5. Also, the evaluation at this time saves the overhead time of reevelauting the authorized objects each time user request is checked. The system also allows the administrator to introduce new users and define their profiles (user name, password..etc). We have implemented a simple parser to process our concept expression, introduced



Add authorization rule

Add Subject

Specify Credential Expression:

Mode: ○ View ● Play ○ Edit

Add Protected Object

Specify Concept Expression: Firearms

Add Restricted Object

Specify Concept Expression: gun DURING operate

Select user

User ID	Name
1	moustafa
2	aref
3	tata
4	tamer
5	ahmed

Figure 10: Add new authorization rule

in definition 4.6. The parsing output is a set of SQL commands that access the underlying relational model used to represent the video model. The SQL commands are then executed to retreive the target video objects. Although, it would be faster to directly evaluate the query, without using SQL, but using SQL provides a uniform and expressive way to express our concept expression. Also, all video operations (Boolean, temporal, spaial or spatio-temporal) introduced in Section 3 and Definition 4.5 are easily expressed in SQL. For example, temporal operations can be specified as Boolean checks against *start* and *end* frames of segement intreval. Also, spatial operation can be expressed using Boolean checks against the points representing the geometric description of the hot objects.

As checking authorization time is considered overhead in terms of query response time. A good design and implementation should try to minimize this time as much as possible. For the case of querying the video database, the set of relevant authorization rules for this subject is obtained. If this set is empty the user query is not executed and the user is notified. Furthermore, only those video streams and segemnts that (temporally) intersect with the authorized video objects need to be checked during query evaluation. This check has the effect of excluding totally unauthorized obejcts (their frame interval is totally disjoint with the authorization intervals). It should be clear that all checks are performed at the frame level, not the concept level. Since concept level is too week to base authorization on as discussed in Section 5. If the user is authorized to play the resulting video segment(s), the system materializes the query result in one or more video streams, to be played latter by the user.

It is obvious that, the process of query evaluation and presentation in video database is completely different from any other types of data. The result here is a real time video that is preprocessed according to the authorization requirements before it's handed to the user. The process of displaying the query results to the user involves concatenating together pieces of video segment into a new video stream to be played by the user. Also, if the segments are from different video streams, different video streams are composed and displayed as a result. Other approches involoves providing representative frames before actually composing the video and can help in reducing query retrieval time, but involves other overhead in terms of extracting the representative frame or storing and retreiving them. Video browsing operation, is much like displaying the result of video query with the additional checking of the type of browsing used.

In our system, we preferred to work on MPEG files and to make all video operation on line. It 's natural that the target of querying or browsing video is to play the result. Although this may introduce overhead in retrieving video results for the first time, but our technique provides greater flexibility and the wasted time is spent in the first retreival of the video, after that any access to the video is applied to the materialized result. Also, MPEG files provides a flexible way to manipulate video in compressed format and as a result eliminates the need to decode and then encode the video to perform any processing.

Figure 11 shows the user query and the resulting video stream after applying any filtering effects.

Our system also allows the introduction of new logical video streams, logical video segments and addition of annotations to them. By this user administrator can specify the authorized intervals and associate the necessary annotation to them, that may be used latter in specifying concept expression. As an extension to our implemenation, we consider building editting tool on video that enables the system adminstrator to assign authorization while adding annottaion or browsing parts of the video.

