P

Part 7 of the MPEG-21 Multimedia Framework

► MPEG-21 Digital Item Adaptation

Partial Video Bit Streams

► MPEG-21 Digital Item Adaptation

Peer-to-Peer Multicast Video

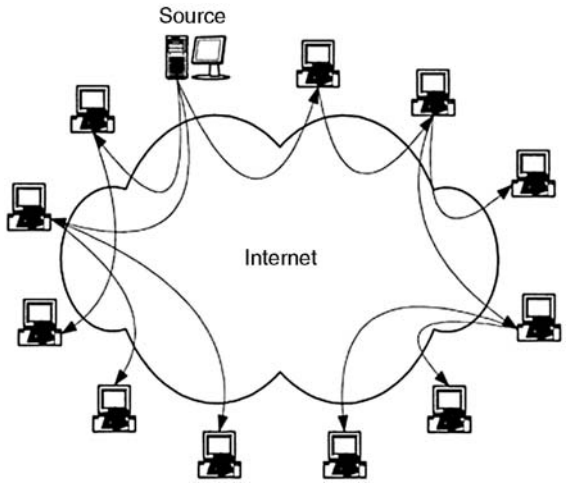
Definition

Peer-to-peer multicast video refers to the concept of peer-to-peer communication, where nodes are both clients and servers.

In the tradition video distribution scheme, a client contacts a server and establishes a unicast session before starting to receive the required content. Although this configuration is enough in many simple scenarios, it presents a number of problems. Indeed, the sources maximum output bandwidth limits the number of parallel clients. Moreover, since this approach follows a centralized configuration, it is vulnerable to attacks. A number of alternative solutions have appeared to overcome these limitations. They generally propose the use of replicated servers to increase both the robustness and the capacity of the system to serve more clients. A more recent approach relies upon the concept of peer-to-peer (P2P) communication, where nodes are both clients and servers. Virtual links associating two IP addresses are established forming an *overlay* network.

In addition to being more resistant to attacks, the inherent distributed nature of P2P is a solution for the problem of multiple clients bottlenecking the source. Peer-to-peer overlays have been rapidly adopted as a promising substrate for video distribution, for both video sharing (in the same way people share general data files) and streaming. In the latter case, a distribution tree is established between the source and the receivers [1]. The particularity here is that intermediate nodes are also end-systems (and in general also a client). In this way, the source sends the video to a number of clients that, in turn, send to other clients, and so on until all clients receive the video. We refer to this approach as *application-layer multicast* (since it forms an overlay distribution tree). Figure 1 shows an example of a video multicast tree using P2P concepts.

Although very attractive, video transmission over P2P networks presents a number of technical challenges. First, the distribution tree cannot grow indefinitely, since nodes closer to the leaves may experience high delays. While in a video on demand service this may not be a real issue, for other applications, such as interactive video communication, delays beyond a certain threshold are unacceptable. Another challenge is inherently related to the dynamic nature of peer-to-peer communications: nodes in an overlay may join/leave whenever they want, without any notice. A third problem concerns the complexity of establishing the distribution tree. Indeed, nodes are extremely heterogeneous in terms of receiving, transmitting, and storage capacities. For a small network, managing such a structure is feasible, but for larger dynamic topologies (even millions of receivers) it is likely that the overlay always operates in a sub-optimum configuration. Different solutions have been (and are being) proposed to overcome such limitations, which include, for instance, advanced algorithms for correctly placing nodes on the overlay. Recent results show that P2P video distribution is a promising solution to contour the limitations of the current Internet architecture [2].



Peer-to-Peer Multicast Video. **Figure 1.** Example of an overlay distribution tree.

Cross-References

► Video Over IP

References

1. Y.-H. Chu, S.G. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," IEEE Journal on Selected Areas in Communications, Vol. 20, No. 8, October 2002, pp. 1456–1471.
2. D.A. Tran, K.A. Hua, and T.T. Do, "A Peer-to-Peer Architecture for Media Streaming," IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, January 2004, pp. 121–133.

Peer-to-Peer Streaming

Roger Zimmermann, Leslie S. Liu
University of Southern California, Los Angeles, CA, USA

Synonyms

► Streaming P2P architectures

Definition

Peer-to-Peer (P2P) architecture for multimedia streaming is emerging in recent years which can eliminate the need for costly dedicated video servers in the traditional client-sever approach.

Introduction

The basic concept of peer-to-peer (P2P) computing is not new and some techniques date back many years when the Internet was first designed. However, the key phrase "peer-to-peer" has become widely and publicly recognized mostly after the pioneering Napster (http://www.napster.com) file sharing network emerged in the late 1990s. Peer-to-peer is a very general term and people associate different concepts with it. Various forms of P2P techniques have been used in the fields of computing, networking, distributed file systems, and others. In this chapter we focus on how P2P techniques are being used for streaming media distribution.

P2P systems have some key characteristics that distinguish them from the traditional and widely used client-server model. The most prominent feature is that a P2P system is composed of a number of member *nodes*, each of which combines the functionality that is traditionally associated with *both* the server and the client. As such, multiple P2P nodes can form a collective that aggregates their resources and functionality into a distributed system. Node *A* may act as a client to node *B*, while at the same time function as a server to Node *C*. Beyond this fundamental characteristic, there are a number of features that are often associated with P2P systems. Note, however, that usually only a subset of the following characteristics holds true for any practical system.

- *Reduced central control.* Many P2P systems work in a fully decentralized fashion where all the nodes have equal functionality. The members are connected based on a system-specific construction policy and form a distributed topology. Exceptions to this model exist. For example, the original Napster file sharing network used a centralized index to locate files; subsequently the data was exchanged directly between individual peers.
- *Heterogeneity.* Members of a P2P system are usually heterogeneous in terms of their computing and storage capacity, network bandwidth, etc. A system may include high performance nodes on a university network and computers owned by residential users with broadband or modem connections.
- *Flat topology.* Members of the P2P network are often treated equally which results in a flat connection topology. However, hierarchical systems exist that introduce the concept of "super-peers."
- *Autonomy.* The time and resources that a member node can or will contribute to the system are dynamic and unpredictable. Often, nodes are under different administrative control. Hence the enforcement of global policies is a challenge.
- *Fault resilience.* P2P members may join or leave the topology at any time. Therefore, not only

is the formed community very dynamic, but no assumptions should be made about the availability of resources or network paths. A P2P system must be able to recover from the unexpected and ungraceful leave of any of its members at any time.

Members of a P2P system are also referred to as nodes because they are often represented as network nodes in topology graph.

Streaming P2P Architectures

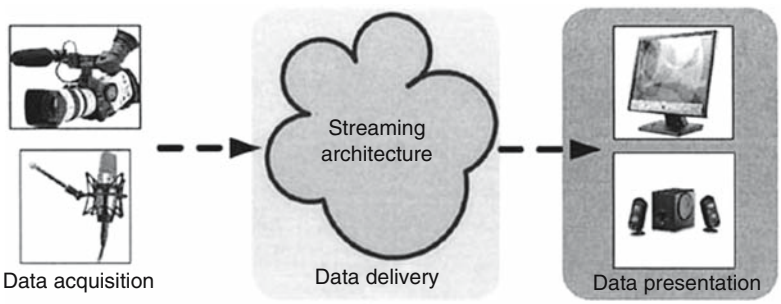
Streaming is a process of generating and delivering a steady, isochronous flow of data packets over networking medium, e.g., the Internet, from a source to a destination. The rendering of the content starts as soon as a small fraction of the data stream has been received. Streaming media usually denotes digital audio and video data, however haptic or other data may be streamed as well. One of the main resource bottlenecks that afflicts large client-server distribution architectures is the massive bandwidth that must be available from the server into the core of the network. This network connection is often very costly (compared to the server and client hardware) and may render a technically feasible solution economically not viable. Peer-to-Peer streaming is an alternative that alleviates the bandwidth cost problem by offering a service to deliver continuous media streams directly between peer nodes. However, the previously listed characteristics of P2P systems influence the design of such decentralized streaming solutions.

Theoretically, P2P architecture can be built over any networking medium and at potentially different layers of the network. However, most of the existing P2P implementations and their associated research have focused on application-level overlay networks. The Internet, as the dominant networking medium for research, business and entertainment, is also the preferred choice for P2P network substrates.

One of the virtues of today's P2P systems is their scalable nature. Peer-to-peer technologies were first widely used and accepted as file-sharing platforms in systems such as Napster, Gnutella and KaZaA. Subsequently, the P2P architecture evolved and was adapted for store-and-forward streaming. Examples of streaming systems that may be used to distribute previously stored content are Narada, HMTP, and Pastry. One distinguishing characteristic among these proposals is the shape of the streaming topology they construct, which will be described later in this chapter. Even though these designs promise good performance in terms of network link stress and control overhead, only a few of them have been implemented in real systems. Next, P2P technology was adapted for live streaming. In this scenario, media streams are generated by live sources (e.g., cameras and microphones) and the data is forwarded to other nodes in real-time. We distinguish two types of live streaming: one-way and two-way. The requirements for the two are quite different and more details follow below.

Streaming Process

A streaming process can be separated into three stages that overlap in time (Fig. 1): data acquisition, data delivery, and data presentation. Data acquisition is the stage that determines how the streaming content is acquired, packetized, and distributed for streaming. The data presentation stage represents the methods on how to buffer, assemble, and render the received data. Data delivery is the process of how the stream data is transported from the source to the destination. The source, the destination and all the intermediate nodes in a streaming system participate in a topology that is constructed based on the specific system's protocol.

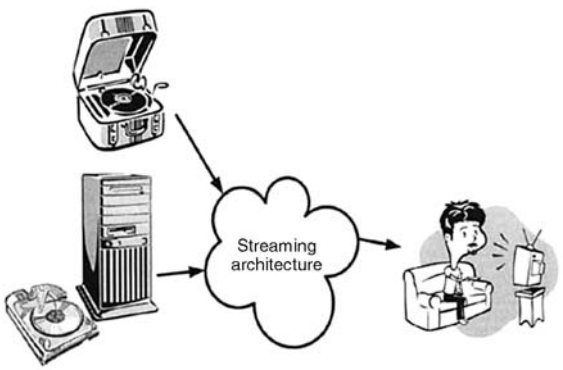


Peer-to-Peer Streaming. Figure 1. Streaming process.

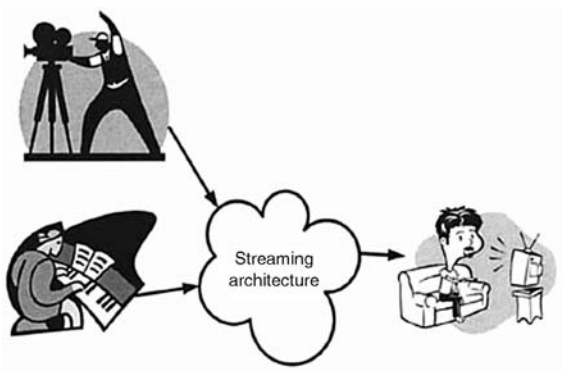
In a P2P streaming system, this network architecture exhibits peer-to-peer characteristics.

Data Acquisition and Presentation

At the streaming source, the content is prepared for distribution. If the data was pre-recorded and is available as files, we categorize this as on-demand streaming (Fig. 2). On the other hand, if the data is acquired in real time from a device, we term this live streaming (Fig. 3). Content for on-demand streaming is pre-recorded and made available at source nodes usually long before the first delivery requests are initiated. This pre-recorded content can be distributed onto a single or multiple source nodes. Compared with a live streaming system, on-demand streaming usually can utilize a more sophisticated distribution process which may mean encoding the content into a processing-intensive, high-quality format and pre-loading it onto multiple source nodes. The efficiency and scalability of on-demand streaming is improved by caching copies of the content at the intermediate peers. With this approach, popular content is automatically replicated many times within the network and a streaming request can often be satisfied by peers in close proximity.



Peer-to-Peer Streaming. Figure 2. On-demand streaming.

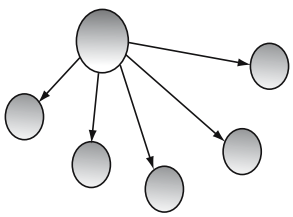


Peer-to-Peer Streaming. Figure 3. Live streaming.

One-way live applications have similar requirements as their on-demand cousins. One obvious difference is that the source data is generated in real time by a source device such as a camera, a microphone or some other sensor. One application is the broadcasting of live events such as sports games. Data may be cached for later on-demand viewing. Two-way live applications have very different requirements. Here, the end-to-end latency is crucial to enable interactive communications. Note that P2P topologies have a disadvantage in terms of minimizing the latency among participants because application-level processing is often required at every node. Skype (http://www.skype.com) was probably the first successful Internet telephony system built on a P2P streaming architecture. It demonstrated that the latency problem can be solved and that P2P technology, with its many advantages, can indeed be used for live streaming purposes. AudioPeer [1], which is built on top of the ACTIVE [2] architecture, is another multi-party audio conferencing tool. It is designed specifically for large user groups. Its design distinguishes active users from passive users and provides low-latency audio service to active users.

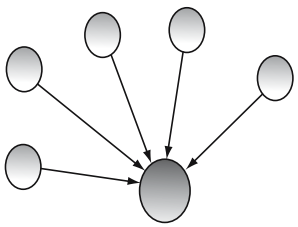
Data Delivery

The transition of one or multiple copies of the content from a source node to a destination node is called a *streaming session*. A streaming session starts when a streaming request is made and ends when all associated destination nodes have received the last byte of the content. Depending on the number of source and destination nodes involved in a streaming session, we can distinguish three types of streaming systems: one-to-many, many-to-one and many-to-many (see Figs. 4–6). All of these three types apply to either live or on-demand streaming. One-to-many streaming is

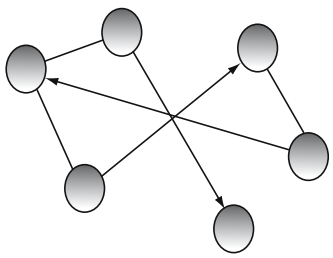


Peer-to-Peer Streaming. Figure 4. One to many streaming.

also called broadcasting. It delivers content from a single source to multiple destination nodes. Much research has focused on how to make the delivery process fast and efficient for one-to-many streaming. P2P systems naturally produce a multi-cast distribution tree since any peer that receives a stream can forward it to multiple other nodes. Many-to-one streaming delivers data from multiple sources to a single destination. A good example is an on-demand movie viewer who simultaneously downloads fragments of the movie clip from multiple peers. Many-to-many streaming combines the features of the previous two designs and usually requires a more complicated delivery network, which we will discuss in detail in following sections.



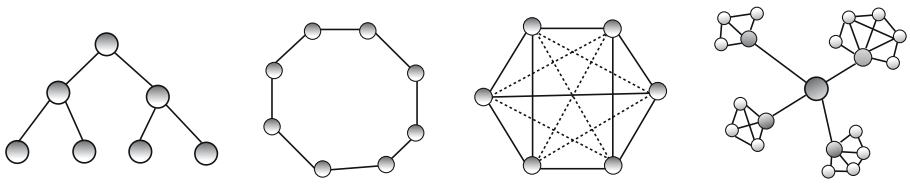
Peer-to-Peer Streaming. **Figure 5.** Many to one streaming.



Peer-to-Peer Streaming. **Figure 6.** Many to many streaming.

The P2P *network architecture* represents the topology how the nodes are inter-connected in a P2P system. P2P *streaming architecture* is the data path over which the streaming content is delivered from source to destination nodes. For a P2P streaming system, the network architecture is not necessary the same as the streaming architecture. For example, Scribe is a P2P network protocol constructing a ring-shaped network architecture and Pastry is the streaming architecture built on top of Scribe. But for most P2P systems, these two architectures are identical and can be represented in a single topology graph.

P2P streaming topologies, including the network architecture and the streaming architecture, can be categorized into four types: tree, mesh, ring, and hybrid (see Fig. 7). Tree structures start with a root node and add new nodes in a parent/children fashion. Many systems are built as tree topologies, e.g., AudioPeer, Yoid [3], and HMTP [4]. A mesh-based topology builds a full interconnect from each node to every other node and constructs a fully-connected map. For example, Narada [5] builds a mesh structure among all the peers and then for each peer constructs a single-source multicast tree from the mesh structure. Due to its centralized nature, Narada does not scale well. A ring-shaped topology links every node in the graph sequentially. This is usually done by assigning each node a unique node ID, which is generated by specific algorithms such as a distributed hash table (DHT). Finally, a hybrid approach combines two or more of the previous designs into their topology graph. Hybrid systems are usually divided into multiple hierarchical layers and different topologies are built at each layer. For example, NICE [6] was developed as a hierarchical architecture that combines nodes into clusters. It then selects representative parents among these clusters to form the next higher level of clusters, which then is represented as a tree topology.



Peer-to-Peer Streaming. **Figure 7.** Peer-to-peer topologies.

