Identification of Objects and Retrieval of their Pose

► Object Recognition and Pose Estimation from 2.5D Scenes

Image and Video Capture

Definition

Image video capture interface is responsible for acquiring both data and control signals from the CMOS or CCD camera sensor as well as providing the appropriate formatting of the data prior to being routed to memory through DMA.

The camera sensor may provide either raw or pre-processed image data to the interface through a variety of programmable options. The interface receives the video/image data stream from the camera sensor and provides all control signaling for this operation as either a Master or Slave device. In the Master mode, the line and frame synchronization signals are provided by the CMOS sensor and in Slave mode, the synchronization signals are provided by the interface (see Fig. 1). Several reduced pin-count alternatives are supported as subsets of the Master mode of operation. These include allowing the data to be serialized and the elimination of the separate synchronization signals for sensors that utilize the ITU-R BT.656 Start-of-Active-Video (SAV) and End-of-Active-Video (EAV) embedded in the data stream. The width of the data bus is also flexible and may be configured in an 8, 9, or 10- wire parallel mode or in a 4 or 5-wire serialized mode.

The pixel data received may be in several possible formats. These formats include RAW, YCbCr 4:2:2, RGB 8:8:8, RGB 6:6:6, RGB 5:6:5, RGBT 5:5:5, and RGB 4:4:4. When a RAW capture mode is enabled, the data may be in 8, 9, or 10-bit formats. The RAW data is de-serialized if necessary, and then packed as either 8-bit or 16-bit elements prior to transfer to memory. In a similar manner the pre-processed RGB and YCbCr image formats are packed into 16-bit or 32-bit elements.

The YCbCr format has the additional option of being planerized prior to being stored to memory. This planarization facilitates SIMD processing (for example, using Wireless MMX) as well as immediate display using the color space conversion engine available in some advanced the LCD controller. The RGB formats may also be formatted to enable immediate preview using the LCD controller. Programmable options are provided to reduce RGB component

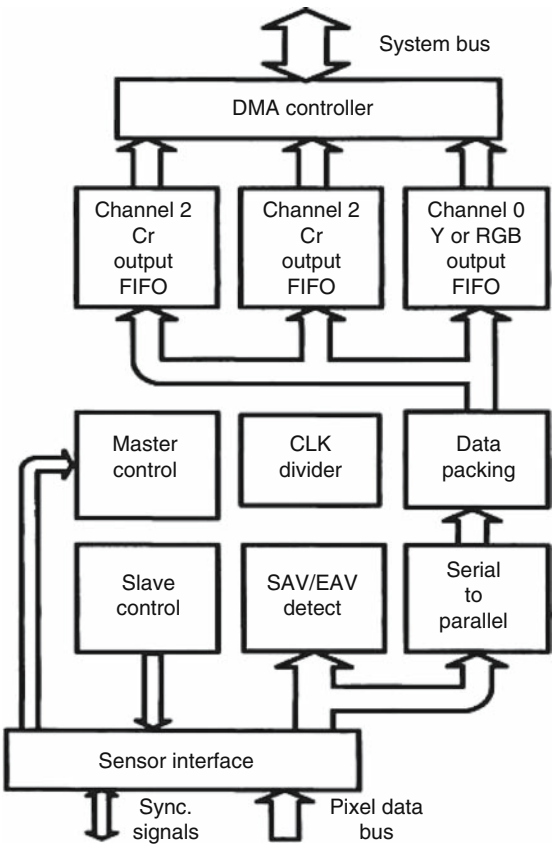


Image and Video Capture. **Figure 1.** Interface for image and video capture.

precisions and provide the transparency management for use with the LCD overlays. The formatting options offer value in reducing power consumption during the digital viewfinder preview sequence for image and video capture.

For the camera sensors with higher pixel counts, only support raw format in favor of reducing camera complexity. Raw pixel processing (i.e., conversion from bayer pattern, correcting for noise and camera irregularities) can be done on the camera interface. Some advanced camera interfaces support these algorithmic operations.

Cross-References

► Multimedia System-on-a-Chip

References

1. Interfaces for Digital Component Video Signals in 525-Line and 625-Line Television Systems operating at the 4:2:2 Level of Recommendation ITU-R BT.601 (Part A), ITU-R Recommendation BT.656-3, 1995.

Image and Video Quality Assessment

Kalpana Seshadrinathan, Alan C. Bovik
University of Texas at Austin, Austin, TX, USA

Synonym

► Quality of images signals; Quality of video signals

Definition

Image and video quality assessment deals with quantifying the quality of an image or video signal as seen by a human observer using an objective measure.

Introduction

In this article, we discuss methods to evaluate the quality of digital images and videos, where the final image is intended to be viewed by the human eye. The quality of an image that is meant for human consumption can be evaluated by showing it to a human observer and asking the subject to judge its quality on a pre-defined scale. This is known as *subjective assessment* and is currently the most common way to assess image and video quality. Clearly, this is also the most reliable method as we are interested in evaluating quality *as seen by the human eye*. However, to account for human variability in assessing quality and to have some statistical confidence in the score assigned by the subject, several subjects are required to view the same image. The final score for a particular image can then be computed as a statistical average of the sample scores. Also, in such an experiment, the assessment is dependent on several factors such as the display device, distance of viewing, content of the image, whether or not the subject is a trained observer who is familiar with processing of images etc. Thus, a change in viewing conditions would entail repeating the experiment! Imagine this process being repeated for every image that is encountered and it becomes clear why subjective studies are cumbersome and expensive. It would hence be extremely valuable to formulate some *objective measure* that can predict the quality of an image.

The problem of image and video quality assessment is to quantify the quality of an image or video signal *as seen by a human observer* using an objective measure. The quality assessment techniques that we present in this article are known as *full-reference* techniques, i.e. it is assumed that in addition to the test image whose quality we wish to evaluate, a "perfect" reference image is also available. We are, thus, actually evaluating the fidelity of the image, rather than the quality. Evaluating the quality of an image without a reference image is a much harder problem and is known as *blind* or *no-reference* quality assessment. Blind techniques generally reduce the storage requirements of the algorithm, which could lead to considerable savings, especially in the case of video signals. Also, in certain applications, the original uncorrupted image may not be available. However, blind algorithms are also difficult to develop as the interpretation of the content and quality of an image by the HVS depends on high-level features such as attentive vision, cognitive understanding, and prior experiences of viewing similar patterns, which are not very well understood. *Reduced reference* quality assessment techniques form the middle ground and use some information from the reference signal, without requiring that the entire reference image be available.

Why Do We Need Quality Assessment?

Image and video quality assessment plays a fundamental role in the design and evaluation of imaging and image processing systems. For example, the goal of image and video compression algorithms is to reduce the amount of data required to store an image and at the same time, ensure that the resulting image is of

sufficiently high quality. Image enhancement and restoration algorithms attempt to generate an image that is of better visual quality from a degraded image. Quality assessment algorithms are also useful in the design of image acquisition systems and to evaluate display devices etc. Communication networks have developed tremendously over the past decade and images and video are frequently transported over optic fiber, packet switched networks like the Internet, wireless systems etc. Bandwidth efficiency of applications such as video conferencing and Video on Demand (VOD) can be improved using quality assessment systems to evaluate the effects of channel errors on the transported images and video. Finally, quality assessment and the psychophysics of human vision are closely related disciplines. Evaluation of quality requires clear understanding of the sensitivities of the HVS to several features such as luminance, contrast, texture, and masking that are discussed in detail in Sect. 4.1. of this article. Research on image and video quality assessment may lend deep insights into the functioning of the HVS, which would be of great scientific value.

Why is Quality Assessment So Hard?

At first glance, a reasonable candidate for an image quality metric might be the Mean-Squared Error (MSE) between the reference and distorted images. Consider a reference image and test image denoted by $R = R(i, j)$ and $T = T(i, j)$ respectively, where $0 \leq I \leq N - 1$, $0 \leq j \leq M - 1$. Then, the MSE is defined by:

$$MSE = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [T(i, j) - R(i, j)]^2$$

MSE is a function of the Euclidean distance between the two vectors *R*and *T*in an *MN*-dimensional space. Since the MSE is a monotonic function of the error between corresponding pixels in the reference and distorted images, it is a reasonable metric and is often used as a quality measure. Some of the reasons for the popularity of this metric are its simplicity, ease of computation and analytic tractability. However, it has long been known to correlate very poorly with visual quality [1]. A few simple examples are sufficient to demonstrate that MSE is *completely unacceptable* as a visual quality predictor. This is illustrated in Fig. 1 which shows several images whose MSE with respect to the reference are identical, but have very different visual quality. The main reason for the failure of MSE as a quality metric is the absence of any kind of modeling of the sensitivities of the HVS.

The difficulties in developing objective measures of image quality are best illustrated by example. Figure 1 (a) and (b) show the original "Caps" and "Buildings" images respectively. Figure 1(c) and (d) show JPEG compressed versions of these images of approximately the same MSE. While the distortion in the "Buildings" image is hardly visible, it is visibly annoying in the "Caps" image. The perception of distortion varies with the actual image at hand and this effect is part of what makes quality assessment difficult. There is enormous diversity in the content of images used in different applications and even within images of a specific category, for example, the class of images obtained from the real world. Consistent performance of a quality assessment algorithm irrespective of the specific image at hand is no easy task. Additionally, different kinds of distortion produce different characteristic artifacts and it is very difficult for a quality assessment algorithm to predict degradation in visual quality *across distortion types*. For example, JPEG produces characteristic blocking artifacts and blurring of fine details (Fig. 1(c) and (d)).

This is due to the fact that it is a block-based algorithm that achieves compression by removing the highest frequency components that the HVS is least sensitive to. JPEG 2000 compression eliminates blocking artifacts, but produces ringing distortions that are visible in areas surrounding edges, such as around the edges of the caps in Fig. 1(e). Sub-band decompositions, such as those used in JPEG 2000, attempt to approximate the image using finite-duration basis functions and this causes ringing around discontinuities like edges due to Gibb's phenomenon. Figure 1(f) shows an image that is corrupted by Additive White Gaussian Noise (AWGN) which looks grainy, seen clearly in the smooth background regions of the image. This kind of noise is typically observed in a lot of imaging devices and images transmitted through certain communication channels. A generic image quality measure should predict visual quality in a robust manner across these and several other types of distortions.

Thus, it is not an easy task for a machine to automatically predict quality by computation, although the human eye is very good at evaluating the quality of an image almost instantly. We explore some state-of-the-art techniques for objective quality assessment in the following sections.

Image and Video Quality Assessment. **Figure 1.** (a) Original "Caps" image (b) Original "Buildings" image (c) JPEG compressed image, MSE = 160 (d) JPEG compressed image, MSE = 165 (e) JPEG 2000 compressed image, MSE = 155 (f) AWGN corrupted image, MSE = 160.

Approaches to Quality Assessment

Techniques for image and video quality assessment can broadly be classified as bottom-up and top-down approaches. Bottom-up approaches attempt to model the functioning of the HVS and characterize the sensitivities and limitations of the human eye to predict the quality of a given image. Top-down approaches, on the other hand, usually make some high-level assumption on the technique adopted by the human eye in evaluating quality and use this to develop a quality metric. Top-down methods are gaining popularity due to their low computational complexity, as they don't attempt to model the functioning of the entire HVS, but only try to characterize the features of the HVS that are *most relevant* in evaluating quality. Also, the HVS is a complex entity and even the low-level processing in the human eye that includes the optics, striate cortex and retina are not understood well enough today, which reflects on the accuracy of existing HVS models. In this chapter, we categorize several state-of-the-art quality assessment techniques into three main categories, namely HVS modeling based approaches, structural

approaches and information theoretic approaches. Each of these paradigms in perceptual quality assessment is explained in detail in the following sections.

HVS-Based Approaches

Most HVS-based approaches can be summarized by the diagram shown in Fig. 2. The initial step in the process usually involves the decomposition of the image into different spatial-frequency bands. It is well known that cells in the visual cortex are specialized and tuned to different ranges of spatial frequencies and orientations. Experimental studies indicate that the radial frequency selective mechanisms have constant octave bandwidths and the orientation selectivity is a function of the radial frequencies. Several transforms have been proposed to model the spatial frequency selectivity of the HVS and the initial step in an HVS-based approach is usually a decomposition of the image into different sub-bands using a filter-bank.

The perception of brightness is not a linear function of the luminance and this effect is known as luminance masking. In fact, the threshold of visibility of a brightness pattern is a linear function of the background luminance. In other words, brighter regions in an image can tolerate more noise due to distortions before it becomes visually annoying. The Contrast Sensitivity Function (CSF) provides a description of the frequency response of the HVS, which can be thought of as a band-pass filter. For example, the HVS is less sensitive to higher spatial frequencies and this fact is exploited by most compression algorithms to encode images at low bit rates, with minimal degradation in visual quality. Most HVS-based approaches use some kind of modeling of the luminance masking and contrast sensitivity properties of the HVS as shown in Fig. 2.

In Fig. 1, the distortions are clearly visible in the "Caps" image, but they are hardly noticeable in the "Buildings" image, despite the MSE being the same. This is a consequence of the contrast masking property of the HVS, wherein the visibility of certain image components is reduced due to the presence of other strong image components with similar spatial frequencies and orientations at neighboring spatial locations. Thus, the strong edges and structure in the "Buildings" image effectively mask the distortion, while it is clearly visible in the smooth "Caps" image. Usually, a HVS-based metric incorporates modeling of the contrast masking property, as shown in Fig. 2.

In developing a quality metric, a signal is first decomposed into several frequency bands and the HVS model specifies the maximum possible distortion that can be introduced in each frequency component before the distortion becomes visible. This is known as the Just Noticeable Difference (JND). The final stage in the quality evaluation involves combining the errors in the different frequency components, after normalizing them with the corresponding sensitivity thresholds, using some metric such as the Minkowski error. The final output of the algorithm is either a spatial map showing the image quality at different spatial locations or a single number describing the overall quality of the image.

Different proposed quality metrics differ in the models used for the blocks shown in Fig. 2. Notable amongst the HVS-based quality measures are the Visible Difference Predictor [2], the Teo and Heeger model [3], Lubin's model [4] and Sarnoff's JNDMetrix technology [5].

Structural Approaches

Structural approaches to image quality assessment, in contrast to HVS-based approaches, take a top-down view of the problem. Here, it is hypothesized that the HVS has evolved to extract *structural information* from a scene and hence, quantifying the loss in structural

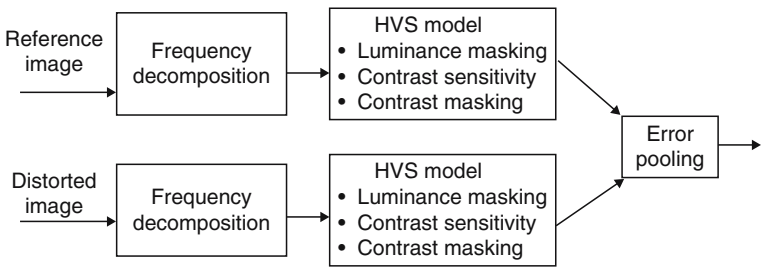


Image and Video Quality Assessment. **Figure 2.** Block diagram of HVS-based quality metrics.

information can accurately predict the quality of an image [6]. In Fig. 1, the distorted versions of the "Buildings" image and the "Caps" image have the same MSE with respect to the reference image. The bad visual quality of the "Caps" image can be attributed to the structural distortions in both the background and the objects in the image. The structural philosophy can also accurately predict the good visual quality of the "Buildings" image, since the structure of the image remains almost intact in both distorted versions.

Structural information is defined as those aspects of the image that are independent of the luminance and contrast, since the structure of various objects in the scene is independent of the brightness and contrast of the image. The Structural SIMilarity (SSIM) algorithm, also known as the Wang-Bovik Index partitions the quality assessment problem into three components, namely luminance, contrast and structure comparisons.

Let $\vec{x}$ and $\vec{y}$ represent N+ dimensional vectors containing pixels from the reference and distorted images respectively. Then, the Wang-Bovik Index between $\vec{x}$ and $\vec{y}$ is defined by:

$$SSIM(\vec{x},\vec{y}) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2+\mu_y^2+C_1}\right)^{\alpha} \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2+\sigma_y^2+C_2}\right)^{\beta} \left(\frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}\right)^{\gamma} \quad (1)$$

where

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)^2\right)^{1/2},$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)$$

C_1, C_2 and C_3 are small constants added to avoid numerical instability when the denominators of the fractions are small. α, β and γ are non-negative constants that control the relative contributions of the three different measurements to the Wang-Bovik Index.

The three terms in the right hand side of (1) are the luminance, contrast and structure comparison measurements respectively. μ_x and μ_y are estimates of the mean luminance of the two images and hence, the first term in (1) defines the luminance comparison function. It is easily seen that the luminance comparison function satisfies the desirable properties of being bounded by 1 and attaining the maximum possible value if and only if the means of the two images are equal. Similarly, σ_x and σ_y are estimates of the contrast of the two images and the second term in (1) defines the contrast comparison function. Finally, the structural comparison is performed between the luminance and contrast normalized signals, given by $(\vec{x}-\mu_x)/\sigma_x$ and $(\vec{y}-\mu_y)\big/\sigma_y$. The correlation or inner product between these signals is an effective measure of the structural similarity. The correlation between the normalized vectors is equal to the correlation coefficient between the original signals $\vec{x}$ and $\vec{y}$, which is defined by the third term in (1). Note that the Wang-Bovik Index is also bounded by 1 and attains unity if and only if the two images are equal.

The structural philosophy overcomes certain limitations of HVS-based approaches such as computational complexity and inaccuracy of HVS models. The idea of quantifying structural distortions is not only novel, but also intuitive, and experimental studies show that the algorithm is competitive with several other state-of-the-art quality metrics.

Information Theoretic Approaches

Information theoretic approaches attempt to quantify the *loss in the information* that the HVS can extract from a given test image, as compared to the original reference image [7]. *Mutual information* between two random sources is a statistical measure that quantifies the amount of information one source contains about the other. In other words, assuming the distorted and reference images to be samples obtained from two random sources, mutual information measures the distance between the distributions of these sources. Information theoretic approaches use this measure to quantify the amount of information that the human eye can obtain from a given image, to develop a metric that correlates well with visual quality. The Visual Information Fidelity (VIF) criterion, also known as the Sheikh-Bovik Index, assumes that the distorted image is the output of a communication channel that introduces errors in the image that passes through it. The HVS is also assumed to be a communication channel that limits the amount of information that can pass through it.

Photographic images of natural scenes exhibit striking structures and dependencies and are far from random. A random image generated assuming an independent and identically distributed Gaussian source, for example, will look nothing like a natural image. Characterizing the distributions and statistical dependencies of natural images provides a description of the subspace spanned by natural images, in the space of all possible images. Such probabilistic models have been studied by numerous researchers and one model that has achieved considerable success is known as the Gaussian Scale Mixture (GSM) model [8]. This is the source model used to describe the statistics of the wavelet coefficients of reference images in the Sheikh-Bovik Index. Let $\vec{R}$ represent a collection of wavelet coefficients from neighboring spatial locations of the original image. Then, $\vec{R} \tilde{z} \vec{U}$, where represents a scalar random variable known as the mixing density and $\vec{U}$ represents a zero-mean, white Gaussian random vector. Instead of explicitly characterizing the mixing density, the maximum likelihood estimate of the scalar x is derived from the given image in the development of the Sheikh-Bovik Index.

Let $\vec{D}$ denote the corresponding coefficients from the distorted image. The distortion channel that the reference image passes through to produce the distorted image is modeled using:

$$\vec{D} = g\vec{R} + \vec{v}$$

This is a simple signal attenuation plus additive noise model, where g represents a scalar attenuation and $\vec{v}$ is additive Gaussian noise. Most commonly occurring distortions such as compression and blurring can be approximated by this model reasonably well. This model has some nice properties such as analytic tractability, ability to characterize a wide variety of distortions and computational simplicity. Additionally, both reference and distorted images pass through a communication channel that models the HVS. The HVS model is given by:

$$\vec{R}_{out} = \vec{R} + \vec{N}_1, \vec{D}_{out} = \vec{D} + \vec{N}_2$$

where $\vec{N}_1$ and $\vec{N}_2$ represent additive Gaussian noise, that is independent of the input image. The entire system is illustrated in Fig. 3.

The VIF criterion is then defined for these coefficients using:

$$VIF = \frac{I(\vec{D}; \vec{D}_{out}/z)}{I(\vec{R}; \vec{R}_{out}/z)} \quad (2)$$

$I(\vec{D}; \vec{D}_{out}/z)$ represents the mutual information between $\vec{D}$ and $\vec{D}_{out}$, conditioned on the estimated value of z. The denominator of Equation (2) represents the amount of information that the HVS can extract from the original image. The numerator represents the amount of information that the HVS can extract from the distorted image. The ratio of these two quantities hence is a measure of the amount of information in the distorted image relative to the reference image and has been shown to correlate very well with visual quality. Closed form expressions to compute this quantity have been derived and further details can be found in [7]. Note that wavelet coefficients corresponding to the same spatial location can be grouped separately, for example, coefficients in each sub-band of the wavelet decomposition can be collected into a separate vector. In this case, these different quality indices for the same spatial location have to be appropriately combined. This results in a spatial map containing the Sheikh-Bovik quality Index of the image, which can then be combined to produce an overall index of goodness for the image.

The success of the information theoretic paradigm lies primarily in the use of accurate statistical models for the natural images and the distortion channel.

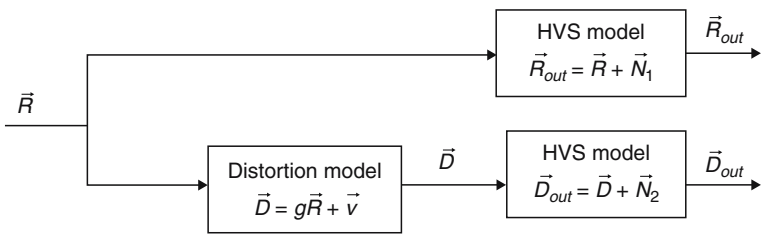


Image and Video Quality Assessment. Figure 3. Block diagram of the Sheikh-Bovik quality assessment system.

Natural scene modeling is in some sense a dual of HVS modeling, as the HVS has evolved in response to the natural images it perceives. The equivalence of this approach to certain HVS-based approaches has also been established. The idea of quantifying information loss and the deviation of a given image from certain expected statistics provides an altogether new and promising perspective on the problem of image quality assessment.

Figure 4 illustrates the power of the Wang-Bovik and Sheikh-Bovik indices in predicting image quality. Notice that the relative quality of the images, as

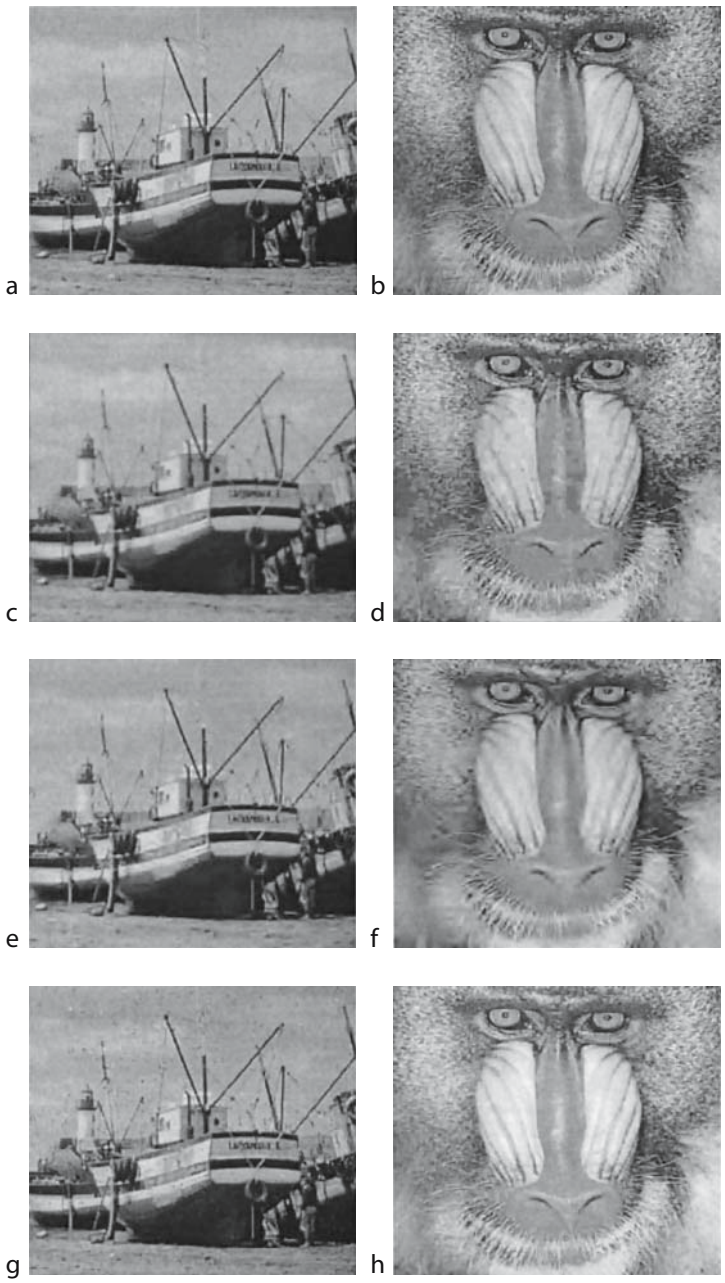


Image and Video Quality Assessment. **Figure 4.** Illustration of the Wang-Bovik and Sheikh-Bovik indices. (a) Original "Boats" image (b) Original "Mandrill" image (c) Gaussian Blurring, SSIM = 0.85, VIF = 0.25 (d) JPEG compression, SSIM = 0.54, VIF = 0.07 (e) JPEG2000 compression, SSIM = 0.78, VIF = 0.11 (f) JPEG2000 compression, SSIM = 0.48, VIF = 0.05 (g) Salt and Pepper noise, SSIM = 0.87, VIF = 0.38 (h) Mean shifted, SSIM = 0.99, VIF = 1.

predicted by both indices, is the same and agrees reasonably well with human perception of quality. In the Video Quality Experts Group (VQEG) Phase I FR-TV tests [9], which provides performance evaluation procedures for quality metrics, logistic functions are used in a fitting procedure to obtain a non-linear mapping between objective/subjective scores first. Hence, the differences in the absolute values of quality predicted by the two algorithms are not important.

Conclusions

In this chapter, we have attempted to present a short survey of image quality assessment techniques. Researchers in this field have primarily focused on techniques for images as this is easier and usually the first step in developing a video quality metric. Although insights from image quality metrics play a huge role in the design of metrics for video, it is not always a straight forward extension of a two-dimensional problem into three dimensions. The fundamental change in moving from images to video is the motion of various objects in a scene. From the perspective of quality assessment, video metrics require modeling of the human perception of motion and quality of motion in the distorted image. Most of the algorithms discussed here have been extended to evaluate the quality of video signals and further details can be found in the references.

Considerable progress has been made in the field of quality assessment over the years, especially in the context of specific applications like compression and halftoning. Most of the initial work dealt with the threshold of perceived distortion in images, as opposed to supra-threshold distortion which refers to artifacts that are perceptible. Recent work in the field has concentrated on generic, robust quality measures in a full reference framework for supra-threshold distortions. No-reference quality assessment is still in its infancy and is likely to be the thrust of future research in this area.

References

1. B. Girod, "What's Wrong with Mean-Squared Error," in A.B. Watson (Ed.), "Digital Images and Human Vision," MIT, Cambridge, MA, 1993, pp. 207–220.
2. S. Daly, "The Visible Difference Predictor: An algorithm for the assessment of image fidelity," Proceedings of the SPIE, Vol. 1616, 1992, pp. 2–15.
3. P.C. Teo and D.J. Heeger, "Perceptual Image Distortion," Proceedings of the SPIE, Vol. 2179, 1994, pp. 127–141.
4. J. Lubin, "The use of Psychophysical Data and Models in the Analysis of Display System Performance," in A.B. Watson (Ed.), "Digital Images and Human Vision," MIT, Cambridge, MA, 1993, pp. 163–178.
5. Sarnoff Corporation, "JNDMetrix Technology," 2003, evaluation version available: http://www.sarnoff.com/products_services/video_vision/jndmetrix/downloads.asp.
6. Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004, pp. 1–14.
7. H.R. Sheikh and A.C. Bovik, "Image Information and Visual Quality," IEEE Transactions on Image Processing, Accepted for publication, September 2004.
8. M.J. Wainwright and E.P. Simoncelli, "Scale Mixtures of Gaussians and the Statistics of Natural Images," in S.A. Solla, T.K. Leen, and K.R. Mueller (Eds.), "Advances in Neural Information Processing Systems," MIT, Cambridge, MA, May 2000, Vol. 12, pp. 855–861.
9. VQEG, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment," March 2000, http://www.vqeg.org.

Image and Video Super Resolution Techniques

Jinjun Wang, Yihong Gong
NEC Laboratories America, Inc., Cupertino, CA, USA

Synonyms

► Medical images for medical diagnosis

Definition

Image and video super resolution techniques refer to creating higher resolution image from single/multiple low-resolution input.

Introduction

Images with high pixel density are desirable in many applications, such as high-resolution (HR) medical images for medical diagnosis, high quality video conference, high definition Television broadcasting, Blu-ray movies, etc. While people can use higher resolution camera for the purpose, there is an increasing demand to shoot HR image/video from low-resolution (LR) cameras such as cell phone camera or webcam, or converting existing standard definition footage into high definition video material. Hence, software resolution enhancement techniques are very desirable for these applications.

The task of software image resolution enhancement is to estimate more pixel values to generate a processed image with higher resolution. The simplest way to produce more pixel values is to use up-sampling method. However, it would only produce a nearest neighbor estimation with serious blocking effect. An alternative is to apply 2D interpolation methods to achieve best approximation based on the values at surrounding pixels, such that the generated image is locally smooth, and the perceptual quality is much more improved. Popular image interpolation methods include bi-cubic interpolation, polynomial spline resampling [1], etc. Due to their robustness and computational efficiency, these interpolation methods have been extensively studied [1,2] and are widely used for image resizing/enlargement.

In theory, the generation process of LR image can be modeled by a combination of atmosphere blur, motion blur, camera blur, and down-sampling. Due to the aliasing effect, the high-frequency components is lost or degraded during the image degrading process. Since the interpolation-based methods do not bring any additional information, they cannot recover these high-frequency components [3], and hence their performance is limited. The image super-resolution (SR) idea was introduced by Tsai and Huang [4] to reconstruct the HR image by recovering additional information from multiple LR images using signal processing techniques. The SR approach has attracted increasing research attentions in recent years, and many improved methods such as single image SR and real-time SR, have been reported [3,5].

Existing SR methods can be divided into three major categories, specifically the reconstruction-, the functional interpolation-, and the learning-based methods. The reconstruction-based methods [6–13, 14] form the largest body of SR research. They assume that, from the same HR scene, a set of downgraded LR images are available for reconstruction. These LR images result from shifting, warping, blurring, and sub-sampling operators performed on the HR image, corrupted by additive noise [13]. To estimate the original HR image, most reconstruction-based methods consist of three major stages as illustrated in Fig. 1: registration, interpolation, and restoration. Registration refers to the estimation of relative shifts between LR images compared to the reference LR image with fractional pixel accuracy. Since the registered HR image does not always match up to a uniformly spaced grid, nonuniform interpolation can always be applied. The restoration stage up-samples the image and removes blurring and noise. Comprehensive overviews of reconstruction-based methods are given by Borman et al. [15] and Park et al. [3].

The second category of existing SR methods includes the functional interpolation approaches. They apply an existing function on the LR input to obtain a processed image [16,17]. Unlike those simple image interpolation methods, which cannot recover the high-frequency components in the magnified image, the functional interpolation-based method encodes certain prior knowledge into the applied functions, such that the produced SR images can have some high-frequency components, and hence are more perceptually pleasing. For instance, Irani and Peleg [16,17] proposed that, if the image down-grading process is known, the desired SR image should be able to "reproduce" an LR image that is identical to the LR input. They introduced an iterative method to generate such an SR image as illustrated in Fig. 2: First the LR input image is interpolated to the desired size. Then given a simulated image down-grading process (such as a Gaussian blur filter), the interpolated image is projected into an LR image. The difference between the original LR input and the projected LR image is back-projected into the interpolated image. The process is

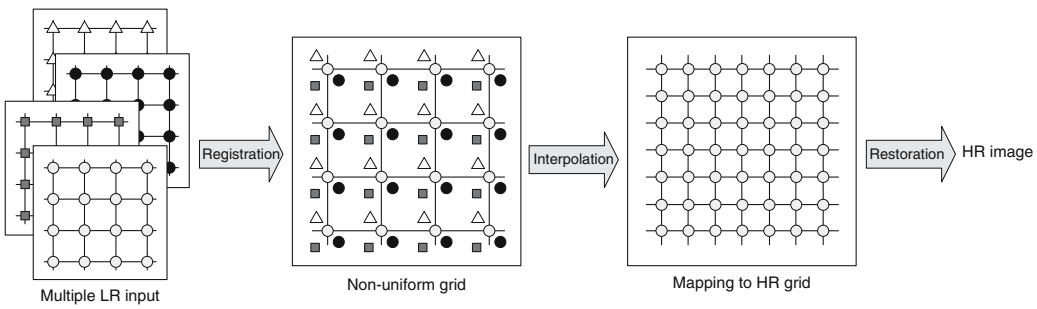


Image and Video Super Resolution Techniques. **Figure 1.** Illustration of reconstruction-based SR methods.

iterated several times, and the final interpolated image after several back-projections is regarded as the generated SR images.