8 Conclusion and Future Work

In this paper we presented an access control model for video databases. The main components of authorization rules have been tailored to the video domain. Our model allows one to specify authorization subjects not only by their identifiers, but also according to their credentials. Credentials have been used in access control mechanism for textual data. Their use in access control for video data is a new application. The use of credentials for specifying subjects makes specification of authorization policies clear and easy to understand for users. Also, credential specifications can exploit the various media types in video data and can include visual and audio characteristic of the users in addition to their normal descriptive features. Our model also provides a clear definition of authorized objects that can be either specified implicitly by using concept expressions, or explicitly by providing a set of logical video elements. We thus provide access control based on the semantics of video data and not just on its physical representation. By providing access control based on semantic contents, we thus exploit the richness of information in video data and enable semantically high-level specifications of authorizations. As far as we know, content-based access control for video has not been previously addressed. The main focus of previous work was on access control based on physical representation of video, such as dropping frames. In our model, we relate access control to semantic contents of video data, in addition to also support of access control based on the physical video. Therefore, our approach subsumes previous approaches based on the physical representation only. A distinct contribution of our work is also to provide access control for different video granularities ranging from whole logical video stream to subframe regions or to hot objects. In order to implement these techniques, filtering effects are incorporated into the access control mechanism. Filtering effects are used to hide a sequence of frames, or to blur subframe regions in these sequences. Filtering effects can be extended to deal with audio and text as well. Our access control model is not acting like a dump guard against user requests but more like a smart manager that tries to fulfill user requests and in the same time satisfy the authorization requirements. The access model also provides a categorization of privileges that are meaningful for video data. The privileges we have devised are abstract and suitable to interact with video. They range from the privilege of just viewing the annotations associated with video to full control on physical video components. Users can be authorized to play the video for a limited time or even with a limited precision. The last type of privilege is important for video data used in advertising applications and in supporting different quality of service for different users. Because our authorization model is based on video contents, it can be easily applied to different video data models. The access control mechanism that we have developed based on such authorization model has a modular architecture with with well-defined interfaces to the various components of a video database system. The architecture modularity enables both centralized or distributed design and allows the system to be interfaced with any video database system providing content-based access to video. This is a relevant feature of our work since several efforts is currently on-going in the area of video content description. A new standard is about to be released, MPEG-7 [11]. Our model can exploit the semantic provisions of MPEG-7 and with some modifications can be integrated with it to introduce semantic video access control. MPEG-7 will use the physical presentation of video provided by MPEG-4 and will introduce a content description language for video. These two features, e.g. physical presentation and content description provision, are the main interfaces of our access control to video database. We implemented a prototype system of our access control model based on a LHVDM [15].

Several issues arise in the implementation of our access control model. First, video processing has real-time constraints. Our access control introduces an overhead that should be maintained within a certain range in order not to affect the quality of service. Also the distributed implementation of access control mechanism is a relevant research issue especially when users access video libraries through the web. Several issues are also related to secure communications between clients and server and request authentication and protection.

Future extensions of our model include exploiting web blocking and censorship technology like PICS. PICS stands for *Platform for Internet Content Selection.* It is a general purpose system for labeling the contents of documents appearing on the World Wide Web [10]. PICS labels contain one or more ratings that are issued by a rating service. For example, some movies accessed through the net can be rated as violent and hence children under age are not allowed to view them if the browser is customized to block those movies. Access control mechanisms for video databases through the web should benefit from this technology and our access model should be able to exploit such labeling service.

References

[1] Frequently asked questions about closed captioning. *http://www.robson.org/capfaq/overview.html.*

[2] N. Adam, V. Atluri, E. Bertino, and E. Ferrari. A content-based authorization model for digital libraries.

[3] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. of ACM, 26(11):832-843, November*, 1983.

[4] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An access control model supporting periodicity constraints and temporal reasoning. *ACM Trans. on Database Systems, 23(3):231-285*, 1998.

[5] E. Bertino, P. Samarati, and S. Jajodia. An extended authorization model. *IEEE Trans. on Knowledge and Data Engineering, 9(1):85-101*, 1997.

[6] S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security.* Addison-Wesley, 1995.

[7] A. K. Elmagarmid and H. Jiang. Multimedia video (chapter). *Wiley Encyclopedia of Electrical and Electronics Engineering. Volume 13*, 1998.

[8] A. K. Elmagarmid, H. Jiang, A. Helal, A. Joshi, and M. Ahmed. *Video Database Systems. Issues, Products and Applications.* Kluwer Academic Publishers, 1997.

[9] E. Fernandez, E. Gudes, and H. Song. A model for evaluation and administration of security in object-oriented databases. *IEEE Transactions on Knowledge and Data Engineering, 6(2): 275-292*, 1994.

[10] Simson Garfinkel and Gene Spafford. *Web Saecurity and Commerce.* O'Reilly and Associates, 1997.

[11] Requirement Group. Mpeg-7 context and objectives. *International Organization of Standardization, ISO/IEC JTC1/SC29/WG11. Coding of Moving Pictures and Audio*, 1999.