Peer-to-Peer System Operation

From the perspective of a peer, the life-cycle of a P2P streaming session can be decomposed into a series of four major processes: finding the service, searching for specific content, joining or leaving the service, and failure recovery when there is an error.

Service Discovery and Content Search

In most P2P systems, service discovery is accomplished through a bootstrap mechanism that allows new nodes to join the P2P substrate. It may be accomplished through some dedicated "super-peers" to act as the well-known servers to help new peers to find other member nodes. These "super-peers" are called Rendezvous Point (RP) servers and are sometimes under the control of the administrator of the P2P system. A new peer finds the existence of the running Rendezvous Point Server from its pre-loaded RP server list. The list can be updated once a peer is connected to one of the RP servers. RP servers can also be used to collect statistic data and in some systems, these "super-peers" are connected to form a backbone streaming platform to make the system more stable.

The next step for a peer, after joining the collective, is to locate a stream or session. The availability of specific content can be discovered in two distinct ways. In an *unstructured* design, streams and files are located by flooding the P2P network with search messages. This technique is obviously wasteful and may result in significant network traffic. The second approach, called *structured*, is to index the content such that search messages can be forwarded efficiently to specific nodes that have a high probability to manage the desired content. To keep with the distributed theme of P2P systems, indexing is often achieved by hashing a content key and assigning that key to nodes with a distributed hash table (DHT) mechanism.

JOIN: After retrieving the necessary information from the RP server, or gaining enough information from the P2P system through some methods such as flood-based search, a new peer can join an existing session by establishing the necessary connections to already joined peers. After the join operation is done, a peer is considered to be a legitimate member.

LEAVE: Every member of a P2P system is usually also serving some other peers as part of the duty to share the load of the whole system. An unexpected departure of a peer can cause disruptions or loss of service for other peers in the system. Ideally a peer should help to reconcile the disconnect in the streaming network caused by its departure. If a system protocol is well designed, this process can be very fast and almost unnoticeable to the end user application.

RECOVERY: In the dynamic environment of a P2P system where peers are under different administrative control, the unexpected departure of peers is unavoidable. A P2P streaming system must cope with these failures and include a robust and efficient recovery mechanism to repair the streaming topology. However, on the positive side, since a robust recovery mechanism is an integral part of the design, this makes P2P systems naturally very tolerant to faults.

Challenges for Peer-to-Peer Streaming

P2P systems are designed to distribute the workload and network traffic among the peers and take advantage of the computing and storage resources of each individual peer. There are two aspects to this approach. One the positive side, a P2P system is very scalable and can potentially serve a very large streaming community where the network and processing load will be a significant challenge for a centralized system. The drawback of P2P systems is that because of the dynamic and unpredictable nature of peers a more complicated, fully distributed protocol is required to constantly maintain the system and recover from errors. A lack of centralized control also introduces difficulties for the administration and security of P2P systems. Below we list a few of the challenges commonly encountered in a P2P streaming system.

Quality of Service (QoS)

The quality of service (QoS) of a streaming system usually refers to the end-users experience. Criteria may include the smoothness of the display, the frequency of visual distortions, and the startup latency from session initiation to the onset of the display. QoS requirements depend on the type of P2P streaming systems. For example, users can tolerate a relatively longer delay in a non-interactive streaming system such as on-demand movie watching. This is in contrast to the requirements of a live, two-way audio conferencing system in which the delay must be bounded at the millisecond scale.

The fact that a P2P system is connected in a distributed topology introduces some challenges that are usually less relevant for a centralized system. For

example, in order to accommodate a large number of members, P2P systems usually build an application-level overlay network among all members. The resulting stream forwarding or processing at the application level increases the end-to-end delay through the additional intermediate hops from source to destination and as a result it is difficult to build a low latency streaming platform using a P2P platform. Some existing work has investigated low-latency P2P streaming. One idea is to distinguish active users who require low latency from passive users who can tolerate longer latency. By clustering the active users logically close together the delay among them can be reduced [2]. The remaining challenge is to distinguish active users effectively and automatically.

Dynamics

One of the biggest challenges for all P2P streaming systems is how to provide a reliable service over an unreliable, constantly changing and most likely, heterogeneous streaming architecture. The members of a P2P system are often of different computing power, network bandwidth, and network connectivity. Some are connected from behind a firewall and some are connected through a network address translation (NAT) device. Peers may join and leave at any moment, leaving some fraction of the P2P streaming network isolated and disconnected. These dynamics make the construction of a reliable and deterministic streaming service very challenging.

A common solution is to maintain redundant information to recover the lost service. For example, in the AudioPeer system, each peer caches information about a fraction of the other online peers and when there is a disruption, this cached peer list is used to repair the network. Another possibility is to ask the rendezvous point server for help. This approach is easier to implement and the service may be more reliable since the server is usually monitored and maintained professionally. However such a centralized recovery design hinders the scalability of a distributed system and may increase its costs. A hybrid approach that combines the above two designs is often a good compromise.

Security

Starting from the early days when P2P systems were mostly used for file sharing until today's blooming online audio conferencing systems that employ P2P as the streaming architecture, security has always been a big concern that affects the acceptance of P2P applications. Members in a P2P system are usually untrusted entities and service is received through such peers. This cooperative model opens the door to unfair service distribution (i.e., a peer only consumes services but does not provide any) and abuse. Since most peers computers are not maintained and configured by network security professionals, a P2P network provides an opportunity for hackers and malicious attacks (e.g., injection of bogus content).

Aside from worries that the P2P service could open a back door to intruders, many people are also concerned about the possibility that confidential information can be obtained by the intermediate peers who are used to relay the content form source to destination. Unfortunately many of current paradigms for P2P systems are limited to the authentication phase and scant research has been done in terms of how to assure the integrity and confidentiality of content being delivered. It will be helpful to revise some research proposals in multicast areas and find the appropriate implementations for P2P streaming platform.

Profit model

Despite the popularity and promising advantages of P2P streaming systems, finding a viable business model remains elusive. One of the challenges is to measure the usage of each individual in a distributed system and charge fairly for the services received. Since each peer in the P2P streaming system is acting as both a customer and a provider, it would not be fair to charge the user by the volume of content received without considering her contribution to help relay the content to other peers. It is also quite challenging to monitor the activities of peers even when permission is given. Various P2P streaming systems may require different profit models. One possible revenue stream is to display advertising on a companion website or in the content itself. Another possibility is to charge a fee for add-on services. Creating a fair and efficient subscription model for P2P streaming systems is a practical challenge that needs to be resolved before P2P streaming systems become a mature commercial platform.

References

1. R. Zimmermann, B. Seo, L.S. Liu, R.S. Hampole, and B. Nash, "AudioPeer: A Collaborative Distributed Audio Chat System,"

Proceedings of the Tenth International Conference on Distributed Multimedia Systems (DMS 2004), San Francisco, CA, September 2004.
2. L.S. Liu and R. Zimmermann, "ACTIVE: A Low Latency P2P Live Streaming Architecture," Proceedings of the SPIE Conference on Multimedia Computing and Networking Conference, San Jose, CA, January 2005.
3. P. Francis, Yoid: Your own internet distribution.
4. B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast to End Users," Proceedings of IEEE Infocom, New York, June 2002.
5. Y. H. Chu, S.G. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture," Proceedings of the ACM SIGCOMM 2001, San Diego, CA, August 2001.
6. S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," Proceedings of the ACM SIGCOMM 2002, pp. 205–217, Pittsburgh, PA.
7. M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "SCRIBE: A Large-scale and Decentralized Application Level Multicast Infrastructure," IEEE Journal on Selected Areas in Communications (JSAC), Vol. 20, No. 8, 2002, pp. 1489–1499.
8. M. Steiner, G. Tsudik, and M. Waidner, "Key Agreement in Dynamic Peer Groups," IEEE Transactions on Parallel and Distributed Systems, Vol. 11, No. 8, August 2000, pp. 769–780.
9. D. Malkhi, M. Merrit, and O. Rodeh, "Secure Reliable Multicast Protocols in a WAN," Proceedings of the International Conference on Distributed Computing Systems (ICDCS 97), 1997, pp. 87–94, Baltimore, MD.
10. S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-level Multicast Using Content-Addressable Networks," Proceedings of the Third International Workshop Networked Group Communications, November 2001, LNCS, Vol. 2233, Springer, London, pp. 14–29.

Peer-to-Peer Systems

Definition

The term peer to peer denotes a class of distributed system architectures, that is, a way of structuring and organizing the work of several computers that communicate through a network.

In a peer to peer system, each computer that participates in the system has some kind of resource (data, computing capacity, disk space, algorithms, etc.) that it offers to the other users of the system. In a multimedia peer to peer system, the most common resource is constituted by data files. A computer needing a particular file or algorithm will send a request for it to all or some of the participants of the system (the details of how this happens depend on the specific architectural details of the systems, and will be debated shortly). The participants that have that resource available will answer, possibly with some additional information about the computer on which the resource is located, such as its computing power, network bandwidth, work load at the moment, etc. Based on this information, the requester contacts one of the participants that answered, and instaurates with it a one to one communication during which the file is exchanged, or the algorithm is invoked, the disk space is allocated and used, etc. . ..

With respect to client server, peer to peer systems present the advantage of avoiding the concentration of network traffic or the computation on a single computer – or on a small group of computers–, namely the server. On the other hand, knowing where a resource is located is a more complicated affair in a peer to peer system, and subject to more compromises. There are, with respect to the localization of a resource, several solutions possible.

The most immediate possibility is that of complete broadcast: whenever a computer needs a resource, it will send a request to *all* the other computers in the system to verify which ones have the resource. Two are the problems with this solution: firstly, the network traffic that it generates: in a system with N computers, for each request, $N-1$ messages (this problem is not too dramatic, since only short messages are sent); secondly, every computer must contain a list of all other participants that, at a given time, are active. In a large system, in which computers come on-line and go off-line frequently, creating and maintaining such lists might be a considerable problem.

A different architectural solution is what might be called *impure* peer to peer. In this solution, there is a central server that contains the location of all available resources. Whenever a participants comes on-line, it communicates its availability to the central server, and whenever the participants goes off-line, it communicates that it will be no longer available; in this way the server can maintain an updated list of active participants. Whenever a participant needs to access a certain resource (e.g., a file), it will first contact the server to obtain the location(s) of that resource (e.g., given the name of the file, a list of computers and the relative directories where files with that name are located) and then contact one of these computers in order to access the resource (viz. copy the file). This solution relieves the problem of the previous one, since only the

server needs to keep a list of available resources, and it also avoids some of the problems of client-server architectures, since the server only deals with short search requests, and not with the actual use of the resources. On the other hand, many simultaneous requests can still result in congestion, and the system is, just as client-server, susceptible to malfunctions in the server.

A third solution involves partial indexing: every computer in the system keeps a list of a certain, usually fixed, number of other computers, called its (first order) *neighbors*. A search request goes from a computer to its neighbors, from these to their neighbors (the second order neighbors of the requester), and so on until a prescribed and fixed order of neighborhood is reached: these are the computers on which the sought resource is searched. The most serious problem of this solution is that a search might fail to find a resource even if it is present in the system, since not the whole system is searched. For peer to peer systems in which resources are duplicated on many computers (e.g., file exchange, in which the number of locations where a given file can be found grows with every copy), and in which absolute reliability is not a requirement, this solution can be usefully applied.

Cross-References

► Multimedia File Sharing

Peer-to-Peer Systems and Digital Rights Management

Definition

Peer to peer file sharing and the management and enforcement of digital rights remain two very important cogs in online multimedia e-commerce.

Peer to Peer File Sharing and Tools

While large multimedia repositories such iTunes and Download.com store media in an organized and *centralized* fashion, other technologies allow multimedia to be replicated and distributed upon thousands of servers. In this latter case – the peer-to-peer case – only the media index may be centralized; the media themselves are stored on distributed servers. When a client wants a particular media the following main steps occur: (1) a search algorithm yields the best source(s) of the media for this user, (2) the transfer occurs (the media may arrive in chunks from different sources), (3) the media is recovered and validated at the client (e-payments may be made at this point), (4) the media and its new location are registered, allowing subsequent clients to be served the media from its new location. Decentralized media indices are another variation of these frameworks in which case no single server indexes all the media. At any rate, such systems (first popularized by Napster) allow the exchange of copyrighted media (e.g., music in mp3 format and movies in MPEG-4 format) and brought legality issues of file-sharing to the forefront in the late 1990s. In 1999 the RIAA (backed by AOL Time Warner, Bertelsmann, EMI, Vivendi Universal and Sony) filed suit against Napster; the suit symbolically ended the era of care-free peer-to-peer exchange of copyrighted material.

Current Landscape

The current peer-to-peer file sharing landscape is dynamic. Napster has been reborn into an outlet for legal file exchange. It is joined by many other legal systems. In 2004, more than 200 million music tracks were downloaded and the related revenues increased six-fold from 2003. Some industry experts estimate the online music market at $660 million for 2005.

Digital Rights Management

Despite the systems and tools listed above experts agree that digital piracy of online multimedia will remain a threat. For this reason, Digital Rights Management (DRM) [1] technologies are important. Simply put, DRM allows media rights holders to restrict the usage of media. Such restrictions are implemented by various technologies; for example, a rights database persists usage rights and policies of use, while a cryptographic layer provides the "lock and key" that the rights holders requires. A common DRM issues is how to restrict content buyers from copying and redistributing media. As a case in point, at the time of writing the Open Mobile Alliance (OMA) – a large and influential forum for mobile service enablers – is the proponent of a DRM standard for mobile wireless content. OMA's DRM specification will consist of (1) a rights expression language, (2) a content format, and (3) a framework for content metadata. For rights holders wishing to allow e-commerce on mobile devices on the basis of their content, efforts like OMA's DRM and others are key stepping stones.

Cross-References

► Online Multimedia E-Commerce

Reference

1. R. Ianella, "Digital Rights Management Architectures," D-Lib Magazine, Vol. 7, No. 6, June 2001.

Peer-to-Peer VoD Architecture

► Large Scale Multimedia Streaming in Heterogeneous Network Enviroments

Perceptural Image Optimization using SSIM

► Structureal Similarity Index Based Optimization

Person Detection in Images and Video

Petros Kapsalas, Yannis Avrithis
National Technical University of Athens, Athens, Greece

Synonyms

► Human detection; ► Person localization

Definition

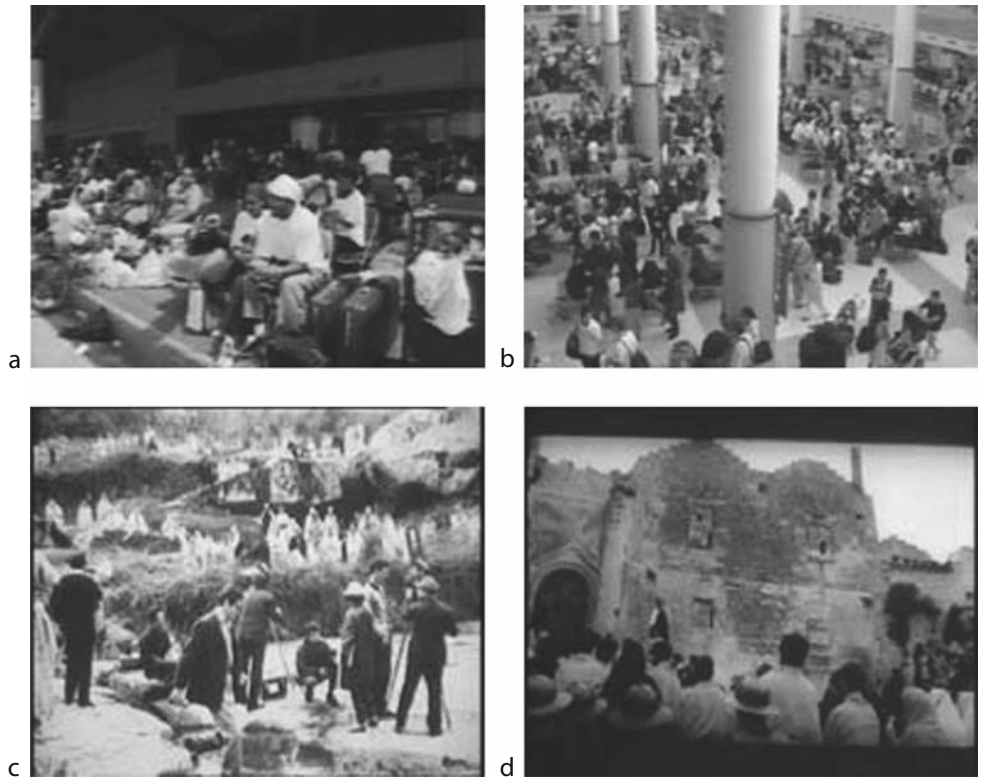
Human detection corresponds to the process of detecting human bodies, either globally or as distinct human body parts. There are numerous challenges that should be considered through the detection process. They are mostly associated with the monitoring conditions and the variability of poses and orientations that the human body can adopt. Thus, person detection can be regarded as a more general problem than human localization in the sense that the number of persons is not known a-priori. The response of an efficient person detector provides a bounding polygon or box at the location of human occurrence.