The third category of SR techniques includes the learning-based SR methods, which are very popular in recent years. It solves the SR problem by first learning the co-occurrence prior between HR and LR image patches or coefficients, and then processing the LR input along with appropriate smoothness constraint [6, 15, 16, 18–, 19–28]. Figure 3 shows a typical framework of the learning-based methods [19]. It is usually assumed that any HR image (Fig. 3a) consist of three frequency layers, specifically the high-frequency layer (Fig. 3b), the middle-frequency layer (Fig. 3c), and the low-frequency layer (Fig. 3d). Since the down-graded LR image results from discarding the high-frequency

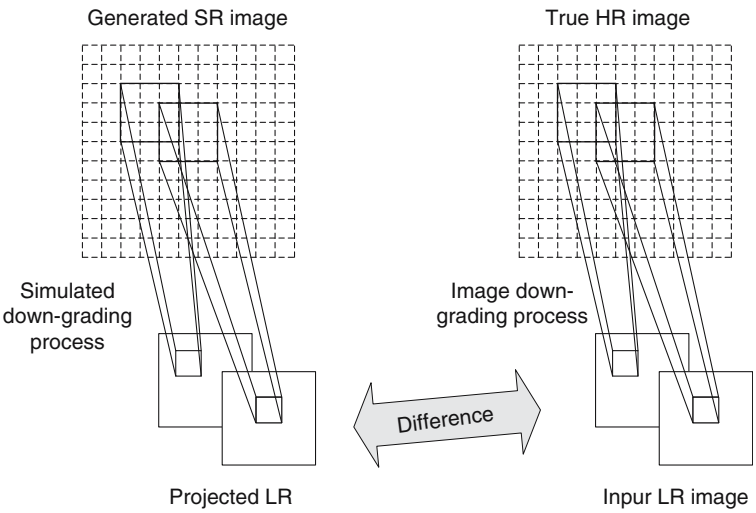


Image and Video Super Resolution Techniques. **Figure 2.** Illustration of back-projection SR method.

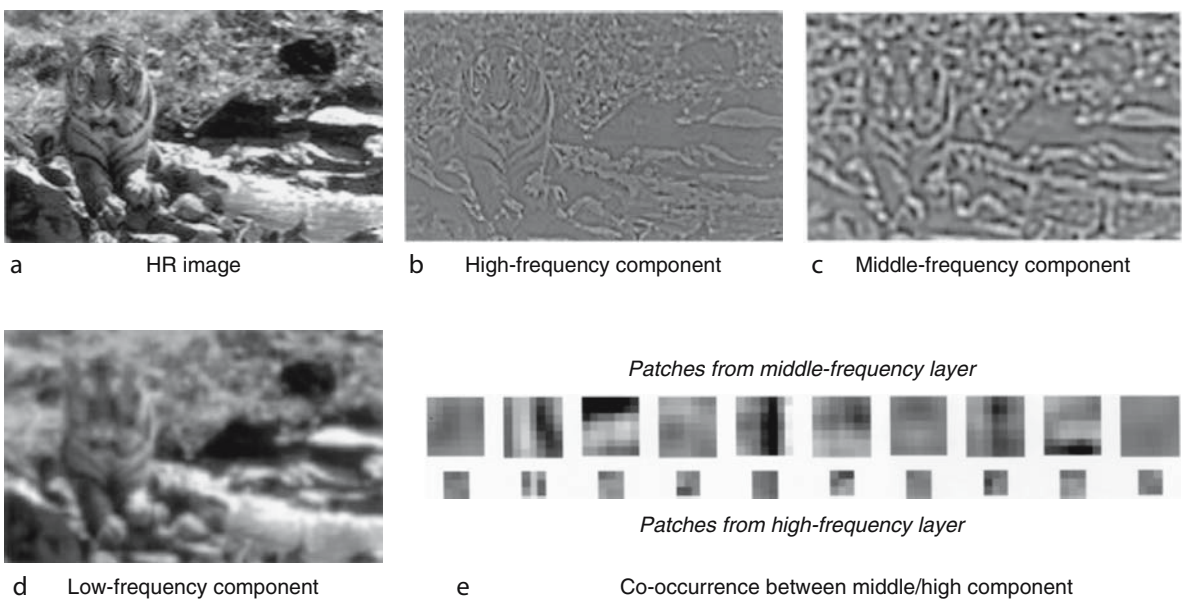


Image and Video Super Resolution Techniques. **Figure 3.** Illustration of learning-based SR method. (a) HR image, (b) high-frequency component, (c) middle-frequency component, (d) low-frequency component, (e) co-occurrence between middle/high component.

components, the remaining low- and middle-frequency layers are correlated to the lost high-frequency components. With suitable machine-learning techniques, such co-occurrence patterns can be discovered automatically from a training set of multiple HR/LR image pairs (Fig. 3e). Based on the learned patterns, when a new LR input is available, the computer can infer what the high-frequency component should be.

In general, since the introduction of SR idea by Tsai and Huang [4] in 1984, image/video SR has become one of the most spotlighted research areas, because it can overcome the inherent resolution limitation of the imaging system and improve the performance of most digital image processing applications [3]. However, certain limitations exist for the methods listed in all the above three categories. For the reconstruction-based methods, they are not able to handle single image SR. In addition, they require that the image pairs are related by global parametric transformations, which restricts their usage in various real scenarios. Besides, the reconstruction constraints provide less and less useful information as the image magnification factor increases [15,29]. For the functional interpolation-based methods, since the parameters of the true image down-grading process are usually unknown, the applied functions sometimes generate serious artifact for certain types of images. Hence, their performance is not robust. For the learning-based methods, the quality of processed images dependents on the training set. Hence, the result is not stable and sometimes produces artifacts in real applications [30]. Besides, the inference process is usually computationally expensive, which restrict the learning-based methods from real-time applications.

In this article, we present two SR approaches that are suitable for both single image and video SR. The first approach belongs to the learning-based category where we introduce a fast image SR method to address the computation efficiency issue of existing learning-based methods. In the second approach, we propose a functional interpolation-based approach where the edge sharpness and image smoothness priors is encoded into the objective function to produce perceptually pleasing SR image. The remainder of the article is organized as follows: Section "Learning-based approach using Gaussian Processes Regression" describes our proposed learning-based SR method, Section on "Functional Interpolation-based approach" presents our proposed functional interpolation-based method. Section "Experiment" lists the experimental results for both the two methods, and Section "Conclusions and Future Works" summarizes the proposed methods and discusses future works.

Learning-Based Approach Using Gaussian Processes Regression

Limitations with Current Learning-Based Methods

As introduced in the previous section, many existing learning-based methods have two major limitations: (1) computational expensive and (2) dependent on the training data set. To understand this, let's look at a typical learning-based approach that is adopted by many researchers [18, 22, 23, 28]. Denoting the high-frequency layer in an image by H, the middle-frequency layer by M, and the low frequency layer by L. The image down-grading process discards the H components. Hence, to obtain the processed SR image, the learning-based methods seek to predict H that maximizes $\Pr(H|M,L)$ based on learned co-occurrence patterns between H, M, and L. In addition, Freeman et al. [18] argued that the high-frequency component H is independent of L, and it is only required to maximize $\Pr(H|M)$, which greatly reduces the variability to be stored in the training data. Hence, a typical training data set usually consists of a set of N patch pairs extracted from the middle-frequency layer M, denoted as $\{\mathbf{P}_i^L\}$, $i = 1,\ldots,N$, and their corresponding high-frequency patches $\{\mathbf{P}_i^H\}$ extracted from the high-frequency layer H. $\mathbf{P}_i^L$ and $\mathbf{P}_i^H$ often have different dimensions.

Given a set of new middle-frequency patches $\{\tilde{\mathbf{P}}_j^L\}$, $j = 1,\ldots,M$ extracted from an LR input image, and denoting the high-frequency patches to be estimated by $\{\tilde{\mathbf{P}}_j^H\}$, the spatial relationship between $\{\tilde{\mathbf{P}}_j^L\}$ and $\{\tilde{\mathbf{P}}_j^H\}$ can be represented by a Markov network as illustrated in Fig. 4.

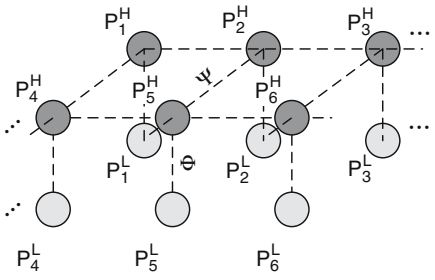


Image and Video Super Resolution Techniques.

Figure 4. Markov network representation.

Hence, the estimation process strives to find $\{\tilde{\mathbf{P}}_j^H\} \in \{\mathbf{P}_i^H\}$ such that (1) $\{\tilde{\mathbf{P}}_j^L\}$ is similar to $\{\mathbf{P}_j^L\}$ (Ψ in Fig. 4), and (2) $\{\tilde{\mathbf{P}}_j^H\}$ follows certain neighboring smoothness constrain (Ψ in Fig. 4). This equals to minimize the following objective function

$$\{\tilde{\mathbf{P}}_j^H\}^* = \underset{\{\mathbf{P}_j^H\} \in \{\mathbf{P}_i^H\}}{argmin} \sum_{j=1}^{M} \left(\Psi(\mathbf{P}_j^H, N(\mathbf{P}_j^H)) + \Phi(\mathbf{P}_j^L, \tilde{\mathbf{P}}_j^L)\right), \tag{1}$$

where $\Phi(\mathbf{P}_j^L, \tilde{\mathbf{P}}_j^L)$ represents the Euclidean distance between $\mathbf{P}_j^L$ and $\tilde{\mathbf{P}}_j^L$, $\Psi(\mathbf{P}_j^H, N(\mathbf{P}_j^H))$ represents the Euclidean distance between $\mathbf{P}_j^H$ and $N(\mathbf{P}_j^H))$, and $N(\mathbf{P}_j^H)$ represents the neighboring patches of $\mathbf{P}_j^H$. Exact inference of Eq. (1) is a #P-complete problem, and thus computationally intractable. To solve the problem, Freeman et al. [18] first considered only the Ψ term in Eq. (1) to find 10 nearest neighbors (NN) of each middle-frequency patches in the training set, then used Bayesian belief propagation to find the global combination that minimizes Ψ in Eq. (1). Although the strategy makes the SR problem computable, selecting 10-NN for each patch from a large number of patch pairs (40,000 in [18]), and then iterating belief propagation for the whole image is very computationally expensive. This restricts the approach from real-time applications or video SR tasks. Besides, when the given training data set is not appropriate, the processed image generates many artifacts, and thus the result is not stable.

These limitations have motivated us to research more efficient SR methods. In the next subsection, we introduce a regression approach instead of nearest neighbor algorithm to overcome the above-mentioned issues.

Gaussian Processes Regression

Instead of searching for the nearest neighbors $\mathbf{P}^L$ for each input patch $\tilde{\mathbf{P}}^L$ to select $\tilde{\mathbf{P}}^H$, in this section we propose to learn a mapping function $\tilde{\mathbf{P}}^H = f(\tilde{\mathbf{P}}^L)$ to allow fast SR processing. To simplify the notation, let $\mathbf{x}_i$ be the D dimensional column vector of $\mathbf{P}_i^L$, and y_i be one component in $\mathbf{P}_i^H$. We define a $D \times N$ matrix $\mathbf{X}$ where the i^{th} column of $\mathbf{X}$ is $\mathbf{x}_i$, and a $N \times 1$ vector $\mathbf{y}$ where the i^{th} element of $\mathbf{y}$ is y_i. From the training data, we want to learn a mapping function $y_i = f(\mathbf{x}_i) + \varepsilon$ that maximizes the posterior $\Pr(\mathbf{y}|\mathbf{X})$. ε is *iid* noise, having distribution of $N(\varepsilon|0, \sigma^2)$.

For simple linear form of $f(\mathbf{x})$, i.e., $y_i = \mathbf{w}^\top \mathbf{x}_i + \varepsilon$, we can assume the prior distribution $Pr(\mathbf{w})$ to be a Gaussian $N(\mathbf{w}|0, I)$. Given the training data $\mathbf{X}$ and $\mathbf{y}$, it can be easily verified that the posterior distribution, $Pr(\mathbf{w}|\mathbf{X}, \mathbf{y})$, is also a Gaussian of $N(\mathbf{w}|\mu_w, \Sigma_w)$, where

$$\mu_w = (\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I})^{-1}\mathbf{X}\mathbf{y}, \tag{2}$$

$$\Sigma_w = \left(\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^\top + \mathbf{I}\right)^{-1}. \tag{3}$$

For a new input vector $\tilde{\mathbf{x}}$, the MAP estimation of $\tilde{y}$, i.e., $Pr(\tilde{y}|\tilde{\mathbf{x}}, \mathbf{X}, \mathbf{y})$, follows a Gaussian distribution of $N(\tilde{y}|\mu_{\tilde{y}}, \Sigma_{\tilde{y}})$, where

$$\mu_{\tilde{y}} = \mu_w^\top \tilde{\mathbf{x}}, \tag{4}$$

$$\Sigma_{\tilde{y}} = \tilde{\mathbf{x}}^\top \Sigma_w \tilde{\mathbf{x}} + \sigma^2. \tag{5}$$

Hence the learned mapping function is $\tilde{y} = \mu_{\tilde{y}} = \mu_w^\top \tilde{\mathbf{x}}$, which yield a MAP estimation. Note that in our notation, $\tilde{y}$ represents only one component in $\mathbf{P}^H$, hence for every other component in $\mathbf{P}^H$ we will learn another set of μ_w and Σ_w. Finally, all the estimated component $\tilde{y}$ are concatenated together to form $\mathbf{P}^H$, which is then pasted into the high-frequency layer to compose the processed SR image. At running time, since only matrix–vector multiplication is needed (matrix–matrix multiplication in batch), the linear model can be implemented very efficiently.

There are two problems with the simple linear approach. First, the method does not consider neighboring dependency such that there might be discontinuity artifact between neighboring high-frequency patches. However, since the LR image is locally smooth, it is observed that, with good estimation of $\tilde{y}$, the discontinuity artifact in the generated HR patches is not significant. $\Sigma_{\tilde{y}_i}$ can be regarded as the uncertainty of the estimation. Hence, we use the $\Sigma_{\tilde{y}_i}$ to decide the order when $\mathbf{P}_i^H$ (as can be seen from Eq. (1), the uncertainties of every component in $\mathbf{P}_i^H$ only depend on $\tilde{x}_i$, hence they are equal) should be pasted into the high-frequency layer. We put more uncertain patches first, then less uncertain patches. When two patches overlap, the part of the more uncertain patch is replaced by the part of the less uncertain patch. In practice, we found this strategy to be very effective.

Another problem with the simple linear regression approach is the low capacity of linear model, which means that even with sufficient training samples provided, the model cannot store enough patterns. The generated results tend to be over smoothed within each patch region, and hence produce zig-zag artifact

by the discontinuity between neighboring patches. In fact, the performance of linear approach is only similar to the bi-linear interpolation.

To increase the capacity of learned function, we resort to Gaussian Processes Regression [31]. Consider $\mathbf{x}$ as a Gaussian process having $\text{cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}')$, where $k(\mathbf{x}, \mathbf{x}')$ is a kernel function. We define an N × N Gram matrix $\mathbf{K}$ for the training data set $\mathbf{X}$ where $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, N$. Now let the mapping function be $y = f(\mathbf{x}) = \theta^\top k(\mathbf{X}, \mathbf{x})$, where the parameter vector θ has a Gaussian prior distribution $N(\theta|0, \mathbf{K}^{-1})$. Then the posterior distribution of $Pr(\theta|\mathbf{X}, \mathbf{y})$ is also a Gaussian of $N(\mu_\theta, \Sigma_\theta)$, and it can be verified that

$$\mu_\theta = (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \tag{6}$$

$$\Sigma_\theta = \left(\frac{1}{\sigma^2}\mathbf{K}\mathbf{K}^\top + \mathbf{K}\right)^{-1}. \tag{7}$$

Therefore, for a new input vector $\tilde{\mathbf{x}}$, the posterior distribution of $Pr(\tilde{y}|\tilde{\mathbf{x}}, \mathbf{X}, \mathbf{y})$ is a Gaussian of $N(\mu_y, \Sigma_y)$, where

$$\mu_{\tilde{y}} = \mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}k(\mathbf{X}, \tilde{\mathbf{x}}), \tag{8}$$

$$\Sigma_{\tilde{y}} = k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - k(\mathbf{X}, \tilde{\mathbf{x}})^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}k(\mathbf{X}, \tilde{\mathbf{x}}) + \sigma^2. \tag{9}$$

Similar to linear regression, we construct patches with $\tilde{y} = \mu_{\tilde{y}}$, then apply the patches onto the LR image in the descending order of average $\Sigma_{\tilde{y}}$. In our current implementation, we use the radial basis function (RBF) kernel, i.e., $k(\mathbf{x}, \mathbf{x}') = exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right)$. Although it is slower than linear model, experimental results showed that the method can be implemented very efficiently to allow for real-time video SR. Also by using the RBF kernel, the method reduced the root mean square (RMS) error as compared with bi-cubic interpolation method, and hence improved the quality of generated SR images.

Functional Interpolation-Based Approach Using Alpha Matting and Alpha Channel SR

Edge Smoothness Prior

In our second approach, instead of modeling the co-occurrence patterns between the HR/LR images to supervise the SR process, we directly reconstruct the SR image from the LR input using prior knowledge about a perceptually pleasing image. One of the most widely used priors is the edge smoothness constrain [32] that prefers a HR image with smooth edges. This is reasonable because the human perception seems to favor this choice. Given the severely under-determined nature of SR, such a prior is especially important for getting rid of zig-zag artifacts at region boundaries, which is a common drawback of simple functional-based methods such as bi-cubic and polynomial spline.

In practice, the edge is much more complex than a single geometric curve. Natural color images exhibit a large variety of edges with different contrast, scale, and orientation. To utilize the edge information to constrain the reconstruction process, this section proposes a novel approach based on alpha matting decomposition of a local patch and alpha channel SR.

Patch Decomposition by Alpha Matting

Alpha-matting is a technique to decompose an image into a linear combination of foreground image and background image through an alpha channel. Assume that the foreground and background regions of a local patch $\mathbf{P}^L$ are $\mathbf{P}_F^L$ and $\mathbf{P}_B^L$, then the following equation holds,

$$\mathbf{P}^L = \alpha^L\mathbf{P}_F^L + (1 - \alpha^L)\mathbf{P}_B^L, \tag{10}$$

where α^L is the foreground opacity. The alpha matting model applies to both the middle- frequency patch and the high-frequency patch. Under alpha model, the down-graded image patch $\mathbf{P}^L$ can be expressed as follows:

$$\begin{aligned}\mathbf{P}^L = (\mathbf{P}^H \otimes G)\downarrow \cong (\alpha^H \otimes G)\downarrow \mathbf{P}_F^H \downarrow \\ + (1 - (\alpha^H \otimes G)\downarrow)\mathbf{P}_B^H \downarrow,\end{aligned} \tag{11}$$

where $\otimes$ is the convolution operator, G is a spatial filter representing the down-grading process, and $\downarrow$ is the down-sampling operator.

By assuming $\alpha^L = (\alpha^H \otimes G)\downarrow$, $\mathbf{P}_F^L = \mathbf{P}_F^H \downarrow$ and $\mathbf{P}_B^L = \mathbf{P}_B^H \downarrow$, Eq. (1) is exactly the same as Eq. (1). This means that we can do alpha matting for $\mathbf{P}^L$ to get α^L, $\mathbf{P}_F^L$ and $\mathbf{P}_B^L$, then α^H, $\mathbf{P}_F^L$ and $\mathbf{P}_B^L$ can be recovered accordingly from them.

Ideally, the alpha matting decomposition should remove the influence of the neighboring background color from the foreground object. However, given a blended patch $\mathbf{P}^L$, solving for $\mathbf{P}_F^L$, $\mathbf{P}_B^L$, and α^L is an under-determined problem. Hence, we used the edge detection result to constrain that the pixels on the same side of the edge should be classified as the same foreground or background region. Then the method

proposed in [33] is used to obtain a closed form solution of the $\mathbf{P}_F^L$, $\mathbf{P}_B^L$, and α^L decomposition.

Alpha Channel Sr

With the smoothness assumption for $\mathbf{P}_F^H$ and $\mathbf{P}_B^H$, they can be interpolated with bi-cubic method given their down-sampled version $\mathbf{P}_F^L$ and $\mathbf{P}_B^L$. Hence the remaining problem is how to recover α^H from α^L. According to Eq. (1), we have assumed that

$$\alpha^L \cong (\alpha^H \otimes G)\downarrow, \tag{12}$$

Hence, the objective is to find the optimal α^{H^*} such that with given G, $(\alpha^H \otimes G)\downarrow$ should be similar to α^L, while satisfying certain smoothness constrain. Hence the objective function can be defined as

$$\alpha^{H^*} = \arg\min_{all\alpha^H}(\Phi(\alpha^L, (\alpha^H \otimes G)\downarrow) + \lambda\Psi(\alpha^H, N_{\alpha^H})), \tag{13}$$

where $\Phi(\alpha^L, (\alpha^H \otimes G)\downarrow)$ represents the Euclidean distance between α^L and $(\alpha^H \otimes G)\downarrow$, $\Psi(\alpha^H, N_{\alpha^H})$ represents the Euclidean distance between each pixel in α^H and its neighbors (8-connectivity), and λ is a parameter to balance these two terms.

In our implementation, the objective function is optimized using steepest descent algorithm as follows,

$$\alpha_{t+1}^H = \alpha_t^H - \beta(d_\Phi + \lambda d_\Psi), \tag{14}$$

and

$$d_\Phi = ((\alpha_t^H \otimes G)\downarrow - \alpha^L)\uparrow \otimes G, \tag{15}$$

$$d_\Psi = \sum_8 \mathrm{sgn}(\alpha_t^H - \alpha_t^H D_k)(1 - D_k), \tag{16}$$

where β is the descent step, d_Φ is the derivative of Φ, which is similar to the update function of back-projection [16,17], $\uparrow$is the up-sampling operator, d_Ψ is the derivative of Ψ, $D_k$is the displacement operator which shifts the entire patch to 1 of the 8 neighbors, and sgn is the sign indication function. This updating strategy is the same as in [32]. α^H is initialized to be the bi-cubic interpolation of α^L.

The above operations described in subsection "Patch Decomposition by Alpha Matting" and subsection "Alpha Channel Sr" can be summarized in Fig. 5.

Experiments

Learning-Based Method

In the first experiment, we did single image SR with 2×2 magnification. Our proposed algorithm in this essay improved the mean square error (MSE) by 10% compared with the bi-cubic interpolation algorithm. Figure 6 shows some input and output image samples. As can be seen from Fig. 6, the proposed method can add more details to the processed images as compared with the bi-cubic interpolation, which proves that the learned mapping functions essentially capture the patterns between the HR/LR images. At the same time, the added details look very natural as compared to Kong et al.'s [28] results where the discontinuity artifacts can be easily observed. This result validates that our strategy of ignoring the neighboring dependency to leverage computation cost while using the estimation uncertainty to make smooth synthesis is effective.

In the second experiment, we evaluated the proposed SR method under real-time video scenario. A DELL workstation with dual-core 2.8 GHz CPU and 2G memory was used to perform on-line SR. The input video is taken by a Creative webcam with an 160×120

Image and Video Super Resolution Techniques. Figure 5. Illustration of Alpha Matting and Alpha Channel SR.

Image and Video Super Resolution Techniques. **Figure 6.** Single image SR examples for learning-based method.

resolution. The selected output is four-by-four time of the input. We can achieve about 10 fps which demonstrated that the proposed method is very computationally efficient.

Reconstruction-Based Method

To evaluate the performance of second proposed SR method, we also did single image SR with 2 × 2 magnification. Since the objective is not to reduce the estimation error as that used in the learning-based method, the proposed functional interpolation-based SR method only reduced the RMS measure by 8%. However, this does not necessarily say that the proposed method is ineffective. As can be seen from Fig. 7, the proposed method can recover some subtle structures, and thus enhance the image quality for the entire image. Compared with the results of bi-cubic interpolation and back-projection, the chessboard artifact is greatly reduced without introducing blur or ringing effect. This validates the importance of edge smoothness constrain for SR reconstruction process.

Discussion

Comparing the two proposed SR methods, it is observed that the first learning-based method produces less abrupt pixel value changes in locally smooth regions; hence, the generated SR image looks very natural. Besides, the method can be implemented very efficiently to allow for real-time applications. On the other side, the second functional interpolation-based method usually generates more perceptually pleasing SR images by producing noticeable edge sharpness enhancement. For images with fine texture region, the results by the second method sometimes look artificial. However, this can be improved by applying post-processing, such as the back-projection method [16,17].

Conclusions and Future Work

In this essay we introduce two image SR methods that are both capable for single image and video SR. In the first method, a learning-based approach is presented. The Gaussian process regression technique is applied to model the co-occurrence patterns between HR/LR images, and an RBF kernel-based mapping function is built to perform fast image SR. In the second approach, a functional interpolation-based method is described. We use alpha matting decomposition and alpha channel SR to utilize the edge smoothness prior to constrain the reconstruction process. Perceptually appealing results for a large variety of images are obtained.

Image and Video Super Resolution Techniques. **Figure 7.** Single image SR examples for reconstruction-based method.

References

1. M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," IEEE Transactions on Image Process, No. 3, Mar. 1995, pp. 247–258.
2. R. Crochiere and L. Rabiner, "Interpolation and decimation of digital signals – a turorial review," Proceedings of IEEE, No. 3, pp. 300–331, Mar. 1981.
3. S.C. Park, M.K. Park, and M.G. Kang, "Super-resolution image reconstruction: a technical overview," IEEE Signal Processing Magazine, pp. 21–36, 2003.
4. R. Tsai and T. Huang, "Multiframe image restoration and registration," Advances in Computer Vision and Image Processing (JAI Press), pp. 317–339, 1984.
5. S. Borman and R. Stevenson, "Super-resolution from image sequences-a review," Proc. of Midwest Symposium on Circuits and Systems, pp. 374–378, 1998.
6. C. Jiji and C. Subhasis, "Single-frame image super-resolution through contourlet learning," EURASIP Journal on Applied Signal Processing, p. 73767(11), 2006.
7. M.K. Ng and N.K. Bose, "Analysis of displacement errors in high-resolution image reconstruction with multisensors," IEEE Transactions on Circuits and Systems (Part I), No. 6, pp. 806–813, 2002.
8. M.K. Ng and N.K. Bose, "Fast color image restoration with multisensors," International Journal of Imaging Systems and Technoloy, No. 5, pp. 189–197, 2002.

9. N. Nguyen, P. Milanfar, and G. Golub, "A computationally efficient super-resolution image reconstruction algorithm," IEEE Transactions on Image Processing, No. 4, pp. 573–583, 2001.
10. R.R. Schultz and R.L. Stevenson, "A bayesian approach to image expansion for improved definition," IEEE Transactions on Image Processing, No. 3, pp. 233–242, 1994.
11. D. Rajan and S. Chaudhuri, "An mrf-based approach to generation of super-resolution images from blurred observations," Journal of Mathematical Imaging and Vision, No. 1, pp. 5–15, 2002.
12. M. Elad and A. Feuer, "Restoration of a single super-resolution image from several blurred, noisy and undersampled measured images," IEEE Transactions on Image Processing, No. 12, pp. 1646–1658, 1997.
13. N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," IEEE Transactions on Image Processing, pp. 1299–1308, Sept. 2001.
14. M.K. Ng, J. Koo, and N.K. Bose, "Constrained total leastsquares computations for high-resolution image reconstruction with multisensors," International Journal of Imaging Systems and Technology, No. 1, 2002, pp. 35–42.
15. S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution image sequences. a comprehensive review with directions for future research," Laboratory Image and Signal Analysis, University of Notre Dame, Technical Report, 1998.
16. M. Irani and S. Peleg, "Improving resolution by image registration," CVGIP: Graphical Models and Image Processing, No. 3, 1991, pp. 231–239.
17. M. Irani and S. Peleg, "Motion analysis for image enhancement: resolution, occlusion, and transparency," Journal of Visual Communication and Image Representation, No. 4, 1993, p. 324–335.
18. W.T. Freeman, E.C. Pasztor, and O.T. Carmichael., "Learning low-level vision," IJCV, No. 1, 2000, pp. 25–47.
19. J. Sun, H. Tao, and H. Shum, "Image hallucination with primal sketch priors," Proceedings of the IEEE CVPR'03, pp. 729–736, 2003.
20. C.V. Jiji, M.V. Joshi, and S. Chaudhuri, "Single-frame image super-resolution using learned wavelet coefficients," International Journal of Imaging Systems and Technology, No. 3, 2004, pp. 105–112.
21. D. Capel and A. Zisserman, "Super-resolution from multiple views using learnt image models," Proceedings of the IEEE CVPR'01, pp. 627–634, December 2001.
22. Q. Wang, X. Tang, and H. Shum, "Patch based blind image super resolution," Proceedings of ICCV'05, No. 1, pp. 709–716, 2005.
23. D.Y.H. Chang and Y. Xiong, "Super-resolution through neighbor embedding," Proceedings of CVPR'04, 2004, pp. 275–282.
24. B. Gunturk, A. Batur, Y. Altunbasak, M. Hayes, and R.M. Mersereau, "Eigenface-domain super-resolution for face recognition," IEEE Transactions on Image Processing, no. 5, 2003, pp. 597–606.
25. X. Wang and X. Tang, "Hallucinating face by eigentransformation with distortion reduction," Proceedings of ICBA'04, pp. 88–94, 2004.
26. C.V. Jiji and S. Chaudhuri, "Pca-based generalized interpolation for image super-resolution," Proceedings of Indian Conference on Vision, Graphics & Image Processing'04, pp. 139–144, 2004.
27. G. Dalley, B. Freeman, and J. Marks, "Single-frame text super-resolution: a bayesian approach," Proceedings of IEEE ICIP'04, pp. 3295–3298, 2004.
28. D. Kong, M. Han, W. Xu, H. Tao, and Y. Gong, "A conditional random field model for video super-resolution," Proceedings of ICPR'06, pp. 619–622, 2006.
29. Z. Lin and H. Shum, "Fundamental limits of reconstruction-based super-resolution algorithms under local translation," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), no. 1, 2004, pp. 83–97.
30. M. Ben-Ezra, A. Zomet, and S. Nayar, "Jitter camera: High resolution video from a low resolution detector," Proceedings of IEEE CVPR'04, pp. 135–142, Jun. 2004.
31. C. Bishop, A. Blake, and B. Marthi, "Super-resolution enhancement of video," Proceedings of the Artificial Intelligence and Statistics, 2003.
32. C. Williams and C. Rasmussen, "Gaussian processes for regression," Advances in Neural Information Processing Systems, MIT Press., Cambridge, MA, 1996, pp. 514–520.
33. S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong, "Soft edge smoothness prior for alpha channel super resolution," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2007.
34. R. Hardie, K. Barnard, and J. Bognar, "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system," Optical Engineering, No. 1, Jan. 1998, pp. 247–260.
35. S. Farsiu, M.Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," IEEE Transactions on Image Processing, pp. 1327–1344, 2004.
36. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting," Proceedings of the IEEE CVPR'06, 2006.

Image Classification

► Detection of High-Level Concepts in Multimedia

Image Compression and Coding

Oge Marques
Florida Atlantic University, Boca Raton, FL, USA

Synonyms

► Visual data compression

Definition

Image compression deals with reducing the amount of data required to represent a digital image by removing of redundant data.

Introduction

Images can be represented in digital format in many ways. Encoding the contents of a 2-D image in a raw bitmap (raster) format is usually not economical and may result in very large files. Since raw image representations usually require a large amount of storage space (and proportionally long transmission times in the case of file uploads/ downloads), most image file formats employ some type of compression. The need to save storage space and shorten transmission time, as well as the human visual system tolerance to a modest amount of loss, have been the driving factors behind image compression techniques.

Compression methods can be *lossy*, when a tolerable degree of deterioration in the visual quality of the resulting image is acceptable, or *lossless*, when the image is encoded in its full quality. The overall results of the compression process, both in terms of storage savings – usually expressed numerically in terms of compression ratio (CR) or bits per pixel (bpp) – as well as resulting quality loss (for the case of lossy techniques) may vary depending on the technique, format, options (such as the quality setting for JPEG), and the image contents. As a general guideline, lossy compression should be used for general purpose photographic images, whereas lossless compression should be preferred when dealing with line art, technical drawings, cartoons, etc. or images in which no loss of detail may be tolerable (most notably, space images and medical images).

We will review the most important concepts behind image compression and coding techniques and survey some of the most popular algorithms and standards.

Fundamentals of Visual Data Compression

The general problem of image compression is to reduce the amount of data required to represent a digital image or video and the underlying basis of the reduction process is the removal of redundant data. Mathematically, visual data compression typically involves transforming (encoding) a 2-D pixel array into a statistically uncorrelated data set. This transformation is applied prior to storage or transmission. At some later time, the compressed image is decompressed to reconstruct the original image information (preserving or lossless techniques) or an approximation of it (lossy techniques).