[12] E. Gudes, E. Fernandez, and H. Song. Evaluation of negative, predicate and instance-based authorizations in object-oriented databases. *In Database Security, IV: Status and Prospects, Elsevier publ*, 1991.

[13] R. Holowczak. Extractors for digital library objects. *PhD Thesis Rutgers University, Department of MS/CIS*, 1997.

[14] H. Jiang. Semantic content-based access to hypervideo databases. *A Ph.D. thesis submitted to the faculty of Purdue university*, 1998.

[15] H. Jiang and A. K. Elmagarmid. Spatial and temporal content-based queries in hypervideo databases. *The VLDB Journal 7 (1998) 4, 226-238*, 1998.

[16] H. Jiang, A. Helal, A. K. Elmagarmid, and A. Joshi. Scene change detection techniques for video database systems. *ACM Multimedia Systems, 6(3):186-195, May*, 1998.

[17] F. Kokkoras, H. Jiang, I. Vlahavas, A. K. Elmagarmid, E. N. Houstis, and W. G. Aref. Smart videotext: A video data model based on conceptual graphs. *Accepted for publishing in ACM Multimedia Systems Journal*, 1999.

[18] P. S. Kumar and G. P. Babu. Intelligent multimedia data: data + indices + inference. *ACM Multimedia Systems, 6:395-407*, 1998.

[19] E. Riloff and W. Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems, 12(3):296-333*, 1994.

[20] A. Rosenthal, J. Williams, W. R. Herndon, and B. M. Thuraisingham. A fine-grained access control model for object-oriented dbmss. *J. Biskup, M. Morgenstern, C. E. Landwehr (Eds.): Database Security, VIII: Status and Prospects, Proceedings of the IFIP WG11.3 Working Conference on Database Security, Bad Salzdetfurth, Germany, 23-26 August 1994. Elsevier.*

[21] P. Samarati, E. Bertino, and S. Jajodia. An authorization model for a distributed hypertext system. *IEEE Trans. on Knowledge and Data Engineering, 8(4):555-562*, 1996.

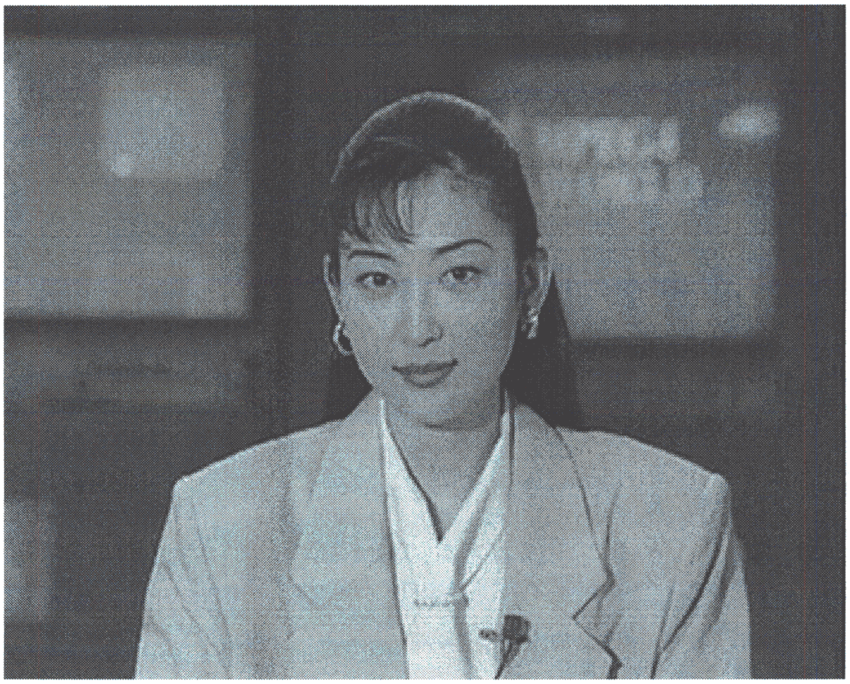
[22] R. sandhu et Al. Role-based access control models. *IEEE Computer, pages 38-47, February*, 1996.