Introduction

The rapidly expanding research in face processing and human body analysis is based on the premise that information about a user's identity, state, and intent can be extracted from multimedia sources. In particular, face detection and analysis corresponds to a well-known problem which has been thoroughly considered through the last several years. On the other hand, human body detection involves numerous challenges which are mainly associated with the variability of poses and orientations that the human body can adopt. Moreover, monitoring conditions such as occlusion, illumination changes, clothing etc should also be faced. Thus, some of the recent human detection systems perform face localization as a preliminary step for the detection of the human body in overall. In this way, false positives induction is eliminated while enhancing localization accuracy. In order to further illustrate the challenges associated with human detection we depict (in Fig. 1 (a) through (d)) some representative cases of human occurrences in complex scenes.

In an effort to categorize the most popular person detection systems we should mention that they can be classified into four broad categories according to the way that they approach the overall problem. However, many of the existing techniques clearly overlap the boundaries. The above mentioned categories are at first specified and only shortly explained. More details on the methods included within each distinct category are discussed in the subsequent sections.

1. *Bottom-Up Feature-Based Approaches.* These algorithms aim to find structural features that exist even when the pose, viewpoint, or lighting conditions vary, and then use them in the detection procedure.
2. *Top-Down Knowledge-Based Methods.* These rule-based methods encode knowledge of what constitutes a typical human body. These methods are designed mainly for human body localization.
3. *Template Matching Methods.* Several standard patterns of humans or human body parts are used to describe either the human body globally or as distinct human body parts (limbs, face, head etc). The correlations between the input image and the patterns are computed for detection. These methods have been used for both localization and detection with considerable accuracy.



Person Detection in Images and Video. Figure 1. (a) crowded scene, human bodies at various poses, aliasing due to camera movements (b) Image taken from upside-down, (c) low contrast, human bodies at variable poses, (d) similar to (c).

4. *Appearance-Based Methods.* In contrast to template matching, the models (or templates) are learned from a set of training images, which should capture the representative variability of human appearance. These learned models are then used for detection.
5. *Integration of Parts detectors.* In contrast to all the techniques described above, this last category fuses the detection results derived by robust part detectors. They are commonly deployed in order to perform reliable approximation of the bodies' shape and extent.

Bottom-Up Feature Based Approaches

Many recent methods of detecting humans involve extracting features from the studied images and selecting the appropriate feature sets to encode human body characteristics. The person detection problem is thus reduced in selecting the appropriate feature sets and subsequently using the extracted features as input for the machine learning module. Many descriptors representing human body's occurrence have been proposed thus far [1–3]. The majority of them are based on local appearance and shape. Thus, low-level descriptors extract visual features associated with local intensity variations, dominant colors, edges orientations or spatial density, motion and other generalized measures.

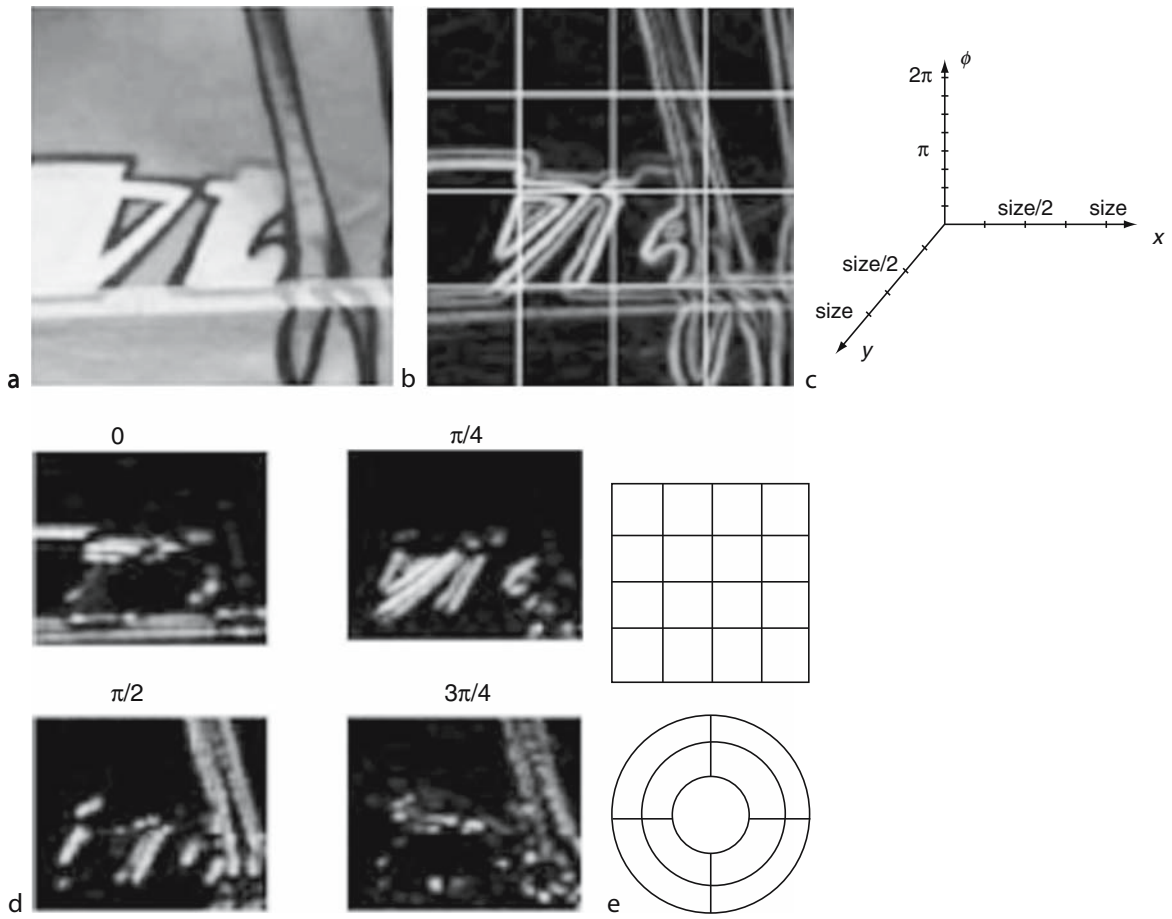
Feature spaces relying on distribution-based descriptors have become quite popular. These techniques use histograms to represent different characteristics of appearance or shape. In this category, numerous gradient and contour based descriptors have been deployed to determine the exact locations of human occurrence. The Scale Invariant Feature Transform (SIFT) and shape context have been extensively used for person/face localization [3–6]. Scale invariant feature transform (SIFT) was proposed by Lowe [4] and works by combining a scale invariant region detector and a descriptor based on the gradient distribution in the detected regions. The descriptor is represented by a 3D histogram of gradient locations and orientations. The quantization of gradient locations and orientations makes the descriptor robust to small geometric distortions and suppresses the errors in the region detection. Geometric histogram [5] and shape context

[6] implement the same idea and are very similar to SIFT descriptor. Shape context is identical to SIFT descriptors but it is based on edges.

Histogram of Gradient Orientations (HOGs) [7] is also a powerful recent approach providing accurate detection results. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients orientations or edge directions. The method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. For better invariance to illumination, shadowing etc., the local responses are contrast normalized before being used. Feature representation of the overall image takes place by tiling the detection window with a dense grid of HOG descriptors and feeding the combined feature vector in an SVM-based window classifier [Fig. 2].

Frequency-based descriptors form a further class of representation techniques. Fourier Transform is the most popular frequency representation approach which works by decomposing the image content into basis functions. However, in this representation, the spatial relations between points are not explicit and the basis functions are infinite; therefore, it is difficult to adapt to a local approach. Moreover intensity peaks or singularities associated with the presence of an object of interest may not be visible when the window of the transform is not appropriately selected. The Gabor transform overcomes these problems. However, a large number of Gabor filters is required to capture small changes in frequency and orientation

Generalized Haar wavelet basis functions provide a further frequency-based representation; that have been extensively used to extract features characterizing the

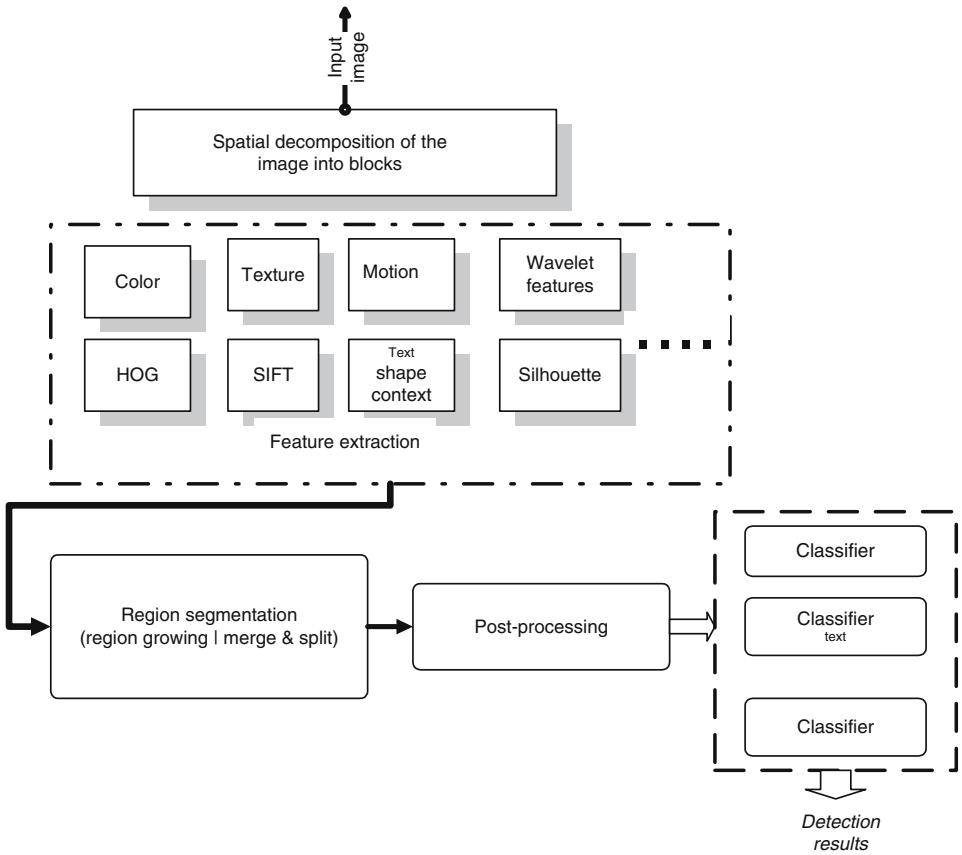


Person Detection in Images and Video. **Figure 2.** SIFT descriptor. (a) Detected region. (b) Gradient Image and Location grid. (c) Dimensions of the histogram. (d) Four of eight orientation planes. (e) Cartesian and the log-polar grid shows nine location bins used in shape context (four in angular direction).

humans' presence in complex scenes [1]. Wavelet features improve the performance of the classification process because, depending on the context, it allows representing the original image only through the coefficients of the sub-bands containing the most discriminating information. Moreover, by selecting only the most important coefficients, the computational complexity is reduced. Papageorgiou and Poggio [8] deployed Haar Wavelet Transform to build a representation that describes person and face classes in terms of over-complete dictionaries of local, oriented, multi-scale intensity differences between adjacent regions. The features derived by the above approach are used to train an example based training system.

AC and DC Discrete Cosine Transform *(DCT)* coefficients have also been used in numerous applications in order to detect humans in single images and video sequences [8–10]. Ozer and Wolf in [9] represent structural aspects of the humans' silhouettes by DCT coefficients. Square image blocks are initially processed in order to obtain the energy by summing up the absolute amplitudes of the first order. This work proves that the sides of the human body have a high response to the vertical harmonics while AC coefficients of the horizontal harmonics capture head, shoulder and belt lines. Furthermore, the corner edges at shoulders, hands and feet contribute to local diagonal harmonics. The goal of their approach is that they find a compact representation of the human silhouette by computing the principal components of the energy distribution of human bodies, or the eigenvectors of the covariance matrix of the human body images. These eigenvectors represent a set of features, which together characterize the variation between human images.

Figure 3 summarizes the general architecture of the majority of bottom-up person detectors. Thus, it can be observed that bottom-up detection approaches are based on encoding local features through a range of robust visual descriptors. The extracted characteristics are used to train independent classifiers. Finally, any given image is scanned (through different ways) to detect humans' location.



Person Detection in Images and Video. Figure 3. General architecture of person detection techniques.

Top-Down Knowledge Based Methods

Top-down approaches of detecting objects in static or dynamic scenes involve an initial step of determining candidate regions and subsequently extracts features from the regions of interest and uses them as inputs to a classification module. Knowledge-based methods have the fundamental advantage of reducing false positive instances of detection as they are eliminated through the verification step. Motion characteristics provide important information in both determining regions of interest and assessing whether an object's motion features resemble to humans motion [10–12].

The majority of the motion based person detection approaches initially extract features from the objects motion and subsequently uses prior knowledge to assess whether the region of interest corresponds to a person or not. Interest has recently increased because of the clear application of these methods to problems in surveillance. Cutler and Davis describe a system that measures motion periodicity robustly and directly from the tracked images. In contrast, all other systems require complex intermediate representations. Viola and Jones [11] proposed a state of the art human detection approach, which considers prior knowledge on the person's motion and appearance. No separate mechanisms of tracking, segmentation and alignment are supported. The system works by simply selecting the feature set, the scale of the training data and the scales used for detection. The training process uses AdaBoost to select a subset of features and construct the classifier. The classifier consists of a linear combination of the selected features. Viola and Jones [11,13] have also proposed a cascade of classifiers architecture to reduce the computational cost.

Template Matching Approaches

These methods usually involve background subtraction to derive the regions of interest and subsequent template matching to a database containing human bodies or human bodies' parts. In the next step, the foreground regions are processed to measure the selected features values. Template matching takes place subsequently by measuring the distance between the object of interest and the pre-stored templates. Whether the distance is beyond a threshold then the studied region is not detected, otherwise it is selected as human. The final verification stage is performed either by SVM classifiers or by analyzing the motion features extracted from the objects of interest and assessing whether the motion characteristics resemble to human motion. Gavrila [12] propose a human detection scheme, which initially involves segmentation of foreground regions and edge-based analysis. At a further step, the algorithm searches for humans in the image by matching the ROIs' edge features to a database of templates of human silhouettes. The matching is realized by computing the average Chamfer distance between the template and the edge map of the target image area.

Wren et al. [14], also describe a person detector based on template-matching. However, their approach requires domain specific scene analysis. Castillo and Chang use fast template matching as a focus of attention [15]. Basically, their algorithm discards locations where there is no silhouette that matches the human body. Their definition of human body occurrence includes two cues: a silhouette and some visible skin.

Integrating Robust Part Detectors

There is a considerable amount of research work that is focused towards developing human detectors based on an assembly of body parts. Earlier approaches consider specific body plans for finding people in general configurations. Ioffe and Forsyth [16] describe a model of assembling body parts with projected classifiers or sampling. However, they rely on simplistic body part detectors- the parts are modeled as bar shaped segments and pairs of parallel edges are extracted. This body part detector fails in the presence of clutter and loose clothing. Considerable improvement on the modeling of body part relations is given in Sigal et al. [17], where these are represented by a conditional probability distribution. However, these relations are defined in 3D, and multiple simultaneous images are required for detection. Mikolajczyk et al. [18] propose the use of local orientation position features extracted by gradient and Laplacian based filters. The spatial layout of the features, together with their probabilistic co-occurrence captures the appearance of the parts and their distinctiveness. The features with the highest co-occurrence and co-occurrence probabilities are learnt using AdaBoost. Finally, the detected parts are combined with a joint probabilistic body model. The features deployed by Mikolajczyk et al has proven to describe the shape better than prior descriptors such as Haar wavelets [8], yet they are simple enough to be computed and provide reliable local information to allow for handling occlusions and close up views.

Appearance-Based Methods

In contrast to template matching methods, where templates are predefined by experts, the "*templates*" in appearance-based methods are learned from examples in images. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of person and non- person images. The learned characteristics are in the form of distribution models or discriminant functions that are consequently used for face detection. Meanwhile, dimensionality reduction is carried out for the sake of computation efficiency and detection accuracy.

Hydra [19] is a real-time system for detecting and tracking multiple people *before*, *during* and *after* occlusions in monochromatic imagery. *Hydra* constructs a silhouette-based shape model to classify whether or not a foreground blob contains humans, and employs second order motion models to track them. First local corner detection is used and its results are analyzed to determine candidate head locations on the boundary. Next, a vertical projection histogram of each silhouette is constructed in order to verify head locations and find the vertical boundaries between people. A dynamic template for each detected head is generated and updated during tracking. The tracking system is used to predict the location of the head in consecutive frames. The silhouette of the foreground blob is segmented into regions representing individual people. Distance values from each pixel to each potential person are used to obtain the normalized distance map for each silhouette and each pixel is assigned to a person according to this map. Finally, an appearance model is generated and updated for each person during tracking so that the person can be identified after interactions or occlusion.