Redundancy

Data compression is the process of reducing the amount of data required to represent a given quantity of information. Different amounts of data might be used to communicate the same amount of information. If the same information can be represented using different amounts of data, it is reasonable to believe that the representation that requires more data contains what is technically called *data redundancy*.

Image compression and coding techniques explore three types of redundancies: *coding* redundancy, *interpixel* (spatial) redundancy, and *psychovisual* redundancy. The way each of them is explored is briefly described below.

- *Coding redundancy*: consists in using variable-length codewords selected as to match the statistics of the original source, in this case, the image itself or a processed version of its pixel values. This type of coding is always reversible and usually implemented using look-up tables (LUTs). Examples of image coding schemes that explore coding redundancy are the Huffman codes and the arithmetic coding technique.
- *Interpixel redundancy*: this type of redundancy – sometimes called spatial redundancy, interframe redundancy, or geometric redundancy – exploits the fact that an image very often contains strongly correlated pixels, in other words, large regions whose pixel values are the same or almost the same. This redundancy can be explored in several ways, one of which is by predicting a pixel value based on the values of its neighboring pixels. In order to do so, the original 2-D array of pixels is usually mapped into a different format, e.g., an array of differences between adjacent pixels. If the original image pixels can be reconstructed from the transformed data set the mapping is said to be reversible. Examples of compression techniques that explore the interpixel redundancy include: Constant Area Coding (CAC), (1-D or 2-D) Run-Length Encoding (RLE) techniques, and many predictive coding algorithms such as Differential Pulse Code Modulation (DPCM).
- *Psychovisual redundancy*: many experiments on the psychophysical aspects of human vision have proven that the human eye does not respond with equal sensitivity to all incoming visual information; some pieces of information are more important than others. The knowledge of which particular types of information are more or less relevant to the final human user have led to image and video compression techniques that aim at eliminating or

reducing any amount of data that is psychovisually redundant. The end result of applying these techniques is a compressed image file, whose size and quality are smaller than the original information, but whose resulting quality is still acceptable for the application at hand. The loss of quality that ensues as a byproduct of such techniques is frequently called *quantization*, as to indicate that a wider range of input values is normally mapped into a narrower range of output values thorough an irreversible process. In order to establish the nature and extent of information loss, different fidelity criteria (some objective such as root mean square (RMS) error, some subjective, such as pair-wise comparison of two images encoded with different quality settings) can be used. Most of the image coding algorithms in use today exploit this type of redundancy, such as the Discrete Cosine Transform (DCT)-based algorithm at the heart of the JPEG encoding standard.

Image Compression and Coding Models

Figure 1 shows a general image compression model. It consists of a source encoder, a channel encoder, the storage or transmission media (also referred to as *channel*), a channel decoder, and a source decoder. The source encoder reduces or eliminates any redundancies in the input image, which usually leads to bit savings. Source encoding techniques are the primary focus of this discussion. The channel encoder increase noise immunity of source encoder's output, usually adding extra bits to achieve its goals. If the channel is noise-free, the channel encoder and decoder may be omitted. At the receiver's side, the channel and source decoder perform the opposite functions and ultimately recover (an approximation of) the original image.

Figure 2 shows the source encoder in further detail. Its main components are:

- *Mapper:* transforms the input data into a (usually nonvisual) format designed to reduce interpixel redundancies in the input image. This operation is generally reversible and may or may not directly reduce the amount of data required to represent the image.
- *Quantizer:* reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. Reduces the psychovisual redundancies of the input image. This operation is not reversible and must be omitted if lossless compression is desired.
- *Symbol (entropy) encoder:* creates a fixed- or variable-length code to represent the quantizer's output and maps the output in accordance with the code. In most cases, a variable-length code is used. This operation is reversible.

Error-Free Compression

Error-free compression techniques usually rely on entropy-based encoding algorithms. The concept of entropy is mathematically described in (1):

$$H(\mathbf{z}) = -\sum_{j=1}^{J} P(a_j) \log P(a_j) \qquad (1)$$

where:

- a_j is a symbol produced by the information source
- $P(a_j)$ is the probability of that symbol

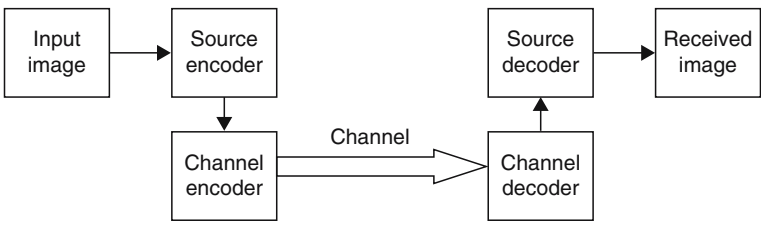


Image Compression and Coding. Figure 1. A general image compression model.

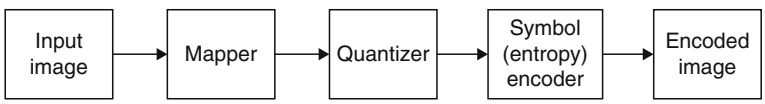


Image Compression and Coding. Figure 2. Source encoder.

- J is the total number of different symbols
- $H(\mathbf{z})$ is the entropy of the source.

The concept of entropy provides an upper bound on how much compression can be achieved, given the probability distribution of the source. In other words, it establishes a theoretical limit on the amount of lossless compression that can be achieved using entropy encoding techniques alone.

Variable Length Coding (VLC)

Most entropy-based encoding techniques rely on assigning variable-length codewords to each symbol, whereas the most likely symbols are assigned shorter codewords. In the case of image coding, the symbols may be raw pixel values or the numerical values obtained at the output of the mapper stage (e.g., differences between consecutive pixels, run-lengths, etc.). The most popular entropy-based encoding technique is the Huffman code [1]. It provides the least amount of information units (bits) per source symbol. It is described in more detail in a separate short article.

Run-Length Encoding (RLE)

RLE is one of the simplest data compression techniques. It consists of replacing a sequence (run) of identical symbols by a pair containing the symbol and the run length. It is used as the primary compression technique in the 1-D CCITT Group 3 fax standard and in conjunction with other techniques in the JPEG image compression standard (described in a separate short article).

Differential Coding

Differential coding techniques explore the interpixel redundancy in digital images. The basic idea consists of applying a simple difference operator to neighboring pixels to calculate a difference image, whose values are likely to follow within a much narrower range than the original gray-level range. As a consequence of this narrower distribution – and consequently reduced entropy – Huffman coding or other VLC schemes will produce shorter codewords for the difference image.

Predictive Coding

Predictive coding techniques constitute another example of exploration of interpixel redundancy, in which the basic idea is to encode only the new information in each pixel. This new information is usually defined as the difference between the actual and the predicted value of that pixel.

Figure 3 shows the main blocks of a lossless predictive encoder. The key component is the predictor, whose function is to generate an estimated (predicted) value for each pixel from the input image based on previous pixel values. The predictor's output is rounded to the nearest integer and compared with the actual pixel value: the difference between the two – called *prediction error* – is then encoded by a VLC encoder. Since prediction errors are likely to be smaller than the original pixel values, the VLC encoder will likely generate shorter codewords.

There are several local, global, and adaptive prediction algorithms in the literature. In most cases, the predicted pixel value is a linear combination of previous pixels.

Dictionary-Based Coding

Dictionary-based coding techniques are based on the idea of incrementally building a dictionary (table) while receiving the data. Unlike VLC techniques, dictionary-based techniques use fixed-length codewords to represent variable-length strings of symbols that commonly occur together. Consequently, there is no need to calculate, store, or transmit the probability distribution of the source, which makes these algorithms extremely convenient and popular. The best-known variant of dictionary-based coding algorithms is the LZW (Lempel-Ziv-Welch) encoding scheme [2], used in popular multimedia file formats such as GIF, TIFF, and PDF.

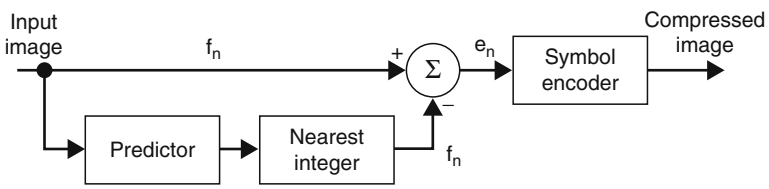


Image Compression and Coding. Figure 3. Lossless predictive encoder.

Lossy Compression

Lossy compression techniques deliberately introduce a certain amount of distortion to the encoded image, exploring the psychovisual redundancies of the original image. These techniques must find an appropriate balance between the amount of error (loss) and the resulting bit savings.

Quantization

The quantization stage is at the core of any lossy image encoding algorithm. Quantization, in at the encoder side, means partitioning of the input data range into a smaller set of values. There are two main types of quantizers: scalar quantizers and vector quantizers. A scalar quantizer partitions the domain of input values into a smaller number of intervals. If the output intervals are equally spaced, which is the simplest way to do it, the process is called *uniform scalar quantization*; otherwise, for reasons usually related to minimization of total distortion, it is called *nonuniform scalar quantization*. One of the most popular nonuniform quantizers is the Lloyd-Max quantizer. Vector quantization (VQ) techniques [3] extend the basic principles of scalar quantization to multiple dimensions. Because of its fast lookup capabilities at the decoder side, VQ-based coding schemes are particularly attractive to multimedia applications.

Transform Coding

The techniques discussed so far work directly on the pixel values and are usually called *spatial domain techniques*. Transform coding techniques use a reversible, linear mathematical transform to map the pixel values onto a set of coefficients, which are then quantized and encoded. The key factor behind the success of transform-based coding schemes many of the resulting coefficients for most natural images have small magnitudes and can be quantized (or discarded altogether) without causing significant distortion in the decoded image. Different mathematical transforms, such as Fourier (DFT), Walsh-Hadamard (WHT), and Karhunen-Loève (KLT), have been considered for the task. For compression purposes, the higher the capability of compressing information in fewer coefficients, the better the transform; for that reason, the Discrete Cosine Transform (DCT) [4] has become the most widely used transform coding technique.

Transform coding algorithms (Fig. 4) usually start by partitioning the original image into subimages (blocks) of small size (usually 8×8). For each block the transform coefficients are calculated, effectively converting the original 8×8 array of pixel values into an array of coefficients within which the coefficients closer to the top-left corner usually contain most of the information needed to quantize and encode (and eventually perform the reverse process at the decoder's side) the image with little perceptual distortion. The resulting coefficients are then quantized and the output of the quantizer is used by a (combination of) symbol encoding technique(s) to produce the output bitstream representing the encoded image. At the decoder's side, the reverse process takes place, with the obvious difference that the "dequantization" stage will only generate an approximated version of the original coefficient values; in other words, whatever loss was introduced by the quantizer in the encoder stage is not reversible.

Wavelet Coding

Wavelet coding techniques are also based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. The main difference between wavelet coding and DCT-based coding (Fig. 4) is the omission of the first stage. Because wavelet transforms are capable of representing an input signal with multiple levels of resolution, and yet maintain the useful compaction properties of the DCT, the subdivision of the input image into smaller subimages is no longer necessary. Wavelet coding has been at the core of the latest image compression standards, most notably JPEG 2000, which is discussed in a separate short article.

Image Compression Standards

Work on international standards for image compression started in the late 1970s with the CCITT

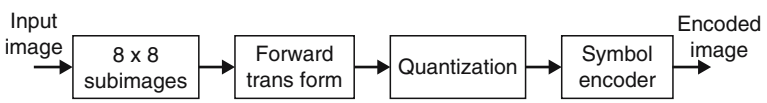


Image Compression and Coding. **Figure 4.** Transform coding.

(currently ITU-T) need to standardize binary image compression algorithms for Group 3 facsimile communications. Since then, many other committees and standards have been formed to produce *de jure* standards (such as JPEG), while several commercially successful initiatives have effectively become *de facto* standards (such as GIF). Image compression standards bring about many benefits, such as: (1) easier exchange of image files between different devices and applications; (2) reuse of existing hardware and software for a wider array of products; (3) existence of benchmarks and reference data sets for new and alternative developments.

Binary Image Compression Standards [5]

Work on binary image compression standards was initially motivated by CCITT Group 3 and 4 facsimile standards. The Group 3 standard uses a non-adaptive, 1-D RLE technique in which the last K-1 lines of each group of K lines (for K = 2 or 4) are optionally coded in a 2-D manner, using the *Modified Relative Element Address Designate* (MREAD) algorithm. The Group 4 standard uses only the MREAD coding algorithm. Both classes of algorithms are non-adaptive and were optimized for a set of eight test images, containing a mix of representative documents, which sometimes resulted in data expansion when applied to different types of documents (e.g., half-tone images). The Joint Bilevel Image Group (JBIG) [6] – a joint committee of the ITU-T and ISO – has addressed these limitations and proposed two new standards (JBIG and JBIG2) which can be used to compress binary and gray-scale images of up to 6 gray-coded bits/pixel.

Continuous Tone Still Image Compression Standards

For photograph quality images (both grayscale and color), different standards have been proposed, mostly based on lossy compression techniques. The most popular standard in this category, by far, is the JPEG standard [5,7], a lossy, DCT-based coding algorithm. Despite its great popularity and adoption, ranging from digital cameras to the World Wide Web, certain limitations of the original JPEG algorithm have motivated the recent development of two alternative standards, JPEG 2000 and JPEG-LS (lossless). JPEG, JPEG 2000, and JPEG-LS are described in separate short articles.

Cross-References

- ▶ Huffman Coding
- ▶ JPEG
- ▶ JPEG-LS

References

1. D.A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proceedings of IRE, Vol. 40, No. 10, 1952, pp. 1098–1101.
2. T.A. Welch, "A Technique for High-Performance Data Compression," IEEE Computer, Vol. 17, June 1984, pp. 8–19.
3. N.M. Nasrabadi and R.A. King, "Image Coding Using Vector Quantization: A Review," IEEE Transactions on Communications, Vol. 36, No. 8, 1988, pp. 957–971.
4. K.R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications," Academic, New York, 1990.
5. W.B. Pennenbaker and J.L. Mitchell, "JPEG Still Image Data Compression Standard," Van Nostrand Reinhold, New York, 1993.
6. JBIG Home page, http://www.jpeg.org/jbig/Accessed July 13, 2005.
7. G. Wallace, "The JPEG Still Picture Compression Standard," Communications of the ACM, Vol. 34, 1991, pp. 30–44.
8. R.B. Arps, "Bibliography on Binary Image Compression," Proceedings of the IEEE, Vol. 68, No. 7, July 1980, pp. 922–924.

Image Data Representations

Oge Marques
Florida Atlantic University, Boca Raton, FL, USA

Synonyms

- ▶ Representation of color images

Definition

At the most basic level, there are two different ways of encoding the contents of a 2-D image in digital format: raster (also known as bitmap) and vector.

Introduction

Images are represented in digital format in a wide variety of ways. At the most basic level, there are two different ways of encoding the contents of a 2-D image in digital format: *raster* (also known as bitmap) and *vector*. Bitmap representations use one or more two-dimensional arrays of pixels (picture elements), whereas vector representations use a series of drawing commands to represent an image. Each encoding method has its pros and cons: the greatest advantages of

bitmap graphics are their quality and display speed; its main disadvantages include larger memory storage requirements and size dependence (e.g., enlarging a bitmap image may lead to noticeable artifacts). In either case, there is no such a thing as a perfect digital representation of an image. Artifacts due to finite resolution, color mapping, and many others will always be present. The key in selecting an adequate representation is to find a suitable compromise between size (in Bytes), subjective quality, and universality/interoperability of the adopted format or standard. We will review several different image representations used by some of the most common file formats currently available.

Binary (1-Bit) Images

Binary images are encoded as a 2-D array, using one bit per pixel, where a 0 usually means "black" and a 1 means "white" (even though there is no universal agreement on that). The main advantage of this representation (Fig. 1(b)) – usually suitable for images containing simple graphics, text or line art – is its small size.

Gray-Level (8-Bit) Images

Gray-level (also referred to as *monochrome*) images are also encoded as a 2-D array of pixels, using eight bits per pixel, where a pixel value of 0 usually means "black" and a pixel value of 255 means "white," with intermediate values corresponding to varying shades of gray. The total number of gray-levels is larger than the human visual system requirements, making this format a good compromise between subjective visual quality and relatively compact representation and storage. An 8-bit monochrome image (Fig. 1(a)) can also be thought of as a collection of bit-planes, where each plane contains a 1-bit representation of the image at different levels of detail.

Color Images

Representation of color images is significantly more complex and varied. The two most common ways of storing color image contents are: *RGB* representation – in which each pixel is usually represented by a 24-bit number containing the amount of its Red (R), Green (G), and Blue (B) components – and *indexed* representation – where a 2-D array contains indices to a color palette (or look-up table – LUT).

24-Bit (RGB) Color Images

Color images can be represented using three 2-D arrays of same size, one for each color channel: Red (R), Green (G), and Blue (B) (Fig. 2). Each array element contains an 8-bit value indicating the amount of red, green, or blue at that point, in a 0–255 scale. The combination of the three 8-bit values into a 24-bit number allows for 2^{24} (16,777,216, usually referred to as 16 million or 16 M) color combinations. An alternative representation uses 32 bits and includes a fourth channel, called the *alpha channel*, which provides a measure of transparency for each pixel and is widely used in image editing effects.

Indexed Color Images

A problem with 24-bit color representations is backward compatibility with older hardware which may not be able to display the 16 million colors simultaneously. A solution devised before 24-bit color displays and video cards were widely accessible was to adopt an indexed representation, in which a 2-D array of the same size as the image contains indices (pointers) to a color palette (or look-up table – LUT) of fixed maximum size (usually 256 colors) (see Fig. 3).

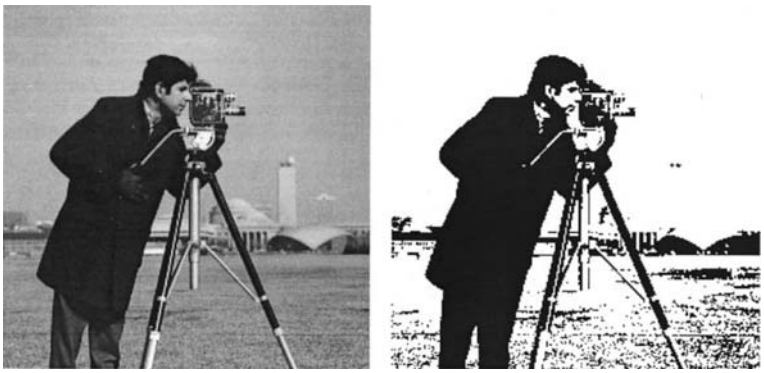


Image Data Representations. Figure 1. (a) A monochrome image (b) Its binary equivalent.

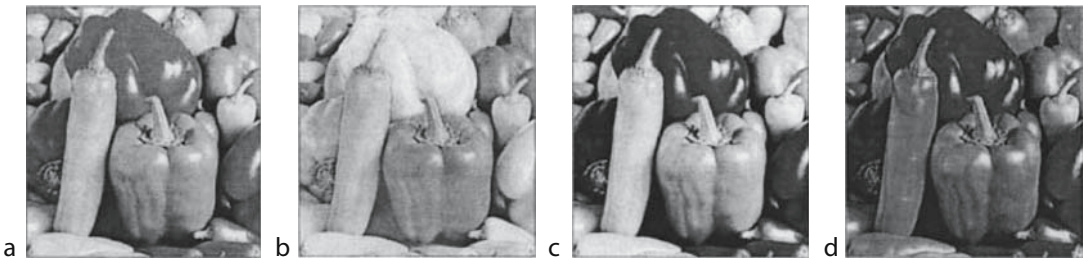


Image Data Representations. **Figure 2.** (a) Color image and its R (b), G (c), and B (d) components.

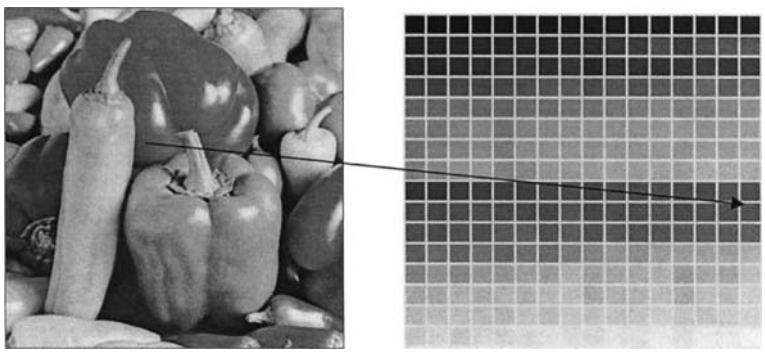


Image Data Representations. **Figure 3.** Indexed color image and its color palette.

Other Color Models

The RGB color model is one of the most popular and straightforward methods for representing the contents of a digital color image, but there are several alternative models, such as: YCbCr – adopted in the JPEG standard – and the CMYK – used to encode color information for printing purposes – among many others.

Compression

Since raw image representations usually require a large amount of storage space (and proportionally long transmission times in the case of file uploads/downloads), most image file formats employ some type of compression. Compression methods can be *lossy* – when a tolerable degree of deterioration in the visual quality of the resulting image is acceptable – or *lossless* – when the image is encoded in its full quality. The overall results of the compression process, both in terms of storage savings – usually expressed as compression ratio – as well as resulting quality loss (for the case of lossy techniques) may vary depending on the technique, format, options (such as the quality setting for JPEG), and the image contents. As a general guideline, lossy compression should be used for general purpose photographic images, whereas lossless compression should be preferred when dealing with line art, drawings, facsimiles, or images in which no loss of detail may be tolerable (most notably, space images and medical images).

Popular File Formats

The selected file formats described below represent some of the most widely used formats at the present time. Research in the field of image compression and encoding remains active and it is possible that other formats may appear and become popular in the future.

Windows BMP

Windows BMP is the native image format in the Windows platform [1]. It is one of the simplest image formats and supports images with 1, 4, 8, 16, 24, and 32 bits per pixel. BMP files are usually uncompressed. Multi-byte integers are stored with the least significant bytes first. The BMP file structure (Fig. 4) contains [2]:

1. A *file header*, implemented as a BITMAPFILEHEADER structure, serves as the signature that identifies the file format (the first two bytes must contain the ASCII characters "B" followed by "M").
2. An *image header*, which may come in two different formats: BITMAPCOREHEADER (for the

Image Data Representations. Figure 4. BMP format: file structure.

old OS/2 BMP format) and BITMAPINFOHEADER (for the much more popular Windows format). This header stores the images width, height, compression method, and number of bits per pixel, among other things.
3. A *color palette*, which contains values for R, G, B, and an additional ("reserved") channel for each pixel.
4. The actual *pixel data*, ordered in rows, from bottom to top. The exact format of the pixel data depends on the number of bits per pixel.

The main advantages of the BMP format are its simplicity, widespread distribution (with Windows) and popularity. Moreover, it is a well-documented patent-free format. Because it is usually uncompressed, its main disadvantage is the resulting file size (in Bytes).

GIF

Graphic Interchange Format (GIF) was originally devised by CompuServe in 1987. It soon became the most widely used format for image storage [2]. The GIF format became popular because it used LZW (Lempel-Ziv-Welch) data compression, a lossless compression algorithm which was more efficient than the run-length encoding (RLE) compression used by (at that time) competing formats such as PCX and MacPaint. Thanks to its improved compression scheme, reasonably large images could be downloaded in a reasonable amount of time, even with very slow modems. GIF's optional interlacing feature, which allows storing image scan lines out of order in such a fashion that even a partially downloaded image is somewhat recognizable, also helped GIF's popularity, allowing a user to stop the download if it was not what was expected.

The first GIF specification was called GIF87a. In 1989, CompuServe released an enhanced version, called GIF89a, which added support for multiple images in a stream and storage of application-specific metadata. The two versions can be distinguished by looking at the first six bytes of the file, whose ASCII characters are "GIF87a" and "GIF89a," respectively.

The LZW compression algorithm on which GIF is based, was covered by US Patent 4,558,302, owned by Unisys, which has led to a legal battle that started when Unisys discovered that GIF used the LZW and announced that they would be seeking royalties on that patent. This led to the development of a patent-free alternative format (PNG, described in a short article) with similar technical characteristics. That patent expired on June 20, 2003, which means that Unisys and CompuServe can no longer collect royalties for use of the GIF format in the United States [3].

The GIF file structure in Fig. 5 contains [2]:

1. A *GIF header*, required, always present at the beginning of the file. It contains a 6-byte signature, either "GIF87a" or "GIF89a."
2. A 7-byte *global screen descriptor*, which specifies the dimensions and background color of the logical screen area in which the individual images in the GIF file are displayed.
3. A *global color table*, an array of structures containing values for R, G, and B for each pixel.
4. One or more *images*, each of which may contain:
   - An image header
   - An optional local color table
   - A collection of data blocks
   - A terminator block
5. A *trailer*.

The GIF89a feature of storing multiple images in one file, accompanied by control data, is used extensively on the Web to produce simple animations and short, low-resolution films, the so-called *animated GIF*. GIF is palette-based, which limits the maximum number of colors to 256. This limitation is usually not an issue for web page logos and design elements such as buttons or banners. For digital photographs, the JPEG format (described in a short article) is preferred.

PNG

The PNG (Portable Network Graphics) [4–6] format was originally designed to replace the GIF format. It is described in more detail in a separate short article.

JPEG

The JPEG format was originally published as a standard (ISO IS 10918-1) by The Joint Photographic Experts Group in 1994. It has become the most widely used format for storing digital photographs ever since. The JPEG specification defines how an image is transformed into a stream of bytes, but not how those bytes are encapsulated in any particular storage medium. Another standard, created by the Independent JPEG Group, called JFIF (JPEG File Interchange Format) specifies how to produce a file suitable for computer storage and transmission from a JPEG stream. JFIF is described in a separate short article.

Even though the original JPEG specification defined four compression modes (sequential, hierarchical, progressive, and lossless), most JPEG files used today employ the sequential mode. The JPEG encoder (Fig. 6) consists of the following main stages:

1. The original RGB color image is converted to an alternative color model (YCbCr) and the color information is subsampled.
2. The image is divided into 8×8 blocks.
3. The 2-D Discrete Cosine Transform (DCT) is applied to each block image; the resulting 64 values are referred to as *DCT coefficients*.
4. DCT coefficients are quantized according to a quantization table; this is the step where acceptable loss is introduced.
5. Quantized DCT coefficients are scanned in a zigzag fashion (from top-left to bottom-right). The resulting sequence is run-length encoded, in preparation for the entropy encoding step.
6. The run-length encoded sequences are converted to variable-length binary codewords using Huffman encoding.

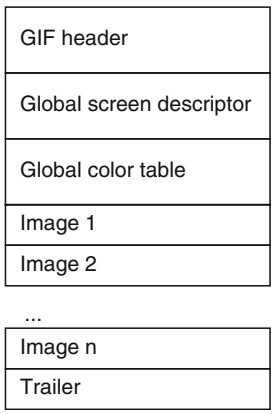


Image Data Representations. **Figure 5.** GIF format: file structure.

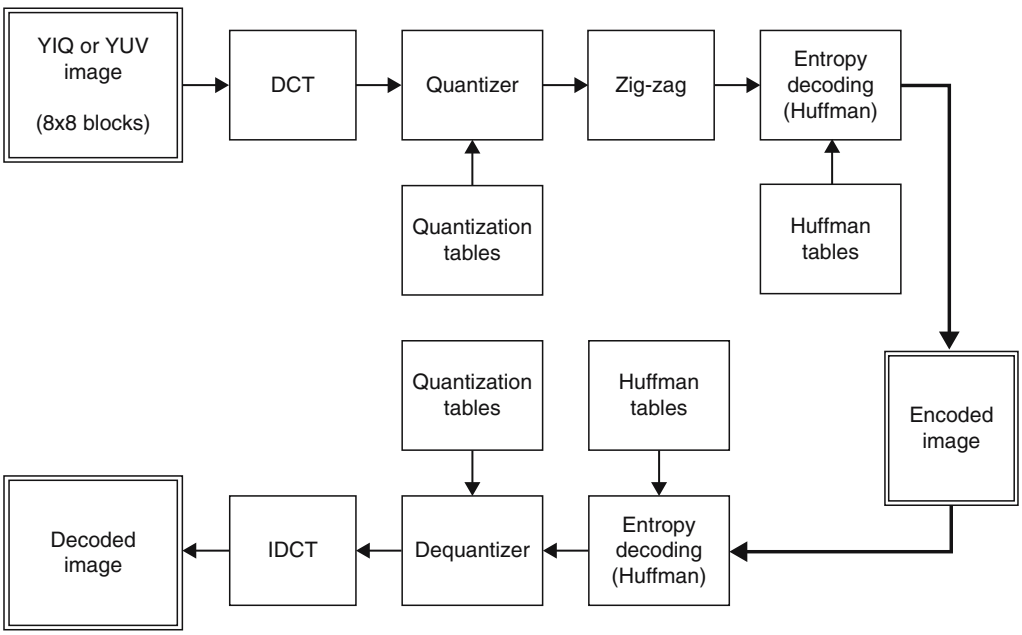


Image Data Representations. **Figure 6.** JPEG encoder and decoder: block diagram.

At the decoder side, the process is reversed; it should be noted that the loss introduced at the quantizer stage in the encoder cannot be canceled at the "dequantizer" stage in the decoder.

JPEG 2000

The JPEG 2000 format [7] is a wavelet-based image compression standard (ISO/IEC 15444–1:2000), created by the Joint Photographic Experts Group committee with the intention of superseding their original DCT-based JPEG standard. The usual file extension is.jp2. It addresses several well-known limitations of the original JPEG algorithm and prepares the way for next-generation imagery applications. Some of its advertised advantages are:

1. Low bitrate compression
2. Superior lossless and lossy compression in a single bitstream
3. Ability to handle very large images without need for tiling
4. Single decompression architecture
5. Error resilience for transmission in noisy environments, such as wireless and the Internet
6. Region of Interest (ROI) coding
7. Metadata mechanisms for incorporating additional non-image data as part of the file

JPEG 2000 is not yet widely supported in web browsers, and hence is not generally used on the World Wide Web.

TIFF

TIFF (Tagged Image File Format) is another popular image file format, developed in the 1980s by Aldus Corp. The TIFF format includes a number of options that can be used to attach additional information (such as images encoded in another file format); these options can be exercised by including specific "tags" in the header. Many of these tags convey basic image information, such as its size, but others define how the data is arranged and various image compression options. As a result, TIFF can be used as a container for JPEG compressed images, and in this respect is completely universal. However, because of the complexity involved in supporting all of its options, a lowest common denominator variant became "the" TIFF, and even today the vast majority of TIFF files, and the programs that read them, are based on a simple 32-bit uncompressed image.

DNG

Digital Negative Specification (DNG) is a recently announced royalty-free raw image file format introduced by Adobe Systems as a response to demand for a unifying digital camera raw file format [8].

Other Formats

It is virtually impossible to cover all image file formats. Other image formats not described here include: ANIM, ART, CGM, CIN, DjVu, DPX, EPS, EXIF, GDF, ILBM, MNG, PCX, PICT, PSD, PSP, XBM, and XPM, among many others. Please refer to [3] for an updated list and useful links.

Cross-References

▶ JFIF
▶ PNG

References

1. MSDN, http://msdn.microsoft.com/library/default.asp?url = /library/en-us/gdi/bitmaps_99ir.asp. Accessed April 25, 2005.
2. J. Miano, Compressed Image File Formats, Addison-Wesley, Reading, MA, USA,1999.
3. Wikipedia, http://en.wikipedia.org/wiki/Graphics_file_format. Accessed April 13, 2005.
4. Official PNG home page, http://www.libpng.org/pub/png/. Accessed April 25, 2005.
5. W3C PNG page, http://www.w3.org/Graphics/PNG/. Accessed April 25, 2005.
6. Official PNG specification, http://www.libpng.org/pub/png/spec/iso/, Accessed April 29, 2005.
7. JPEG 2000, http://www.jpeg.org/jpeg2000/, Accessed April 29, 2005.
8. B. Fraser, "Real World Camera Raw with Adobe Photoshop CS2," Peachpit, 2005.

Image Device Movements

▶ Camera Motions

Image Fidelity Measurement

▶ Advances in Image and Video Quality Assesment

Image Fidelity Optimization using SSIM

► Structural Similarity Index Based Optimization

Image Inpainting

Definition

Image inpainting refers to the process of filling-in missing data in a designated region of the visual input.

Image inpainting [1–3] refers to the process of filling-in missing data in a designated region of the visual input (Fig. 1). The object of the process is to reconstruct missing parts or damaged image in such a way that the inpainted region cannot be detected by a causal observer. Applications range from the reconstruction of missing blocks introduced by packet loss during wireless transmission, reversing of impairments, such as cracks, scratches, and dirt, in scanned photographs and digitized artwork images, to removal/introduction of image objects such as logos, stamped dates, text, persons, and special effects on the scene. Typically, after the user selects the region to be restored, the inpainting algorithm automatically repairs the damaged area by means of image interpolation.