[23] V. S. Subrahmanian. *Principles of Multimedia Database Systems.* Morgan Kaufmann, 1997.

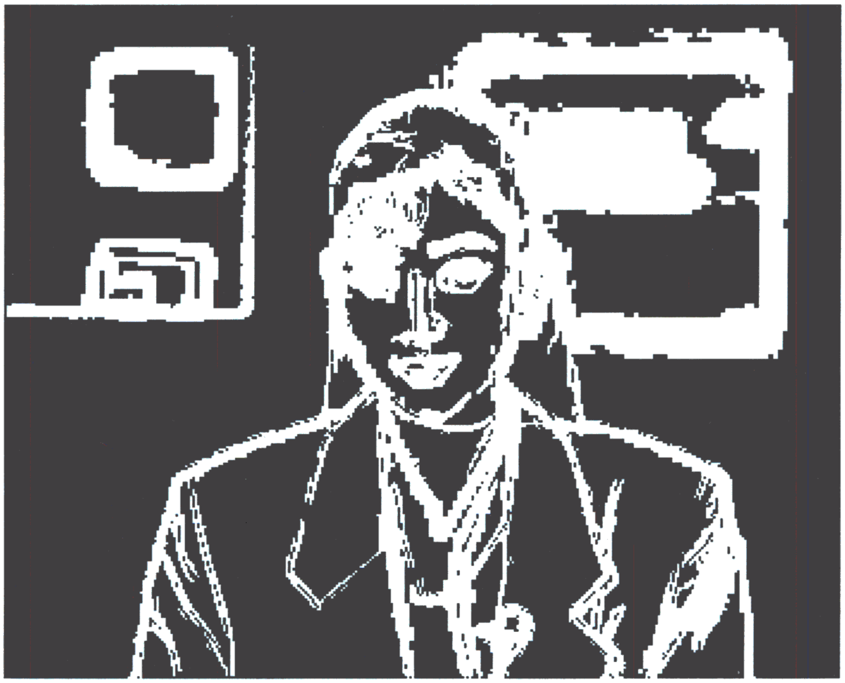
[24] M. Winslett, N. Ching, V. Jones, and I. Slepchin. Using digital credentials on the world-wide web. *Journal of Computer Security, 5*, 1997.

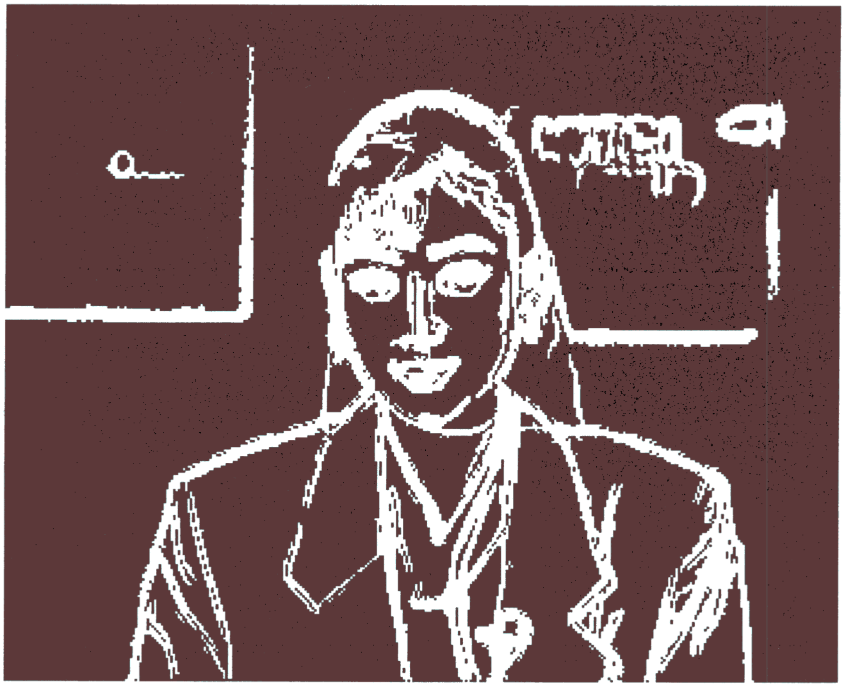
[25] K. Yamaashi, Y. Kawamata, M. Tani, and H. Matsumoto. User-centered video: Transmitting video images based on the user's interest. *CHI'95 Proceedings:325-330 DBLP*, 1995.