Results and Discussion

This article provided an overview of state-of-the-art person detection approaches while organizing them

Person Detection in Images and Video. **Figure 4.** Illustration of the results provided by the person detection approaches. (a) Detection of more than one persons in the same image, (b & c) human detection under difficult monitoring conditions (aliasing in (b) and illumination problems & shadowing in (c)), (d) results provided by the Hydra system [19], (e) part-based person detection [18].

based on the way they approach the overall detection issue. Figure 4 illustrates the potential of several person detection approaches.

Figures 4 (a) through (e) illustrate the results provided by some recent human detection systems. As it can be observed, the performance is quite efficient in detecting humans under a wide variability of monitoring conditions. However, the problem cannot be considered as resolved. Recent research is focused towards developing accurate part-based detectors able to deal with complex occlusion phenomena. Moreover, the induction of powerful feature sets (robust to geometric distortions) able to represent efficiently human body characteristics is also an aspect under investigation. Many researchers also investigate accurate localization of human bodies through providing the exact locations while suppressing the borders of the bounding rectangle that usually depict background regions.

References

1. C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," International Journal of Computer Vision, Vol. 38, No. 1, 2000, pp 15–33.
2. C.P. Papageorgiou, M. Oren, and T. Poggio, "A General Framework for Object Detection," Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, pp. 555–562, 1998.
3. J.W. Davis, V. Sharma, "Robust Background Subtraction for Person Detection in Thermal Imagery," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04), Vol. 8, 2004, pp. 128–135.
4. D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, Vol. 2, No. 60, 2004, pp. 91–110.
5. A. Ashbrook, N. Thacker, P. Rockett, and C. Brown, "Robust Recognition of Scaled Shapes Using Pairwise Geometric Histograms," Proceedings of Sixth British Machine Vision Conference, pp. 503–512, 1995.
6. S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, No. 4, 2002, pp. 509–522.
7. N. Dalal and B Triggs, "Histograms of Oriented Gradients for Human Detection," Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2005, pp. 886–893.
8. C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," International Journal of Computer Vision, Vol. 38, No. 1, 2000, pp. 15–33.
9. I.B. Ozer and W.H. Wolf, "A Hierarchical Human Detection System in (Un) Compressed Domains," IEEE Transactions on Multimedia, Vol. 4, No. 2, 2002, pp. 283–300.
10. J.K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," International Journal of Computer Vision and Image Understanding, Vol. 73, No. 3, 1999, pp. 428–440.
11. P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, Hawaii, USA, Vol. 1, 2001, pp. 511–518.
12. D.M. Gavrila and V. Philomin, "Real-Time Object Detection for Smart Vehicles," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Fort Collins, Colorado, USA, pp. 87–93, 1999.
13. P. Viola, M.J. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," Proceedings of the Ninth International Conference on Computer Vision, Nice, France, Vol. 1, 2003, pp. 734–741.
14. C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. "Real-Time Tracking of the Human Body," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, 1997, pp. 780–785.
15. C. Castillo and C. Chang. "An Approach To Vision-Based Person Detection in Robotic Applications," Proceedings of the Second Iberian Conference on Pattern Recognition and Image Analysis, Vol. 2, 2005, pp. 209–216.
16. S. Ioffe and D. Forsyth, "Probabilistic Methods for Finding People," International Journal of Computer Vision, Vol. 43, No. 1, 2001, pp. 45–68.
17. L. Sigal, M. Isard, B.H. Sigelman, and M.J. Black, "Attractive People: Assembling Loose-Limbed Models Using Non-Parametric Belief Propagation," Advances in Neural Information Processing Systems, NIPS Vancouver, Canada, Vol. 16, 2003.
18. K. Mikolajcyk, C. Schmid, and A. Zisserman, "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors," Proceedings of the European Conference in Computer Vision (ECCV), pp. 69–81, 2004.
19. I. Haritaoglu, D. Harwood, and L. Davis, "W4S: A Real Time System for Detecting and Tracking People in 2.5D," Proceedings of the European Conference on Computer Vision (ECCV), pp. 877–892, 1998.

Person Information Analysis

► Face Detection, Tracking, and Recognition for Broadcast Video

Person Localization

► Person Detection in Images and Video

Personalized Educational Hypermedia

Apple W. P. Fok, Horace H. S. Ip
City University of Hong Kong, Hong Kong, China

Synonyms

► Sematic Web for educational hypermedia

Definition

Enabling technologies that support the personalization of Web-based applications include technologies for user profiling, intelligent search, filtering and recommendation of hypermedia contents, adaptive hypermedia, tracking and characterization of user browsing behaviors, and adaptive user interface.

Introduction

The emergence of personalization technologies on the Web allows customized information to be delivered to users according to their pertinent needs and interests as perceived by the system. Enabling technologies that support the personalization of web-based applications include technologies for user profiling, intelligent search, filtering and recommendation of hypermedia contents, adaptive hypermedia, tracking and characterization of user browsing behaviors, and adaptive user interface. In order to boost the commercial advantages and to retain internet-based customers' loyalties, personalization technologies have been adopted and driven predominantly by e-commerce applications and information portals such as MyYahoo, Amazon, Persona [1] etc. Enterprises increase their profits by providing personalization services to their customer on the web, for example, by providing personalized features such as making suggestions based on users' transaction and historical records. The idea of personalization can be summarized into three main themes: "Building a meaningful one-to-one relationship" [2], "Delivering appropriate content and services to fulfill user's needs" [3], and "Understanding where and when to suggest the 'right' things" [4]. The ultimate goal of personalization is "User satisfaction." User satisfaction means getting the right thing at the right time in the right place.

In order to create a personalized experience, a general Personalized Hypermedia System must perform several distinct tasks: (1) Identify the user; (2) Store and Update user information; (3) Learn and identify the user preferences, needs and interests; and (4) Provide and recommend specific personalized services. In summary, personalization techniques serve to enable the system to know the user, remember the user, and adjust their personal memory of the user according to the user's changing needs.

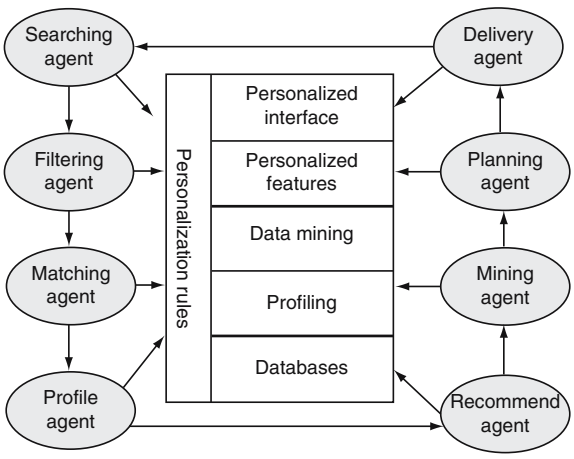
With the increasing demand of learning just-in-time and just-in-place, the Web has become a new and effective channel for education and the acquisition of knowledge, particularly in adult education and distant learning. Streaming technology allows the delivery of continuous media such as video and audio content for educational purpose. The process of learning in the cyber space is more complex than conventional class learning for many reasons. From the educational point of views, the educational philosophers and psychologists are interested in the influences of technologies in human's beliefs, knowledge and learning. Research in the learning environment, on the other hand, focused on the changes of traditional classroom settings to computer-based learning and the impacts of web-based learning have on the learning process.

The demand of adaptability in a computer-based educational system is due to the fact that every person possesses distinctive collection of talents, abilities, and limitations. One-size-fits-all approach to education forgets that individuals are different and have different needs. In response to this demand, research interest toward personalized education hypermedia systems has evolved. An emerging approach is the Adaptive Educational Hypermedia Systems, which inherits and combines two earlier paradigms of computer-based educational systems: Intelligent Tutoring Systems (ITSs) and Adaptive Hypermedia Systems (AHSs). Adaptive educational hypermedia systems (AEH) interact with user through a learner model of the goals, preferences and knowledge of each individual learner and adapt the hypermedia to the needs of that user. A challenge to the development of adaptive hypermedia system is the authoring of adaptive hypermedia materials. In additional to hypermedia content, the presentation of the content should also be adaptive to the user needs through mechanisms such as link annotation link hiding, and conditional inclusion of materials. The Adaptive Hypermedia Architecture (AHA!) provides a set of authoring tools and a runtime environment for creating and presenting adaptive hypermedia courseware.

Conceptual Framework and Functionality Layers of Personalized Education System

Instead of focusing on the creation and presentation of adaptive hypermedia materials, a Personalized Educational System (PES) [5] takes advantage of the vast resource of educational hypermedia on the Web and integrates a range of supportive tools for participants to conduct their teaching or learning activities in a personalized manner. These supports range from automatic search of relevant teaching and learning hypermedia materials from the Web, anticipating needs of the participants based on his/her individual descriptive profile and the filtering, matching and sequencing of the retrieved materials based on dynamic monitoring of the student's progress supported by a learner model or based on the teacher's intended pedagogy. Due to the diverse but inter-related functions that need to be provided in a Personalized Educational System (PES), software agent technology has been applied to the design and development of a PES. Particularly, an agent architecture has been proposed [6] that deploys a team of Personalized agents to achieve the goals of a PES. Figure 1 shows the conceptual agent framework of PES.

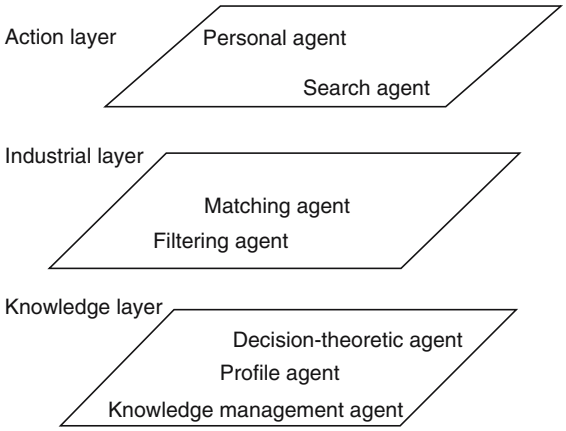
The functionality of PES can be divided into three functional layers. Action Layer: Within this layer, personal agents positioned between the browser and the Web, capture a complete clickstream history for all hypermedia resources visited by the user, and the complete navigational history of the user.



Personalized Educational Hypermedia. Figure 1. Conceptual framework of a personalized education system.

A user profile is generated together with its schema that is both human readable and machine processable. Web servers use the profile information, obtained from the client to deliver personalized information. Domain-specific Search agents as intelligent searching assistant focus on getting relevant hypermedia or documents in one specific subject domain (i.e. educational websites or teaching/learning resources). These agents can recognize not only the types of information provided, but also automatically classified those retrieved documents and hypermedia into different categories based on the subject ontologies as well as the personalized attributes stored in the personalized knowledge base/libraries. Industrial Layer: this is where collaboration work takes place. The actual process matching retrieved content to the learner's needs is carried out by a Matching Agent. The task of the matching agent is to match document metadata against registered profiles. Also in this layer, intelligent filtering agent transforms static data and information into meaningful and useful information for knowledge construction.

Various filtering and data mining techniques would be used. Demographic filtering technique uses to identify the relationship between a particular resource and the type of users. Content-based filtering technique is used to learn the relationship between the content and a single user. Collaborative filtering technique uses the like minded approach, in which the navigational history of each user is kept to determine the learning dependent behaviors to make recommendations based on the user profile groups. Knowledge Layer: Educational psychologies, philosophies and instruction design theories/principles as the cognitive knowledge domain of the decision-theoretic agents, these agents are capable of handling queries from other agents for expert advices or diagnosing learner behaviors. To generate different types of profiles and accurately locate the relevant information, the profiling agents take up the responsibilities of handling different representation of profiles and automatically construct an active profile for the other agents. Each learner has his/her own knowledge status which is constructed throughout his/her own development process (i.e. learning or experience). Various pieces of data form different episodes. The Knowledge Management agent helps to manage and organize the acquired knowledge for personalization processes. Figure 2 illustrates the layer structure of the PES agent functionality.



Personalized Educational Hypermedia. **Figure 2.** Three functionality layers of PES.

Semantic Web for Educational Hypermedia and Standards for Interoperability of PES

Learning technologies standardization helps to increase the integration, usability and reusability of heterogeneous systems for education. Many professional organizations and institutions have contributed toward such standardization efforts to strengthen the process of delivering educational services, to define learning data and metadata as well as recommendations for the development of software architectures devoted to computer-based education. One of the main contributors to this effort is the IEEE Learning Technology Standards Committee (LTSC) (http://ltsc.ieee.org/) Learning Objects Metadata (LOM) working group.

The LOM specification (http://ltsc.ieee.org/doc/wg12/LOM_WD4.PDF) describes learning content cataloguing information. This specification is the result of the effort of many contributors. The European ARIADNE project (http://ariadne.unil.ch) uses LOM for indexing and exploiting its network of interconnected knowledge pools (KPS) and the IMS project (http://www.imsproject.org/metadata) provides the IMS Learning Resources Metadata Specifications which would be incorporated into the IEEE specifications. Apart from the learning metadata definition studies, the description of learner profiles and records (http://edutool.com/papi/), course structure formats, course packaging (http://www.imsproject.org/content), questions and tests interoperability (http://www.imsproject.org/question), learning architectures and run time environments, have also been investigated with a view to provide recommendations and specifications to enhance interoperability between different educational platforms.

In order for PES to effectively search and make use of the vast amount of potential learning resources that exist in the Web, it exploits the Semantic Web (http://www.w3.org/2001/sw/) and the development of appropriate ontology for education hypermedia for various subject domains so that content and its inherent educational values can be understood, shared and used across educational platforms. For hypermedia content description, eXtensible Markup Language (XML) allows users to add arbitrary structure to their documents, the meaning of the structure is expressed by Resource Description Framework (RDF), which encodes it in sets of triples, each triple being rather like the subject, verb and object of an elementary sentence. These triples can be written using XML tags. Subject and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a Web page. (URLs, Uniform Resource Locators, are the most common type of URI.) In order to handle the problem between two different education platforms that use different identifiers for what is in fact the same concept, Semantic Web provides a basic component – Ontology. Ideally, an ontology helps a system to discover the common meanings in a subject domain. A typical kind of ontology for the Web consists of taxonomy and a set of inference rules. The taxonomy defines classes of objects and relations among them. For example, English may be defined as a type of subjects, and topic codes may be defined to apply only to subject, and so on. Classes, subclasses and relations among entities are a very powerful tool for Web use. A recent development in this area is a set of ontology for high school subjects, e.g. EduOnto (http://web.syr.edu/~jqin/eduonto/eduonto.html). The emergence of ontology for educational content allows educational content available on the Web to be searched and identified effectively for specific subject domains and for meeting specific learning goals.

One of the outcome of a personalized education system is an individualized learning plan that specifies the sequence for which a set of learning materials should be studied by a learner in order to achieve his/her learning goal or objective. The Sharable Content Object Reference Model (SCORM), first proposed by The Advanced Distributed Learning (ADL) initiative (http://www.adlnet.org/) is an emerging standard for

sharing and defining learning sequence of educational multimedia content. (Link to SCORM, T Shih). By adhering to such standards and making use of the ontology defined for various subject domains, PES is not only able to search and reuse educational multimedia content generated from desperate sources from the Web, its personalized output such as a plan of individualized learning sequence would also be (re-) used by different educational platforms.

Cross-References

► Creating (and Delivering) Adaptive Course Texts with AHA!
► Definition of Adaptive Educational Hypermedia Systems
► Streaming Media and its Applications in Education
► The Sharable Content Object Reference Model (SCORM)

References

1. F. Tanudjaja and L Mui, "Persona: A contextualized and Personalized Web Search," Proceedings of the 35 Annual Hawaii International Conference on System Sciences (HICSS'02), Big Island, Hawaii, Vol. 3, 2002.
2. D. Riecken, "Personalized Views of Personalization," Communications of the ACM, Vol. 43, No. 8, 2000.
3. M. Bonett, "Personalization of Web Services: Opportunities and Challenges," http://www.ariadne.ac.uk/issue28/personalization/.
4. "The Art of Personalization," An Oracle White Paper, August 2003.
5. A.W.P. Fok and H.H.S Ip, "Personalized Education (PE) – Opportunities and Challenges in Technology Integration for Individual Learning," Proceedings of IASTED International Conference on Web-Based Education (WBE 2004), Innsbruck, Austria, pp.48–53, February 2004.
6. A.W.P. Fok and H.H.S. Ip, "Personalized Education (PE) – An Exploratory Study of Learning Pedagogies in Relation to Personalization Technologies," in W.-Y. Liu, Y.-C. Shi, and Q. Li, "Advances in Web-Based Learning (ICWL 2004)," Lecture Notes in Computer Science (LNCS 3143), Springer, Berlin, pp. 407–415, 2004.