To recover the color, structural, and textural content in a large damaged area, inpainted (output) pixels are calculated using the available data from the surrounding undamaged areas. The required input can be automatically determined by the inpainting technique or supplied by the user. Since different inpainting techniques focus on pure texture or pure structure restoration, both the quality and cost of the inpainting process differ significantly. For example, exemplar-based techniques effectively generate new texture by sampling and copying color values from an undamaged source [3]. Such an approach produces good results in replicating consistent texture seen in artificially generated imagery, but fails when it comes to reconstruct missing parts in photographs of natural scenes. This is due to the fact that most image areas consist of both texture and structure. Boundaries between image regions constitute structural (edge) information which is a complex, nonlinear phenomenon produced by blending together different textures. Therefore, it is not surprising that the state-of-the-art inpainting methods attempt to simultaneously perform texture and structure filling-in [1–3].

Cross-References

► Color Image Filtering and Enhancement
► Inpainting in Virtual Restoration of Artworks
► Video Inpainting

References

1. S.-D. Rane, G. Sapiro, and M. Bertalmio, "Structure and Texture Filling-In of Missing Image Blocks in Wireless Transmission and Compression Applications," IEEE Transactions on Image Processing, Vol. 12, No. 3, March 2003, pp. 296–303.
2. C.-A.-Z. Barcelos, M.-A. Batista, A.-M. Martins, and A.-C. Nogueira, "Level Lines Continuation Based Digital Inpainting," Proceedings of XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'04), October 2004, pp. 50–57.
3. A. Criminisi, P. Perez, and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," IEEE Transactions on Image Processing, Vol. 13, No. 9, September 2004, pp. 1200–1212.

Image Inpainting. Figure 1. Image inpainting: (a) original image, (b) damaged image with missing rows (*blocks*) due to wireless transmission, (c) reconstructed image by an inpainting technique.

Image Retrieval

William I. Grosky
University of Michigan at Dearborn, Dearborn, MI, USA

Definition

Image retrieval techniques integrate both low-level visual features, addressing the more detailed perceptual aspects, and high-level semantic features underlying the more general conceptual aspects of visual data.

Introduction

The emergence of multimedia technology and the rapid growth in the number and type of multimedia assets controlled by public and private entities, as well as the expanding range of image and video documents appearing on the web, have attracted significant research efforts in providing tools for effective retrieval and management of visual data. Image retrieval is based on the availability of a representation scheme of image content. Image content descriptors may be visual features such as color, texture, shape, and spatial relationships, or semantic primitives.

Conventional information retrieval is based solely on text, and these approaches to textual information retrieval have been transplanted into image retrieval in a variety of ways, including the representation of an image as a vector of feature values. However, "a picture is worth a 1,000 words." Image contents are much more versatile compared with text, and the amount of visual data is already enormous and still expanding very rapidly. Hoping to cope with these special characteristics of visual data, content-based image retrieval methods have been introduced. It has been widely recognized that the family of image retrieval techniques should become an integration of both low-level visual features, addressing the more detailed perceptual aspects, and high-level semantic features underlying the more general conceptual aspects of visual data. Neither of these two types of features is sufficient to retrieve or manage visual data in an effective or efficient way. Although efforts have been devoted to combining these two aspects of visual data, the gap between them is still a huge barrier in front of researchers. Intuitive and heuristic approaches do not provide us with satisfactory performance. Therefore, there is an urgent need of finding and managing the latent correlation between low-level features and high-level concepts. How to bridge this gap between visual features and semantic features has been a major challenge in this research field.

The different types of information that are normally associated with images are:

1. Content-independent metadata: data that is not directly concerned with image content, but related to it. Examples are image format, author's name, date, and location.
2. Content-based metadata:
   - Non-information-bearing metadata: data referring to low-level or intermediate-level features, such as color, texture, shape, spatial relationships, and their various combinations. This information can easily be computed from the raw data.
   - Information-bearing metadata: data referring to content semantics, concerned with relationships of image entities to real-world entities. This type of information, such as that a particular building appearing in an image is the *Empire State Building*, cannot usually be derived from the raw data, and must then be supplied by other means, perhaps by inheriting this semantic label from another image, where a similar-appearing building has already been identified.

Low-level visual features such as color, texture, shape, and spatial relationships are directly related to perceptual aspects of image content. Since it is usually easy to extract and represent these features and fairly convenient to design similarity measures by using the statistical properties of these features, a variety of content-based image retrieval techniques have been proposed. High-level concepts, however, are not extracted directly from visual contents, but they represent the relatively more important meanings of objects and scenes in the images that are perceived by human beings. These conceptual aspects are more closely related to users' preferences and subjectivity. Concepts may vary significantly in different circumstances. Subtle changes in the semantics may lead to dramatic conceptual differences. Needless to say, it is a very challenging task to extract and manage meaningful semantics and to make use of them to achieve more intelligent and user-friendly retrieval.

High-level conceptual information is normally represented by using text descriptors. Traditional

indexing for image retrieval is text-based. In certain content-based retrieval techniques, text descriptors are also used to model perceptual aspects. However, the inadequacy of text description is very obvious:

1. It is difficult for text to capture the perceptual saliency of visual features.
2. It is rather difficult to characterize certain entities, attributes, roles, or events by means of text only.
3. Text is not well suited for modeling the correlation between perceptual and conceptual features.
4. Text descriptions reflect the subjectivity of the annotator and the annotation process is prone to be inconsistent, incomplete, ambiguous, and very difficult to be automated.

Although it is an obvious fact that image contents are much more complicated than textual data stored in traditional databases, there is an even greater demand for retrieval and management tools for visual data, since visual information is a more capable medium of conveying ideas and is more closely related to human perception of the real world. Image retrieval techniques should provide support for user queries in an effective and efficient way, just as conventional information retrieval does for textual retrieval. In general, image retrieval can be categorized into the following types:

1. Exact Matching — This category is applicable only to static environments or environments in which features of the images do not evolve over an extended period of time. Databases containing industrial and architectural drawings or electronics schematics are examples of such environments.
2. Low-Level Similarity-Based Searching — In most cases, it is difficult to determine which images best satisfy the query. Different users may have different needs and wants. Even the same user may have different preferences under different circumstances. Thus, it is desirable to return the top several similar images based on the similarity measure, so as to give users a good sampling. The similarity measure is generally based on simple feature matching and it is quite common for the user to interact with the system so as to indicate to it the quality of each of the returned matches, which helps the system adapt to the users' preferences. Figure 1 shows three images which a particular user may find similar to each other. In general, this problem has been well-studied for many years.
3. High-Level Semantic-Based Searching — In this case, the notion of similarity is not based on simple feature matching and usually results from extended user interaction with the system. Figure 2 shows two images whose low-level features are quite different, yet could be semantically similar to a particular user as examples of peaceful scenes. Research in this area is quite active, yet still in its infancy. Many important breakthroughs are yet to be made.

For either type of retrieval, the dynamic and versatile characteristics of image content require expensive computations and sophisticated methodologies in the areas of computer vision, image processing, data visualization, indexing, and similarity measurement. In order to manage image data effectively and efficiently, many schemes for data modeling and image representation have been proposed. Typically, each of these schemes builds a symbolic image for each given physical image to provide logical and physical data independence. Symbolic images are then used in conjunction with various index structures as proxies for image comparisons to reduce the searching scope. The high-dimensional visual data is usually reduced into a lower-dimensional subspace so that it is easier to

Image Retrieval. Figure 1. Three similar images based on simple feature matching.

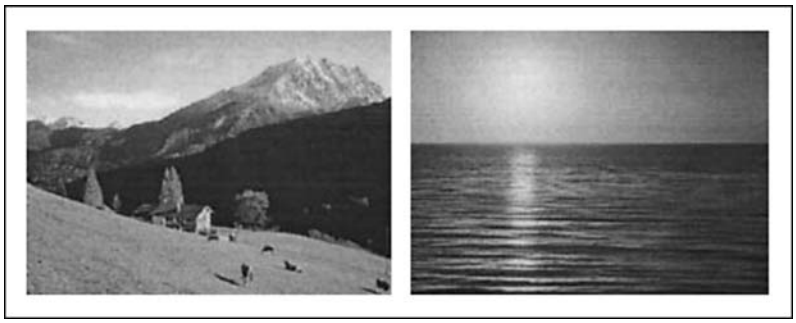


Image Retrieval. Figure 2. Two semantically similar images ("peaceful scenes") with different low-level features.

index and manage the visual contents. Once the similarity measure has been determined, indexes of corresponding images are located in the image space and those images are retrieved from the database. Due to the lack of any unified framework for image representation and retrieval, certain methods may perform better than others under differing query situations. Therefore, these schemes and retrieval techniques have to be somehow integrated and adjusted on the fly to facilitate effective and efficient image data management.

Existing Techniques

Visual feature extraction is the basis of any content-based image retrieval technique. Widely used features include color, texture, shape, and spatial relationships. Because of the subjectivity of perception and the complex composition of visual data, there does not exist a single best representation for any given visual feature. Multiple approaches have been introduced for each of these visual features and each of them characterizes the feature from a different perspective.

Color is one of the most widely used visual features in content-based image retrieval. It is relatively robust and simple to represent. Various studies of color perception and color spaces have been proposed, in order to find color-based techniques that are more closely aligned with the ways that humans perceive color. The color histogram has been the most commonly used representation technique, statistically describing combined probabilistic properties of the various color channels (such as the (R)ed, (G)reen, and (B)lue channels), by capturing the number of pixels having particular properties. For example, a color histogram might describe the number of pixels of each red channel value in the range [0, 255]. Figure 3 shows an image and three of its derived color histograms, where the particular channel values are shown along the x-axis, the numbers of pixels are shown along the y-axis, and the particular color channel used is indicated in each histogram. It is well known that histograms lose information related to the spatial distribution of colors and that two very different images can have very similar histograms. There has been much work done in extending histograms to capture such spatial information. Two of the well-known approaches for this are correlograms and anglograms. Correlograms capture the distribution of colors of pixels in particular areas around pixels of particular colors, while anglograms capture a particular signature of the spatial arrangement of areas (single pixels or blocks of pixels) having common properties, such as similar colors. We note that anglograms also can be used for texture and shape features.

Texture refers to the patterns in an image that present the properties of homogeneity that do not result from the presence of a single color or intensity value. It is a powerful discriminating feature, present almost everywhere in nature. However, it is almost impossible to describe texture in words, because it is virtually a statistical and structural property. There are three major categories of texture-based techniques, namely, *probabilistic/statistical*, *spectral*, and *structural* approaches. Probabilistic methods treat texture patterns as samples of certain random fields and extract texture features from these properties. Spectral approaches involve the sub-band decomposition of images into different channels, and the analysis of spatial frequency content in each of these sub-bands in order to extract texture features. Structural techniques model texture features based on heuristic rules of spatial placements of primitive image elements that attempt to mimic human perception of textural patterns.

The well known *Tamura features* include *coarseness*, *contrast*, *directionality*, *line-likeness*, *regularity*,

and *roughness*. Different researchers have selected different subsets of these heuristic descriptors. It is believed that the combination of *contrast*, *coarseness*, and *directionality* best represents the textural patterns of color images. Figure 4 illustrates various textures.

Shape representation is normally required to be invariant to *translation*, *rotation*, and *scaling*. In general, shape representations can be categorized as either *boundary-based* or *region-based*. A boundary-based representation uses only the outer boundary characteristics of the entities, while a region-based representation uses the entire region. Shape features may also be *local* or *global*. A shape feature is local if it is derived from some proper subpart of an object, while it is global if it is derived from the entire object. See Fig. 5 for an illustration of these concepts.

A combination of the above features are extracted from each image and transformed into a point of a high-dimensional vector space. Using this representation, the many techniques developed by the information retrieval community can be used to advantage. As the dimensionality of the underlying space is still quite high, however, the many disadvantages caused by the *curse of dimensionality* also prevail.

Originally devised in the context of estimating probability density functions in high-dimensional spaces, the curse of dimensionality expresses itself in high-dimensional indexing by causing log time complexity indexing approaches to behave no better than linear search as the dimensionality of the search space increases. This is why there has been so much effort spent in the development of efficient high-dimensional indexing techniques, on the one hand, and in dimensional reduction techniques which capture the salient semantics, on the other hand.

As the ultimate goal of image retrieval is to serve the needs and wants of users who may not even know

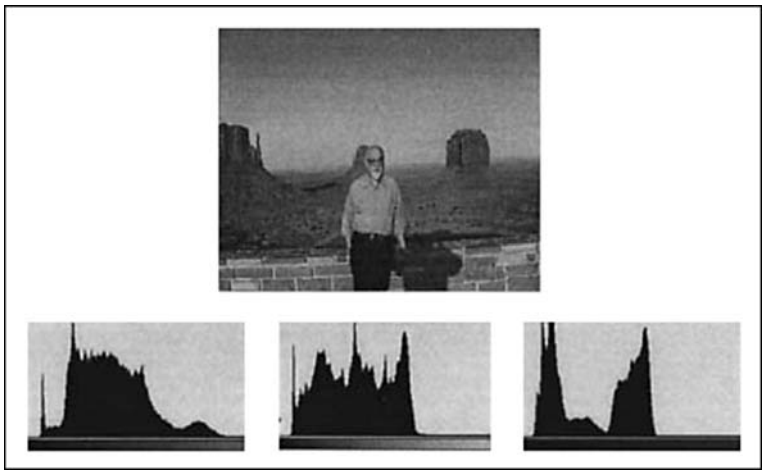


Image Retrieval. **Figure 3.** An image and three of its derived color histograms over the *red*, *green*, and *blue* color channels.

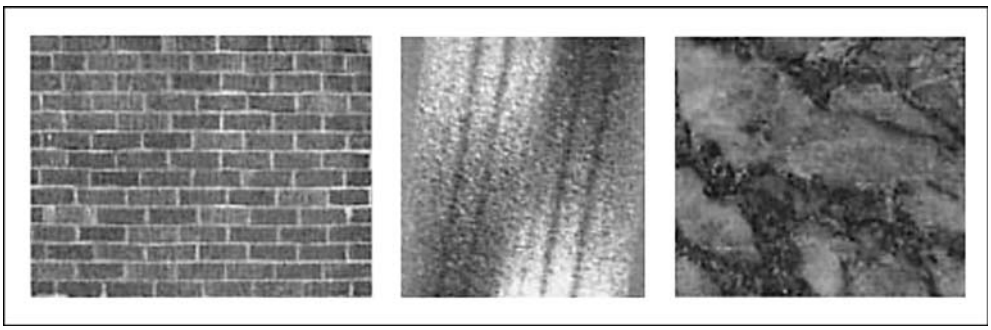


Image Retrieval. **Figure 4.** Examples of textures.

Image Retrieval. Figure 5. The shape of an object. A local shape feature is the set of angles formed by the red portion of the object's boundary, while a global shape feature is the object's center of mass.

what they are looking for but can recognize it when they see it, there has been much work done in trying to discover what is in the mind of the user. A very common technique for this is *relevance feedback*. Originally advanced in the information retrieval community, it has become a standard in most existing image retrieval systems, although some researchers believe that more involved user interactions are necessary to discover user semantics. This technique helps the system refine its search by asking the user to rank the returned results as to relevance. Based on these results, the system learns how to retrieve results more in line with what the user wants. There have been many new approaches developed in recent years, but the classical techniques are *query refinement* or *feature reweighting*. Query refinement transforms the query so that more of the positive and less of the negative examples will be retrieved. Feature reweighting puts more weight on features which help to retrieve positive examples and less weight on features which aid in retrieving negative examples. This process continues for as many rounds as is necessary to produce results acceptable to the user.

Needless to say, human beings are much better than computers at extracting and making use of semantic information from images. Many researchers believe that complete image understanding should start from interpreting image objects and their relationships. The process of grouping low-level image features into meaningful image objects and then automatically attaching correlated semantic descriptions to image objects is still a challenging problem in image retrieval. One of the earliest examples of such an approach is that used in the ImageMiner system. Their method is structural in nature, using graph grammars, and generates scene descriptions with region labels. Current techniques in this area use Baysian approaches which integrate textual annotations and image features.

Content-Based Image Retrieval (CBIR) Systems

There are several excellent surveys of content-based image retrieval systems. We mention here some of the more notable systems. The first, QBIC (Query-by-Image-Content), was one of the first prototype systems. It was developed at the IBM Almaden Research Center and is currently folded into DB2. It allows queries by color, texture, and shape, and introduced a sophisticated similarity function. As this similarity function has a quadratic time-complexity, the notion of dimensional reduction was discussed in order to reduce the computation time. Another notable property of QBIC was its use of multidimensional indexing to speed-up searches. The Chabot system, developed at the University of California at Berkeley, brought text and images together into the search task, allowed the user to define concepts, such as that of a *sunset*, in terms of various feature values, and used the post-relational database management system Postgres. Finally, the MARS system, developed at the University of Illinois at Urbana-Champaign, allowed for sophisticated relevance feedback from the user.

Cross-References

▶ Emergent Semantics

References

1. L.D.F. Costa and R.M. Cesar Jr., "Shape Analysis and Classification: Theory and Practice," CRC, Boca Raton, 2000.
2. M. Flickner, H. Sawhney, W. Niblack et al. "Query by Image and Video Content: The QBIC System," IEEE Computer, Vol. 28, No. 9, September 1995, pp. 23–32.
3. W.I. Grosky, "Multimedia Information Systems," IEEE Multimedia, Vol. 1, No. 1, Spring 1994, pp. 12–24.
4. M.L. Kherfi and D. Ziou, "Image Retrieval from the World Wide Web: Issues, Techniques, and Systems," ACM Computing Surveys, Vol. 36, No. 1, March 2004, pp. 35–67.
5. O. Marques and B. Furht, "Content-Based Image and Video Retrieval," Springer, Berlin, 2002.
6. V. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images," IEEE Computer, Vol. 28, No. 9, September 1995, pp. 40–48.
7. Y. Rui, R.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 5, September 1998, pp. 644–655.
8. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early

Years," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, December 2000, pp. 1349–1380.
9. R.C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey," http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/index.html.
10. C. Wang and X.S. Wang, "Indexing Very High-Dimensional Sparse and Quasi-Sparse Vectors for Similarity Searches," The VLDB Journal, Vol. 9, No. 4, April 2001, pp. 344–361.
11. I.H. Witten, A. Moffat, and T.C. Bell, "Managing Gigabytes: Compressing and Indexing Documents and Images," 2nd edn, Morgan Kaufmann, Los Altos, CA, 1999.

Image Search Engine

Mingjing Li, Wei-Ying Ma
Microsoft Research, Beijing, China

Synonyms

► Web Image search engine

Definition

Image search engines are Web-based services that collect and index images available on the Internet.

Abstract

Some commercial image search engines have indexed over one billion images so far. Like general web search, image searching is mostly based on text information associated with images, which can be automatically extracted from containing web pages. In addition to text-based image search engines, there are also some content-based image retrieval systems that index images using their visual characteristics. Those systems are mainly developed for research purpose and usually limited to small image collections.

Introduction

Due to improved digital imaging technologies and convenient accessibility facilitated by the Internet, the popularity of digital images is rapidly increasing. As most of those images are not annotated with semantic descriptors, it might be a challenge for general users to find specific images from the Internet.

Image search engines are such systems that are specially designed to help users find their intended images. In general, image search engines may adopt two approaches to achieve this goal. One is text-based; the other is content-based.

Text-based image search engines index images using the words associated with the images. Depending on whether the indexing is done automatically or manually, image search engines adopting this approach may be further classified into two categories: Web image search engine or collection-based search engine. Web image search engines collect images embedded in Web pages from other sites on the Internet, and index them using the text automatically derived from containing Web pages. Most commercial image search engines fall into this category. On the contrary, collection-based search engines index image collections using the keywords annotated by human indexers. Digital libraries and commercial stock photo collection providers are good examples of this kind of search engines.

Content-based image retrieval (CBIR) has been an active research topic since 1990s. Such systems index images using their visual characteristics, such as color, texture and shape, which can be extracted from image itself automatically. They can accept an image example as query, and return a list of images that are similar to the query example in appearance. CBIR systems are mainly experimental and often limited to small image collections.

In the following, we briefly describe how those image search engines work.

Web Image Search Engine

The Word Wide Web (WWW) may be the largest repository of digital images in the world. The number of images available on the Internet is increasing rapidly and will continue to grow in the future. Image search engine is a kind of Web-based services devoted to collect and index those Web images.

There are a number of image search engines commercially available, such as AltaVista *Image Search* (http://www.altavista.com/image), Google Image Search (http://images.google.com/) and Yahoo! Image Search (http://search.yahoo.com/images). AltaVista is the first search engine in the world that launches image search functionalities. It also supports video and music search as well. Yahoo! claims to have indexed over 1.6 billion images in August 2005, while Google claims over 1 billion. Those engines are based on existing search engine technology in the sense that they index images using the text information associated with images.

Such search engines take the text in hosting Web pages as approximate annotation of Web images, assuming that images are embedded into Web pages to complement the text information. Some sources of information might be relevant to the content of embedded images. These include, in the decreasing order of usefulness, image file names, image captions,

alternate text, which is an HTML tag used to replace the image when it cannot be displayed, surrounding text, the page tile and others [1,2]. Surrounding text refers to the words or phrase that are close to the image, such as those in the above, below, left or right areas. However, it is difficult to determine which area is more relevant and how much should be considered. Thus the extraction of surrounding text is somewhat heuristic and subjective. As such information can be extracted automatically from Web pages, the indexing process is automated.

To build a commercial image search engine, a lot of functionalities should be implemented. Among those, at least the following four should be provided.

Image crawler is used to collect Web images, and usually implemented as software robots that run on many machines. Those robots scan the entire Web to identify images and then download images and hosting Web pages. As Web images are usually protected by copyrights, image search engines only keep a thumbnail for each Web image. Original images can be accessed via links in the search result.

Page parser is used to analyze Web pages so as to find informative images and extract associated text for indexing. Not all Web images are useful for search. Some are too small or used for decoration or function only, such as background images, banners and buttons. Some are advertisements that are not relevant to hosting Web pages at all. Those images should be excluded from the indexing. Page parser also tries to determine which parts of the hosting Web page are likely relevant to the contained images and extract corresponding text as for indexing.

Index builder is used to build indexing structure for efficient search of images. The methods adopted are quite similar to general web search, except that each image is treated as a text document rather than a Web page.

Image searching is actually processed in the server side. It accepts users' queries and compares with indexed images. When generating the final search result, it considers a number of factors, e.g. the similarity between the query and an indexed image, the image quality, etc. Google claims to present high-quality images first so as to improve the perceived accuracy.

The search result is organized as a list of thumbnails, typically 20 in one page, along with additional information about the retrieved images, such as the file name, the image resolution, the URL of the host webpage, etc. Some search engines provide advanced search options to limit the search result by size, file type, color or domain, and to exclude adult content.

Image searching service is usually provided via a Web-based user interface. Users may access image search engine in a Web browser, such as Microsoft Internet Explorer.

Collection-Based Search Engine

Unlike Web image search engine, collection-based search engines index image collections using manually annotated keywords. Images in such collections are usually of high-quality, and the indexing is more accurate. Consequently, the search results from collection-based engines are more relevant and much better than those from Web search engines. Large digital libraries and commercial stock photo or clip art providers offer image searching facilities in this way.

Those image collections are often held in databases, and cannot be easily accessed by Web crawlers. Therefore, they are usually not covered by general search engines.

Among those engines, Corbis (http://www.corbis.com/) and Getty (http://creative.gettyimages.com/) are probably the two largest ones that specialize in providing photography and fine art images to consumers. Corbis collection currently contains 25 million images with more than 2.1 million available online. Its search result can be limited by categories and collections, or even by date photographed or created, by number of people in the image. Getty Images offer localized image data and contextual search capabilities in six local languages.

Microsoft Office Online (http://office.microsoft.com/clipart/) also provides a large collection of clip art and multimedia data with well annotated keywords in multiple languages. The images in this collection can be used in creating documents. Figure 1 shows an example image from this site with its keyword annotations.

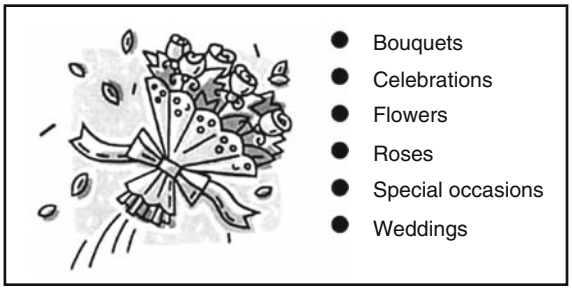


Image Search Engine. Figure 1. A clip art image from Microsoft Office Online and its annotations.

Image Search Engine. Figure 2. The mismatch between low-level features and high-level concepts.

In fact, there are many stock photo or clip art collections available online. A list is provided in TASI's image search engine review [3].

Content-Based Image Retrieval

Content-based image retrieval was initially proposed to overcome the difficulties encountered in keyword-based image search in 1990s. Since then, it has been an active research topic, and a lot of algorithms have been published in the literature. In keyword-based image search, images have to be manually annotated with keywords. As keyword annotation is a tedious process, it is impractical to annotate so many images on the Internet. Furthermore, annotation may be inconsistent. Due to the multiple contents in a single image and the subjectivity of human perception, it is also difficult to make exactly the same annotations by different indexers. In contrast, CBIR systems extract visual features from images and use them to index images, such as color, texture or shape. Color histogram is one of the most widely used features. It is essentially the statistics of the color of pixels in an image. As long as the content of an image does not change, the extracted features are always consistent. Moreover, the feature extraction can be performed automatically. Thus, the human labeling process can be avoided.

In a CBIR system, each image is represented as a vector, which is the feature automatically extracted from the image itself. During the retrieval process, the user may submit an image example as query to the system. After that, the system calculates the similarity between the feature vector of the query and that of each database image, rank images in the descending order of their similarities, and returns images with the highest similarities as the search result.

However, those features often do not match human perception very well. Images with similar concepts may have totally different appearance, while images having similar features may be irrelevant to each other at all. This is the so-called semantic gap, which limits the applicability of CBIR techniques. Figure 2 shows an example. Images A and B should be more semantically similar to each other since both are the image of a butterfly. However, images A and C are closer in the feature space because they contain more similar colors. If A is used as a query, it is more likely to retrieve C as the search result.

Because the features used in CBIR are usually of high dimensionality and there is no efficient indexing method, current CBIR systems only index small image collections. So far, the largest CBIR system reported in the literature is Cortina [4], which indexes over 3 million images. The overall performance of CBIR systems is not satisfactory.

Conclusion

There are so many images available on the Internet that users do need efficient tools to browse and search for those images. The current image search engines can partially fulfill this need. In the future, a proper combination of textual and visual features may produce better image searching experience.

References

1. C. Frankel, M.J. Swain, and V. Athitsos, "WebSeer: An Image Search Engine for the World Wide Web," Technical Report, University of Chicago, July 1966.
2. Z. Chen, W. Liu, F. Zhang, and M. Li, "Web Mining for Web Image Retrieval," Journal of the American Society for Information Science and Technology, Vol. 52, No. 10, August 2001, pp. 831–839.
3. "A Review of Image Search Engine," Available online at: http://www.tasi.ac.uk/resources/searchengines.html (accessed on October 2004).
4. T. Quack, U. Monich, L. Thiele, and B.S. Manjunath, "Cortina: A System for Large-scale, Content-based Web," Proceedings of the 12th Annual ACM International Conference on Multimedia, 2004, pp. 508–511.

Image Secret Sharing

Rastislav Lukac[1], Konstantinos N. Plataniotis[1], Ching-Nung Yang[2]
[1]University of Toronto, Toronto, ON, Canada
[2]National Dong Hwa University, Shou-Feng, Taiwan

Synonyms

► Visual cryptography; ► Secret image

Definition

Among numerous cryptographic solutions proposed in the past few years, secret sharing schemes have been found sufficiently secure to facilitate distributed trust and shared control in various communication applications.

Introduction

Digital rights management systems are used to protect intellectual property rights of Digital media itself or secure its transmission over untrusted communication channels [1–3]. The required protection is achieved by employing either cryptography or Steganography. Steganography hides the secret message inside a cover signal producing its stego-version with an imperceivable difference from the original cover, whereas Cryptography alters the meaningless of the secret message through its encryption Necessitating the use of a decryption key to recover the original content.

Among numerous cryptographic solutions proposed in the past few years, secret sharing schemes have been found sufficiently secure to facilitate distributed trust and shared control in various communication applications, such as key management, conditional access, message authentication, and content encryption [4–6]. Due to the proliferation of imaging-enabled consumer electronic devices and the extensive use of digital imaging in networked solutions and services, secret sharing concepts have been used to secure transmission and distribution of personal digital photographs and digital document images over public networks [2,7–10].

A (k,n)-Threshold Scheme

Most of the existing secret sharing schemes are generalized within the so-called (k,n)-threshold framework where $k \leq n$. The framework confidentially divides the content of a secret message into n shared in the way which requires the presence of at least k shares for the secret message reconstruction [11,12]. If $k = n$, then all the shares are required in the (n,n)-threshold scheme to recover the secret. However, the lost of any of the shares produced using the (n,n)-threshold scheme results in inaccessible secret message.

Therefore, apart from the simplest (2,2)-schemes commonly used as a private key cryptosystem solution [13,14], general (k,n)-threshold schemes with $k < n$ are often the object of interest due to offer their ability to recover the secret message even if several shares are lost. In this case, any of $[n!/(k!(n-k)!)]$ possible combinations of k shares can be used to recover the secret message. Since protection against cryptoanalytic attacks, including brute force enumeration, should remain unchanged regardless of how many shares are available until the threshold k is reached, the use of $k-1$ shares should not reveal additional information compared to that obtained by a single share.

Basis Matrices in Visual Cryptography

Probably the most well-known (k,n)-secret sharing schemes for encryption of visual data are those based on visual cryptography (VC) [11–13,15–19]. Among

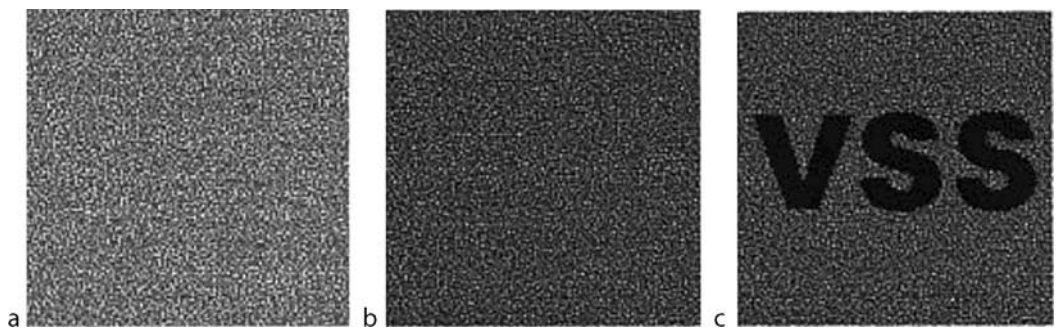


Image Secret Sharing. Figure 1. Demonstration of (3,3)-VC: (a) one binary share, (b) two binary shares stacked together, (c) three binary shares stacked together.

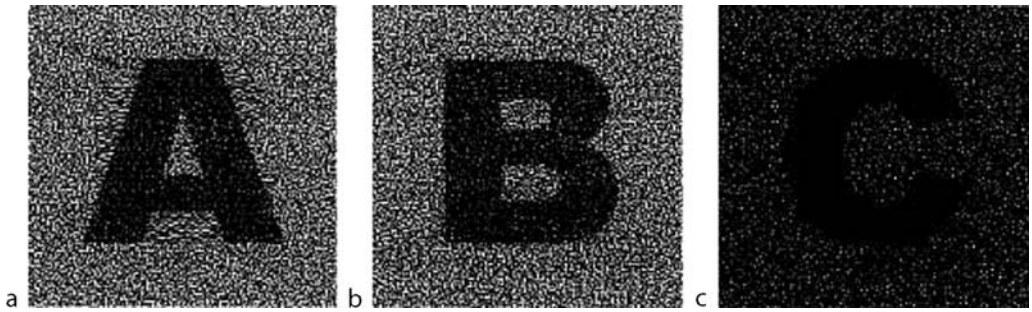


Image Secret Sharing. Figure 2. Demonstration of (2,2)-extended VC: (a,b) meaningful binary shares, (b) image produced by stacking of the shares shown in (a,b).

the numerous VC-based schemes (Figs. 1 and 2), such as the (k,n), (k,n,c), and extended VC schemes, the so-called (k,n)-VC framework is the most widely used. The traditional VC schemes (Fig. 1) produce meaningless, noise-like, shares as opposed to the meaningful shares obtained using complex, extended VC schemes (Fig. 2). Furthermore, compared to (k,n,c)-VC schemes constructed using colors, the (k,n)-VC framework uses frosted/transparent (or binary) representation of the shares keeping the simplicity of the approach at the level suitable for practical implementation. It should be noted that within the (k,n)-VC framework, the special cases of (2,2), (2,n), and (n,n)-VC schemes can be obtained.

Operating on a $K_1 \times K_2$ binary or binarized secret image, a (k,n)-VC scheme encrypts, via an encryption function, each binary pixel into a $m_1 \times m_2$ block of binary pixels in each of the n binary shares. The spatial arrangement of bits in the produced blocks varies depending on the value of secret pixel to be encrypted and the (random) choice of the matrix from the so-called matrices' generation set C_0 or C_1. The sets C_0 and C_1 include all matrices obtained by permuting the columns of the $n \times m_1\ m_2$ basis binary matrices A_0 and A_1 respectively [2]. Examples of the basis matrices are listed here for the most widely used (k,n)-VC schemes:

The value $m_1\ m_2$ is the so-called expansion factor and therefore, the basis matrices are constructed in the way to minimize the expansion factor as much as possible [11,15]. By repeating the encryption operations at each spatial location of the secret image, a $K_1 \times K_2$ secret image is encrypted into n binary shares with dimensions of $m_1\ K_1 \times m_2\ K_2$ pixels.