③ Object Extraction and Face Detection

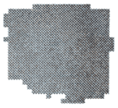


(a)





(c)

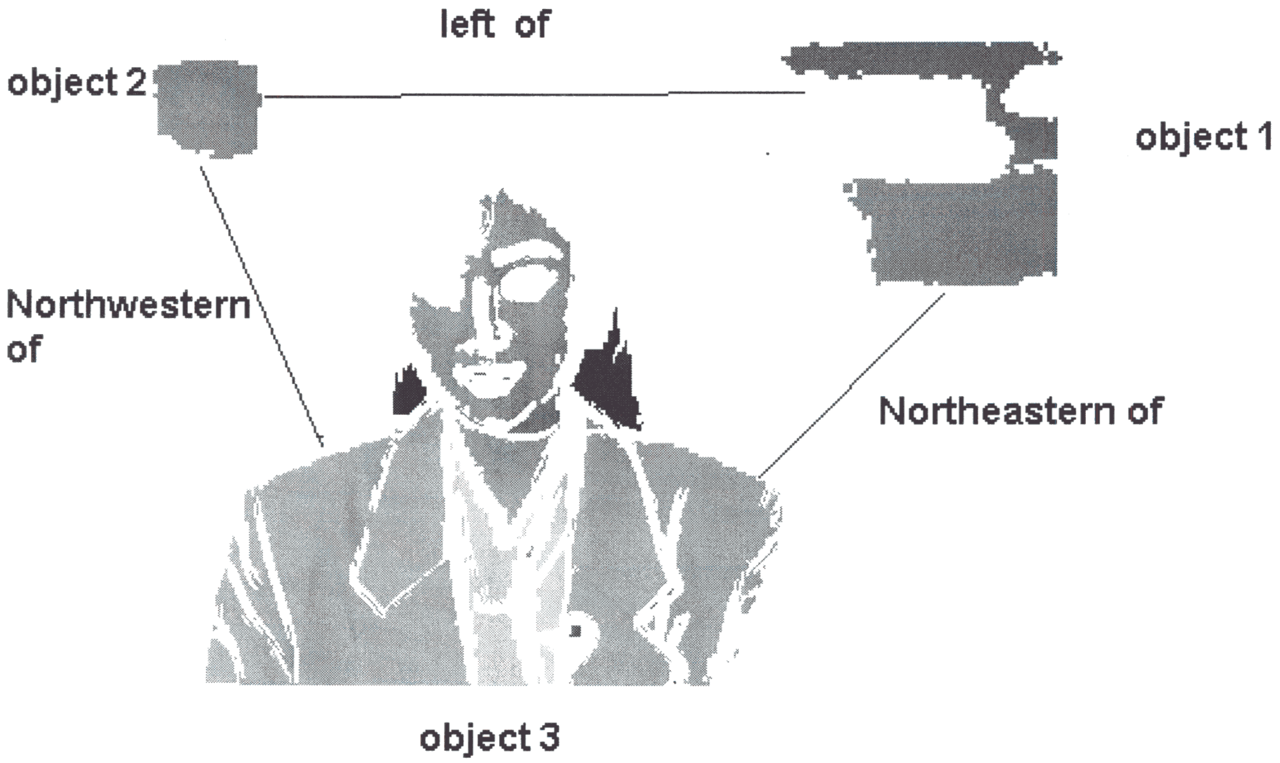


(e)

(f)