Photo Defect Detection

Definition

Image inpainting techniques include photo defect detection, where ink traces, scratch, and damage from ink pens are automatically detected.

The concept of image inpainting is discussed in the article entitled "Digital Inpainting." Most existing inpainting mechanisms allow users to select a defect area or a mask of object to be removed. To cope with the inconveniency, we developed a nave photo defect detection mechanism. Ink traces, scratch, and damage from ink pens are automatically detected. We consider the intensity and shape of photo defects. Though, it is almost not possible to discriminate ink regions from objects in photos. The intensity variation of ink regions is usually quite smooth and steady while comparing with objects in photos. Thus, we use the HSI color space and use intensity in the first filter. In the second filter, we record the number of pixels been detected in each intensity variation. We calculate the variance of pixel numbers detected between two different continuous steps. If the variance is low, it means that pixels detected in the last adjustment are not much affected by the present adjustment of intensity. And the pixels been detected are possible to be pixels of ink spray because of its steady intensity. Keeping on the calculation of variance of pixels detected, a collection consists of consecutive adjustments of intensity can be constructed. The collection needs to be analyzed since sometimes the variance of pixels been detected is low only because it has low discrimination in a step of adjustment. If so, the collection been record has no use and we can't detect the ink. Thus, if the collection of adjustment contains too many passes of adjustment, we just leave it and go on to analysis the photo image. After the calculation and analysis, a set of

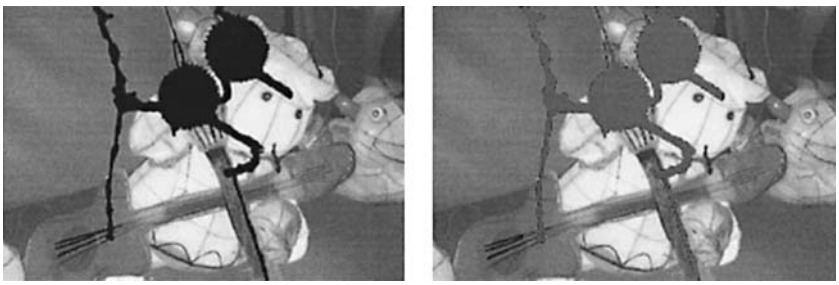


Photo Defect Detection. **Figure 1.** Damages from black ink are detected and shown.

Photo Defect Detection. Figure 2. Damages from ink pen (a) are detected (b) Scratch (c) is detected (d).

adjustment of intensity can be found out to separate pixels of defects from objects in the photo. Figure 1 and 2 illustrates our results.

Cross-References

- ▶ Digital Inpainting
- ▶ Motion Picture Inpainting on Aged Films
- ▶ Multi-Resolution Image Inpainting

Placement of Continuous Media in Ad-Hoc Networks of Devices

Definition

Using the ad-hoc network, we deliver data from the local storage of one or more neighboring devices to reduce the demand for the network infrastructure to remote servers. This mode of delivery requires the devices to collaborate with one another by sharing a fraction of their available storage.

One may enhance availability of a clip by bringing it closer to the device that displays it. To elaborate, in [1], it was noted that the overall bandwidth required to implement an interactive video-on-demand solution based on a naive design that employs one centralized server would be as high as 1.54 Peta bits/s for the entire United States. Using the ad-hoc network, we deliver data from the local storage of one or more neighboring devices to reduce the demand for the network infrastructure to remote servers. This mode of delivery requires the devices to collaborate with one another by sharing a fraction of their available storage. In return, the storage manager provides physical data independence which means the physical organization of data can be modified without causing application programs to be rewritten. It empower authenticated users to stream a clip to any device as long as it has either wired or wireless network connectivity to devices containing the referenced clip. This means the system (instead of the user) resolves the identity of the device that delivers the data requested by the user. Moreover, the system may offer each user a larger amount of storage capacity than that offered by one device. The exact capacity is dictated by the total storage of devices connected in the ad-hoc network and the capacity of remote servers. This storage might be shared by service providers that provide households with on-demand entertainment content.

Bringing clips closer to a user means that an individual who employs others' devices to share his or her experiences may have personal content (typically a fraction of it) pre-staged on many devices in anticipation of future access. This may raise a host of privacy, copyright, and legal issues. While there are techniques for some of these challenges, a significant amount of future research is necessary to ensure privacy of users' personal libraries. We believe the advantages of physical data independence will usher-in a new host of techniques to address these challenges.

Placement of data consists of a collection of techniques to: (1) collect statistics about the environment and how the application references data, (2) place data across devices, (3) re-organize placement of data in response to changes in access profiles and the environment.

Each topic is vast and most research to date has focused on topic number 2. Formally, given a repository of C clips with a pre-specified frequency of access (f_i) for each clip i, a data placement strategy addresses the following questions. First, what is the granularity of placement for data (a block or a clip)? Second, how many replicas of a granule should be constructed in the system? Third, how should these replicas be placed across devices? Answers to these questions are a trade-off between (1) the average startup latency and (2) the number of simultaneous H2O devices that can display clips. These two metrics constitute the dimensions of a recent experimental study used to evaluate three data placement strategies for an ad-hoc network of stationary H2O devices: Simple, Halo-Clip, and Halo-Block [2]. Granularity of data placement is a clip with both Simple and Halo-Clip. It is a block with Halo-Block. Simple employs the profile of a device to pack its storage with its most frequently accessed clips. If the demographics of each device is identical then Simple assigns the same collection of clips to each device. Halo-Clip strives to maintain a replica of the clips that constitute the repository across the devices participating in the ad-hoc network. This is possible when the storage capacity of the participating devices exceeds the repository size. Similar to Halo-Clip, Halo-Block also strives to maintain a copy of every clip in the ad-hoc network. When sufficient storage is available, it replicates the first few blocks of each clip aggressively in order to enhance startup latency.

Mobile Devices

In [3], we investigated how many replicas of different clips should be constructed in a mobile environment that provides on-demand access to audio and video clips. This environment consists of vehicles equipped with a Car-to-Car Peer-to-Peer (C2P2) device that might serve as a component of the vehicle's entertainment system. Mobility is the key difference between a C2P2 and a H2O environment. With C2P2 devices, a vehicular entertainment system offers its user a list of available movie titles during the car's journey. A particular title is available only if sufficient replicas of that title are expected to be encountered in the vicinity of the car to enable successful viewing. However, a title may have a certain time delay after which it is available. We define a related QoS metric for content availability, termed availability latency, defined as the earliest time after which the client vehicle encounters a replica of its referenced title. To minimize this metric, the system may replicate popular clips more aggressively than the less popular clips.

A family of replication techniques to compute the number of replicas for a title as a power law function of its popularity, i.e., frequency of access, is presented in [3]. The exponent value (n) identifies a specific technique. Three distinct exponent values are studied in [3]: random (n = 0), square root (n = 0.5), and linear (n = 1). Availability latency is impacted by a large number of system parameters such as density of C2P2 devices in a geographical area, title display time, size of clip repository, trip duration, the mobility model, storage per C2P2 device and the popularity of the titles. We refer the interested reader to [3] for details.

In [4], we explore the use of mobile C2P2s that carry a referenced data item from a mobile C2P2 containing that data item to a client C2P2 that requested it. Such devices are termed zebroids. A device acts as a zebroid when it is in close vicinity of a server C2P2 and travels along a path that rendezvous with the client C2P2. A key finding of [4] is that zebroids enhance the availability latency of a client with a random mobility model. An investigation of this finding with other mobility models is a future research direction.

Cross-References

▶ Data Management Techniques for Continuous Media in Ad-hoc Networks of Wireless Devices

References

1. J. Nussbaumer, B.V. Patel, F. Schaffa, and J.P.G. Sterbenz, "Networking Requirements for Interactive Video on Demand," IEEE Journal of Selected Areas in Communications, Vol. 13, No. 5, 1995, pp. 779–787.
2. S. Ghandeharizadeh, T. Helmi, T. Jung, and S. Kapadia, "A Comparison of Alternative Data Placement Strategies for Continuous Media in Multi-hop Wireless Networks," Submitted for publication, August 2005.
3. S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari, "Comparison of Replication Strategies for Content Availability in C2P2 Networks," Proceedings of the Sixth International Conference on Mobile Data Management (MDM'05), Ayia Napa, Cyprus, pp. 107–115, May 2005.
4. S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari, "Zebroids: Carrier-based Replacement Policies to Minimize Availability Latency in Vehicular Ad-hoc Networks," Submitted for publication, August 2005.

Portable Network Graphics (Png)

Definition

Portable Network Graphics (PNG) format uses lossless data compression and contains support for device-independent color through gamma correction and the XYZ color model.

The PNG (Portable Network Graphics) [1] format was originally designed to replace the GIF format. PNG, now on version 1.2, is an International Standard (ISO/IEC 15948:2003), also released as a W3C Recommendation on November 10, 2003 [2].

PNG uses lossless data compression and contains support for device-independent color through gamma correction and the XYZ color model. A PNG file consists of an 8-byte signature (89 50 4E 47 0D 0A 1A 0A in hexadecimal) followed by a number of *chunks*, each of which conveys certain information about the image. Chunk types can come from three main sources: the PNG standard, registered public chunk types maintained by the PNG Development Group, and private chunks, defined by some applications. This chunk-based structure is designed to allow the PNG format to be extended while maintaining compatibility with older versions. Chunks follow the format shown in Table 1.

The MIME media type for PNG is image/png. Most current Web browsers support (most features of) the PNG format, paving the way for the PNG format to finally replace GIF for still images.

An extension of PNG, called APNG (Animated Portable Network Graphics), has been proposed to allow for animated PNG files, similar to their animated graphics interchange format (GIF) counterpart, but compatible with non-animated PNG files.

References

1. Official PNG home page, Available online at: http://www.libpng.org/pub/png/ (accessed on April 25, 2005).
2. W3C PNG page, Available online at: http://www.w3.org/Graphics/PNG/ (accessed on April 25, 2005).

Portals

Definition

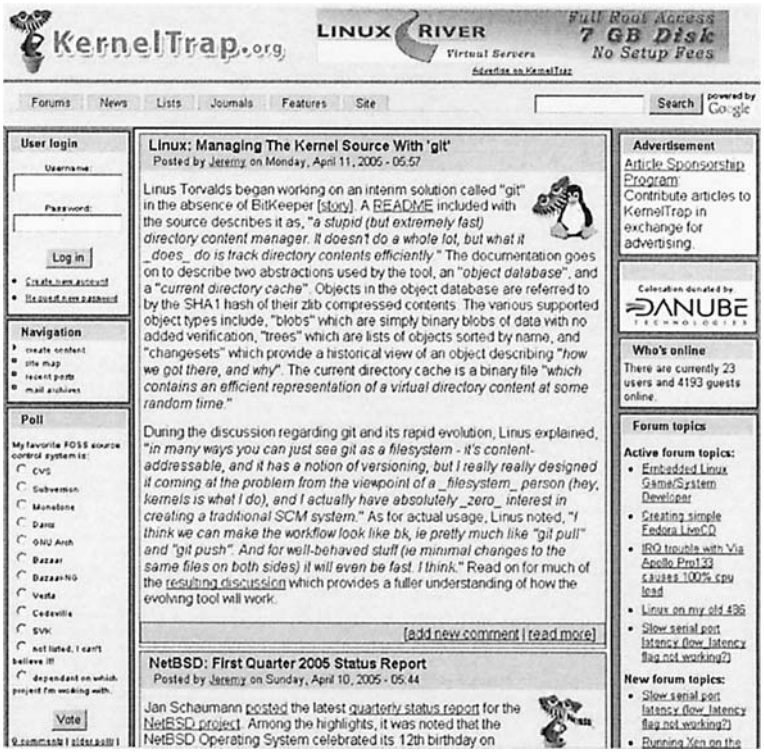
Portals serve as entry points to public and private IP-based networks, including the Internet.

Borrowing from its historical definition as a "grand doorway," the modern technological interpretation of "portal" is as an entry point (or gateway) to a broad array of collaborative network-based resources and services. In the early days of personal computing, portals served as the means of access to bulletin board services (BBS) such as CompuServe, Prodigy and the like. Today, portals have evolved to serve as entry points to public and private IP-based networks, including the Internet. As part of this transition, portals now typically utilize browser technology (rather than proprietary software) to provide a standard Web interface not just to HTML pages, but to various information management, communication and collaborative services. As such, they are an example of the contemporary reality of collaborative computing.

Typical functionality exposed through a portal ranges from accessing information stores (including traditional web sites and databases, as well as document, information and knowledge repositories) to providing wide-ranging search functionality, access to (group) communication tools such as e-mail,

Portable Network Graphics (Png). **Table 1.** PNG chunk fields

Field	Size (in Bytes)	Description
Length	4	Number of bytes in the chunk's *Data* field
Type	4	Chunk name. Each byte of a chunk type is restricted to the decimal values 65–90 and 97–122. These correspond to the uppercase and lowercase ISO 646 letters (A-Z and a-z) respectively for convenience in description and examination of PNG datastreams
Data	*Length*	The data bytes appropriate to the chunk type, if any. This field can be of zero length
CRC	4	A four-byte CRC (Cyclic Redundancy Code) calculated on the preceding bytes in the chunk, including the chunk type field and chunk data fields, but not including the length field. The CRC can be used to check for corruption of the data. The CRC is always present, even for chunks containing no data



Portals. **Figure 1.** Example portal.

chat/instant messaging, audio and video conferencing, discussion and news groups, blogging and so forth. Workflow support, calendaring, voting/polling and on-line meeting systems are also typical functionalities enabled through portal systems. Effectively, they serve as a user-centric access point to the user's "electronic world" – thus necessitating per-user configuration (i.e., profiles supporting custom user interface layout, information service specification, content selection/aggregation, unified logon to the various services, and so forth). Many of the specifics on what functionality is offered and how it is accessed often depends on whether the portal is intended for a private or public audience.

Private, organization-specific portals are typically used to provide a logically centralized access to organizational information, tools and services. Such a common forum provides for increased awareness and faster dissemination of organizational information by providing a common front-end to employees. For example, by providing a familiar interface to the organization through a content management system (vs. shared file systems), issues of scalability and knowledge-management (e.g., search and audit via version control, meta-tagging, classification and categorization) can be addressed. The result is that portals can help stimulate new ways of working, including Communities of Practice.

Conversely, public portals typically offer services of wide-spread interest to large, diverse audiences. These include: Web site directories, access to news, weather, stock quotes, phone and map information, as well as e-mail, chat/instant messaging and community forums according to individual interests. For organizations that provide a public interface to their business process, a portal can supplement traditional means of communicating and interacting with clients; examples include electronic banking, account management for utility companies and e-commerce support for retail outlets. Scientific and engineering organizations, such as ACM, IEEE and numerous others, also use portals both to facilitate their operation as well as their interaction with members and the wider scientific/engineering communities. Examples of portal technology being used in a wide array of specific scientific research projects, ranging from geo-science and on-line biology

labs to astrophysics and grid computing management can be found in [1] (Fig. 1).

In terms of implementation, the technologies and tools used to build portals range significantly in capability and cost. Some common portal technologies are commercial (Sharepoint, Livelink, WebLogic, WebSphere, 10*g*, amongst others) while others are open source (e.g., Plone, Nuke/PHP-Nuke, Drupal, XOOPS, eXo). Most are built using a modular approach in which additional or enhanced functionality can be added to the "portal server" via implementation-specific add-on modules. The end-user generally accesses the portal server using a standard browser interface which typically employs any number of common browser technologies. These include XML, HTML/DHTML, CGI, Java, ASP/JSP, PHP, Python and so forth. Recent efforts towards a more interoperable approach for portal development have lead to an container-based architecture for portal and portal extensions (called portlets) via the JSR 168 portlet specification [2] and WSRP (Web Services for Remote Portlets) [3].

In all cases, by employing "thin client" architecture, portals benefit from the ubiquity of browser technology on virtually every computer system as well as user comfort with the browser interface. Such an approach also promotes acceptance through easier software management (less difficult installation and upgrading) along with potentially fewer security considerations (when compared to custom software). Even so, the specifics of implementation technology can have considerable impact on the portal's acceptability within organizations, as some do not allow downloadable technologies (e.g., applets) to be used in certain situations, often based on network topology and inter-organizational connectivity.

Cross-References

► Collaborative Computing – Area Overview

References

1. GridSphere, "GridSphere Portal Framework," Available online at: http://www.gridsphere.org, 2005.
2. Sun Microsystems, "Introduction to JSR 168: The Java Portlet Specification," Available online at: http://developer.sun.com, 2003.
3. T. Schaeck and R. Thompson, "Web Services for Remote Portlets Whitepaper," Available online at: http://www.oasis-open.org/committees/wsrp, 2003.