VC-based decryption, which has to be performed over the set of $\zeta \leq n$ shares, can be modeled through a decryption function similar to the one proposed in [2]. Following the formulation of a $\{k,n\}$-threshold schemes, the secret image is revealed only if $\zeta \geq k$. Due to the utilization of the transparent/frosted concept: (1) the VC decryption process recovers the decrypted pixel as black if any of the share pixels at the same spatial location of the shares stacked for decryption is black, or recover it as white if all the pixels at the same spatial location in the stacked shares are transparent, and (2) do not recover the original secret image.

ISS with Perfect Reconstruction of the Secret Image

To recover the original secret image and prevent the introduction of visual impairments, a different decryption strategy should be used. Following the approach

$$(2,2)\ A_0 = \begin{bmatrix} 0,1,0,1 \\ 1,0,1,0 \end{bmatrix}\ A_1 = \begin{bmatrix} 0,1,0,1 \\ 0,1,0,1 \end{bmatrix}$$

$$(2,3)\ A_0 = \begin{bmatrix} 1,0,0,0 \\ 0,1,0,0 \\ 0,0,1,0 \end{bmatrix}\ A_1 = \begin{bmatrix} 1,0,0,0 \\ 1,0,0,0 \\ 1,0,0,0 \end{bmatrix}$$

$$(3,4)\ A_0 = \begin{bmatrix} 0,1,1,1,1,1,1,0,0 \\ 0,1,1,1,1,1,0,0,1,1 \\ 0,1,1,0,0,1,1,1,1,1 \\ 0,0,0,1,1,1,1,1,1,1 \end{bmatrix}\ A_1 = \begin{bmatrix} 1,1,1,1,1,1,1,0,0,0 \\ 1,1,1,1,0,0,1,1,1,0 \\ 1,1,1,0,1,0,1,0,0,1 \\ 1,1,1,0,0,1,0,1,1,1 \end{bmatrix}$$

$$(4,4)\ A_0 = \begin{bmatrix} 1,0,0,1,0,0,1,0,1 \\ 1,0,1,0,0,0,1,1,0 \\ 1,0,1,0,0,1,0,0,1 \\ 0,1,1,0,0,0,1,0,1 \end{bmatrix}\ A_1 = \begin{bmatrix} 1,0,0,0,0,0,1,1,1 \\ 1,0,1,0,0,1,1,0,0 \\ 1,1,0,0,0,1,0,1,0 \\ 1,1,1,0,0,0,0,0,1 \end{bmatrix}$$

$$(2,6)\ A_0 = \begin{bmatrix} 0,1,0,1 \\ 1,0,1,0 \\ 1,1,0,0 \\ 0,0,1,1 \\ 1,0,0,1 \\ 0,1,1,0 \end{bmatrix}\ A_1 = \begin{bmatrix} 0,1,0,1 \\ 0,1,0,1 \\ 0,1,0,1 \\ 0,1,0,1 \\ 0,1,0,1 \\ 0,1,0,1 \end{bmatrix}$$

introduced in [2,7], the decryption function should observe the contrast properties of the share blocks when stacked together. Similarly to VC decryption, the difference between the encrypted "black" and "white" binary values reveals only if $\zeta \geq k$. In this case, the decryption process recovers the corresponding original binary pixel. By decrypting the binary share blocks over the image domain the procedure recovers the original binary secret image, as it is shown in Fig. 3. This suggests that such an approach can be used to construct an image encryption scheme which satisfies the essential perfect reconstruction property [2, 7, 8].

Built on the framework presented in [2,20] a number of solutions with different design characteristics and performance can be obtained. For example, instead of the utilization of a halftoning module in processing a $K_1 \times K_2$ continuous-tone image as it is suggested by traditional VC, the encryption operations can be performed (Fig. 4a) at the decomposed bit-levels of the secret image with B-bits per pixel representation. Each of B bit-levels represents a $K_1 \times K_2$ binary image which is commonly required in the input of the VC encryption procedure. By repeating the VC-based encryption operation at each bit-level, the generated share bits are used to obtain the B-bit share pixel and thus, the bit-level processing based encryption process splits the B-bit secret image into n, seemingly random, m_1 $K_1 \times m_2$ K_2 shares. Each of the constructed shares has B-bit representation identical to the bit-level representation of the secret image (Fig. 5).

The decryption process (Fig. 4b) decomposes the bit levels of all B-bit shares which are available for decryption. If $\zeta \geq k$, then the procedure recovers the individual bit levels which are then used by to reconstruct the original, continuous tone secret image.

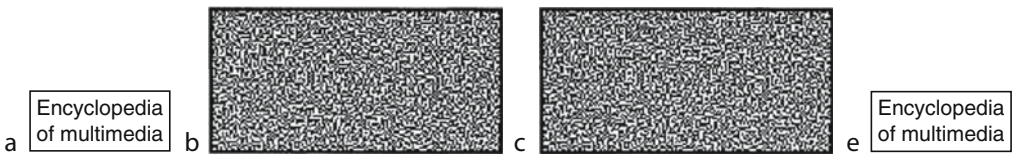


Image Secret Sharing. **Figure 3.** A ISS scheme constructed using 2 × 4 basis matrices with $m_1 = 2$ and $m_2 = 2$: (a) secret binary image, (b,c) binary shares, (d) output image decrypted using the share inputs shown in (b,c).

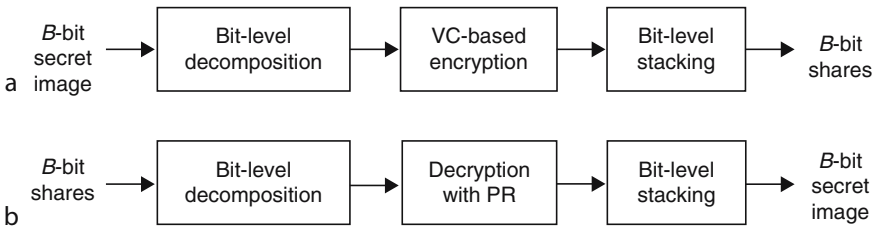


Image Secret Sharing. **Figure 4.** An ISS concept: (a) encryption procedure, (b) decryption procedure satisfying the perfect reconstruction (PR) property.

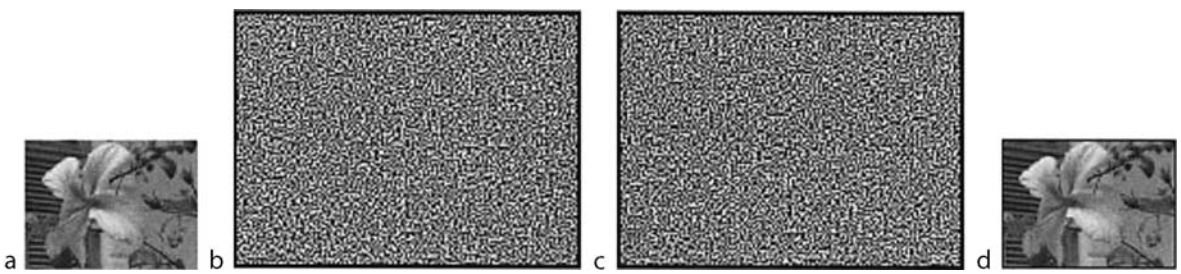


Image Secret Sharing. **Figure 5.** An (2,2) ISS scheme constructed using 2 × 4 basis matrices with $m_1 = 2$ and $m_2 = 2$: (a) secret grayscale image, (b,c) gray-scale shares, (d) decrypted grayscale image.

Thus, the decryption process recovers the original secret image (Fig. 5) in a digital form making the framework ideal for modern, digital, multimedia systems.

The ISS framework allows for the design of both input-agnostic and input-specific solutions [20]. Such solutions: (1) differ in their design characteristics and complexity, (2) may introduce different protection levels during the encryption process, and (3) can be used to process the secret image using an arbitrary $\{k,n\}$ configuration and expansion factor m_1 m_2.

Input-Agnostic ISS Solutions

The input-agnostic (IA) ISS solution follows the representation of the input and process sequentially all bit-levels decomposed from the secret image [2,20]. As it is shown in Fig. 6, the IA-ISS solution encrypts: (1) the binary secret image into the binary shares, (2) the gray-scale secret image into the gray-scale shares, or (3) the color secret image into the color shares. As the result, the IA-ISS solution always produces shares with the image (bit) representation identical to that of the secret image, and decrypts the original secret image with perfect reconstruction.

Input-Specific ISS Solutions

The input specific (IS)-ISS solution encrypts the secret image arranged in the specific image format [7,8,20]. For example, IS-ISS solution can require to convert: (1) the binary or grayscale input image into the color image when the solution is color image specific to produce color shares, (2) the binary or color input image into the gray-scale image when the solution is gray-scale image specific to produce gray-scale shares, and (3) the color or gray-scale input image into the binary image when the solution is binary image specific to produce binary shares. Thus, any IS-ISS solution can be used to process the image with less rich or richer bit-representation than that of the required in the IS input. The approach requires: (1) format conversion such as the replication of the input (Fig. 7(a)) or reduction of image representation (Fig. 7(b)) in order to meet the requirements for the input, (2) the procedure requires to transmit Q times more (Fig. 7(a)) or less (Fig. 7(b)) share information compared to the share information produced by the IA-ISS solution, and (3) inverse format conversion is necessary to recover the original (Fig. 7(a)) or due to the loss in input format conversion approximated (Fig. 7(b)) secret image.

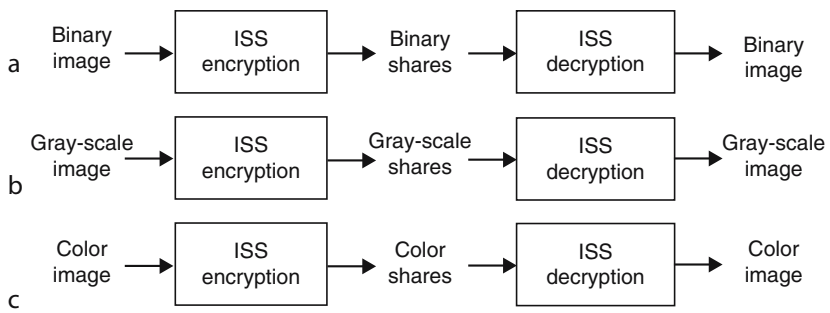


Image Secret Sharing. Figure 6. Input-agnostic ISS solution applied to encrypt: (a) binary secret image, (b) gray-scale secret image, (c) color secret image.

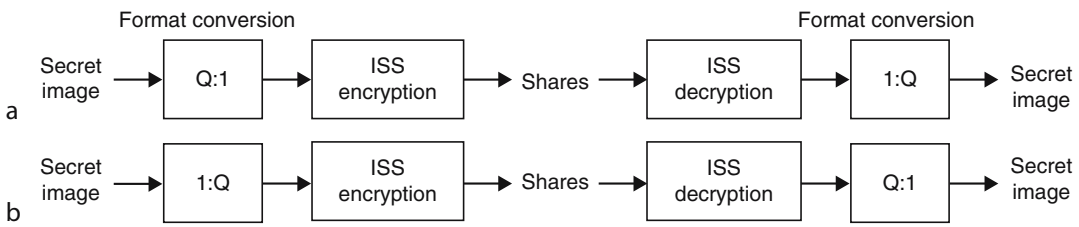


Image Secret Sharing. Figure 7. Input-specific ISS solution applied to encrypt: (a) secret image with less rich representation, or (b) richer representation than that of required in the input to the ISS encryption/decryption module.

Image Secret Sharing. Figure 8. Examples of the share formats: (a) binary, (b) gray-scale, (c) full-color share.

Security Characteristics of the ISS Framework

The randomness of the encryption processes in the ISS framework is fortified by the depth of the B-bit representation of the secret image makes the original pixel and the share pixels significantly different [2,20]. The variety in the share pixels, which can be viewed as the degree of protection, increases with the value of B which denotes the number of bits used to represent the image pixel (Fig. 8). Assuming that N denotes the number of unique matrices either in C_0 or C_1, the B-bit pixel is encrypted using one of N^B unique $m_1 \times m_2$ share blocks of B-bit pixels instead of one of only N unique share blocks of binary pixels used in the traditional and halftoning based VC schemes. In this way, the IS-ISS solution in Fig. 7(a) increases the protection by generating "richer" noise compared to the shares obtained using the IA-ISS solution. Similarly, by reducing the image representation, the IS-ISS solution in Fig. 7(b) reduces protection of the secret image compared to the case when the IA-ISS solution is used. Depending on the application, security requirements, available computational resources, and nature of the secret image, the end-user may select the particular cryptographic solution designed within the proposed framework to process the input (secret) image [20].

Cross-References

- ▶ Compression in Image Secret Sharing
- ▶ Halftoning Based VSS
- ▶ Image Watermarking Using Visual Cryptography
- ▶ Private-Key ISS Solution
- ▶ Threshold Schemes With Minimum Pixel Expansion
- ▶ Visual Cryptography

References

1. M. Barni and F. Bartolini, "Data Hiding for Fighting Piracy," IEEE Signal Processing Magazine, Vol. 21, No. 2, March 2004, pp. 28–39.
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. E.-T. Lin, A.M. Eskicioglu, R.-L. Lagendijk, and E.-D. Delp, "Advances in Digital Video Content Protection," Proceedings of the IEEE, Vol. 93, No. 1, January 2005, pp. 171–183.
4. A.-M. Eskicioglu, E.-J. Delp, and M.-R. Eskicioglu, "New Channels for Carrying Copyright and Usage Rights Data in Digital Multimedia Distribution," Proceedings of the International Conference on Information Technology: Research and Education (ITRE'03), August 2003, pp. 94–98.
5. E.-D. Karnin, J.-W. Greene, and M.-E. Hellman, "On Secret Sharing Systems," IEEE Transactions on Information Theory, Vol. 29, No. 1, January 1983, pp. 35–41.
6. A. Biemel and B. Chor, "Secret Sharing with Public Reconstruction," IEEE Transactions on Information Theory, Vol. 44, No. 5, September 1998, pp. 1887–1896.
7. R. Lukac and K.-N. Plataniotis, "Colour Image Secret Sharing," IEE Electronics Letters, Vol. 40, No. 9, April 2004, pp. 529–530.
8. R. Lukac and K.-N. Plataniotis, "A New Encryption Scheme for Color Images," Computing and Informatics, submitted.
9. C.-S. Tsai, C.-C. Chang, and T.-S. Chen, "Sharing Multiple Secrets in Digital Images," Journal of Systems and Software, Vol.64, No.2, 2002, pp. 163–170.
10. C.-C. Chang and J.-C. Chuang, "An Image Intellectual Property Protection Scheme for Gray-Level Images Using Visual Secret Sharing Strategy," Pattern Recognition Letters, Vol. 23, No. 8, June 2002, pp. 931–941.
11. P.-A. Eisen and D.-R. Stinson, "Threshold Visual Cryptography Schemes with Specified Levels of Reconstructed Pixels," Design, Codes and Cryptography, Vol. 25, No.1, January 2002, pp. 15–61.
12. E.-R. Verheul and H.-C.-A. Van Tilborg, "Constructions and Properties of k out of n Visual Secret Sharing Schemes," Designs, Codes and Cryptography, Vol. 11, No. 2, May 1997, pp. 179–196.
13. 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.
14. 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.
15. M. Naor and A. Shamir, "Visual Cryptography," Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol. 950, 1994, pp. 1–12.
16. C.-N. Yang, "New Visual Secret Sharing Schemes Using Probabilistic Method," Pattern Recognition Letters, Vol. 25, No. 4, March 2004, pp. 481–494.

17. C.-N. Yang and T.-S. Chen, "Aspect Ratio Invariant Visual Secret Sharing Schemes with Minimum Pixel Expansion," Pattern Recognition Letters, Vol. 26, No. 2, January 2005, pp.193–206.
18. C.-N. Yang and T.-S. Chen, "Size-Adjustable Visual Secret Sharing Schemes," IEICE Transactions on Fundamentals, Vol. E88-A, No. 9, September 2005, pp. 2471–2474.
19. H. Yamamoto, Y. Hayasaki, and N. Nishida, "Securing Information Display by Use of Visual Cryptography," Optics Letters, Vol. 28, No. 17, September 2003, pp. 1564–1566.
20. R. Lukac and K.-N. Plataniotis, "Image Representation Based Secret Sharing," Communications of the CCISA (Chinese Cryptology Information Security Association), Special Issue on Visual Secret Sharing, Vol. 6, April 2005.

Image Watermarking

Definition

Image watermarking deals with creating a metadata (a watermark) about the image content and hiding it within the image.

Challenges and Benchmarking

Digital images are often printed and scanned again, for example when they are used in magazines and readers want to store digitally. When used as web site illustrations, they are often compressed by lossy compression algorithms like JPEG. Common processing operations include softening, sharpening, denoising, scaling, cropping and color corrections. A number of benchmarking suits address image watermarking robustness evaluation, Stirmark [1], Checkmark and Certimark are well known examples.

The most challenging attacks on the robustness of image watermarks today are nonlinear transformations, rendering a watermark in an image undetectable for the watermarking algorithm. But even common image processing operations like scaling and rotation can be a serious challenge for the re-synchronization of an image watermark.

Typical challenges in practical solutions are printing and scanning with low quality devices like customer ink jet printers leading to the additions of noise, color changes, small rotations and changes in image resolution.

Another important aspect of image watermarking is the broad variety of image types. There are photos, also called natural images in the literature, figures based on line drawings, rendered synthetic images, bitmap representations of textual information, just to give the most common examples. Challenges with respect to transparency and robustness often depend on the characteristics of these images. Many watermarking algorithms address only one image type, like e.g., photographs. Often the dependency can be derived from the embedding principle: An algorithm which needs textures to embed information will not be able to deal with a black and white drawing but performs perfectly with most photographs.

Advanced Image Watermarking

While all watermarking methods described in this article are known for image watermarking, there are also a number of innovative approaches which are especially suited for image watermarking. They usually could be transferred to video watermarking, but the long computation time caused by high complexity and the stronger compression in digital video are a serious hindrance here. As examples for advanced watermarking methods we describe two innovative algorithms combining existing approaches with new techniques.

Region of Interest Watermarking

A good example of advanced fragile watermarking for integrity protection is region of interest (ROI) watermarking. The basic idea is distinguish between semantically important and unimportant regions of an image. It is often sufficient to protect the important regions as manipulations changing the meaning of the image will only occur in these. Modifications taking place not in the RIO may be of cosmetic nature or could be simple name tag annotations. An example of ROI watermarking is the face detection approach discussed in [2] protecting only automatically detected faces and the relative positions in an image.

Self-Correcting Images

While many approaches to identify changes in images are known, some algorithms even allow re-creating the original from the watermark to a certain degree. This enables a comparison of the content of an image before and after an attack. One known approach is based on an algorithm described in [3], using a block-based pseudo-random distribution of pixel information in over the image thereby shuffling and spreading a kind of parity checksum over the whole image. The idea here is to use the spread information about a block of pixels to identify small local changes by identifying parity errors after attacks.

References

1. F. Petitcolas and R. Anderson, "Evaluation of copyright marking systems," Proceedings of IEEE Multimedia Systems, Multimedia Computing and Systems, Florence, Italy, Vol. 1, June 1999, pp. 574–579.
2. H. Liu, H. Sahbi, L. Croce Ferri, and M. Steinebach, "Image Authentication using automatic detected ROIs," WIAMIS 2004, Fifth International Workshop on Image Analysis for Multimedia Interactive Services, Instituto Superior Técnico, Lisboa, Portugal, April 2004.
3. M. Wu and B. Liu, "Digital watermarking using shuffling," ICIP 99, Proceedings of International Conference on Image Processing, Vol. 1, 1999, pp. 291–295.

Image Watermarking using Visual Cryptography

Definition

A new class of digital watermarking techniques is based on visual cryptography.

Recent works have introduced a new class of digital watermarking schemes which employ visual cryptography (VC) concepts to secure watermarked content [1–3]. In addition, VC-based watermarking may be used to robustify recognition of an extracted watermark from images which have been subjected to attacks [2,3]. For example, in the approach shown in Fig. 1, instead of embedding a binary logo directly to the host image, a VC-based watermarking scheme first encrypts the binary logo into two noise-like binary shares. One of the two generated shares can be viewed as a private watermark share and is kept by the owner. The other share represents a public watermark share and is being embedded to the host image using a conventional watermarking technique which operates either in the spatial or frequency domain of the host image [2,3].

As it is shown in Fig. 1, the resulting watermarked image is to be stored, transmitted via public channel, and thus is vulnerable to various signal processing and cryptoanalysis attacks. After extracting the public watermark share from the attacked watermarked image, a private watermark share is used as a private key and stacked together with a public watermark share to visually reveal a binary logo.

Cross-References

- ▶ Image Secret Sharing
- ▶ Private-Key Cryptosystem.
- ▶ Visual Cryptography

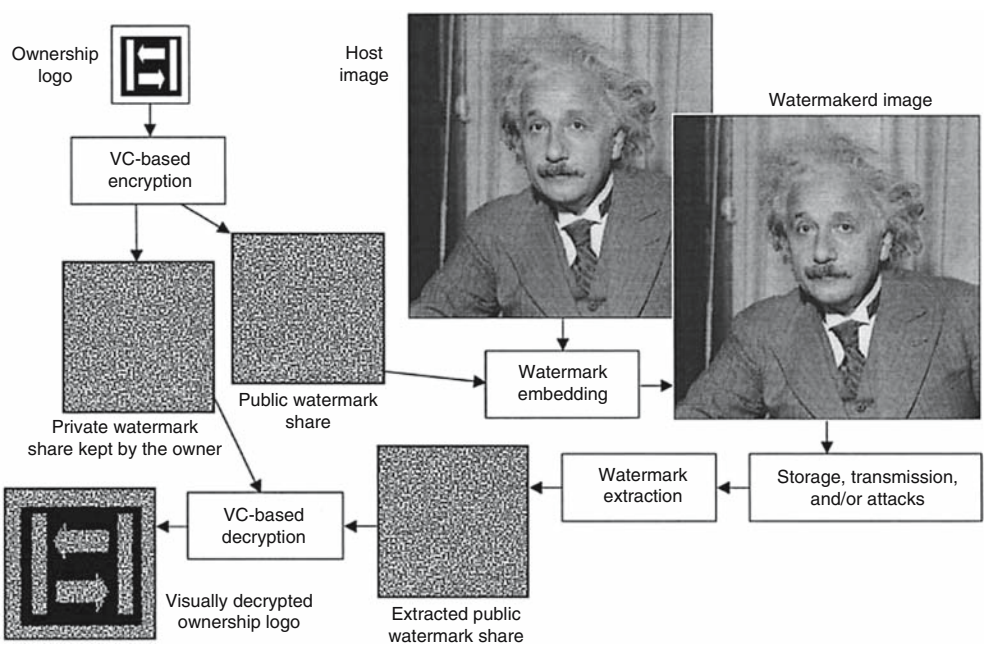


Image Watermarking using Visual Cryptography. **Figure 1.** Visual cryptography based digital watermarking.

References

1. R. Lukac and K.-N. Plataniotis, "Digital Image Indexing Using Secret Sharing Schemes: A Unified Framework for Single-Sensor Consumer Electronics," IEEE Transactions on Consumer Electronics, submitted.
2. G.-C. Tai and L.-W. Chang, "Visual Cryptography for Digital Watermarking in Still Images," Lecture Notes in Computer Science, Vol. 3332, 2004, pp. 50–57.
3. C.-S. Tsai and C.-C. Chang, "A New Repeating Color Watermarking Scheme Based on Human Visual Model," Eurasip Journal of Applied Signal Processing, Vol. 2004, No. 13, October 2004, pp. 1965–1972.

Immersive Telepresence

► Networked Collaboration Environments

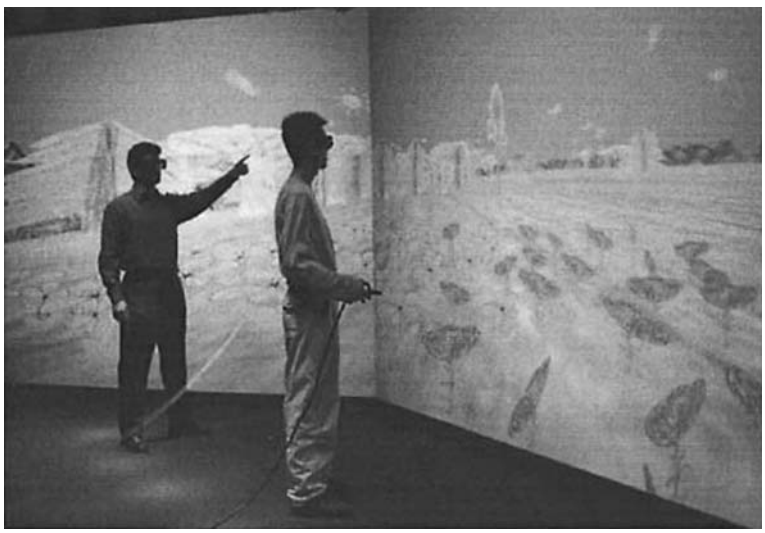
Immersive Virtual Reality

Definition

The goal of Immersive Virtual Reality is to completely immerse the user inside the computer generated world, giving the impression to the user that he/she has "stepped inside" the synthetic world.

Virtual Reality (VR) is the technology that provides almost real and/or believable experience in a synthetic or virtual way. The goal of *Immersive VR* is to completely immerse the user inside the computer generated world, giving the impression to the user that he/she has "stepped inside" the synthetic world. This can be achieved by using either the technologies of *Head-Mounted Display* (HMD) or multiple projections. Immersive VR with HMD uses HMD to project VR just in front of the eyes and allows users to focus on display without distraction. A magnetic sensor inside the HMD detects the users' head motion and feeds that information to the attached processor. Consequently, the user turns his or her head; the displayed graphics can reflect the changing viewpoint. The virtual world appears to respond to head movement in a familiar way.

Immersive VR with multiple projections uses multiple projectors to create VR on a huge screen, which might be a hemispherical surface, in a room where users might ware polarized glasses to maximize the feeling of being present at the scene in standstill. The form of this immersive graphical display is known as the CAVE (stands for Computer-Aided Virtual Environment), where the immersion occurs by surrounding the body on all sides by images, rather than just the eyes. Early versions of these technologies were demonstrated at SIGGRAPH' 92 in Chicago by Sun Microsystems and University of Illinois. The CAVE is essentially a five sided cube. The participant stands in the middle of cube, and images are projected onto the walls in front, above, below and on either side of the participant, utilizing full 270° peripheral vision. As the user travels through the virtual environment, updated images are projected onto the



Immersive Virtual Reality. Figure 1. Immersive virtual reality in a CAVE.

CAVE's walls to give the sensation of smooth motion. Figure 1 shows a CAVE at the University of Ottawa's DISCOVER Lab [1].

Cross-References

► Virtual and Augmented Reality

References

1. DISCOVER Lab, http://www.discover.uottawa.ca.

In Home, in Car, in Flight Entertainment

Definition

Entertainment has exited from traditional places and is now an anywhere activity for people at homes, cars, and flights.

Home entertainment can be considered as a part of a more complex demotic system. *Domestics* is the combination of technologies for improved living in the areas of safety, comfort and security management. New systems can provide consumers with personalized services tailoring to their well-being, communication and entertainment needs. In an entertainment-equipped home, a high-speed, always-on, Internet access combined with WLAN is quite common. At the same time, home theaters, game consoles and other entertainment equipments are widely adopted. The presence of a broadband network and of different entertainment equipments encourage a drastic revolution in future home entertainment promoting the design of a *magic box* where new exciting multimedia applications will be all available for instant enjoyment. According to [1], today, the magic box does not exist yet. While in the computer industry all majors have understood the need to simplify the life for the consumers, the home theater industry has not become aware of this topic. The devices composing a home theater live as fragmented pieces of the same picture, still having many companies fighting to impose their technical choices as standard. Hence, to obtain a fully inclusive system (video, stereo, decoder, console, Web) it is still necessary to acquire a variety of equipments made by different manufacturers. Therefore, many efforts should be devoted should develop a control protocol that permits to configure each device and to build-up a *plug and play* system for new devices.

From home to car, entertainment becomes, usually, an attempt to define a mobile home theatre [2]. For example, Dolby provides an enhanced audio/video experience for all passengers in a car. Other simple multimedia entertainments are based on the use of GPS systems that show the localization of the car on a map, or in alternative, by the use of digital radios that provide dynamically update information. Sophisticated applications exploiting wireless solutions implement *network of cars* that offer multimedia distribution services.

From car to airplane, technical problem becomes more urgent. Entertaining people during an intercontinental flight is really a hard work [3]. Typically, an on-board broadcasting system offers to travelers music and movies. New forms of entertainment are possible, simply equipping the plane with on-board multimedia services and network connections. For example, 757 Boing jet of the Song Airlines is equipped with an in-flight multimedia system that offers GPS tracking, connecting gate, digital shopping, in-seat Internet connectivity, SMS/email messaging, streamed MP3 play lists, 24 channels of live network TV, and interactive networked games for free.

Cross-References

► Multimedia Entertainment Applications

References

1. D.A. Norman, "Home Theater: Not Ready for Prime Time," IEEE Computer, Vol. 35, No. 6, 2002, pp. 100–102.
2. A. Gilroy, "Car A/V Spotlights iPod Interfaces, MP3 on DVD, Mini TFTs," Special Report CES, Las Vegas, USA,
3. G. Lui-Kwan, "In-Flight Entertainment: the Sky's the Limit," IEEE Computer, Vol. 33, No. 10, October 2000, pp. 98–101.

Indexing Three Dimensional Scenes

Ioan Marius Bilasco, Jérôme Gensel,
Hervé Martin, Marlène Villanova-Oliver
Laboratoire LSR IMAG, Grenoble, France

Synonyms

► Construction of 3D scenes

Definition

Semantic queries on indexed objects allow the reuse of the 3D scenes.

Introduction

Nowadays, the 3D is a highly expanding media. More particularly with the emergence of dedicated standards such as VRML and X3D, 3D animations are widely used on the Web. The continuous evolution of computing capabilities of desktop computers is also a factor that facilitates the large deployment of 3D information contents. At the same time the demand in term of 3D information is becoming more and more sustained in various domains such as spatial planning, risks management, telecommunications, transports, defense, and tourism. 3D information should represent a real world scene as accurately as possible and should exhibit properties (like, topological relations) to allow complex spatial analysis [1].

The construction of a 3D scene is a complex and time consuming task. Thus, being able to reuse the 3D scenes is a very important issue for the multimedia community. In order to meet this goal, the indexing process is essential. Indexing implies the enrichment of the raw information contained in a multimedia document. In general, indexing is achieved by means of signal analysis or manual or semi-automatic annotations. Signal indexing supposes the automatic extraction of implicit features from the document. For instance, if one analyses a 2D image the signal indexing results in the extraction of the dominant color, the color histogram, etc.

Usually, in a 3D scene, one can model only the geometric features of the scene paying very little attention to semantic information that should guide and help the reuse. Identification of interesting/reusable *objects* in the scene is part of the indexing process. The granularity of a reusable *object* can vary from a simple geometric element (e.g. a cube, etc.) to a full scene (e.g. a casual office scene, a building). Since a 3D scene is built up from different geometric elements, identification of objects is performed by localizing its geometric elements. In order to facilitate the reuse of 3D objects, some semantic information should be added. Semantic queries on indexed objects would yield the most appropriate result according to the intent of reuse.

3D Scenes as Pure Geometric Worlds

The 3D community benefits from the support of a highly interactive consortium called the Web3D consortium, involving many companies (NASA, HP, nVIDIA, Sun Microsystem) and academic communities (Communications Research Center of Canada, GIS Research Center at Feng Chia University, and others). The research efforts of the consortium are directed towards the development of a widely adopted standard for deploying 3D information all over the Web.

The Extensible 3D (X3D) [2] standard has emerged in the mid 2002. X3D was proposed as a revised version of the Virtual Reality Modelling Language (VRML97) [3]. In July 2002 the Web3D Consortium made available the final working draft version of X3D. The X3D was accepted by the ISO as an ISO/IEC 19775 standard of communication for 3D real-time scenes in August 2004. The final specifications were produced by the end of October 2004.

This standard defines a runtime environment and a delivery mechanism for 3D content and applications running over a network. It combines geometry descriptions, runtime behavioral descriptions and control features. It proposes numerous types of encodings including an Extensible Modeling Language (XML) [4] encoding.

X3D is extensible as it is built on a set of components organized in profiles. Each profile contains a set of components. A component introduces a specific collection of nodes. The next extensions of the standard will be made by defining new components and organizing them into new profiles.

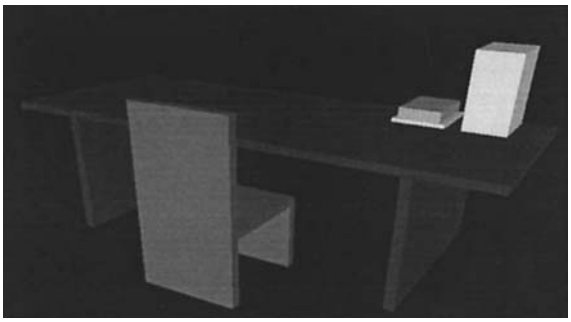
An X3D document represents the scene as an n-ary tree. The tree is composed of nodes supported by the selected profile. Among the most important nodes are found: geometric primitives (*Cube*, *Box*, *IndexLineSet*, etc), geometric transformations (*Transform* assuring *translations* and *rotations*), composite objects (*Group*), alternate content (*Switch*), multi level representation (*LOD*), etc. The tree also contains ambiental elements: *lights*, *viewpoints*, etc, and a meta-data node (*WorldInfo*).