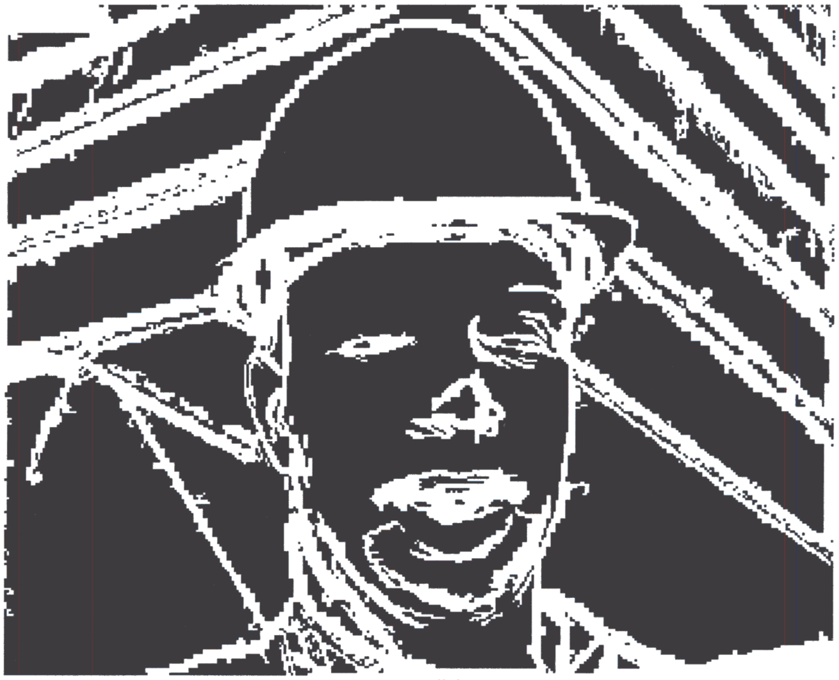
(g)

(h)

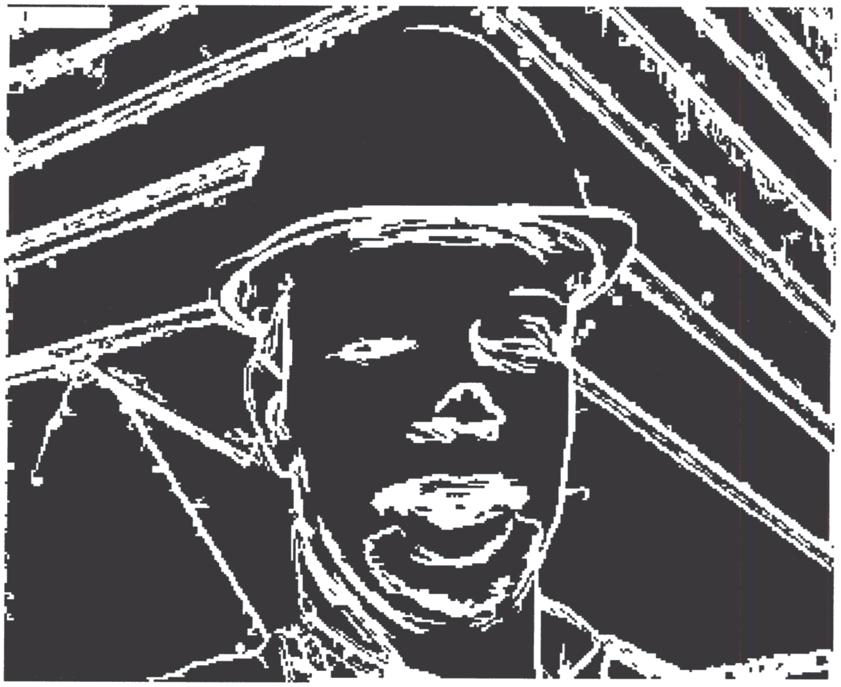

left of
object 2
object 1
Northwestern of
Northeastern of
object 3

(i)

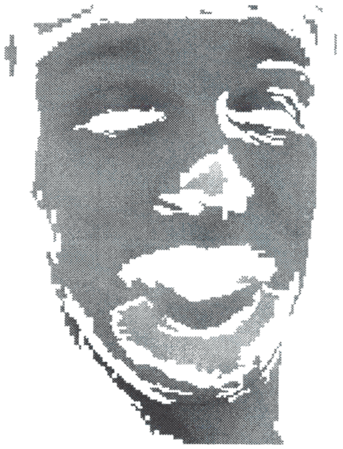
(a)



(b)