Power-Rate Distortion Analysis for Wireless Video

Definition

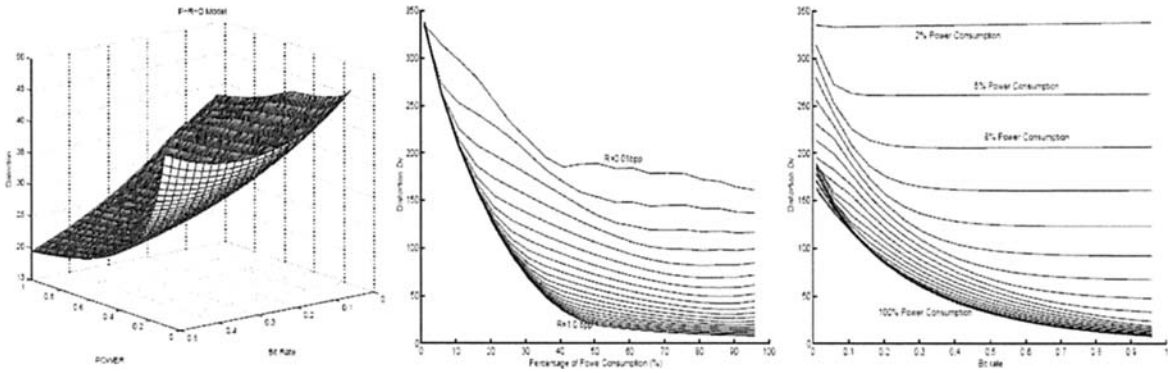
Power-rate distortion analysis is important for wireless video communication applications in energy management, resource allocation, and QoS provisioning, especially over wireless video sensor networks.

To design an energy-scalable standard video encoder, we take three major steps:

1. We group the encoding operations into several modules, such as motion prediction, pre-coding (transform and quantization), and entropy coding, and then introduce a set of control parameters $\Gamma = [\gamma_1, \gamma_2, \ldots \gamma_L]$ to control the power consumption of each module. The encoding power consumption, denoted by P, is then a function of Γ, denoted by $P(\gamma_1, \gamma_2, \ldots \gamma_L)$. The expression of this function also depends on the power consumption model of the specific micro-processor [1,2].
2. We analyze the rate-distortion behavior of each control parameter, and integrate these models into a comprehensive parametric rate-distortion model for the video encoder, denoted by $D(R; \gamma_1, \gamma_2, \ldots \gamma_L)$.
3. We perform optimum configuration of the power control parameters to maximize the video quality (or minimize the video distortion) under the power constraint. This optimization problem can be mathematically formulated as follows:

$$\min_{\{\gamma_1,\gamma_2,\ldots\gamma_L\}} D = D(R;\gamma_1,\gamma_2,\ldots\gamma_L), \quad \text{s.t. } P(\gamma_1,\gamma_2,\ldots\gamma_L) \le P \tag{1}$$

where P is the available power consumption for video encoding. The optimum solution, denoted by $D(R; P)$, describes the P-R-D behavior of the video encoder. To view the P-R-D model in more detail, we plot the D-P curves for different bit rates, ranging from 0.01 to 1.0 bpp in Fig. 1(a). Figure 1(b) shows the D-P curves at different bit rates R_s, and Figure 1(c) shows the D-R curves at different power consumption levels P_s (in percentages of the maximum power consumption level). We can see that when the power supply level is low, the $D(R)$ function is almost flat, which means the video processing and encoding efficiency is very low; hence, in this case, more bandwidth does not improve



Power-Rate Distortion Analysis for Wireless Video. Figure 1. (a) The P-R-D Model; (b) the D-P curves at different bit rates *R* (c) The D-R curves for different power consumption levels.

the video presentation quality. The P-R-D model has direct applications in energy management, resource allocation, and QoS provisioning in wireless video communication, especially over wireless video sensor networks. For detailed energy-scalable encoder design and P-R-D analysis, please refer to [2].

Cross-References

► Wireless Video

References

1. T. Burd and R. Broderson, "Processor Design for Portable Systems," Journal of VLSI Signal Processing, Vol. 13, No. 2, August 1996, pp. 203–222.
2. Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraint," IEEE Transactions on Circuits and System for Video Technology, Vol. 15, No. 5, May 2005, pp. 645–658.

Practical Video Processing Framework

► MPEG-21 Based Video Adaptation with Encryption and Authentication

Presentation Recording

► Privacy and Video Surveillance

Privacy and Video Surveillance

Paula Carrillo, Hari Kalva
Florida Atlantic University, Boca Raton, FL, USA

Synonyms

► Selective encryption; ► Privacy protection

Definition

This article discusses privacy related to anonymous identity in regions of video surveillance.

Introduction

One of the main concerns of the wide use of video surveillance is the loss of individual privacy. Individuals who are not suspects need not be included on camera recordings. The record-everything-and-process-later approach has serious privacy implications. The same privacy issues arise when surveillance cameras routinely record highway traffic and vehicle tags. Mechanisms that protect identity while ensuring legitimate security needs are necessary. Moreover, selectively hiding objects that reveal identity (e.g., faces or vehicle tags) are necessary to preserve individuals' right to privacy; while simultaneously preserving, the general video context. There are many challenges in ensuring privacy protection in video processing. One of them is the complexity of encrypting huge amount of video data. Other challenges appear when applications require real time, total recovery of the hidden objects, compression and/or transcoding for distribution and archiving. Ensuring privacy implies some

form of video encryption. Depending on stage at which encryption is performed the process can be classified as pixel domain, transform domain, or bitstream domain. Figure 1 highlights the types of encryption in a general video encoder.

Techniques Used in Video Privacy Systems

The main techniques used in video privacy systems are summarized below.

Obfuscation

The system presented in [1], a privacy preserving video console, uses a rendering face images technique in the pixel domain and leaves the face unrecognizable by identification software. Based on computer vision techniques, the video console determines the interesting information components of a video and then obscures that piece of information, or its components, such that face recognition software cannot recognize the faces. With this method the privacy is obtained but the surveillance and security needs are not met due to the irreversibility of the obfuscation process. In [2] a medical environment algorithm for automatic patient detection, tracking, labeling and obscuring (the obscuring option in the case the patient does not want to be involved in the research) in real time has been developed. In this particular case, reversibility is not required or desired.

Transform-Domain Coefficient Scrambling

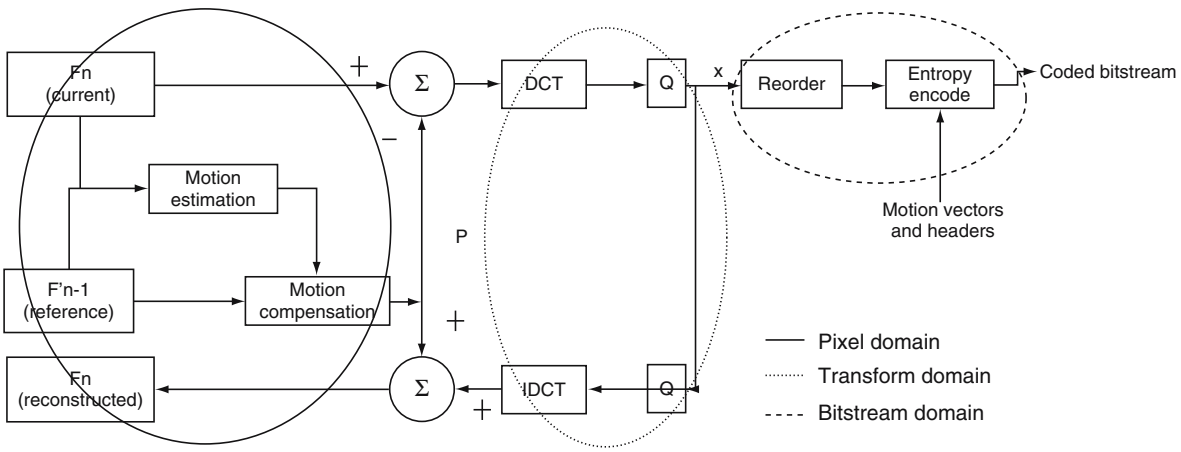
This technique, in the transform domain for JPEG/MPEG video standards was presented in [3]. The region of interest is detected and then the signs of selected transform coefficients are scrambled. More specifically for JPEG2000 Discrete Wavelet Transform (DWT) and for MPEG the Discrete Cosine Transform (DCT) coefficients, corresponding to the regions of interest (ROIs), are scrambled by pseudo-randomly inverting their signs. Consequently, the scene remains understandable, but the ROIs are unidentifiable. The decoded video will have blocky regions unless a proper key is used for de-scrambling. This process is reversible but it is specific to video compression used and cannot survive operations such as transcoding that may be necessary to distribute video.

Invertible Cryptographic Obfuscation

Another technique proposed in [4] is privacy through a cryptographic obfuscation; it uses DES/AES to encrypt regions of JPEG images during the compression stage, before Huffman encoding, in the bitstream domain. This is similar to the transform coefficient sign scrambling. This method also suffers from the same drawbacks: it is compression algorithm specific and cannot survive transcoding.

Cameras Detecting/Replacing Skin

In [5] the approach to privacy protection is based on detecting skin tones in images and replacing it with other colors, hence making it impossible to determine the race of the individual. This process works in the pixel domain (see Fig. 1). Cameras systems based on this method have been developed; the idea is to detect



Privacy and Video Surveillance. Figure 1. Types of encryption in a general video encoder.

the face and then overlay this information with a dark patch or a mosaic or any other obfuscation technique before the video is recorded. At the end no copies of the original faces will exist. This method is compression and transcoding independent. However, specifically in the case of just color replacement, it does not hide the identity completely and since the cameras perform the replacement before recording the video, the method is not reversible. Another issue is that the skin replacement method is applicable only for privacy involving human identity and cannot be used in applications where identity of non-human objects has to be protected; for example, a car's license tag.

ROI in Low Quality Layers

In the privacy system proposed in [6], the authors propose to decrease the ROI quality in JPEG2000, locating this information in the lowest quality layer of the codestream. This ensures poor visual quality in lossy compression, up to invisibility if required. This proposal is in the bitstream domain and hence specific to compression standards used and it is not reversible. Hence, when a suspicious activity needs to be investigated, the identities can not be uncovered to meet the security needs.

Row-Columns Permutation

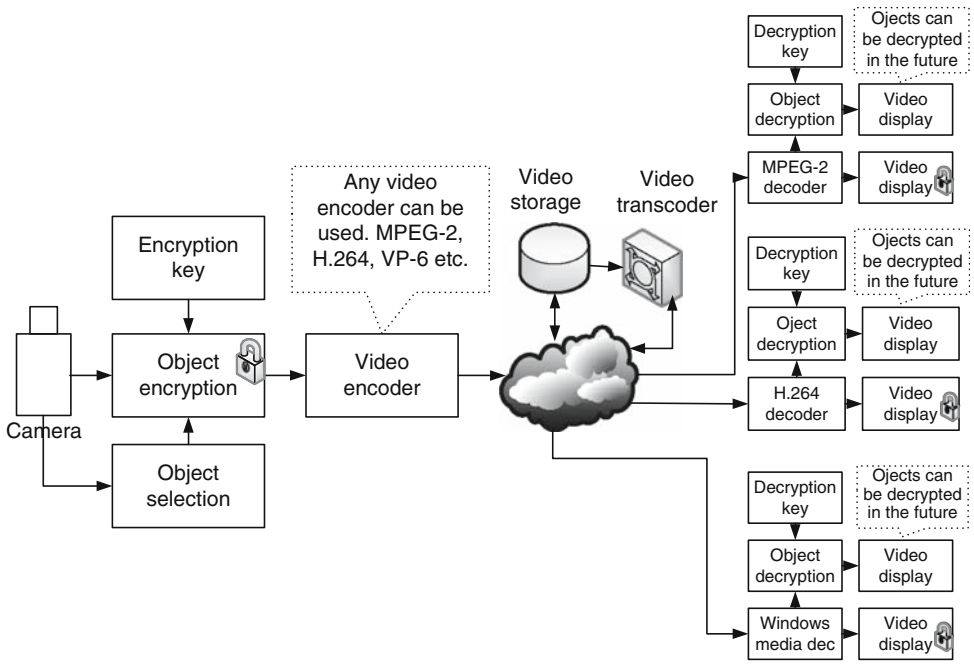
A low complexity system proposed in [7] uses a block based encryption for the regions of interest based on row-columns permutations. These permutations are calculated according to secret keys, in the pixel domain (see Fig. 1), and permit a reversible process if the key is known. This method is independent of the image and video compression algorithms used. This allows the use of standard video encoders and decoders and also enables smart-cameras that output encrypted video. The proposed solution also survives video transcoding and recoding, allowing a normal video distribution chain with multiple video encoding and decoding operations. The innovation in the proposed approach is the use of permutation based encryption that can survive lossy compression.

Figure 2 shows snapshots of the face encryption using row-column permutations algorithms. Under extreme compression, the decoded and decrypted video looks blurry and leaves the video without the possibility of recovering the encrypted regions. For that reason, in this work, the authors implemented a high quality option for the regions of interest, see Fig. 3. Thus, the system ensures reliability and recovery for any level of compression condition.

A more general architecture for a reversible privacy video surveillance system would consist of an early stage video object detection and tracking. Then, the system hides the selected regions of video with obfuscation, scrambling, replacing skin or encryption (the encryption is possible not only in the pixel domain, but also in the transform domain or video layers). Finally, the video is compressed/transcoded for storing and transmission purpose. Later, in the decoding stage, there are two options depending on the user rights. If the user has the rights and the correct keys, a complete

Privacy and Video Surveillance. **Figure 2.** Original frame (left), and the same frame with encrypted faces (right).

Privacy and Video Surveillance. **Figure 3.** Decoded and decrypted frame under extreme compression (left), and the same decoded and decrypted frame with a fixed ROI high quality (right).



Privacy and Video Surveillance. **Figure 4.** General system architecture of a video surveillance system.

video will be shown. If not, a video with distorted information will be shown. This general scheme is shown in the Fig. 4.

In general, non-reversible methods permit privacy but not security in surveillance. However, for a complete transmission chain, methods should permit different kinds of encoding and transcoding. For a complete and general scheme of privacy and security, the system should be reversible, object independent (humans and things), and coding/transcoding independent. A comparison of the privacy protection schemes is shown in Table 1.

Privacy and Video Surveillance. Table 1. Comparison of privacy protection schemes

Video selective encryption method	Privacy completeness	Reversibility	Transcoded capability	Domain	General bit-rate increases
Obscuration	Yes	No	No	Pixel	No
Transform-domain scrambling coefficients	Yes	Yes	No	Transform	Yes
Invertible cryptographic obscuration	Yes	Yes	No	Bit-stream	Yes
Cameras detecting/replacing skin	Not always	No	Yes	Pixel	No
ROI in Low quality layers	Yes	No	No	Bit-stream	No
Row-column permutation	Yes	Yes	Yes	Pixel	Yes

References

1. Senior, et al. "Enabling Video Privacy Through Computer Vision," IEEE Security and Privacy Magazine, Vol. 3, No. 3, May–June 2005, pp. 50–57.
2. I. Martínez-Ponte, X. Desurmont, J. Meessen, and J. Delaigle, "Robust Human Face Hiding Ensuring Privacy," Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (WIAMIS'05), Montreux, Switzerland, April 2005.
3. F. Dufaux and T. Ebrahimi, "Scrambling for Video Surveillance with Privacy," Conference on Computer Vision and Pattern Recognition Workshop, pp. 160–160, June 2006.
4. T.E. Boult, "PICO: Privacy through Invertible Cryptographic Obscuration," Proceedings of the Computer Vision for Interactive and Intelligent Environment, pp. 27–38, Nov. 2005.
5. M. Berger, "Privacy Mode for Acquisition Cameras and Camcorders," US. Patent 6,067,399 May 23, 2000.
6. D. Chen, Y. Chang, R. Yan, and J. Yan, "Tools for Protecting the Privacy of Specific Individuals in Video," Eurosip Journal in Advances in Signal Processing, Vol. 2007, 2007, Art ID75427.
7. P. Carrillo, H. Kalva, and S. Magliveras, "Compression Independent Objects Encryption for Ensuring Privacy In Video Surveillance," ICME, 2008.

Privacy Protection

► Privacy and Video Surveillance

Private-Key Cryptosystem

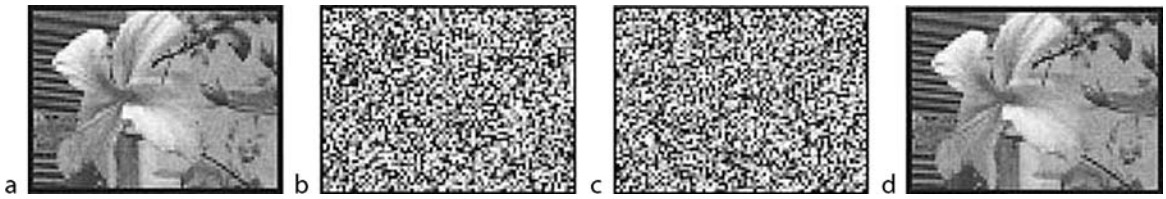
Definition

A private-key cryptosystem can be obtained using the visual cryptography concepts or perfect-reconstruction based image secret sharing.