A view of the scene is shown in Fig. 1. The office contains a desk and a chair. On the desk, there are two stacks of books and papers. Hereafter, in Fig. 2, we present the tree corresponding to a 3D scene describing a researcher's office. An excerpt of the X3D code corresponding to the materialization of the chair can be found in Fig. 3.

The desk is built up using three boxes: one for the desk top and two for the desk legs. The chair is composed of three boxes as well: one for the back side and the back-side legs, one for the front legs and one to sit on. The books are modeled by three boxes: two small ones for the books on the left and a bigger one for the stack of papers.

The resulting model of a 3D scene using the X3D treats exhaustively the geometry, the environmental aspects (lights, etc.), and describes levels of user's interactions. However, the scene is not self-descriptive. The scene does not contain any information on the real-world objects included into the scene. The main scope of an X3D scene model is to ensure the delivery of the scene to the user's rendering device. The semantics aspects are scarcely included.

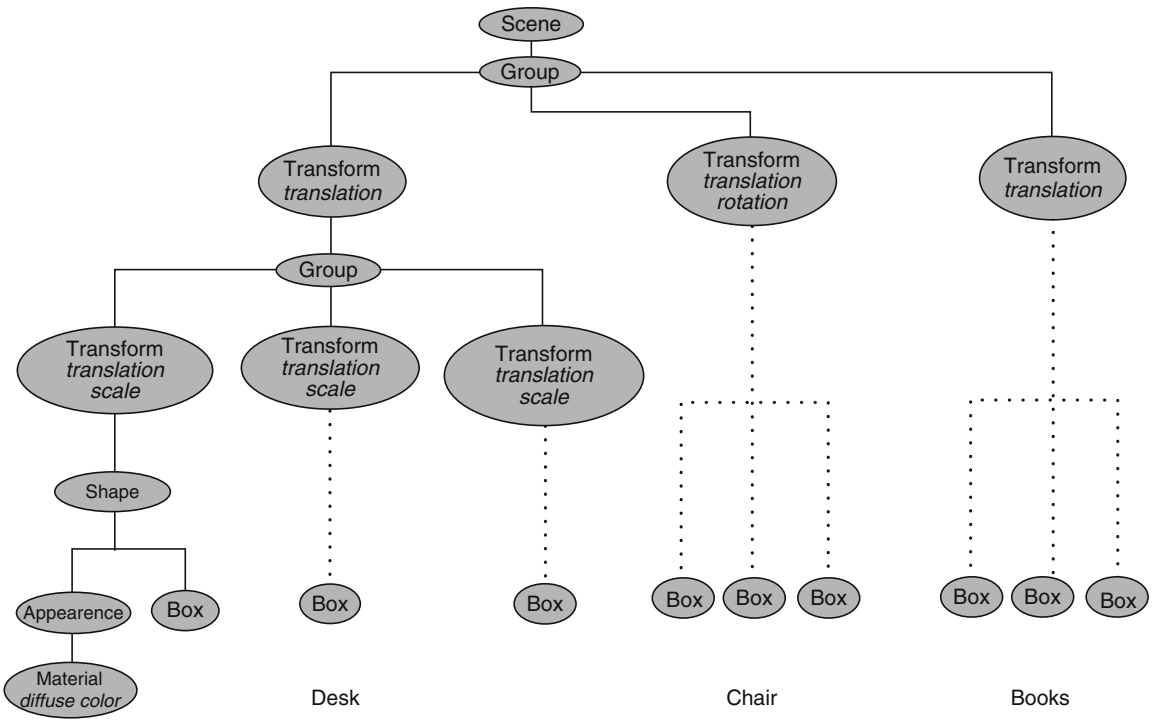
A 3D scene is a condensed information space (geometric description, spatial organization, etc.). The richness of a 3D scene cannot be fully exploited if information is not explicitly materialized. In our example, the fact that the chair is under the table cannot be directly deduced from the X3D description of the scene. Geometric constructs embed information that it is easily understood by human actors, but raw information, globally, remains unexploitable by means of queries without further analysis. Consequently, some spatial analysis is required.



Indexing Three Dimensional Scenes. **Figure 1.** A researcher's office modeled using X3D.

Even though the structural organization of an X3D file facilitates the retrieval of attributary information (*position, color, appearance*), a series of spatial or semantic queries still remain unanswered. A complementary description, by means of annotations, should take into account the semantic aspects of information associated with the scene (e.g. the chair is allowing the researcher to sit at the table – semantic relations, the chair cannot support a weight heavier than 200 kg – semantic properties). Firstly, the annotations should allow identifying the geometric elements that correspond to the representation of a real world object. Secondly, semantic properties should be associated with the related real-world object, as well as relations it has with its environment.

The boxes are imposed an adequate size using the *scale* attributes of *Transform* nodes as in Fig. 3. They



Indexing Three Dimensional Scenes. **Figure 2.** A 3D scene tree.

```
<X3D profile="Core">
  <Scene>
    <Group>
      <Group id ="desk">
        <!--desk top-->
        <Transform translation="09-50"
                              scale="25 0.5 10">
          <Shape>
            <Box />
            <Appearance>
              <Material diffuseColor="0 0 1" />
            </ Appearance>
          </ Shape>
        </ Transform>
        <!--left leg -->
        <Transform translation="–17.50 –50"
                              scale="0.5 9 10">
          <Shape>
                    ...
          </Shape>
                  </Transform>
                  <!--right leg-->
                  <Transform translation="17.50 –50"
                            scale="0.5 9 10">
                    <Shape>
                            ...
                    </Shape>
                  </Transform>
                </Group>
                  ..
                <Transform translation = "50 2 –40"
                                      rotation="0 1 0 1.57">
                  <Group id="chair>
                   ...
                  </Group>
                 </Transform>
                  ...
               </Group>
             </Scene>
</X3D>
```

Indexing Three Dimensional Scenes. Figure 3. An excerpt of an X3D scene model.

are grouped together into *Group* nodes, placed at the right position using the *translation* attribute, and are imposed a fixed orientation using the *rotation* attributes.

Localizing Real World Objects in a 3D Scene

In order to explain the localization of real world objects in a 3D scene one should take also into account the work done in the domain of 2D images. The localization supposes the identification of a set of elements that constitutes the representation of the object. For instance, in the case of a 2D image objects are associated to sub-regions delimited by sets of 2D polygons.

The nature of localized elements can vary from pure geometric elements (*points, lines, surfaces, volumes*) to complex document entries (e.g. a cluster of geometric elements). Hence, we can observe two types of localization, both *geometric* and *structural* localizations.

The Structural Localization uses the structure of a document to indicate the document entries that corresponds to the target element. For instance, if the document is XML-like, then a structural localization would be likely composed of a series of XPATH [5] expressions indicating the parts of the object in the scene.

Let us consider the scene described in Fig. 3. The localization of the desk legs can be expressed as the combination of the two following XPATH expressions: /X3D/Scene/Group/Group[@id = "desk"]/Transform [position() = 2] and //Group[@id = "desk"]/Transform[position() = 3].

This localization contains information about elements that materialize the legs (*Shape*) as well as their position (*Transform*) inside the local cluster (*Group id = "desk"*). The structural localization should target the smallest structural element that offers an adequate description for the object.

Sometimes, this condition cannot be satisfied. It is possible that the structure of the document is not as fine-grained as necessary. In our example, we could imagine that the bottom of the stack of papers contains articles published in the last month while the top of the stack contains articles about the multimedia indexing. The structural localization does not allow localizing the articles at the bottom of the stack since no structural element corresponds to the respective object. The whole stack is defined as a box, and, hence the same structural element is used to represent two real world objects. The only solution to identify the object is to complete the structural localization using a geometric localization. In our example, we can isolate the multimedia indexing related papers with a volume such as, a cube having the same basis as the box and a fixed height. The cube coordinates are defined relatively to the box. In this case, there is a mix between one structural localization and a geometric one. However, the geometric localization could be completely separated from the structure of the scene. In this case, the volume should have been defined according to the global reference system. The situation presented above has many similarities with multimedia documents that do not have any internal structure which

is often the case in the presence of raw images, simples Digital Terrain Models [6], etc.

When the geometric localization is used, geometric queries are applied on the scene elements in order to obtain the precise geometric elements composing the object. The granularity of the retrieved geometric elements (pixels, primitives, etc.) – subject of negotiation – is chosen in order to meet specific requirements.

As in the case of 2D images region identification, one can think of three ways of performing localization:

1. Manual localization: the user defines the set of objects and their localization in the scene.
2. Semi-automatic localization: the machine suggests to the user a series of possible objects in the scene. In the case of an X3D structured scene the machine could associate to each Group node a virtual object leaving then the user choose the relevant ones. In the case of an unstructured 3D scene (a Digital Terrain Model), a signal-level analysis could yield interesting regions using algorithmic methods similar to those employed in 2D contour detection for instance [7]. Other propositions suggest the use of specific tools – like intelligent scissoring of 3D meshes [8], etc. – in order to define precise geometric localization.
3. Automatic localization: the system performs all the work. In order to achieve this degree of generality dependant domain knowledge should be implemented in the localization process.

Indexing Multimedia Content using MPEG-7

The Moving Picture Experts Group (MPEG) is a working group of the ISO/IEC in charge of developing standards for coded representation of digital audio and video. Established in 1988, the group has produced MPEG-1, MPEG-2, MPEG-4, MPEG-7 and MPEG-21 "Multimedia Framework." MPEG-1, MPEG-2 and MPEG-4 are basically standards concerning the encoding and the transmission of audio and video streams. MPEG-7 is a standard that addresses the semantic description of media resources. MPEG-21 is much more considered as the description of a multimedia framework (coding, transmission, adaptation, etc.) than a specific standard. Even if the MPEG-7 and MPEG-21 were proposed in the context of digital audio and video data, they are highly extensible and could cover other areas. Due to its high capability of evolution, we consider MPEG-7 as a valuable candidate for fulfilling requirements in terms of semantic annotations inside a 3D scene. Hereafter, we focus on the MPEG-7.

MPEG-7 [9,10] formally named "Multimedia Content Description Interface," was officially approved as a standard in 2001. It provides multimedia content description utilities for the browsing and retrieval of audio and visual contents. This standard provides normative elements, such as, Descriptors (Ds), Descriptors Schemes (DSs) and a Description Definition Language (DDL).

The Ds are indexation units describing the visual, audio and semantic features of media objects.

They allow the description of low-level audio-visual features (color, texture, animation, sound level, etc.) as well as attributes associated with the content (localization, duration, quality, etc). Moreover, visual and audio features are automatically extracted from the encoding level of the media object as described in [11,12] for visual features and [13,14] for audio. Semantic features are mainly added manually.

The DSs are used to group several Ds and other DSs into structured, semantic units. A DSs models a real life entity and the relations it holds with its environment. DSs are usually employed to express high-level features of media content like: objects, events, and segments, metadata linked to the creation, generation and usage of media objects. As for the semantic features, DSs are scarcely filled in automatically.

In order to offer an important degree of extensibility to the standard the DDL is included as a tool for extending the predefined set of Ds and DSs formerly proposed. The DDL defines the syntax for specifying, expressing and combining DSs and Ds allowing creating new DSs.

The existing DSs cover the following areas: visual description (VDS), audio description (ADS), multimedia content description (MDS) (general attributes and features related to any simple or composed media object). We focus on the MDS as it addresses organization aspects that could serve as a valuable starting point in order to extend the indexing capabilities towards 3D documents. The VDS and the ADS are matched against the physical/logical organizations or semantics of the document.

MDSs propose metadata structures for annotating real world or multimedia entities. MDS is decomposed on the following axis: Content Organization,

Navigation and Access, User Interaction, Basic Elements, Content Management, Content Description.

We discuss more in detail the *Content Description* axis as it offers DSs for characterizing the physical and the logical structures of the content. It ensures also the semantic description using real world concepts. The basic structural element is called a *segment*. A *segment* corresponds to a spatial, temporal or spatio-temporal partition of the content. A segment (Audio Segment DS, Visual Segment DS, AudioVisual Segment DS, and Moving Regions DS) is associated with a *section*. It can be decomposed in smaller segments creating a hierarchical segmentation of the media content. Each segment is individually indexed using the available tools (visual DSs, audio DSs, &).

The conceptual aspects of the media content are formulated using the Semantic DS. Objects (*ObjectDS*), events (*EventDS*), abstract concepts (ConceptDS), places (*SemanticPlaceDS*), moments of time (*SemanticTimeDS*) are all parts of the Semantic DS. As for the segment-based description of the content, the semantics can be organized as trees or graphs. The nodes represent semantic notions/concepts and the links define semantic relations between concepts. A semantic characterization of the office scene (Fig. 1) using MPEG-7 tools is illustrated below (Figure 4).

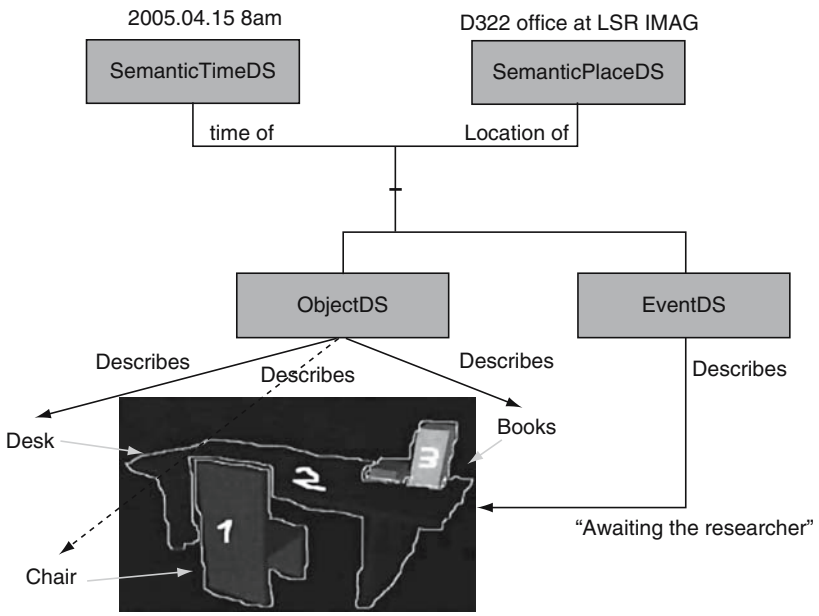
The structural schemas and the semantic ones could be linked to each other. Hence, the content description could be made out of content structure and semantic structure all together.

Conclusions

In this article we presented some problems that are inherent to the process of indexing and reuse of 3D scenes. Widely accepted standards exist for the modeling of 3D scenes (X3D) and multimedia indexing (MPEG-7). They enhance, respectively, the Web deployment of 3D information and the management of multimedia content. However, at our knowledge, no largely accepted research project aims at solving interoperability issues between the two standards in order to support the management – and notably the reuse – of 3D multimedia content.

The interoperability issues concern the capacity of addressing 3D content inside a multimedia document. Specific 3D region/object locators must be designed. A set of specific 3D spatial descriptions schemes have to be provided in order to facilitate the spatial analysis required by most of 3D application domains (spatial planning, transports).

Research efforts will address the interoperability issues between standards in order to improve in flexibility and reuse of 3D scenes. Work is to be performed in order to enhance the capacities of MPEG-7 to address and to characterize 3D content. The semantic added to the pure X3D geometric modeling of the



Indexing Three Dimensional Scenes. **Figure 4.** A semantic description of the image issued from the office scene using MPEG-7 tools.

scene enhances the reuse process of 3D objects. Complex geometric, spatial and/or semantic queries could then be formulated in order to extract the most appropriate 3D content according to specific needs of applications or scene designers.

References

1. S. Zlatanova and K. Templi, "Modeling for 3D GIS: Spatial Analysis and Visualisation through the Web," in M. Molennar and K.J. Beek (Eds.) "International Archives of Photogrammetry and Remote Sensing" Vol. 33, 19th Congress ISPRS, Amsterdam, 2000, pp. 1257–1264.
2. Web3D Consortium, "Information technology – Computer Graphics and Image Processing – Extensible 3D (X3D) – Part 1: Architecture and Base Components," ISO/IEC FDIS 19775–1:200x, 2001.
3. R. Carey, G. Bell, and C. Marrin, "Virtual Reality Modelling Language (VRML97) Functional Specification," ISO/IEC 14772–1, 1997, Available at http://www.web3d.org/x3d/specifications/vrml/.
4. T. Bray, J. Paoli, C.M. Sperberg-McQueen, and E. Maler, "Extensible Markup Language (XML) 1.0 (2nd edn.)," W3C Standard, Available at http://www.w3.org/TR/REC-xml, 1998.
5. J. Clark and S. DeRose, "XML Path Language (XPath) Version 1.0," *W3C* Recommendation, Available at http://www.w3.org/TR/xpath, 1999.
6. R. Weibel and M. Heller, "Digital Terrain Modeling," in D.-J. Maguire, M.F. Goodchild, and D.W. Rhind, (Eds.) "Geographical Information Systems: Principles and Applications," Longman: London, 1991, pp. 269–297.
7. D. Ziou and S. Tabbone, "Edge Detection Techniques – An Overview," International Journal of Pattern Recognition and Image Analysis, Vol. 8, 1998, pp. 537–559.
8. T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and David Dobkin, "Modelling by Examples," ACM Transactions on Graphics (SIGGRAPH'2004), Los Angeles, CA, pp. 652–663, August 2004.
9. J.M. Martinez and R. Koenen, "MPEG-7: The Generic Multimedia Content Description Standard, Part 1," IEEE Multimedia, Vol. 9, No. 2, April 2002, pp. 78–87.
10. J.M. Martänez, "MPEG-7: Overview of Description Tools, Part 2," IEEE Multimedia, Vol. 9, No. 3, July–September 2002, pp. 83–93.
11. A. Yamada, M. Pickering, S. Jeannin, L. Cieplinski, J.R. Ohm, and M. Kim, "MPEG-7 Visual Part of Experimentation Model Version 9.0," ISO/IEC JTC1/SC29/WG11 N3914, 2001.
12. MPEG-7 XM Software, Institute for Integrated Circuits, Technische Universität Munchen, Germany, June 2001, Available at http://www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/e_mpeg7.html.
13. M. Casey, "MPEG-7 sound-recognition tools," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, June 2001, pp. 737–747.
14. A. Zils and F. Pachet, "Automatic Extraction of Music Descriptors from Acoustic Signals using EDS," Proceedings of the 116th AES Convention, May 2004.

Industrial Virtual Trainer

▶ Interactive Virtual Humans in Mobile Augmented Reality

Infrared Face Recognition

Definition

Infrared face recognition systems use infrared sensors to measure the thermal radiation emitted in the infrared spectrum range.

One of the major problems of traditional face recognition systems is constant performance under uncontrolled environments, and especially under extreme variations in illumination conditions, e.g., operating in total darkness or full daylight in an open area surveillance scenario. Such problems may be alleviated using infrared (IR) images for face recognition. Unlike conventional visual cameras, which measure the electromagnetic energy in the visible spectrum range, infrared sensors measure the thermal radiation emitted in the infrared spectrum range (0.7–0.14 μm) [1]. Thermal images of the human face represent patterns caused from superficial blood vessels up to 4 cm below the skin surface, which transport warm blood throughout the body and heat the skin just above them by an average of 0.1°C [2]. The vein and tissue structure of an individual is unique, even for identical twins [1], and thus ensures that except in case of aging, arterial problems, injury or surgery the vascular patterns acquired by IR cameras can be used for identification. Some examples of infrared face images can be seen in Fig. 1.

IR face recognition systems are unaffected by variations in illumination and unlike systems using visual light, they can work without any problem under all lighting conditions, even in complete darkness. They are also unaffected by skin color, suntan, use of cosmetics and colored eye lenses or even plastic surgery. The latter, although it would defeat a visual face recognition system since it would change drastically facial appearance (e.g., facial lift, removal of wrinkles, use of silicon implants, etc), it would not affect IR face recognition, because it does not intervene with the network of blood vessels [2]. IR systems are very robust to impostors using masks or make-up or other means of forgery, because they can readily distinguish between

Infrared Face Recognition. Figure 1. Examples of IR facial images (taken from Equinox IR face database [3]).

real and artificial skin, hair, etc, based on different values of emissivity. Nevertheless, the extremely high cost of IR sensors, makes the use of IR face recognition systems prohibitive for every day applications.

Like visual face images, thermal images are processed for recognition using appearance-based techniques like PCA, or feature based techniques that locate and use features like the corners of the eye where the upper and lower eyelids meet (canthi), the curves produced by the main facial arteries of the two cheeks, the position and angles of main arteries under the forehead, etc [1]. Contour matching techniques are also suitable for IR recognition. The use of multimodal visual and IR face recognition systems has also been proposed.

Cross-References

► Face Recognition

References

1. S.G. Kong, J. Heo, B.R. Abidi, J. Paik, and M.A. Abidi, "Recent Advances in Visual and Infrared Face Recognition – A Review," Computer Vision and Image Understanding, Vol. 97, No. 1, January 2005, pp. 103–135.
2. F.J. Prokoski and R.B. Riedel, "Infrared Identification of Faces and Body Parts," BIOMETRICS: Personal Identification in Networked Society, Kluwer Academic, Dordecht,
3. Equinox public access IR face database: http://www.equinoxsensors.com/products/HID.html.

Infrastructure and Engineering

Wayne Robbins
Defense R&D Canada, Ottawa, ON, Canada

Definition

Multimedia systems rely on a wide variety of infrastructural technologies to enable their communication, processing, and interface/display needs.

In addition, particular classes of media (e.g., continuous media such as audio and video) often require special computational support to ensure their correct rendering. As such, different categories of infrastructure can be defined as to how systems enable and influence synchronized media playback. These include: operating and real-time systems, middleware and networking, database and data management as well as system and software engineering.

Operating and Real-Time Systems

The operating system (OS) [1–5] is a key element in multimedia synchronization due its fundamental infrastructural role within end-user and network equipment. Specifically, a number of OS issues can cause significant impact in situations where continuous media are utilized. Such time-sensitive data, typically large in volume and isochronous in nature, often require significant computing power to provide a responsive system with predictable behavior. Specific issues which need to be considered include:

1. Process Management
2. Time Management
3. Memory Management
4. Inter-Process Communication (IPC)
5. Resource and Device Management
6. User Interface and Display Management

Process management addresses the area of process (program) execution and processor allocation. For continuous time-based media, predictable processing is required to ensure media readiness for display in accordance with its isochronous nature. Relevant process management issues therefore include scheduling algorithms and priority mechanisms, such that inappropriate scheduling and misuse of priorities can introduce asynchrony into a multimedia application due to inappropriate delays in processing (e.g., other applications "grabbing" the processor).

Time management addresses the issue of whether the OS can both ensure adequate temporal accuracy for application-level synchronization efforts as well as if the OS itself can operate in a timely and synchronized manner. Included in this area are issues of clock management as well the availability of synchronization primitives (for process and resource/device management).

Memory management addresses how the OS controls memory allocation and applications' access to memory spaces. Memory protection, virtual memory (VM), shared versus non-shared models, dynamic versus static allocation and automatic garbage collection (in conjunction with certain language/run-time environments) falls in this category. These issues influence media synchronization by affecting data transfer between applications and devices as well as possibly inducing asynchrony into process execution due to automatic system overhead (e.g., use of VM swap files and automatic "cleanup" techniques sometimes associated with dynamic memory/resource allocation).

Inter-Process Communication (IPC) plays a significant role in multimedia synchronization due to the potential to incur delays when transferring large amounts of data between processes. Delays can result from both the data duplication itself but also the incidental (and often hard-to-determine) costs of process management (context switching) and memory management overhead. The result is unexpected and undeterministic delays within the operating system's own execution which (then) ultimately affect the multimedia application.

Resource management refers to how the operating system provides and controls access to any resource (e.g., piece of hardware/software construct). *Device management* specifically refers to the appropriate means to control and facilitate data flow to/from devices; multimedia-oriented examples include capture and rendering devices, hardware codecs, storage and communication equipment and so forth. Because multimedia applications require the timely delivery of data to end-users, devices must enable fine-grain data and control flow in order to ensure asynchrony is not introduced at the final stages just prior to being rendered to the user. Similarly, when media are generated in real-time, any asynchrony resulting from the capture process (i.e., via input devices) can create timing errors which may affect data flow and be difficult to account for in subsequent processing.

User interface and display management issues are also important to multimedia synchronization in much the same way as device management. That is, the display management and user interface subsystems need to provide low-overhead rendering and user interaction with multimedia components. Slow rendering times and unresponsive user interfaces are not viable for time-based media or interactive systems. It is also important to consider the affects of manipulating the visual data (vis-à-vis the aforementioned OS issues) and how synchronization methodologies are compatible with user interaction (e.g., mouse pointer synchronization) [6].

The previous discussion illustrates that multimedia synchronization can be intimately affected by a system's low-level infrastructure. As such, many of these requirements can best addressed in the context a *real-time system* [7] – one whose correct operation depends both on its logical results as well as the temporal properties of its behavior. Such systems are typically characterized as deterministic, with the ability to provide timely responses and flexible scheduling abilities while also providing for security, fault tolerance and robustness. Classic examples include factory robot control and avionic subsystems.

Consequently, the key to real-time systems is highly accurate, temporal predictability; therefore, real-time systems are not necessarily fast but "temporally pedantic," since early event occurrence can be just as damaging as incurring delays. Characterized in Table 1, two classes of real-time systems are defined based on the severity of temporal errors, namely "hard" and "soft" real-time systems. Hard real-time systems are those in which any violation of a timing constraint is considered a system failure. Timely execution is guaranteed through resource allocation based on the worst-case situation, usually resulting in under-utilized resources during normal operation, possibly requiring complete system shutdown when any anomalies occur. Conversely, soft real-time systems are those in which a violation of a temporal constraint does not constitute a system failure.

Accordingly, multimedia systems are generally classified as soft real-time systems because their temporal performance requirements are usually not so restrictive; for example, asynchrony in a presentation

Infrastructure and Engineering. Table 1. Real-time system classification comparison

Aspect	Traditional Hard Real-Time	Multimedia Soft Real-Time
Data Characterization	Small, often local or with controlled distribution	Large amounts, often heavily distributed
Temporal Accuracy	Strict and static deadlines	Approximate and dynamic timelines
Error Severity	Worst case must be met	Quality of service considerations

may degrade its quality and annoy its viewers, but no physical damage results. The human-centric character of multimedia systems also facilitates a range of "acceptable" playback quality which varies with the media, the context of their use and ultimately, the individual users. Consequently, human perceptual limitations can be used to relax certain timing constraints, enabling a choice between which playback characteristics are most important and facilitating potential trade-offs between functionality and resource usage. Such an approach maps well to soft real-time systems, in which performance is not guaranteed by worst-case resource allocation. For example, if the "bandwidth" of a video channel is constrained to only 15 fps at a specific resolution, the user could decide to accept the provided quality or adjust select parameters more aptly suit his/her needs.

Middleware and Networking

Middleware and networking are also important to multimedia synchronization in that they affect the delivery of media data between end (client) systems.

At a most basic level, the communications infrastructure must ensure data availability to enable synchronized rendering and timely user interaction. Typically, this issue is addressed by providing for a reasonable and ideally predictable quality of service (QoS). Therefore, network QoS can be seen as an enabler for "temporal composition" by which media can be assembled together and playback organized according to individual and group timing constraints. The provision of appropriate network and application level protocols also support synchronized data transfer (in cooperation with any provided QoS). A large body of work on protocols for multimedia synchronization exists [8–13], ranging from lower-level adaptive, feedback-based techniques to those provisioned at the application level, such as RTP (Real-Time Protocol) and RTCP (Real-Time Control Protocol). Additional network-oriented considerations include issues of data buffer management and protocol stack implementation which can impact on synchronization vis-à-vis the OS issues described above (e.g., data copying overhead).

Beyond the basic communications level, middleware [14–16] addresses the need to bridge network and client functionality through the provision of centrally-based services and abstractions. As such, middleware is a "glue" layer of software between the network and applications, intended to ease application programming, application integration and system management tasks while also promoting standardization and interoperability of services by lessening multiple, independently developed implementations. In terms of multimedia synchronization, middleware offers a logically centralized, service-oriented approach to synchronization (orchestration) logic. It also provides support for useful abstractions and constructs for communicating multimedia data, ranging from publish and subscribe models, to streams, flows, sources and sinks.

Database and Data Management

Database and data management [8,17–19] are relevant to multimedia synchronization in how their design and implementation provide flexible and responsive data access. For aspects of spatial and content synchronization, issues of multimedia querying and multimedia data semantics (e.g., image analysis vs. keyword meta-descriptors) are of interest. For purposes of temporal synchronization, a broad array of other issues includes disk scheduling and storage models for particular classes of data (e.g., continuous media). This last aspect also includes how the fundamental database structure impacts the means by which the actual multimedia data is accessed; that is, do the media reside within the database itself (in which access is constrained to the database management system and query engine) or is the data stored independently on separate systems (and the database only contains

references to the external data). Such design considerations must be accounted for due to two primary issues: (1) timing considerations in terms of media data retrieval strictly through the database and its overhead (e.g., the potential effects of multiple, concurrent database queries on the timeliness of continuous media streams); and (2) timing considerations in terms of database efficiency resulting from large data objects (such as video) and/or objects of variable and indeterminate size (e.g., how to represent live instances of media, such as a camera capture).

System and Software Engineering

System and software engineering issues are important to multimedia synchronization in how they can affect the real-time implementation of multimedia systems. To provide an actual real-time environment, systems must be appropriately engineered not only to facilitate the necessary structural and behavioral aspects of a system, but also to ensure inappropriate behavior is not inadvertently introduced and that any such anomalies can be corrected as required.

First, a system should be based on the appropriate hardware and software infrastructure, such as a QoS-enabled communication backbone and a real-time operating system. Systems based on inappropriate infrastructure risk reduced quality in the user experience due to unsuitable substrate behavior [20,21]. Second, system structure (both design and implementation) must provide a flexible and extensible architecture capable of real-time performance. This requirement includes using flexible architectural techniques and technologies, including middleware and component-oriented architectures, along with the appropriate programming interfaces, useful abstractions and developmental paradigms (such as object orientation). Third, the system, application and various software components should have the ability to monitor their behaviors [22,23] (i.e., the actual performance of its various components). This is a necessary step in creating a system which can adapt (i.e., "tune") itself to address structural or behavioral deficiencies. An example is a video system which provides the ability to dynamically change playback frame rate based on monitoring the degree of asynchrony that develops during playback. Doing so illustrates the benefit of building systems that address behaviors as first-class considerations and facilitate adaptive behavior management [24].

As a result of better engineering, the potential exists for more flexible and higher quality systems, based on the increased use of common multimedia infrastructures. The end result would be better interoperability and compatibility across the user community, ultimately aiding in the acceptance and continued growth of multimedia technology across broader audiences.

Cross-References

► Multimedia Synchronization – Area Overview

References

1. T.M. Burkow, "Operating System Support for Distributed Multimedia Applications: A Survey of Current Research," Technical Report (Pegasus Paper 94–8), Faculty of Computer Science, University of Twente, 1994.
2. R. Steinmetz, "Analyzing the Multimedia Operating System," IEEE Multimedia, Vol. 2, No. 1, 1995, pp. 68–84.
3. M. Singhal and N.G. Shivaratri, "Advanced Concepts in Operating Systems: Distributed, Database and Multiprocessor Operating Systems," McGraw-Hill, New York, 1994.
4. R. Govindan and D.P. Anderson, "Scheduling and IPC Mechanisms for Continuous Media," Proceedings of the 13th ACM Symposium on Operating System Principles, ACM, 1991, pp. 68–80.
5. V. Baiceanu, C. Cowan, D. McNamee, C. Pu, and J. Walpole, "Multimedia Applications Require Adaptive CPU Scheduling," Technical Report, Department of Computer Science and Engineering, Oregon Graduate Institute of Science and Technology, 1996.
6. S. Greenberg and D. Marwood, "Real-Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface," Research Report 94/534/03, Department of Computer Science, University of Calgary, 1994.
7. S-T. Levi and A.K. Agrawala, "Real-Time System Design," McGraw-Hill, New York, 1990.
8. D.P. Anderson and G. Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism," IEEE Computer, Vol. 24, No. 10, 1991, pp. 51–57.
9. T.D.C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," IEEE Journal on Selected Area in Communications, Vol. 9, No. 12, 1991, pp. 1368–1382.
10. S. Ramanathan and P. Rangan, "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems," The Computer Journal, Vol. 36, No. 1, 1993, pp. 19–31.
11. I.F. Akyildiz and W. Yen, "Multimedia Group Synchronization Protocols for Integrated Services Networks," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, 1996, pp. 162–173.
12. J. Escobar, C. Partridge, and D. Deutsch, "Flow Synchronization Protocol," IEEE/ACM transactions on Networking, Vol. 2, No. 2, 1994, pp. 111–121.

13. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Request for Comments rfc1889, 1996.
14. A. Campbell, G. Coulson, F. Garcia, and D. Hutchinson, "Orchestration Services for Distributed Multimedia Synchronization," Technical Report MPG-92–52, Distributed Multimedia Research Group, University of Lancaster, 1992.
15. I. Herman, N. Correia, D.A. Duce, D.J. Duke, G.J. Reynolds, and J. Van Loo, "A Standard Model for Multimedia Synchronization: PREMO Synchronization Objects," Multimedia Systems, Vol. 6, No. 2, 1998, pp. 88–101.
16. S. Didas, "Synchronization in the Network-Integrated Multimedia Middleware (NMM)," Project Report, Universität des Saarlandes, Saarbrücken,
17. S. Marcus and V.S. Subrahmanian, "Foundations of Multimedia Database Systems," Journal of the ACM, Vol. 43, No. 3, 1990, pp. 474–523.
18. B. Özden, R. Rastogi, and A. Silberschatz, "The Storage and Retrieval of Continuous Media Data," in V.S. Subrahmanian and S. JajodiaMultimedia (Eds.), Database System: Issues and Research Direction, Springer, New York, 1996, pp. 237–261.
19. W. Klas and K. Aberer, "Multimedia and its Impact on Database System Architectures," in P.M.G. Apers, H.M. Blanken, and M.A.W. Houtsma (Eds.), Multimedia Databases in Perspective, Springer, Heidelberg, Germany, 1997, pp. 31–61.
20. D.C.A. Bulterman and R. van Liere, "Multimedia Synchronization and Unix," in R.G. Herrtwich (Ed.), Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video, 1991, pp. 108–119, Springer, Heidelberg, Germany.
21. R.W. Robbins, "A Model for Multimedia Orchestration," M. Sc. Thesis, Department of Computer Science, University of Calgary, 1995.
22. J.A. Boucher and T.D.C. Little, "An Adaptive Real-time Software-Only Player for the MPEG-I Video Stream," Technical Report, Multimedia Communications Lab, Boston University, 1994.
23. F. Jahanian, "Run-Time Monitoring of Real-Time Systems," in S.H. Son (Ed.), Advances in Real-Time Systems, Prentice-Hall, Englewood Cliffs, 1995, pp. 435–460.
24. R.W. Robbins, "Facilitating Intelligent Media Space Collaboration via RASCAL: The Reflectively Adaptive Synchronous Coordination Architectural Framework," Ph.D. Thesis, School of Information Technology and Engineering, University of Ottawa, 2001.

Inpainting in Virtual Restoration of Artworks

Definition

Inpainting techniques are used to enhance the quality of the digitized artwork during virtual restoration.

Digital imaging techniques are widely used to restore, interpret and preserve collections of visual cultural heritage [1–3]. However, although proper storage conditions and controlled access to the cultural heritage pieces are routinely applied, materials, such as mural, canvas, vellum, photography, and paper medium are invariably exposed to various aggressive environmental factors which lead to the deterioration of the perceived image quality. Therefore, digital solutions for archiving and popularization of heritage collections through the online access to digitized artwork virtual museums are of paramount importance.

One of the most critical issues in digitized artwork images restoration is that of crack removal and fading color enhancement. Cracks are breaks in the media, paint, or varnish of the original artwork caused mostly by aging, drying or mechanical factors [1,3]. Fading colors and obscure shadows are caused mostly by environmental conditions, such as sunshine, oxidation, temperature variations, humidity, and the presence of bacteria, affect the original artwork causing [1,2]. Both undesirable effects result in significant variation in the color characteristics and pigmentation of the artwork, preventing proper recognition, classification and dissemination of the corresponding digitized artwork images.

During virtual restoration the quality of the digitized artwork can be enhanced using inpainting techniques (Fig. 1). For example, in the system presented in [1], the user manually selects a point on each crack in order to achieve the proper differentiation between damaged areas and regular pixels. Once the reference pixel is defined, an automated procedure uses the threshold values to compare the intensity of the reference pixel and its neighbors in localized image area in order to determine the crack's mask. Note that the degree of user interaction can be reduced by utilizing a sophisticated crack detection module [3]. After a crack is completely localized, image inpainting is used to fill-in the corresponding spatial location with image interpolated values. This step completes the restoration process [1,3]. Similarly to the crack removal, a region with faded colors and obscure shadows must be first localized [2]. Then, the user by selecting target colors from a color template and an impainting method fills-in the detected gaps and restores both intensity and color information.

Inpainting in Virtual Restoration of Artworks. Figure 1. Virtual restoration of artworks: (a) digitized artwork image with presented cracks, (b) image with localized cracks, (c) image reconstructed using an inpainting technique.

Cross-References

- ▶ Color Image Filtering and Enhancement
- ▶ Digital Inpainting
- ▶ Image Inpainting
- ▶ Video Inpainting

References

1. M. Barni, F. Bartolini, and V. Cappelini, "Image Processing for Virtual Restoration of Artworks," IEEE Multimedia, Vol. 7, No. 2, April-June 2000, pp. 34–37.
2. X. Li, D. Lu, and Y. Pan, "Color Restoration and Image Retrieval for Donhuang Fresco Preservation," IEEE Multimedia, Vol. 7, No. 2, April-June 2000, pp. 38–42.
3. I. Giakumis, N. Nikolaidis, and I. Pitas, "Digital Image Processing Techniques for the Detection and Removal of Cracks in Digitized Paintings," IEEE Transactions on Image Processing, Vol. 15, No. 1, January 2006, 178–188.

Integral Images for Fast Covariance Computation

▶ Object Tracking in Video using Covarience Matrices

Integrated QoS Architecture of LOMSS

Definition

QoS architecture of large-scale object-based multimedia storage systems is established by extending the standard T10 OSD protocol using am new type of objects.

In Large-scale Object-based Multimedia Storage Systems (LOMSS), a multimedia file is striped and then, stored as objects, which may contain the requirements of Quality of Service (QoS). Many QoS solutions that have been developed successfully for networks cannot be applied directly to LOMSS and thus, more researchers have focused on the QoS in this domain recently. For example, ODIS (Object Disk I/O Scheduler) handled this issue with the bandwidth "maximizer" adaptation technique [1]. A QoS provisioning framework for OSD-based storage systems has been proposed in the literature, which extends the existing OSD and iSCSI protocol in order to support QoS specifications [2].

To establish an integrated QoS in LOMSS, the standard T10 OSD protocol [3] shall be extended and a new type of objects shall be introduced: *Method Object*, which is a special object that can be executed in the OSDs to perform operations defined by users. Normally, the *Method Objects* can be classified into two categories: *System Method Object* and *User-defined Method Object*. System Method Objects are open to all the users and can be installed into the OSDs during the setup time; whereas, User-defined Method Objects can be uploaded to the OSDs by users through registration when the system is running. Integrated QoS can support the QoS requirement intelligently at storage device level, which can be further divided into three-level schedulers in an OSD: Upper Level Scheduler (ULS), Middle Level Scheduler (MLS), and Low Level Scheduler (LLS). These schedulers can work cooperatively to satisfy the QoS requirements of the requests under the constraints of limited hardware resources. ULS classifies the arriving object requests according to their required communication QoS and assigns each class of requests with different priority, and then, schedules the object requests according to their priorities. MLS transforms object read/write requests arriving from ULS into corresponding block read/write requests, and controls the block rate of read/write requests deliberately that will be sent to LLS. LLS

takes charge of scheduling block read/write requests and provides the QoS assurance in disk drive level.

Another important extension is to deal with the object attributes. In LOMSS, the objects have various attributes; whereas, in the traditional block-based storage systems, there are no attributes associated with blocks. With the object attributes, an object can maintain more details about the data, e.g., QoS requirements. Access information and control information of the objects may be recorded and evaluated by LOMSS explicitly or implicitly, which can provide the QoS information for scheduling operations in the system. Object attributes have broken through the limits of traditional block-based storage systems where there is no information to describe the data. By adding the attributes of objects that are not defined in OSD-R10 [3], the state-of-the-art integrated QoS architecture in Large-Scale Object-Based Multimedia Storage Systems (LOMSS) aims at providing consistent, stable, class-based, and robust QoS services for the large-scale multimedia applications.

Cross-References

► Object-based Storage Devices of LOMSS

References

1. J.C. Wu and S.A. Brandt, "The Design and Implementation of AQuA: An Adaptive Quality of Service Aware Object-Based Storage Device," Proceedings of the 23rd IEEE/14th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 06), pp. 209-218, May 2006.
2. Y.P. Lu, D.H.C. Du, and T. Ruwart, "QoS Provisioning Framework for an OSD-Based Storage System," In Proceedings of the 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 05), pp. 28-35, 2005.
3. T10 Technical Editor. Information technology-SCSI object-based storage device commands. T10/1355-D, Revision 10, 30 July, 2004.
4. Intel Corporation, "Intel iSCSI Reference Implementation," http://www.intel.com/technology/computing/storage/iscsi/index.htm.

Integration of Collaborative Computing With Modeling and Simulation Technologies

Definition

The blending of collaborative computing together with modeling and simulation (M&S) technologies continues to gain popularity amongst the wide range of distributed work teams that routinely utilize M&S in the course of their work.

With examples ranging from the traditional scientific and engineering communities to business process and management domains, this context of collaborative computing addresses a very broad basis for "working together," ranging from the use of shared workspaces and participant awareness/interaction via conferencing [1,2] to the use of distributed modeling and development environments [3].

Traditionally, M&S technologies have been extremely popular in scientific research, military and defense arenas, systems design and engineering. When compared to standalone systems and practices, M&S executed in a collaborative manner offers the benefits and challenges of participating in a distributed system – specifically the availability of additional resources. The result is typically an increased simulation capability along with more productive and richer discovery and evaluatory processes.

Depending on the audience, collaboration can be applied at different levels in relation to the simulation activity. For example, collaboration could involve use of distributed simulation tools (such as HLA or DIS-based simulators) or it could involve the use of "facilitation-type tools" to augment the communication and interaction between collaborating teams. While not mutually exclusive, and in fact mutually beneficial, the choice of approach is often organizationally inclined; for example, scientists may not have the same fondness for teleconferencing as business process modelers. Successful collaborative M&S experiences are therefore often studied and evaluated in a case-specific manner [4].

Current trends in this area relate to portal-based access to simulations, collaborative tools and shared work spaces [1,2], visual development [3], and grid-based computing initiatives [5].

Cross-References

► Collaborative Computing – An Overview

References

1. S.J.E. Taylor, "Netmeeting: A Tool for Collaborative Simulation Modelling," International Journal of Simulation, Systems, Science and Technology, Vol. 1, No. 1, 2001, pp. 59–68.
2. J. Lee, J.F. Nunamaker Jr, and C. Albrecht, "Experiences with Collaborative Applications that Support Distributed Modeling," Proceedings of the 34th Hawaii International Conference on System Sciences, Maui, Hawaii, 2001.

3. W.A. Filho, C.M. Hirata, and E.T. Yano, "GroupSim: A Collaborative Environment for Discrete Event Simulation Software Development for the World Wide Web," Simulation, Vol. 80, No. 6, 2004, pp. 257–272.
4. R. Maghnouji, G.J. de Vreede, A. Verbraeck, and H.G. Sol, "Collaborative Simulation Modeling: Experiences and Lessons Learned," In Proceedings of the 34th Hawaii International Conference on System Sciences, Maui, Hawaii, 2001.
5. M. Bubak, G.D. van Albada, P.M.A. Sloot, and J.J. Dongarra (Eds.), Proceedings of the Fourth International Conference on Computational Science, Kraków, Poland, Lecture Notes in Computer Science, Vols. 3036–3039, 2004.

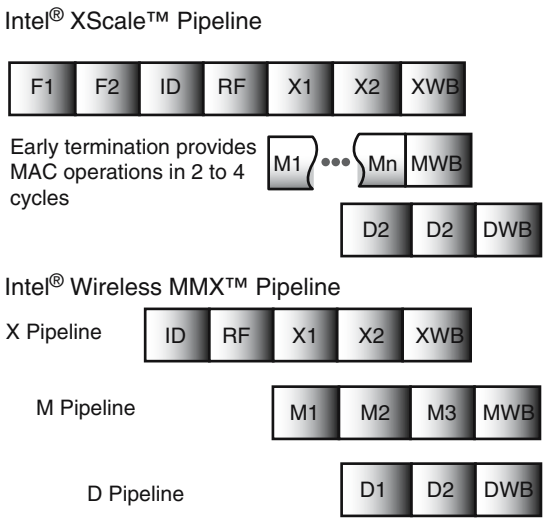
Intel® XScale® Micro-Architecture

Definition

The Intel XScale micro-architecture is an implementation of the ARM V5TE architecture.

The XScale core supports both dynamic frequency and voltage scaling with a maximum frequency today in handheld devices of 624MHz (and increasing going forward). The design is a scalar, in-order single issue architecture with concurrent execution in three pipes that support out-of-order return. To support the frequency targets, a 7-stage integer pipeline is employed with dynamic branch prediction supplied to mitigate the cost of a deeper pipeline (see Fig. 1).

In favor of memory access efficiency, the Intel XScale micro-architecture contains instruction and data caches (32KB each). Also, in order to hide memory latency the micro-architecture supports software issued prefetch capability coupled with advanced load and store buffering. Load buffering allows multiple data/cache lines request from the memory concurrently, thus reducing the data loading overhead. Similarly, store buffers combine multiple neighboring store operations to improve the memory and bus utilization. For virtual memory address translation, the Microarchitecture provides instruction and data translation look-aside buffer with 32-entry for each. Dynamic branch prediction with a target buffer of 32-entries significantly reduces the branch-penalty for deeper pipelines.



Intel® XScale® Micro-Architecture. Figure 1. Pipelines applied in Intel XScale processor.

Intel Xscale Microarchitecture supports standard ARM* coprocessor framework. Intel Wireless MMX™ technology is incorporated as a coprocessor on the Intel XScale® micro-architecture. The ARM architecture specifies that the main core is responsible for fetching instructions and data from memory and delivering them to the coprocessor. An instruction can be issued to the main core pipeline or coprocessor pipeline. For example, an instruction can be issued to the load pipeline while a MAC operation completes in the multiply pipeline.

The architecture allows instructions to be retired out of order. The load buffering combined with the out-of-order completion, allows non-dependent load instructions to execute, reducing the impact of memory latency in system on a chip applications.

Figure 2 shows the XScale core supports a debug interface, JTAG and PMU (Performance Monitoring Unit) in addition to a high-speed interface to the Wireless MMX unit.

Cross-References

▶ Multimedia System-on-a-Chip
▶ SIMD Data Processing

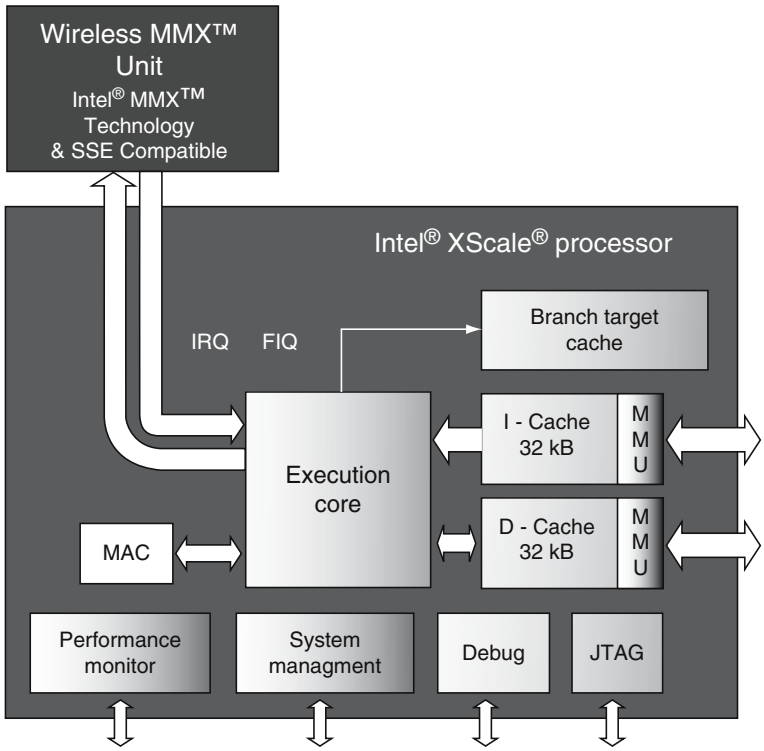
References

1. D. Seal, "Advanced RISC Machines Architecture Reference Manual," Prentice Hall, Englewood, Cliffs, NJ, 1996.

Interactive Story Telling

Definition

Interactive story telling shifts the narrative form from linear to dialectical, creating new stories on the basis of discussions and using dialogue as a method of interactions.



Intel® XScale® Micro-Architecture. **Figure 2.** The architecture of Intel XScale processor.

Telling stories is one of the most ancient forms of human communication. Stories allow people to share life experiences through real or imaginary tales. Stories in a book have usually a fixed structure and a specific order. Rather, told stories are more flexible than read ones, while the storyteller can change order and content in response to the audience feedback. The need for playing games pretending to be immersed in fantastic tales has existed for thousand years, with both children and adults subject to this human rite. *Interactive Story Telling* (IST) represents a digital version of this habit [1]. IST shifts the narrative form from linear to dialectical, creating new stories on the basis of discussions and using dialogue as a method of interactions. Integrating new technologies with traditional storytelling is pushing the development of this kind of narrative towards a direction where people may have a role on how the story progresses. Information technologies, ranging from ubiquitous networking to multimodal interfaces, provide support to augment the interactivity degree, thus enhancing user perception. IST has determined a significant change in forms of *collective intelligence*. This means that individuals can share their abilities and generate a more creative final product. Two kinds of IST applications exist, specifically: (1) those ISTs that resemble traditional entertainment contents (e.g. movies, fiction, soap, music, etc.) yet give the possibility to users to modify fragments of the multimedia presentation, and (2) applications that involve users in playing a role in a story, composing the final plot through the interaction among them.

In the first category of IST applications, fragments of multimedia contents can be mixed together to create one or more sequences to be used as visual or auditory stories. Fragments are needed to be classified to support users during the composition of sequences composed of multimedia scraps [2]. Further, this form of IST can support cooperative composition and orchestration through collaborative interfaces. As an evolution of this trend, tools exist that permit users to write a story at a virtually unlimited scale and distance. An example of this is Moblogs that creates anthologies of multimedia resources (text, photo, sounds or small videos) directly collected from people's everyday life (by using mobile terminals).

The second class of IST applications involves users in playing a role within a story, a game, and a virtual environment. Also Mixed Reality environments can be merged with IST (MR-IST) with the aim of immersing real actors in virtual settings and deriving the final plot of the story from the interactions among users.

As in games, authors have control on the story but its evolution may be influenced by external factors [3]. To this aim, many Mixed Reality based technologies are exploited such as mobile devices, multimodal interfaces, 3D input/output systems and localization tools. Involving people in the story is frequently supported by telling well known stories, such as successful fictions or soaps by using VR artifices to improve sensorial experience.

Cross-References

► Multimedia Entertainment Applications

References

1. J. Schell, "Understanding Entertainment: Story and Gameplay are One," The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications Archive, Lawrence Erlbaum, NJ, 2002, pp. 835–843.
2. J. Lambert, "Digital Storytelling," Digital Diner Press, Berkeley, 2002.
3. L. Romero, J. Santiago, and N. Correia, "Contextual Information Access and Storytelling in Mixed Reality Using Hypermedia," Computers in Entertainment (CIE), Vol. 2, No. 3, July 2004.

Interactive Virtual Humans in Mobile Augmented Reality

Nadia Magnenat-Thalmann, George Papagiannakis, Parag Chaudhuri
University of Geneva, Switzerland

Synonyms

► Virtual humans; ► Mixed reality; ► Industrial virtual trainer

Definition

Virtual humans are used as interfaces as well as real-time augmentations (three-dimensional computer-generated superimpositions) in real environments, as experienced by users though specialized equipment for enhanced mobility (e.g. ultra mobile PCs and video see-through glasses).

Recent advances in hardware and software for mobile computing have enabled a new breed of mobile Augmented Reality systems and applications featuring interactive virtual characters. This has resulted from the convergence of the tremendous progress in mobile computer graphics and mobile AR interfaces. In this paper, we focus on the evolution of our algorithms and their integration towards improving the presence and interactivity of virtual characters in real and virtual environments, as we realize the transition from mobile workstations to ultra-mobile PC's. We examine in detail three crucial parts of such systems: user-tracked interaction; real-time, automatic, adaptable animation of virtual characters and deformable pre-computed radiance transfer illumination for virtual characters. We examine our efforts to enhance the sense of presence for the user, while maintaining the quality of animation and interactivity as we scale and deploy our AR framework in a variety of platforms. We examine different AR virtual human enhanced scenarios under the different mobile devices that illustrate the interplay and applications of our methods.

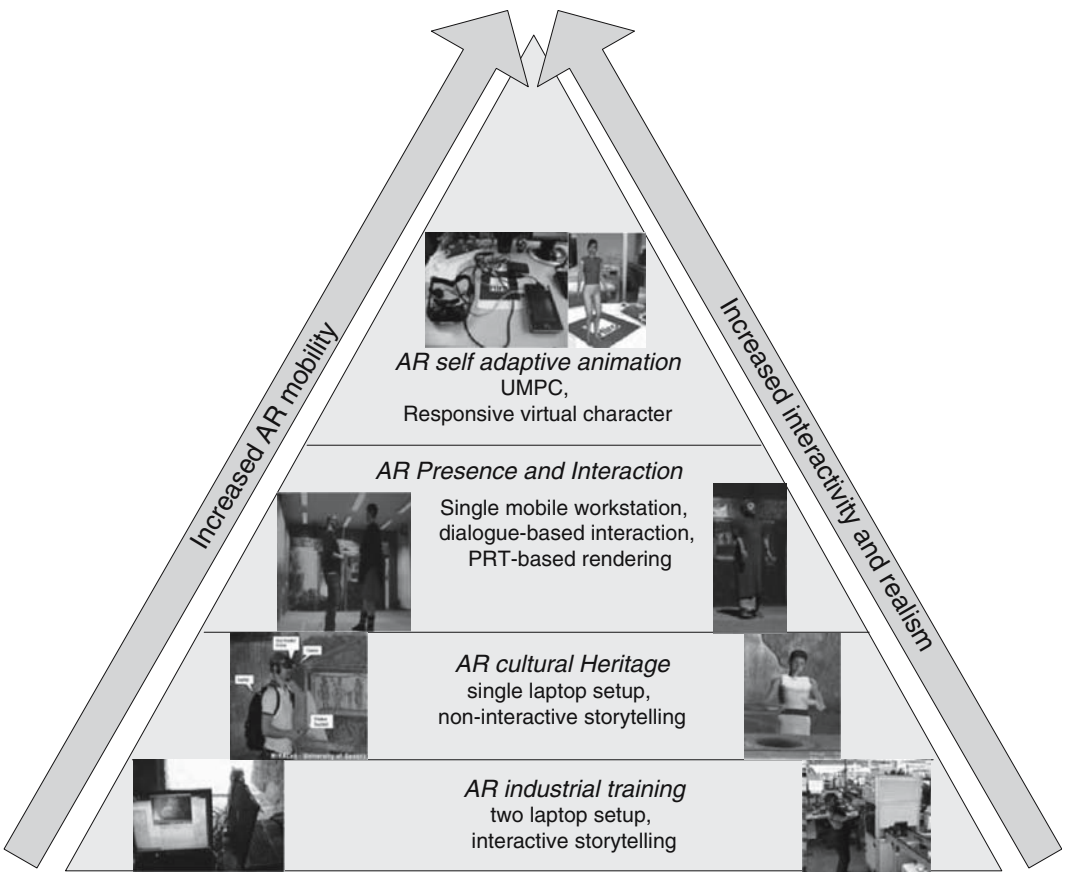
Introduction

Mixed Reality [1] has been depicted as a continuum that includes both Virtual Reality (VR) as well as Augmented Reality (AR). Traditionally the rich content needed for the complex, immersive simulations of VR dictated a desktop hardware setup, whereas the 2D or static 3D superimpositions on a real scene allowed for mobile, wearable (albeit cumbersome) systems. As the expectations and applications of AR have increased with the recent performance boost of mobile graphics on mobile workstations, modern mobile AR simulations have reached unparalleled levels of complexity, featuring advanced 3D simulations with animations, deformations and more realistic real-time rendering. However, these effects were achieved at the expense of the mobility of AR systems, as they were based on combinations of mobile workstations. Recently a new class of mobile devices has arrived, the Ultra Mobile PC (UMPC) that includes similar hardware and software capabilities of mobile workstations. Moreover, PDAs have also been merged with mobile phones, allowing for new opportunities based on their advanced hardware as well as more programming friendly software environments, operating systems and APIs. In this work we summarize our research efforts for the last 5 years, where advanced real-time 3D augmentations of fully simulated virtual characters (body, face and cloth simulation) were brought into mobile AR. In order to achieve such simulations in real-time, we initially commenced with a set-up of two

mobile workstations, then migrated to a single workstation. Very recently our efforts have resulted in adapting our interactive virtual character simulations to a UMPC. This progress toward a more mobile AR system is depicted in Fig. 1. It is interesting to note, however, that it is based on the same component-based 3D simulation framework [2].

Mobile Workstations and UMPCs for Mobile AR

A number of systems [3] have employed mobile workstations (high-end laptops), often aggregated together with other mobile equipment, often carried in a backpack (weighting 1–6 kg), so that the user can freely move in the real environment and have their hands free for input and interaction. These backpacks include amongst others, mobile workstations such as Dell™ Inspiron and Precision, Alienware™ and JVC™ subnotebooks. Although severe ergonomic issues are apparent due to the size and weight of the backpack, it allows researchers to focus on their research without the constraints that smaller devices often present, namely computational power, operating system and hardware connectivity. Almost all of the desktop computing system can be made mobile by using high-end notebook computers. However, due to the backpack setup, the use of head mounted displays (HMDs) is enforced as opposed to handheld display that other devices can offer. The next step towards this direction is the employment of ultra mobile PCs (UMPCs), that could provide both handheld as well as HMD viewing capabilities. The usage of ultra mobile PCs is a very recent trend in mobile AR systems is. UMPCs are based on the Microsoft Origami™ specification released in 2006 and have been developed jointly by Microsoft™, Intel™, and Samsung™, among others. UMPCs are basically small-factor mobile PCs capable of running Microsoft Windows XP™ or Vista™. A number of researchers have started employing them in AR simulations [4].



Interactive Virtual Humans in Mobile Augmented Reality. **Figure 1.** Evolution of interactive virtual humans and their presence, interaction and animation on mobile AR systems.

The VHD++ AR/VR Mobile Framework

AR systems rely heavily on the interplay of heterogeneous technologies. Because of that interdisciplinary character, the AR domain can be viewed as a melting pot of various technologies, which although complementary, are non-trivial to put together. The VHD++ framework [2] is a software development framework that supports production of high performance, real-time, interactive, audio-visual applications. Traditionally it had a core composed of 3D graphics, 3D sound and advanced synthetic character simulation but recently many other technologies like networking, database access, artificial intelligence, content creation and diagnostics have been added. Instead of conventional development the applications are being created by reuse and customization of the existing, fully operational design and code. For more details the reader is invite to look at [2,5]. In the following sections we present four works that show the evolution of our AR technologies as we move toward a more mature mobile AR platform. We will see how these helped us improve our hardware and software development platform significantly as we consistently moved toward more believable and rich AR experiences on mobile platforms.

Instructive Virtual Industrial Trainer Based on Dual Mobile Workstations

In this work, a novice user is trained to use complex machines by a virtual teacher showing them how to correctly operate machinery. Including real machinery and surroundings into the interactive simulation increases realism and decreases time that is required to build complex virtual environments and the computation cost involved in rendering them. Figure 2 illustrates this approach: a virtual worker demonstrates how to use a machine in a complex industrial environment.

The user of the AR system can change the point of view at any time, and since the camera position is correctly registered to the real scene, the virtual worker is always correctly blended into the streaming video. We therefore make a twofold contribution:

- An accurate real-time vision-based camera tracker, which is responsible for the registration of the virtual humans into streaming video and does not require any engineering of the environment.
- It's integration into an existing VR system, called VHD++ that provides the interface with the user and the rendering of realistic virtual humans.

Since VHD++ is a modular component-based framework, the integration of the tracking part was done as a plug-in and was straightforward. The VR part that has been used in this application integrates technologies, such as real-time 3D rendering, skeleton and skin animation and behavioral control together. VHD++ virtual humans show large range of animation capabilities. In our case we used keyframe animation of the virtual human body. For some general movement, such as walking, pointing and grabbing, an inverse kinematics module can also be used. The framework aims to act as a real-time, extensible audiovisual framework with dedicated support to VR/AR real-time virtual characters.

Mobile AR Cultural Heritage Guide on a Single Laptop

In this application, we migrate from a configuration of two connected laptops to a single laptop configuration. This work is centered on the innovative revival of life in ancient frescos-paintings in ancient Pompeii and creation of narrative spaces [6]. The revival is based on real scenes captured on live video augmented with real-time autonomous groups of 3D virtual fauna and flora (Fig. 3).

Interactive Virtual Humans in Mobile Augmented Reality. **Figure 2.** A virtual human demonstrating the use of a real machine.

The metaphor, which inspires the project approach, is oriented to make the "transportation in fictional and historical spaces," as depicted by frescos/paintings, as realistic, immersive and interactive as possible. Thus the ancient characters of the frescos/paintings (including humans and plants) will be revived and simulated in real-time 3D, exhibiting in a new innovative manner their unique aesthetic, dramaturgical and emotional elements. The whole experience is presented to the user on-site in Pompeii during their visit, through an immersive, mobile Augmented Reality-based Guide. This AR platform is also based on the VHD++ component-based framework. The various technologies used in this work include a plug-and-play combination of different heterogeneous technologies such as: Real-time character rendering in AR, real-time markerless camera tracking, facial simulation and speech, body animation with skinning, 3D sound, real-time cloth simulation and behavioral scripting of actions. To meet the hardware requirements of this aim, a single Dell Pentium 4 M50 Mobile Workstation was used, with a Quadro 4 500 GL Nvidia graphics card, a firewire Unibrain camera or USB Logitech web camera for fast image acquisition send on a video-see-through TekGear monoscopic HMD, for an advanced and immersive simulation. We started, as before, on a client-server distributed model, based on two mobile workstations. However, to achieve the requirement of "true mobility," we migrated to a single mobile workstation. This is now used in our current demonstrations, after improvements in the streaming image capturing and introduction of hyper-threading in the platform code. We based our system on a real-time markerless camera tracking method from 2d3™ where the integrated camera tracker is able to self-initialize anywhere within the tracking environment without any intervention from the user as well as recover immediately in case of degenerate tracking (i.e., looking out of the designated area). In effect this means that instead of calculating relative changes in rotation and translation, we calculate absolute rotation and translation for every frame. This has the advantage of avoiding the problem of drift, and also ensures instant recovery after tracking was lost due to excessive motion blur or occlusion. The basic algorithm is based on "structure from motion" techniques and described more in detail in [5].

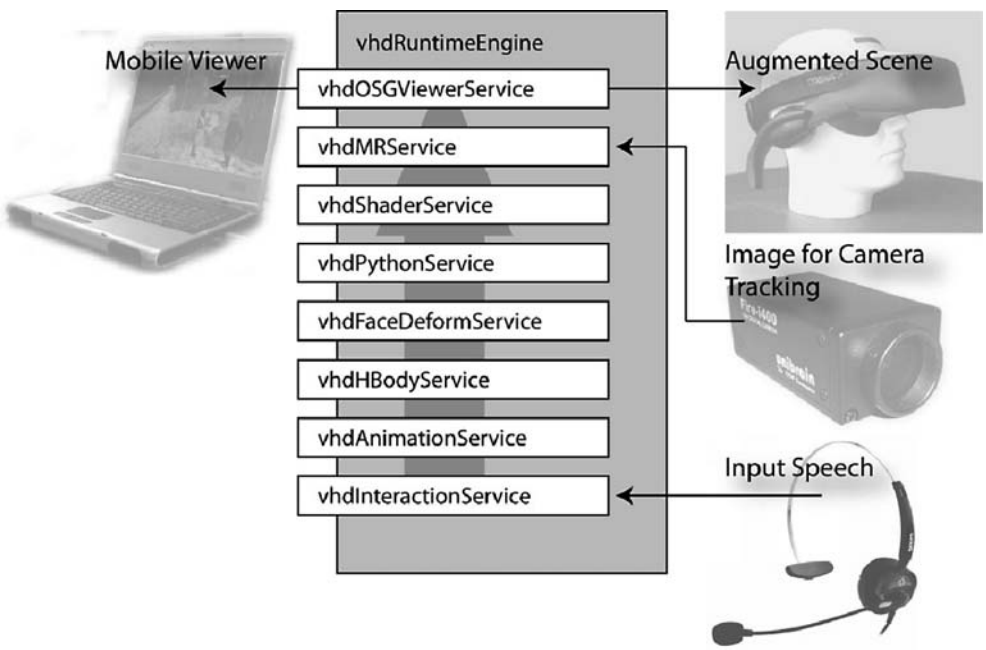
Interactive, Dialogue Based and Advanced Rendered Virtual Characters on a Single Mobile Workstation

In this work, the previous approaches are extended to allow for interaction, animation and global illumination of virtual humans for an integrated and enhanced presence in AR. The interaction system comprises of a dialogue module that is interfaced with a speech recognition and synthesis system. In addition to speech output, the dialogue system also generates face and body motions, which are in turn passed on to the virtual human animation layer. All these different motions are generated and blended online, resulting in a flexible and realistic animation. Our robust rendering method operates in accordance with this animation layer and is based on an extension for dynamic virtual humans. The extended Precomputed Radiance Transfer (PRT) illumination model used results in a realistic display of such interactive virtual characters in complex mixed reality environments. The presented

Interactive Virtual Humans in Mobile Augmented Reality. **Figure 3.** A single laptop employed in a backpack for autonomous mobile AR.

scenario illustrates the interplay and application of our methods, glued under the VHD++ framework for presence and interaction in mixed reality. It features a real human that engages in conversation with a virtual one in AR and witnesses the virtual human, correctly registered and aligned in natural size (human scale) in the real environment, based on the markerless camera tracker from [5] (see Fig. 4).