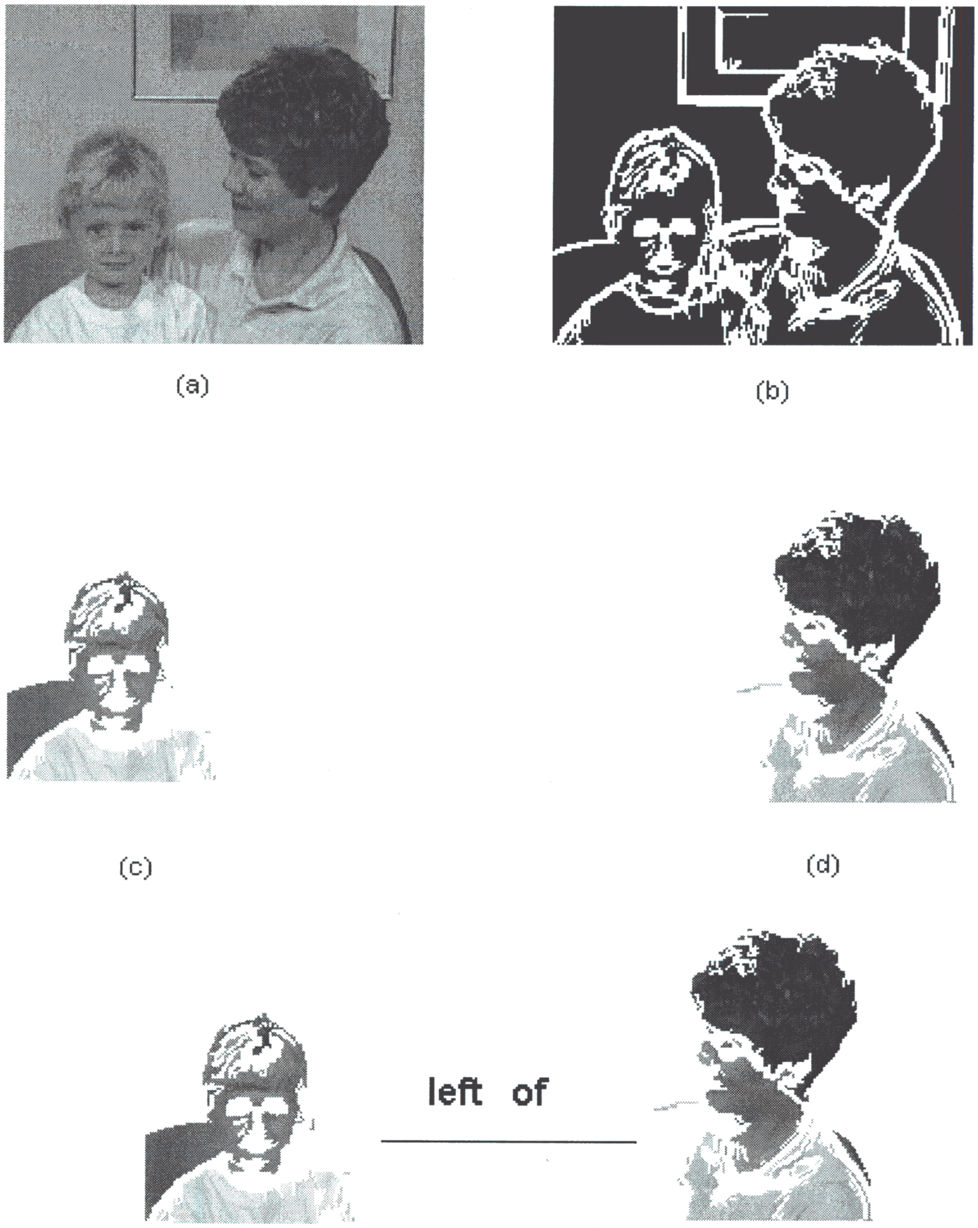


(c)



(d)

(e)