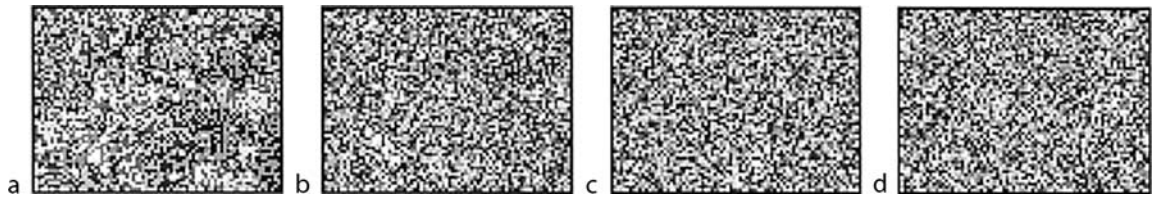
A (2, 2)-scheme is the most popular solution within the (*k, n*) framework due to its common acceptance as a private-key cryptosystem [1–3]. Such a cryptographic solution encrypts the secret image into two noise-like shares. One of the two generated shares can be viewed as a private share or private-key, and is kept by the owner. The other share represents a public share which can be transmitted over an untrusted communication channel. The secret image is reconstructed only if both the public and private shares are used for decryption.

A private-key cryptosystem can be obtained using the visual cryptography concepts [1] or perfect-reconstruction based image secret sharing [2]. Both these solutions encrypt each pixel of the secret image into a block of share pixels and because of their expansion nature, such solutions produce the shares with the enlarged spatial resolution. To reduce the complexity, a cost-effective private-key cryptosystem in [3] uses pixel operations to both encrypt and decrypt the images. As a result, the solution produces the shares with spatial dimensions identical to those of the secret image (Fig. 1). In addition, operating on the bit-levels of the processed images, the solution satisfies the essential perfect reconstruction property. Thus, similarly to the (*k, n* image secret sharing framework proposed in [2]), the private-key cryptosystem in [3] recovers the original secret image (Fig. 1).

The utilization of the bit-level processing operations in [2,3] allows for selective encryption of image bit-planes. Such an approach offers solutions which differ in their security characteristics. For example, as it is shown in Fig. 2, sufficient protection is usually



Private-Key Cryptosystem. **Figure 1.** A cost-effective private-key cryptosystem: (a) secret gray-scale image, (b,c) gray-scale shares, (d) secret image decrypted using the share images shown in (b,c).



Private-Key Cryptosystem. **Figure 2.** A public share obtained by encrypting: (a) the most significant bit only, and (b) two, (c) three, (d) four most significant bits of the secret image.

obtained when the two or three most significant bits of the secret image's pixels are encrypted.

Cross-References

- ▶ Compression in Image Secret Sharing
- ▶ Image Secret Sharing
- ▶ Threshold Schemes with Minimum Pixel Expansion
- ▶ Visual Cryptography

References

1. G. Ateniese, C. Blundo, A. de Santis, and D.-G. Stinson, "Visual Cryptography for General Access Structures," Information and Computation, Vol. 129, No. 2, September 1996, pp. 86–106.
2. R. Lukac and K.-N. Plataniotis, "Bit-Level Based Secret Sharing for Image Encryption," Pattern Recognition, Vol. 38, No. 5, May 2005, pp. 767–772.
3. R. Lukac and K.-N. Plataniotis, "A Cost-Effective Private-Key Cryptosystem for Color Image Encryption," Lecture Notes in Computer Science, Vol. 3514, May 2005, pp. 679–686.

Profiling MPEG-7

- ▶ Multimedia Metadata Profiles

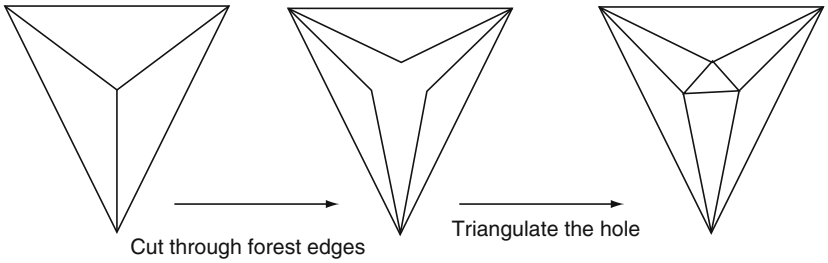
Progressive Forest Split

Definition

Progressive Forest Split is an efficient encoding technique for a simple polygon; it groups the decimations into a batch to achieve a high compression ratio.

PFS (Progressive Forest Split) [1] is much more efficient in encoding at the expense of looser granularity. Similar to CPM (Compressed Progressive Meshes) [2], it groups the decimations into a batch to achieve a high compression ratio. PFS cuts the mesh through the forest edges, triangulates each tree boundary loop, and displaces vertices to new positions. The geometric data contains the error between the predicted and the real vertex positions. The amortized connectivity encoding takes 10 bits and geometry encoding takes 30 bits per vertex. MPEG-4 accepts PFS as the standard compression scheme. However, PFS is not widely implemented in current 3D players [3].

PFS provides an efficient encoding for a simple polygon (triangulated with no internal vertices). For each refinement, at compression, certain simple polygons are selected for removal. Each simple polygon is simplified to a set of connected edges. All these edge sets form a forest. At decompression, the decoder cuts through the edges in the forest and fills in the hole with the encoded corresponding simple polygon. Figure 1



Progressive Forest Split. Figure 1. PFS cuts through forest edges and fills in the hole with a simple polygon.

shows the simplification/triangulation of a simply polygon with four triangles.

Cross-References

▶ Middleware for Streaming 3D Meshes

References

1. G. Taubin, A. Gueziec, W. Horn, and F. Lazarus, "Progressive Forest Split Compression," Proceedings of the 25th Annual Conference on Computer Graphics, SIGGRAPH 98, Orlando, FL, pp. 123–132, 1998.
2. R. Pajarola and J. Rossignac, "Compressed Progressive Meshes," IEEE Transactions on Visualization and Computer Graphics, Vol. 6, No. 1, 2000, pp. 79–93.
3. M. Isenburg and J. Snoeylink, "Coding polygon meshes as compressible ASCII," Proceedings of the Seventh International Conference on 3D Web Technology (Web 3D), Tempe, AZ, pp. 1–10, 2002.

Protection of Multimedia Data in Distribution and Storage

Ahmet M. Eskicioglu
Brooklyn College, New York, NY, USA

Definition

Multimedia data needs to be protected from unauthorized duplication and consumption, from unauthorized disclosure and misuse, and from unauthorized use and exploitation.

Introduction

Multimedia can be defined as a combination of different types of media (e.g., text, images, audio, video, and graphics) to communicate information in a given application.

Recent advances in digital technologies have drastically increased the capacity of both data channels and storage:

1. When compare with floppy disks and CDs, the capacity of DVDs, digital tapes, and hard disk is much larger. Personal computers can be configured with a processor speed of 3 GHz, main memory of 2 GB, and a hard disk of 500 GB. DVDs manufactured with a double sided format, each side having a dual layer, have a data capacity of 17 GB. Digital Linear Tapes (DTLs) come with a storage space above 200 GB for compressed data.
2. Asymmetric digital subscriber line (ADSL) is a new technology that allows more data to be sent over existing copper telephone lines, supporting data rates up to 9 Mbps when receiving downstream data. Very High Speed Digital Subscriber Line (VDSL) transmits data in the 13–55 Mbps range over short distances, usually between 1000 and 4500 feet.

The transition from analog to digital technologies started in 1990s. With the higher capacity of storage devices and data communication channels, multimedia content has become a part of our daily lives. This type of data is now commonly used in many areas such as education, entertainment, journalism, law enforcement, finance, health services, and national defense. The lowered cost of reproduction, storage, and distribution has added an additional dimension to the complexity of the problem. In a number of applications, multimedia needs to be protected for several reasons. Table 1 includes three applications where the data should be protected.

Encryption and watermarking are two groups of complementary technologies that have been identified by content providers to protect multimedia data [1–3]. Watermark embedding and detection are

Protection of Multimedia Data in Distribution and Storage. Table 1. Protection of data in three applications

Application	Provider	Data	What needs to be prevented?
Entertainment	Content owners (e.g., movie studios and recording companies) and service providers (e.g., cable companies and broadcasters)	Copyrighted movies and songs	Unauthorized duplication and consumption
Health services	Hospitals	Medical data for patients (e.g., X-ray pictures and history of illnesses)	Unauthorized disclosure and misuse
Finance	Investment bankers	Financial data (stocks and mutual funds)	Unauthorized use and exploitation

sometimes considered to be analogous to encryption and decryption [4].

1. *Encryption* makes the content unintelligible through a reversible mathematical transformation based on a secret key [5,6]. In secure multimedia content distribution, the audio/visual stream is compressed, packetized, and encrypted. In symmetric key encryption, which is commonly used for protecting multimedia elements, each encryption transformation E_K is defined by an encryption algorithm E and a key K. Given a plaintext M, the transformation produces the ciphertext $C = E_K(M)$. Each decryption transformation D_K is defined with a decryption algorithm D and K. For a given K, $D_K = E_K^{-1}$ such that $D_K(E_K(M)) = M$. One of the most challenging problems in distribution architectures is the delivery of the decryption key.
2. *Watermarking (data hiding)* [7] is the process of embedding data into a multimedia element such as image, audio or video. The embedding transformation E_K is defined by an embedding algorithm E and a key K. In watermarking, the usual approach is to use a symmetric key although there is a recent trend to use asymmetric techniques. Given a cover image I and a watermark W, the transformation produces the watermarked image $I_W = E_K(I,W)$. Each detection (or extraction) transformation D_K is defined with a detection (or extraction) algorithm D and K. For a given K and the watermarked image I_W, the watermark is either detected (or extracted): $W = D_K(I_W)$.

Encryption

Figure 1 shows five primary means of multimedia delivery to consumers: satellite, cable, terrestrial, Internet and prerecorded media (optical and magnetic).

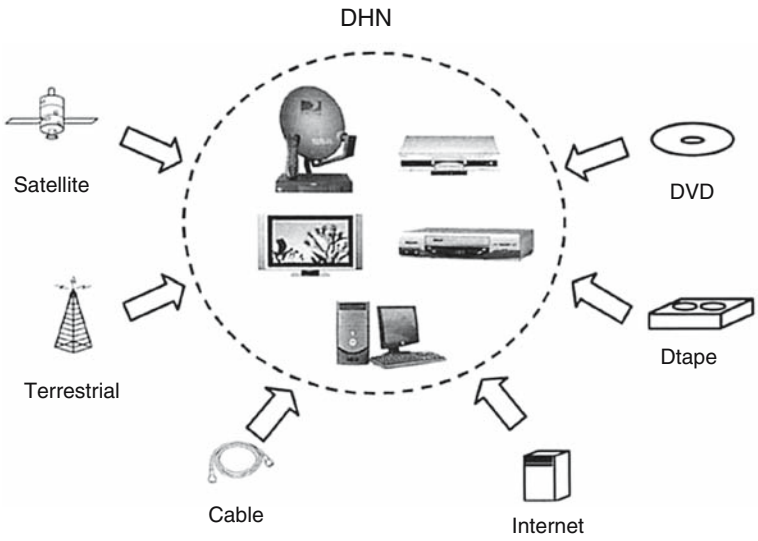
For end-to-end security from the source to the final destination, the most important requirements are:

1. Secure distribution of multimedia content
2. Secure distribution of access keys
3. Authentication of source and sink consumer devices in home networks
4. Association of digital rights with content
5. Manufacturing of licensed devices that have the protection technology
6. Renewability of secure solutions

In the last 10 years, three industries (consumer electronics, information technology, and motion picture) have been working on solutions for protecting copyrighted multimedia content. Some of the key players with interest in developing or implementing secure solutions are ATSC (Advanced Television Systems Committee), CableLabs, CPTWG (Copy Protection Technical Working Group), DVB (Digital Video Broadcasting) Organization, DVD Forum, EIA (Electronics Industries Association), IETF (Internet Engineering Task Force), MPAA (Motion Picture Association of America), MPEG (Moving Pictures Expert Group), North American Broadcasters Association (NABA), RIAA (Recording Industries Association of America), and SCTE (Society of Cable Television Engineers).

In digital distribution networks, copyrighted multimedia content is commonly protected by encryption:

1. Cable, satellite, and terrestrial distribution [8–10]: A conditional access (CA) system provides the encryption technology to control access to digital television services. Digital content ("program") is compressed, packetized, encrypted and multiplexed with the entitlement messages. Two types



Protection of Multimedia Data in Distribution and Storage. **Figure 1.** Multimedia distribution systems.

of entitlement messages are commonly used associated with each program: the Entitlement Control Messages (ECMs) and the Entitlement Management Messages (EMMs). ECMs carry the decryption keys ("control words") and a short description of the program while EMMs specify the authorization levels related to services. The programs are usually encrypted using a symmetric cipher such as the Data Encryption Standard (DES) or any other public domain or private cipher. The CA providers often protect the ECMs privately although public-key cryptography and one-way functions are useful tools for protecting access keys. Authorized users can use the appropriate decoder to decrypt the programs. Because of their secure features, smart cards are a good option for set-top boxes.

2. Internet distribution [10–12]: Digital Rights Management (DRM) refers to the protection, distribution, modification, and enforcement of the rights associated with the use of digital content. The primary responsibilities of a DRM system include secure delivery of content, prevention of unauthorized access, enforcement of usage rules, and monitoring of the use of content. A customer obtains an encrypted file from a server on the Internet for viewing purposes. To be able to decrypt the file, a license (that contains the usage rights and the decryption key) needs to be downloaded from a clearing house. A major responsibility of the clearing house is to authenticate the customer based on his credentials. The client device should have a player that supports the relevant DRM system to play the file according to the rights included in the license. Superdistribution is a process that allows a customer to send the encrypted file to other people. However, as licenses are not transferable, each new customer has to purchase another license for playback. Today, interoperability of DRM systems is a major problem.
3. Distribution in digital home networks [10,13]: A digital home networks is a cluster of consumer electronics devices (e.g., DTV, DVD player, DVCR, and STB) that are interconnected. The multimedia content is encrypted in transmission across each digital interface, and on storage media. The technical solutions developed in recent years are listed in Table 2. In a digital home network, multimedia content moves from one device to another for storage or display. These devices need to authenticate each other to make sure that they are equipped with the licensed protection technology.

Watermarking

A digital watermark is a pattern of bits inserted into a multimedia element such a digital image, an audio or video file. The name comes from the barely visible text or graphics imprinted on stationery that identifies the manufacturer of the stationery. There are several proposed or actual watermarking applications [4]: broadcast monitoring, owner identification, proof of ownership, transaction tracking, content authentication, copy control, and device control. In particular,

watermarks appear to be useful in plugging the analog hole in consumer electronics devices [21].

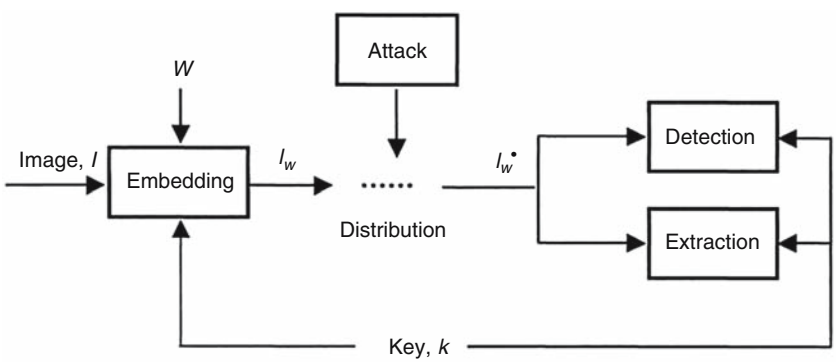
The components of a watermark embedding/detection/extraction system are depicted in Fig. 2. A watermarking system consists of watermark structure, a marking algorithm that inserts some data into multimedia and an extraction or detection algorithm that extracts the data from, or detects the data in, a multimedia element.

In applications such as owner identification, copy control, and device control, the most important properties of a watermarking system are perceptual transparency, robustness, security, high data capacity, and unambiguousness. The relative importance of these properties depends on the requirements of a given application.

1. *Perceptual transparency:* An embedded watermark should not introduce a significant degree of distortion in the cover image. The perceived degradation of the watermarked image should be imperceptible.
2. *Robustness*: Robustness refers to the ability to detect the watermark after normal A/V processes or intentional attacks. A watermark can still be detected or extracted after the image has undergone some common signal processing operations. These operations include special filtering, lossy compression, printing/scanning, and geometric distortions such as rotation, translation, cropping, and scaling.
3. *Security:* Security is the ability to resist unauthorized removal, embedding, or extraction. A hostile attack is any process specifically intended to thwart the watermark's purpose.
4. *Capacity:* Data capacity can be defined as the amount of data that can be embedded. A watermarking system should be able to embed relatively high amount of data without affecting perceptual transparency.