The software architecture is extended here for maintenance of the consistent simulation and interactive scenario state that can be modified with python scripts at run-time. To keep good performance, the system utilizes five threads. One thread is used to manage the updates of all the services that we need to compute, such as human animation, cloth simulation or voice (sound) management. A second thread is used for the 3D renderer that obtains information from the current scenegraph about the objects that must be drawn, in addition to the image received from the camera. It changes the modelview matrix accordingly to the value provided by the tracker. The third thread has the responsibility of capturing and tracking images. The fourth thread manages the update process of the interaction system, by parsing the script to see whether or not an action needs to be taken by the virtual human. The last thread is the python interpreter, which allows us to create scripts for manipulating our application at the system level, such as generating scenario-based scripted behaviors for the human actions (key-frame animation, voice, navigation combined to form virtual short plays). The AR system presented in Fig. 4 features immersive, real-time, interactive simulation supplied with proper information during the course of the simulation. This, however, requires the components to be very diversified and thus their development is an extremely laborious process involving long and complex data processing pipelines, multiple recording technologies, various design tools and custom made software. The various 3D models to be included in the virtual environments like virtual humans or auxiliary objects have to be created manually by 3D digital artists. The creation of virtual humans require to record motion captured data for realistic skeletal animations as well as a database of gestures for controlling face and body animations. Sound environments, including voice acting, need to be recorded in advance based on storyboards. For each particular scenario, dedicated system configuration data specifying system operational parameters, parameters of the physical environment and parameters of



Interactive Virtual Humans in Mobile Augmented Reality. **Figure 4.** A single mobile workstation for advanced presence, PRT-based rendering and dialogue based interaction in AR.

the VR devices used have to be defined as well as scripts defining behaviors of simulation elements, in particular virtual humans, have to be created. These scripts can modify any data in use by the current simulation in real-time.

AR Self Adaptive Animations on a UMPC

We explained in the previous sections that creating more and more believable and interactive characters becomes an increasingly laborious process. To alleviate this problem, we present in this section, our recent work on a simple and fast method to author self adaptive character animations that respond automatically to changes in the user's perspective or point of view in real-time, suitable for simulation in a UMPC. The animator creates a set of example key animations for the characters assuming the user is viewing the animation from different key viewpoints in the world. When the user actually interacts with the character, the user's actual point of view is tracked in real-time by using computer vision techniques or by simple user controlled input methods. The tracked position of the user's viewpoint, with respect to the key viewpoints, is then used to blend the example key animations, in real-time (see Fig. 5). Thus, the animation of the character adapts itself in response to the changes in the user's viewpoint [7]. We demonstrate that our method is simpler and more efficient than other techniques that can be used to obtain similar results. We also show a working, prototype implementation of our method with a simple example in mobile Augmented Reality on a UMPC. The basic aim of this work, in comparison to the previous approaches is to (see Fig. 6):

- Allow for the same AR framework to be utilized effectively on a mobile PC as well as on a UMPC
- Allow for virtual characters to react to users presence during an interactive session, a shortcoming of previous methods

Due to the limited graphics acceleration of the UMPC, allow for 3D content adaptation of the same 3D augmentation. For e.g. The UMPC used did not support OpenGL 2.0 or the OpenGL Texture Rectangle extensions. Thus, both virtual character animation and skinning had to be calculated without accelerated vertex buffer objects and the video see-through HMD camera grabbing had to utilize simple Texture2D objects.

Implementation

A character pose is a hierarchical tree of rigid transformations. If we linearize this tree by doing a fixed traversal on it, we get a list of transformations. Rigid transformations can be represented as unit dual quaternions [7].

We define a character pose as a list of dual quaternions. An animation is only a time varying sequence of poses or by extension, a time varying list of dual quaternions. An animator creates a set of key animations that represent the way the character should react when the user approaches or looks at the character from different directions. We refer to these directions as key viewpoints or key cameras. Now the user's point of view is tracked in real-time by using known camera tracking algorithms. In this example, we have used ARToolkit for this purpose. We recover the current transform for a tracked marker and place our character on that position. The pose of the marker is used to infer the position of the current camera looking at the scene. We can also use our previous markerless camera tracking method for this. The current tracked camera is used to compute a weighted blend of the key

Interactive Virtual Humans in Mobile Augmented Reality. **Figure 5.** Self-adaptive animation based on AR user perspective.

Interactive Virtual Humans in Mobile Augmented Reality. Figure 6. A wearable UMPC with the i-glasses HMD, battery pack and usb webcam.

animations to get the current animations that the user can see. The weights are computed on the basis of the position of the current camera in the space of key cameras defined earlier.

Recently, UMPCs provide unique opportunities for mobile applications in terms of code portability as well as performance. Due to performance capabilities (CPU and GPU) several allowances have to be made and content has to be adapted to better fit the mobile experience that the UMPC offers. As we have recently witnessed the merging of PDAs with phones, it could be possible in the near future to witness a further merge between UMPCs with mobile phones.

Conclusions and Acknowledgments

Here we have presented an overview of our research work that has been carried out over the last years on presence, interaction and animation issues of 3D virtual characters in mobile AR systems. This effort is a material witness to the evolution and progress of state of the art in such systems, bringing richer and more believable content within the grasp of modern day mobile AR systems, networked media and computer graphics based 3D simulations. The currently presented work has been supported by the INTERMEDIA 38419 EU Project in frame of the EU IST FP6 Programme.

Cross-References

► Virtual and Augmented Reality

References

1. R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent Advances in Augmented Reality," IEEE Computer Graphics and Applications, Vol. 21, No. 6, 2001, pp. 34–47.
2. M. Ponder, G. Papagiannakis, T. Molet, N. Magnenat-Thalmann, and D. Thalmann, "VHD++ Framework: Extendible Game Engine with Reusable Components, for VR/AR R&D Featuring Advanced Virtual Character Simulation Technologies," Proceedings of Computer Graphics International 2003, pp. 96–104, IEEE Computer Society, 2003.
3. G. Papagiannakis, G. Singh, and N. Magnenat-Thalmann, "A Survey of Mobile and Wireless Technologies for Augmented Reality Systems," Journal of Computer Animation and Virtual Worlds, Wiley, 2008, (to appear).
4. I. Barakonyi and D. Schmalstieg, "Ubiquitous Animated Agents for Augmented Reality," Proceedings of ISMAR 2006 (IEEE and ACM International Symposium on Mixed and Augmented Reality), 2006, pp. 145–154.
5. G. Papagiannakis, and N. Magnenat-Thalmann, "Mobile Augmented Heritage: Enabling Human Life in Ancient Pompeii," International Journal of Architectural Computing, Multi-Science, Vol. 5, No. 2, 2007, pp. 395–415.
6. G. Papagiannakis, S. Schertenleib, B. O'Kennedy, M. Arevalo-Poizat, N. Magnenat-Thalmann, A. Stoddart, and D. Thalmann, "Mixing Virtual and Real Scenes in the Site of Ancient Pompeii," Computer Animation and Virtual Worlds, Vol. 16, No. 1, pp. 1–24, Wiley, 2005.
7. J.M. Mccarthy, "Introduction to Theoretical Kinematics," MIT, Cambridge, MA, USA, ISBN:0-262-13252-4, 1990.
8. L. Vacchetti, V. Lepetit, M. Ponder, G. Papagiannakis, P. Fua, D. Thalmann, and N. Magnenat-Thalmann, "Stable Real-time AR Framework for Training and Planning in Industrial Environments," Virtual Reality and Augmented Reality Applications in Manufacturing, Ong, Soh K., Nee, ISBN:1-85233-796-6, Springer, Berlin, London, 2004.
9. A. Egges, G. Papagiannakis, and N. Magnenat-Thalmann, "Presence and Interaction in Mixed Realities," Springer, Berlin, London, Visual Computer, Vol. 23, No. 5, 2007, pp. 317–333.
10. P. Chaudhuri, P. Kalra, and S. Banerjee, "View-Dependent Character Animation," Springer, Berlin, London, ISBN:978-1-84628-591-2, 2007.

Interactivity in Multimedia Documents and Systems

Synonyms

► User interaction; ► User input

Definition

Multimedia interactivity describes the set of possible actions a user can do to change the state of a multimedia system, e.g., the course of a multimedia document's playback.

Introduction

The rapid growth of network capabilities in the last decade has fostered the diffusion of a wide set of interactive multimedia documents and systems, ranging from multimedia portals and distributed e-learning applications to interactive games and systems for multimedia data retrieval. These applications usually allow users to actively interact with the system rather than being passive recipients of information.

The problem of describing, supporting and helping user interaction has been largely investigated in literature. Multimedia systems usually allow two types of navigation facilities to control a presentation [1]: to adjust the current time reference in a presentation playback or to follow a hyperlink. In the first case, the user interaction is similar to the VCR functionalities for videocassettes: the user can start, stop, pause, fast-forward or rewind a presentation playback. In the second case, the user may jump to a completely different section of the document.

This second situation is more difficult to achieve and support, since the interaction may affect and change the overall structure of the presentation. Moreover, multimedia applications contain both static objects, like text pages and images, and continuous media items, like video and audio files. A hyperlink can be easily inserted into a static object, as an example, the user easily recognizes the presence of an anchor in an underlined word in a text page, whereas in multimedia system, the association of links with e within components is considerably complex for continuous media, for which a common way of defining anchor is not already available. Consider, for example, the problem of inserting a hyperlink inside an audio file. A proposed solution is the use of *hotspots*, i.e., an icon which remains on the user screen for a determined time interval, on which the user may click to follow the link. Despite many efforts, this problem has not already found a suitable solution, and the discovery of hyperlinks inside continuous media is often still hard for the user.

From the synchronization point of view, a multimedia application may have different behavior when a user follows a link, since multimedia systems do not clearly define a notion of how much information the reader leaves: the link can bring the user to a new document, which completely replaces the source, or both destination and source (or a part of) may coexist in the final presentation.

The Amsterdam Hypermedia Model [1] solves this problem by defining the notion of *context of a hyperlink* which clearly states which components survive after the user interaction, and which others are replaced and added.

The standard SMIL, *Synchronized Multimedia Integration Language* [2,3], whose third version is currently under definition, is a markup language which allows authors to design highly interactive multimedia presentations. In fact, multimedia documents designed with SMIL can contain hyperlinks to other documents, or to some of their components, but they can also modify their behavior according to user interactions. A SMIL document contains both the description of the spatial layout of media items and their temporal synchronization.

The user can start the playback of a SMIL presentation and passively follows its natural evolution, or she can play a more active role. The interactions allowed are not limited to the choice of a link, but the user can freely move the mouse around the user interface and click on the media items displayed. The SMIL standard in fact allows to synchronize the behavior of an item to external events: as an example, the playback of a media item can be started, or stopped, when the user clicks on a particular image, or moves the mouse over, or out, of a video file.

Moreover, unlike other encodings like MPEG-4, SMIL can be considered an integration format, since it does not store the entire presentation in a composite "sealed container," but each stream is a single file which can be distributed across the network and easily reused: the SMIL specification simply describes how the components are displayed on the user screen and synchronized. In MPEG-4 all the files are defined and controlled by the content creator, while SMIL allows the specification of a set of alternatives which can be chosen by the user according to her preferences and settings: as an example, an user can personalize a multimedia document playback by selecting the language or the presence of subtitles like in a DVD environment. In this sense, according to Bulterman [4], the standard SMIL enhances the possibilities of interactions offered to the user who can now partially control and select the multimedia contents. Moreover, since the insertion of new media items in a SMIL

presentation is very easy, it helps the implementation of tools which allow the user to annotate multimedia items.

We must note here that the new possibilities offered by interactivity in multimedia systems bring also some drawbacks. The presence of too much information in the user screen may lead to their cognitive overload and disorientation [5]: multimedia systems offer users freedom to navigate into a very large information space by selecting an own path, but the pluralism of available choices may overawe the user that is no longer able to manage it without proper navigation aids. Therefore with freedom comes complexity, and so, disorientation.

For this reason, interactivity enhances user experience only when implemented properly, and must be carefully designed for each multimedia application: the user should always be able to see what are the consequences of the available choices and receive a suitable feedback on the taken action. Moreover the author must provide guided tours or navigational aids that guide the user through the fulfillment of her goal.

Finally, since interactivity in multimedia systems deals with creating experiences which allow the user to do or make something, a new form of interaction between users and multimedia systems is the content adaptation process. By content adaptation process we mean the set of actions performed to adapt a multimedia presentation to the user context, i.e., the device, the screen resolution, the network connection, the user preferences as well as the situation in which she is immersed: as an example, audio files cannot be played in a silent ambient like a library.

This is a particular form of interaction, since this process is usually transparent to the user, who simply receives, in response to her request, a suitable document to be rendered in the surrounding environment, and does not take note of the selection process that has built the final result from a set of rather content equivalent alternatives. Nevertheless, it is not less important since a multimedia application which is not able to adapt its content to the user context (or at least, to her device) cannot be played in any given situation.

Cross-References

- ▶ Multimedia Content Adaptation
- ▶ Multimedia Synchronization – Area Overview
- ▶ Multimodal Interfaces

References

1. L. Hardman, D. Bulterman, and G. van Rossum, "The Amsterdam hypermedia model: adding time and context to the Dexter model," Communications of ACM, Vol. 37, No. 2, February 1994, pp. 50–62.
2. Bulterman et al., "Synchronized Multimedia Integration Language (SMIL) 3.0 Candidate Recommendation," http://www.w3.org/TR/SMIL3/, 2008.
3. D.F. Zucker and D. Bulterman, "Open standard and open sourced SMIL for interactivity," ACM Interactions, Vol. 14, No. 6, November 2007, pp. 41–46.
4. D. Bulterman, "User-Centered Control within Multimedia Presentations," Multimedia Systems Journal, Vol. 12, No. 4–5, March 2007, pp. 423–438.
5. D. Kirsh, "Interactivity and MultiMedia Interfaces," Instructional Sciences, Springer, Netherlands, Vol. 25, No. 2, 1997, pp. 79–96.

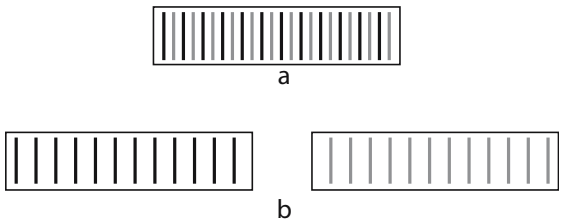
Interleaved Audio

Definition

Interleaved audio transmission is a packet loss resilience technique based on sending alternate audio samples in different packets.

▶ Interleaved audio transmission is a technique that is sometimes used to alleviate network loss and act as a packet loss resilience mechanism [1,2]. The idea is to send alternate audio samples in different packets, as opposed to sending consecutive samples in the same packet. The difference between the two approaches is shown in Fig. 1.

In Fig. 1(a) we see 24 consecutive samples being transmitted in one packet. If this packet is lost, there will be a gap in the audio equal to the duration of the samples. In Fig. 1(b) we see the interleaved approach for the same audio sample in 1a, where alternate



Interleaved Audio. Figure 1. (a) Transmission of consecutive samples, (b) Transmission of alternate samples.

samples are being sent in separate packets. This way if one of the packets is lost, we only lose every other sample and the receiver will hear a somewhat distorted audio as opposed to hearing a complete gap for that duration. In case of stereo audio, we can adapt this technique to send the left channel and the right channel in separate packets, so that if the packet for one of the channels is lost, the receiver still hears the other channel.

Cross-References

► Audio Conferencing

References

1. D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," IETF RFC 2250, January 1998.
2. R. Finlayson, "A More Loss-Tolerant RTP Payload Format for MP3 Audio," IETF RFC 3119, June 2001.

Interoperable Description Formats Facilitating the Adaptation of Multimedia Content

► MPEG-21 Digital Item Adaptation

IP Telephony

Abdulmotaleb El Saddik
University of Ottawa, Ottawa, ON, Canada

Synonyms

► Internet telephony

Definition

Internet telephony is the process of making telephone calls over the Internet.

Introduction

Internet telephony, also referred to as IP telephony (IPT), is the process of making telephone calls over the Internet, regardless of whether traditional telephones (POTS, GSM, ISDN, etc.), single use appliances, or audio-equipped personal computers are used in the calls. IPT is highly appealing for many reasons, the most important of which is the ease of implementing its services. Internet Telephony Service Providers (ITSP) can use a single IP-based infrastructure for providing traditional Internet, as well as Internet telephony access.

The Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is a signaling protocol for Internet Telephony. It is documented in (RFC3261, 2002) by the Internet Engineering Task Force (IETF), and is ideal for real-time multimedia communication signaling [1]. It is an end-to-end application layer signaling protocol that is used to setup, modify, and teardown multimedia sessions such as audio/videoconferencing, interactive gaming, virtual reality, and call forwarding over IP networks. By providing those services, SIP enables service providers to integrate basic IP telephony services with Web, e-mail, presence notification and instant messaging over the Internet. It is clear that SIP is rapidly changing the way that people make telephone calls and is therefore becoming a real threat to traditional plain old telephone service (PSTN) network [2]. SIP works with many other protocols that were designed to carry the various forms of real time multimedia applications data by enabling endpoints to discover one another and to agree on a characterization of a session that they would like to share. Even though SIP was designed to work with other internet transport protocols such as UDP, TCP when it was developed by the IETF as part of the Internet Multimedia Conferencing Architecture, it is very much a general purpose signaling protocol that works independently of underlying protocol, and regardless of the type of session that is being established [1]. SIP is a text based client server protocol that incorporates elements of two widely used Internet protocols: HTTP and the Simple Mail Transport Protocol (SMTP), used for web browsing and e-mail respectively [1]. HTTP inspired a client server design in SIP, as well as the use of URL's and URI's [1], however, in SIP a host may well act as client and server. From SMTP, SIP borrowed a text -encoding scheme and header style. For example, SIP reuses SMTP headers like To, From, Date and Subject [1].

SIP extensions supports mobility and detects presence to allow users to communicate using different devices, modes, and services, anywhere that they are connected to the Internet. Third-Generation

Partnership Project (3GPP) group accepted SIP as the signaling protocol for Multimedia Applications in 3G Mobile Networks.

SIP's Protocol Architecture

As can be seen in Fig. 1, SIP does not rely on any particular transport protocol; it can run indifferently over TCP (Transport Control Protocol), UDP (User Datagram Protocol), TLS (Transport Layer Security), SCTP (Stream Control Transport Protocol), and conceptually any other protocol stack, like ATM (Asynchronous Transfer Mode) or Frame Relay. SIP does not dictate the data flow between peers, the Session Description Protocol (SDP) does that and negotiates and determines the format of data exchanged between them. SDP defined in RFC2327 is intended for describing multimedia sessions for the purposes of session announcement, invitation, and other forms of multimedia session initiation.

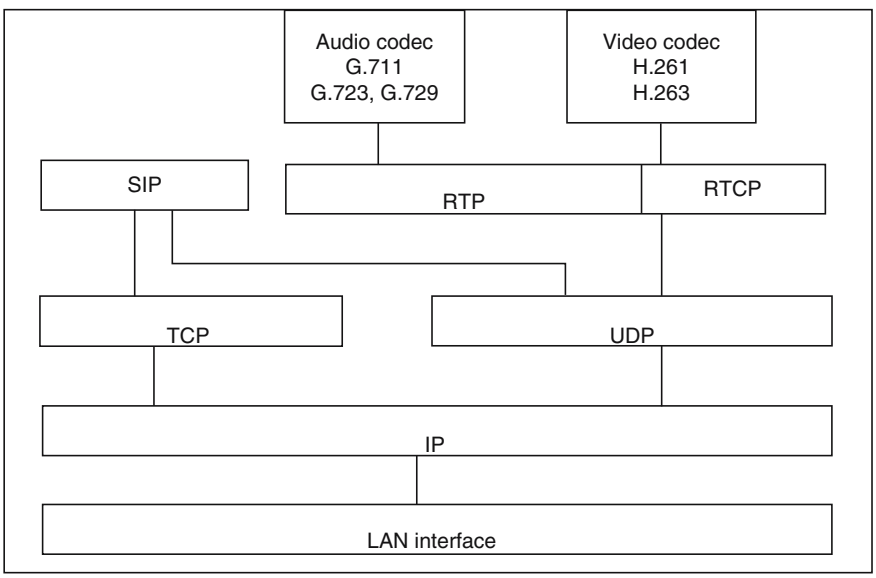
The benefits of the SIP protocol can be summarized by the following:

- Because it utilizes existing IP architecture, services based on SIP are scalable
- Since SIP was built as an IP protocol, it integrates seamlessly with other IP protocols and services
- Global connectivity can be achieved with SIP protocol. Any SIP user can be reached by another SIP user over the Internet, regardless of their location, service provider, and whether they have registered with central services or not
- Simplicity is hallmark of SIP, due to its text coded, highly readable messages, and its simple transactions models, except for few cases that have special conditions
- Statelessness: Depicted by the ability of SIP servers to store minimal information about the state or existence of a media session in a network [6]
- Flexibility: Protocols can be used in any host applications that are not restricted to telephony [6]

SIP Transport

User Datagram Protocol (UDP): A single UDP datagram or packet carries a SIP request, or response. This implies that a SIP message must be smaller in size than the Message Transport Unit (MTU) of the IP network. Since SIP doesn't support fragmentation at the SIP layer, TCP is used for larger messages. The UDP Source port is chosen from a pool of available port numbers (above 49,172) or sometimes the default SIP port, 5060 is used. Lack of a reliable transport mechanism in UDP may cause SIP message to be lost. To tackle this problem, SIP comes with its own reliability mechanism to handle the retransmission, in case a SIP message is lost.

Transport Control Protocol (TCP): Not only does it provide reliable transport, TCP also offers congestion control. It can also transport SIP messages of arbitrary



IP Telephony. Figure 1. IETF's SIP protocol architecture.

size. As in UDP, TCP uses SIP port number 5060 for the destination port. The source port is chosen from an available pool of port numbers. The main disadvantages of TCP are: the setup delay incurred when establishing a connection, and the need to maintain the connection at the transport layer by the server.

Transport Layer Security Protocol (TLS): SIP employs TLS over TCP for encrypted transport with additional capabilities of authentication. The default SIP port number for TLS is 5061. This use of TLS by SIP takes advantage of the encryption and authentication services. However, encryption and authentication are only useful on a single hop. If a SIP request involves multiple hops, TLS becomes useless for end-to-end authentication.

SIP Network Components

SIP network components include User Agents, Servers and Gateways. The following section discusses these components in detail.

User Agent: A SIP enabled end-device is called a SIP user agent. A User Agent takes directions from a user and acts as an agent on that user's behalf to make or accept and teardown calls with other user agents. The UA terminates the SIP call signaling and acts as the interface to the SIP network. It also maintains the state of calls that it initiates or participates in. UA must support UDP transport, and also TCP if it sends messages that are greater than 1,000 octets in size [6]. Also, a SIP user agent must support SDP for media description. A SIP user agent contains both a client application, and a server application. The two parts are designated as User Agent Client (UAC) and User Agent Server (UAS). The UAC initiates requests on behalf of the user and UAS processes incoming requests and generates appropriate responses. During a session, a user agent will usually operate as both a UAC and a UAS [4].

Server: The SIP server assists in call establishment, call teardown, and mobility management. Some SIP servers (proxy and redirect) can be stateless. Usually, logical SIP servers are often co-located within the same physical device. SIP servers must support TCP, TLS and UDP for transport. There is no protocol distinction between these servers, and also a client or proxy server has no way of knowing which it is communicating with. The distinction lies only in function: a proxy or redirect server cannot accept or reject a request, where as a user agent server can. Following are the different types of SIP servers:

- *Register Server*: The SIP registration server, also known as registrar allows SIP agents to register their current location, retrieve a list of current registrations, and clear all registrations. The registrar accepts a user SIP registration request (REGSITER) message and responds with an acknowledgement message (200 OK) for successful registration, otherwise, it responds with an error message. In a registration request message, the "To" header field contains the name of the resource being registered, and the "Contact" header field contain the alternative addresses or aliases. The registration server creates a temporary binding between the Address of Record (AOR) URI in the "To" and the device URI in the "Contact" fields. Registration servers usually require the registering user agent to be authenticated for security reasons. Thus, registered information will be made available to other SIP servers within the same administrative domain, such as proxies and redirect servers. The registrar is responsible for keeping information up-to-date within the location service by sending updates.
- *Proxy Server*: The proxy server forwards SIP requests on behalf of SIP User Agents to the next hop server, which may be another proxy server or the final user agent. A proxy does not need to understand a SIP request in order to forward it. After receiving the SIP request, the proxy will contact the location server to determine the next hop to forward the SIP requests to. The proxy may well rewrite the SIP message before forwarding it to its correct location. For incoming calls, it proxy will interrogate the location service to determine how to forward the call. The proxy may use SIP registration, SIP presence, or any other type of information to determine a user's location. The proxy can also be configured to provide authentication control and act as a point of control between the internal private network and an outside public network. A proxy server may be stateful or stateless. A stateless proxy server processes each SIP request or response based solely on the message contents. Once the message has been parsed, processed, and forwarded or responded to, no information about the messages is stored. A stateless proxy server

never retransmits a message and doesn't use any SIP timers. A stateful proxy server keeps track of requests and responses that were received in the past, and uses that information in processing future requests and responses. A stateful proxy starts a timer when a request is forwarded. If no response to the request is received within the timer period, the proxy will retransmit the requests, relieving the user agent of this task.

- *Redirect Server*: The redirect server responds to a UA request with redirection response, indicating the current location of the called party, so that UA can directly contact it. In this case, the UA must establish a new call to the indicated location. A redirect server does not forward a request received by the UA. Redirect server uses a database or location service to look up a user. The user location information is then sent back to caller in a redirection message response [4].
- *Location Server*: A redirect or proxy server uses a location server to obtain information about a user's whereabouts. The service can be co-located with other SIP servers. The interface between the location service and other servers is not defined by SIP [4].
- *Conference Server*: Conferencing server is used to aid the multiparty conference call establishment. A conferencing server mixes the media received and sends it out to all the participants using one multicast address or all of the participants' unicast addresses, depending on the mode of conference that was setup.

SIP Gateways: A SIP gateway is an application that provides an interface for a SIP network to another network utilizing another signaling protocol. SIP supports internetworking with PSTN and H.323 via SIP-PSTN gateway and SIP-H.323 gateway respectively. In terms of SIP protocol, a gateway is just a special type of user agent, where the user agent acts on behalf of another protocol rather than a human user. A SIP gateway terminates the SIP signaling path and may sometimes also terminate the media path. SIP Gateway can support hundreds or thousands of users and does not register every user it supports.

- *SIP-PSTN* gateway terminates both signaling and media paths. SIP can be translated into, or made to inter-work with common Public Switched Telephone Network (PSTN) protocols such as Integrated Service Digital Network (ISDN), ISDN User part (ISUP), and other Circuit Associated Signaling (CAS) protocols. A PSTN gateway also converts RTP media stream in the IP network into a standard telephony trunk or line. The conversion of signaling and media paths allows calling to and from the PSTN using SIP.
- *SIP-H.323 gateway*: SIP to H.323 terminates the SIP signaling path and converts the signaling to H.323, but the SIP user agent and H.323 terminal can exchange RTP media information directly with each other without going through the gateway.

SIP's Role in Multimedia Services

SIP is heavily involved in today's multimedia services, especially in the following categories:

User Presence Notification and Personalization: SIP Signaling functions request, detect, and deliver presence information and provide presence detection and notification. SIP presence functionality gives the opportunity to know who is online among a given contacts list before the session is established. SUBSCRIBE, NOTIFY messages are used to subscribe and notify users for presence detection and notification in an instant messaging application. A User agent sends a SUBSCRIBE message to another UA with a series of event requests indicating the desire of the sender to be notified by another UA. The NOTIFY message is used to indicate the occurrence of the event to the requested UA [4,6].

Instant Messaging and Collaborative environment: Instant messaging enables User agent to send short messages to another User Agent. It is very useful for short requests and responses. Instant messaging has better real-time characteristics than e-mail. MESSAGE method is used to support instant messaging. Its short messages are sent form UA to UA without establishing a session between them. The messages are sent in multi-part MIME format (similar to e-mail) and can also contain multimedia attachments [6].

Multimedia conference call setup and management: This can be divided into end-to-end call setup and conference setup:

SIP – End to End Call Setup

- *Proxy*: After receiving the SIP request from the User agent, the proxy contacts the location server to determine the next hop to forward the SIP

requests to. Once it receives the next hop information from the location server, it forwards the UA SIP request message. The proxy then updates the INVITE request message with its host address before forwarding it.

- *Redirect*: SIP Redirect Server responds to a UA request with a redirection response, indicating the current location of the called party.

SIP – Conference Setup: Conferencing where many parties can participate in the same call is now a common feature of multimedia communication systems. SIP supports three different multiparty conferencing modes:

1. *Ad hoc/Full Mesh*: In this mode, every participant establishes session with every other participant with a series of INVITE messages and sends an individual copy of the media to the others. This mechanism only scales to small groups [3].
2. *Meet me/Mixer*: In this mode, each participant establishes the point-to-point session to the Conferencing Bridge (or mixer). A mixer or bridge takes each participant's media streams and replicates it to all other participants as a unicast message. This mechanism is idea if all participants are interactive, however, it doesn't scale for a large number of participants [3].
3. *Interactive Broadcast/Network layer multicast*: In this mode, each participant establishes the point-to-point session to the Conferencing Bridge (or mixer). A Conferencing Bridge is used but mixed media is sent to a multicast address, instead of being unicast to each participant. This mechanism can involve active and passive participants. SIP signaling is required for interactive participants only. This mode works well for large-scale conferences [4].

User Mobility: One of the powerful features of SIP is its ability to support terminal mobility, personal mobility, and Service mobility to a SIP user.

- *Terminal Mobility (Mobile IP – SIP)*: **A** SIP user agent will be able to maintain its connections to the Internet as its associated user moves from network to network, and possibly changes its point of connection. The user's generic and location-independent address enables it to access services from both, stationary end devices, or from mobile end-devices [6].
- *Personal Mobility (SIP – REGISTER)*: SIP Personal mobility allows the user to access Internet services from any location by using any end devices. Since SIP URI (similar e-mail address) is device-independent, a user can utilize any end-device to receive and make calls. Participants can also use any end-device to receive and to make calls [6].
- *Service Mobility*: SIP service mobility provides a feature to a SIP user to keep the same services when mobile as long as the services/tools residing in the user agent can be accessed over Internet (e.g., Call Forwarding etc). Participant can interrupt the session and later on, continue at a different location [6].

Conclusion

SIP is a powerful and flexible application layer signaling protocol for multimedia applications. Its applications are not limited to Internet Telephony, although telephony applications are the main driving forces behind SIP development. Another popular application of SIP is Instant Messaging and Presence (IMP). IETF SIMPLE working Group is working on developing standards for IM and Presence. IP has been adopted by the third Generation Partnership Project (3GPP) for establishing, controlling, and maintaining real-time wireless multimedia sessions using Internet Protocol. SIP is an ASCI text based protocol, and SIP messages are long: up to and exceeding 800 bytes. This is not a problem for fixed networks with a high bandwidth, but it is for wireless cellular networks, where the bandwidth is very limited. For this reason, the SIP messages should be compressed in wireless networks. A number of proposals for SIP message compression have been submitted to the Robust Header Compression (ROHC) working group of the Internet Engineering Task Force (IETF). TCCB (Text Based Compression using Cache and Blank Approach) is a compression technique ideal for the compression of long ASCII text-based messages, such as SIP message bodies. Therefore, SIP message compression using TCCB has the potential to reduce request/response delays [5].

References

1. J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.
2. M. Handley et al., "SIP: Session Initiation Protocol," IETF RFC 2543, March 1999.
3. J. Arkko et al., "Security Mechanism Agreement for the Session Initiation Protocol (SIP)," IETF RFC 3329, January 2003.

4. J. Bannister et al., "Convergence Technologies for 3G Networks: IP, UMTS, EGPRS and ATM," Wiley, New York, 2004.
5. J. Sweeney et al., Efficient SIP based Presence and IM Services with SIP message compression in IST OPIUM, available online at: http://www.ist-opium.org/bluepapers/CIT%20-%20Blue%20Paper.doc, September 2003.
6. A.B. Johnston, "SIP Understanding the Session Initiation Protocol," Artech House, Norwood, MA, 2004.

ISO Standard Enabling Device and Coding Format

► MPEG-21 Digital Item Adaptation

ISO/IEC 21000

► MPEG-21 Multimedia Framework

ISO/IEC 21000-7:2007

► MPEG-21 Digital Item Adaptation