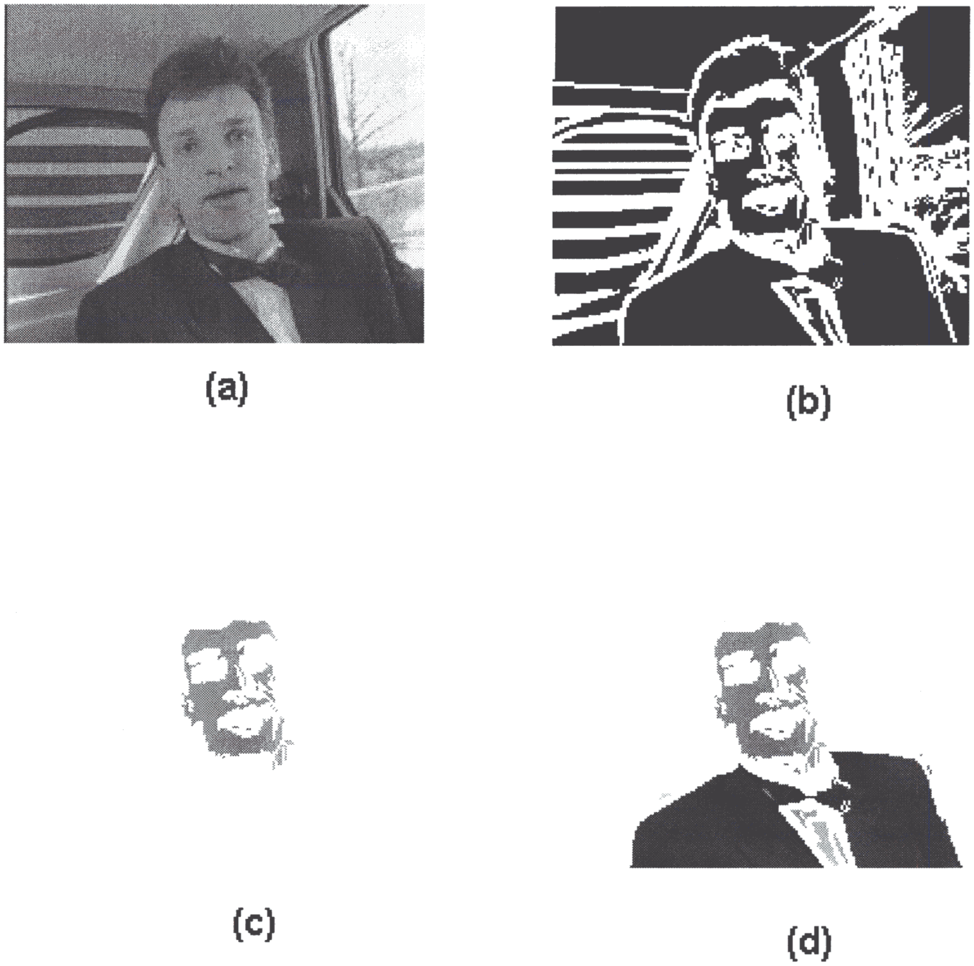
(a)

(b)

(c)

(d)

(e)

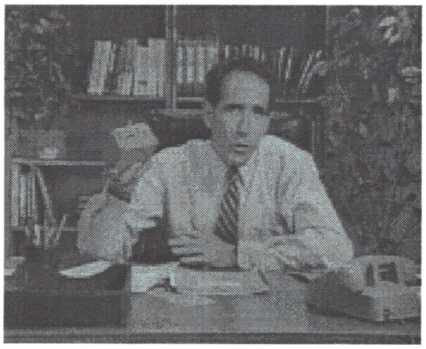


(a)

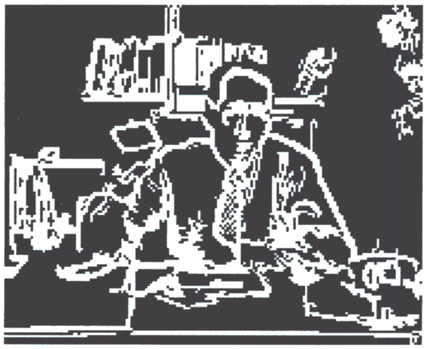
(b)

(c)

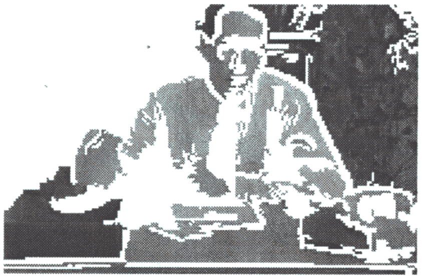
(d)



(a)



(b)



(c)