Protection of Multimedia Data in Distribution and Storage. **Table 2.** Content protection solutions for digital home networks

Media	Solution	What is protected?
Optical media	CSS [14]	Video on DVD-ROM
	CPPM [15]	Audio on DVD-ROM
	CPRM [16]	Video or audio on DVD-R/RW/RAM
	4C/Verance Watermark [17]	Audio on DVD-ROM
	To be determined.	Video on DVD-ROM/R/RW/RAM
Magnetic media	High Definition Copy Protection (HDCP) [18]	Video on digital tape
Digital interfaces	DTCP [19]	IEEE 1394 serial bus
	HDCP [20]	Digital Visual Interface (DVI)



Protection of Multimedia Data in Distribution and Storage. **Figure 2.** Watermarking system.

Protection of Multimedia Data in Distribution and Storage. Table 3. Classification of image watermarking systems

Criterion	Class	Brief description
Domain type	Pixel [22]	Pixels values are modified to embed the watermark.
	Transform [23]	Transform coefficients are modified to embed the watermark. Recent popular transforms are Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT).
Watermark Type	Pseudo random number (PRN) sequence (having a normal distribution with zero mean and unity variance) [24]	Allows the detector to statistically check the presence or absence of a watermark. A PRN sequence is generated by feeding the generator with a secret seed.
	Visual watermark [25]	The watermark is actually reconstructed, and its visual quality is evaluated.
Scheme type	Reversible [26]	Exact restoration of the original unwatermarked image is possible
	Irreversible [27]	The distortion in the watermarked image is small but irreversible.
Information type	Non-blind [28]	Both the original image and the secret key(s)
	Semi-blind [29]	The watermark and the secret key(s)
	Blind [30]	Only the secret key(s)
Algorithm type	Additive Algorithm [31]	Additive algorithm performs linear modification of the host image and the correlative processing in the detection process.
	Quantization Algorithm [32]	Quantization algorithm performs nonlinear modification of the host image and quantizing the received samples to map in the detection process.

5. *Unambiguousness:* The watermark should unambiguously identify the owner. It is desired that the difference between the extracted and the original watermark is as low as possible. For accuracy of identification, the system should exhibit a graceful degradation irrespective of the type of attack.

Several criteria can be used to classify image watermarking systems. Five of such criteria are the type of watermark, the type of domain, the type of watermarking scheme, type of algorithm, and the type of information needed in the detection or extraction process. The classification according to these criteria is listed in Table 3. In general, systems that embed the watermark in the pixel domain are simpler but are less robust to image manipulations. On the other hand, frequency domain watermarking techniques are more complex and robust.

Embedding multiple watermarks in a transform domain using the coefficients in several frequency bands drastically increases the overall robustness of a watermarking scheme [33–35]. For one group of attacks, detection or extraction in lower frequencies is better, and for another group of attacks, detection or extraction in higher frequencies is better. Since the advantages and disadvantages of low and middle-to-high frequency watermarks are complementary, embedding multiple watermarks in an image (namely, one in lower frequencies and the other in higher frequencies) would result in a scheme that is highly robust with respect to a large spectrum of image processing operations.

References

1. M. Eskicioglu and E.J. Delp, "Overview of Multimedia Content Protection in Consumer Electronics Devices," Signal Processing: Image Communication, Vol. 16, No. 7, April 2001, pp. 681–699.
2. M. Eskicioglu, J. Town, and E.J. Delp, "Security of Digital Entertainment Content from Creation to Consumption," Signal Processing: Image Communication, Special Issue on Image Security, Vol. 18, No. 4, April 2003, pp. 237–262.

3. E.T. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp, "Advances in Digital Video Content Protection," Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery, Vol. 93, No. 1, 2004, pp. 171–183.
4. I.J. Cox, M.L. Miller, and J.A. Bloom, "Digital Watermarking," Morgan Kaufmann, San Fransisco, CA, 2002.
5. J. Menezes, P.C. van Oorschot, and S.A. Vanstone, "Handbook of Applied Cryptography," CRC Press, Boca Raton, FL, 1997.
6. B. Schneier, "Applied Cryptography," Wiley, New York, 1996.
7. M. Arnold, M. Schmucker, and S.D. Wolthusen, "Techniques and Applications of Digital Watermarking and Content Protection," Artech House, Norwood, MA, 2003.
8. R. de Bruin and J. Smits, "Digital Video Broadcasting: Technology, Standards and Regulations," Artech House, Norwood, MA, 1999.
9. W. Mooij, "Advances in Conditional Access Technology," International Broadcasting Convention, IEEE Conference Publication, No. 447, September 1997, pp. 461–464.
10. A.M. Eskicioglu, J. Town, and E.J. Delp, "Security of Digital Entertainment Content from Creation to Consumption," Signal Processing: Image Communication, Special Issue on Image Security, Vol. 18, No. 4, April 2003, pp. 237–262.
11. Microsoft Windows Media DRM, Available online at: http://www.microsoft.com/windows/windowsmedia/drm.aspx.
12. Helix DRM, Available online at: http://www.realnetworks.com/products/drm/index.htm.
13. Content Protection System Architecture (CPSA), Available online at: http://www.4Centity.com.
14. Content Scramble System, Available online at: http://www.dvdcca.org.
15. Content Protection for Prerecorded Media, Available online at: http://www.4Centity.com.
16. Content Protection for Recordable Media, Available online at: http://www.4Centity.com.
17. 4C/Verance Watermark, Available online at: http://www.verance.com.
18. High Definition Copy Protection, Available online at: http://www.jvc-victor.co.jp/english/products/vcr/D-security.html.
19. Digital Transmission Content Protection, Available online at: http://www.dtcp.com.
20. High-bandwidth Digital Content Protection, Available online at: http://www.digital-CP.com.
21. Content Protection Status Report III, November 7, 2002, Available online at http://judiciary.senate.gov/special/mpaa110702.pdf.
22. W. Bender, D. Gruhl, and N. Morimoto, "Techniques for Data Hiding," IBM Systems Journal, Vol. 35, No. 3–4, 1996, pp. 313–336.
23. Y. Wang, J.F. Doherty, R.E. Van Dyck, "A Wavelet-based Watermarking Algorithm for Ownership Verification of Digital Images," IEEE Transactions on Image Processing, Vol. 11, No. 2, February 2002.
24. W. Zhu, Z. Xiong, and Y.-Q. Zhang, "Multiresolution Watermarking for Images and Video," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 4, June 1999, pp. 545–550.
25. R. Liu and T. Tan, "A SVD-Based Watermarking Scheme for Protecting Rightful Ownership," IEEE Transactions on Multimedia, Vol. 4, No. 1, March 2002, pp.121–128.
26. J. Fridrich, M. Goljan, and R. Du, "Lossless Data Embedding-New Paradigm in Digital Watermarking," EURASIP Journal on Applied Signal Processing, Special Issue on Emerging Applications of Multimedia Data Hiding, Vol. 2002, No. 2 February 2002, pp. 185–196.
27. D. Kundur and D. Hatzinakos, "Toward Robust Logo Watermarking Using Multiresolution Image Fusion Principles," IEEE Transactions on Multimedia, Vol. 6, No. 1, February 2004.
28. I.J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," IEEE Transactions on Image Processing, Vol. 6, No. 12, December 1997, pp. 1673–1687.
29. R. Dugad, K. Ratakonda, and N. Ahuja, "A New Wavelet-Based Scheme for Watermarking Images," Proceedings of 1998 International Conference on Image Processing (ICIP 1998), Chicago, IL, Vol. 2, pp. 419–423, 1998.
30. S. Pereira and T. Pun, "Robust Template Matching for Affine Resistant Image Watermarks," IEEE Transactions on Image Processing, Vol. 9, No. 6, June 2000, pp. 1123–1129.
31. M.L. Miller, G.J. Doerr, I.J. Cox, "Applying Informed Coding and Embedding to Design a Robust, High Capacity Watermark," IEEE Transactions on Image Processing, Vol. 13, No. 6, June 2004, pp. 792–807.
32. B. Chen and G.W. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," IEEE Transactions on Information Theory, Vol. 47, No. 4, 1999, 1423–1443.
33. R. Mehul and R. Priti, "Discrete Wavelet Transform Based Multiple Watermarking Scheme," Proceedings of IEEE Region 10 Technical Conference on Convergent Technologies for the Asia-Pacific, Bangalore, India, October 2003.
34. P. Tao and A. M. Eskicioglu, "A Robust Multiple Watermarking Scheme in the Discrete Wavelet Transform Domain," Proceedings of Optics East 2004 Symposium, Internet Multimedia Management Systems V Conference, Philadelphia, PA, October 2004.
35. E. Ganic, S. D. Dexter, and A. M. Eskicioglu, "Embedding Multiple Watermarks in the DFT Domain Using Low and High Frequency Bands," Proceedings of IS&T/SPIE's 17th Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII Conference, San Jose, CA, January 2005.

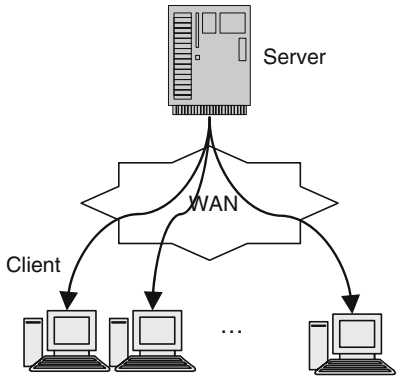
Proxy-Caching for Video Streaming Applications

Definition

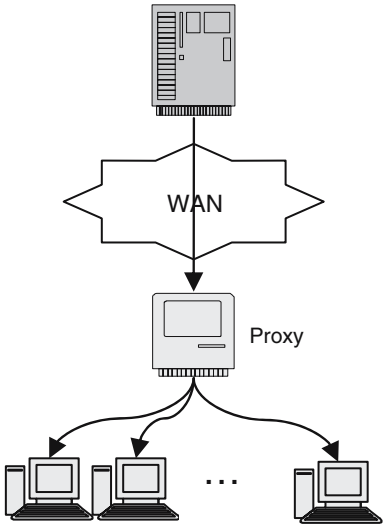
Proxy-caching is a mechanism that can be used to leverage the workload of the central server. To accomplish this operation, an intermediate device called proxy is normally placed between the central server and clients.

A VoD system, which provides service for users, typically consists of two main components: the central server and clients (Fig. 1). The central server has a large storage space to store all the available videos for clients connected via a wide area network (WAN) or a local area network (LAN). In such a framework, all the requests from clients are handled at the central server. The request process starts with generating a request message from clients to the central server. In response to the client's request, the central server serves each request individually with a dedicated channel. This operation is simple to implement. However, this architecture is excessively expensive and nonscalable because the bandwidth bottleneck of the central server limits the number of clients it can serve. Furthermore, the introduction of long service latencies is another critical factor affecting the system performance, which is especially significant when the video is transmitted over the WAN.

To leverage the workload of the central server and reduce the service latencies, an intermediate device called proxy is placed between the central server and clients (Fig. 2). In the proxy-based architecture, a portion of video is cached in the proxy. The request generated by a client is served by the proxy if it has a cached portion of the requested video. Meanwhile, the central server also delivers the uncached portion of the video to the client directly. Existing caching mechanisms can be mainly classified into four categories [1]: sliding-interval caching, prefix caching, segment caching, and rate-split caching. Sliding-interval [2] caches the playback interval between two requests. Prefix caching [3] divides the video into two parts named prefix and suffix. Prefix is the leading portion of the video which is cached in the proxy, while the suffix is the rest of the video which is stored in the central server. Upon receiving a client's request, the proxy delivers the prefix to the client, meanwhile, it also downloads the suffix from the central server and then relays to the client. Segment caching [4] generalizes the prefix caching by partitioning a video object into a number of segments. The proxy caches one or several segments based on the caching decision algorithm. In rate-split [5], the central server stores the video frame with the data rate, which is less than a threshold called cutoff rate. If the data rate of the video frame is higher than the cutoff rate, it is partitioned into two parts where the cutoff is the boundary such that the transmission rate of the central server can keep constant. Content distribution network (CDN) is an extension of the proxy caching. In such architecture, a number of CDN servers are deployed at the edge of the network core. Unlike proxy, which only stores a portion of the video, a full copy of the video is replicated in each CDN server. Then, clients request the video from their closest CDN servers directly. This architecture significantly reduces the workload of the central server and provides a better quality of service (QoS) to clients.



Proxy-Caching for Video Streaming Applications.
Figure 1. Architecture of a typical VoD system.



Proxy-Caching for Video Streaming Applications.
Figure 2. Architecture of VoD system with proxy caching.

Cross-References

► Large Scale Multimedia Streaming in Heterogeneous Network Environments

References

1. J.C. Liu and J.L. Xu, "Proxy caching for media streaming over the Internet," IEEE Communications Magazine, Vol. 42, No. 8, Aug. 2004, pp. 88–94.
2. R. Tewari et al., "Resource-Based Caching for Web Servers," Proceedings of the MMCN'98, San Jose, CA, Jan. 1998.
3. S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," Proceedings of the IEEE INFOCOM'99, New York, NY, Mar. 1999.
4. S. Chen et al., "Designs of High Quality Streaming Proxy Systems," Proceedings of the IEEE INFOCOM'04, Hong Kong, Chain, Mar., 2004.
5. Z.L. Zhang et al., "Video staging: a proxy-server-based approach to end-to-end video delivery over wide-area networks," IEEE/ACM Trans. Net., Vol. 8, No. 9, 2000, pp. 429–442.

Pseudo-Color Image Processing

► Color-Mapped Imaging

Public Key Versus Secret Key Encryption

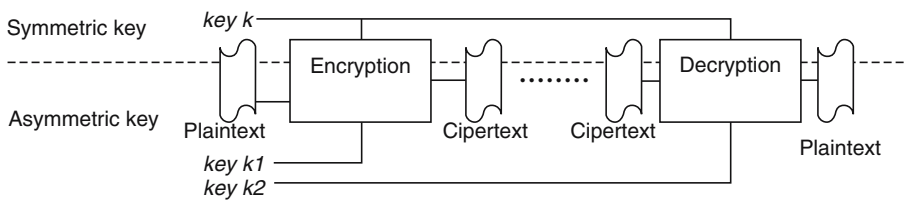
Definition

The purpose of a key in encryption systems is to ensure privacy by keeping information hidden from whom it is not intended. There are two types of encryption systems: secret-key and public-key systems.

Encryption is the transformation of data, the *plaintext*, into a form that is as close to impossible as possible to read, the *ciphetext*, without the appropriate knowledge (a key). Its purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, even those who have access to the encrypted data, the *ciphetext* [1].

There are two types of encryption systems: *secret-key* and *public-key* systems (see Fig. 1). In secret-key encryption, also referred to as symmetric cryptography, the same key is used for both encryption and decryption. The most popular secret-key cryptosystem in use today is the *Data Encryption Standard* (*DES*, see [1–3]). In general, the security of a secret-key system depends on the strength of the algorithm and the length of the key. In a brute-force attack, an N bits long key, which implies 2^N possible keys, will take 2^N attempts for the hacker to find the correct key. That is, the longer the key is, the longer it will take for a certain computer to find the correct key among the 2^N possible keys. Besides the key, there is another factor to determine the interval of a successful brute force attack: the speed of each test which relies on the speed of the computer and the computational complexity of the encryption algorithm. Unfortunately in most cases, the longer the key is, the longer it takes to encrypt/decrypt the message, and the more one need to pay for it in terms of cost. To determine the length of a proper key for a certain application, one needs to look at several aspects: the intended security, the current and future computer power, and the speed of the state-of-the-art factoring algorithms. That is, you need a key long enough to be secure but short enough to be computationally feasible and low cost.

On the other hand, in a public-key system, each user has a *public key* and a *private key*. The public key maybe made public while the private key remains secret. Encryption is performed with the one key while decryption is done with the other. Today's dominant public key encryption algorithms are factorization based [2]. That is, the algorithms are based on the difficulty of factoring large numbers that are the product of two large prime numbers. One of the most popular public-key encryption algorithms, the *RSA public-key cryptosystem*, (see [1–3]) is a typical



Public Key Versus Secret Key Encryption. Figure 1. Illustration of symmetric key encryption versus asymmetric key encryption.

factorization based algorithm. To break such systems, one need to factor out the large number. Intuitively, the large the number is, the harder it is for a hacker to factor it.

Cross-References

► Multimedia Encryption

References

1. RSA Laboratories, "RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1," RSA Security Inc., 2000.
2. B. Schneier, "Applied Cryptography," 2nd Ed., Wiley, New York, 1996.
3. R.K. Nichols, "ICSA guide to cryptography," McGraw-Hill, New York, 1999.