Image Analysis for Digital Media Applications

Hong Yan
City University of Hong Kong

Image analysis plays an important role in many digital-media-related applications, as this overview of analysis techniques explains. Although researchers have developed a variety of algorithms to solve many problems, numerous challenges remain.

Images, integral to digital media, are largely the reason why the World Wide Web's popularity quickly surpassed that of other Internet-based systems such as Usenet News and Gopher. Images, however, present designers with problems associated with creation, processing, recognition, storage, retrieval, and transmission. Specifically, we must consider how to describe properties, or extract features, of an image; how to classify objects in the image; and how to compare objects in different images. These tasks require a variety of analysis techniques, with which we can then enable computers to understand image features and contents. Table 1 summarizes the basic image analysis techniques and their applications.

Traditional image analysis applications include robot vision, automated product inspection, and optical character recognition.[1] The task of image analysis systems—to accurately describe a given image—is almost effortless for humans but difficult for computers to perform adequately. Nevertheless, many applications have successfully applied image analysis under restricted conditions.

In this broad-based overview, we review image analysis for three classes of digital media applications:

- traditional signal processing related procedures, including image enhancement, restoration and compression;
- relatively new media creation and data retrieval problems, including computer animation and multimedia data retrieval; and
- pattern recognition, including document imaging and electronic books.

We discuss specific problem areas, and techniques that address some of these, and briefly identify opportunities for future research.

Signal processing applications

Low-level image analysis procedures treat an image as a two-dimensional signal. A video sequence can be considered a time-varying 2D, or a 3D, signal. The operations to analyze the sequence include enhancement, filtering, restoration, transformation, reconstruction and compression, which are traditional image processing problems. Most techniques are developed as an extension of one-dimensional signal processing methods. Some image processing algorithms such as edge detection, a form of 2D filtering, are developed specifically for 2D applications. In this section, we consider three problems of image processing most relevant to media applications.

Image enhancement and restoration

For gray-image enhancement, we commonly use methods such as histogram equalization and filtering to improve image contrast and reduce noise.[1] A color image complicates enhancement because we must retain the color appearance and consider both luminance contrast and color contrast. To solve the problem, we can scale the $R(x, y)$, $G(x, y)$, and $B(x, y)$ components, where x and y represent the pixel position, using the same factor.[14] The new color components become $k(x,y)R(x,y)$, $k(x,y)G(x, y)$, and $k(x,y)B(x,y)$, respectively. Function $k(x, y)$ can be determined from the spatial luminance and color information of the image.[14] Figure 1 shows an example.

Color image enhancement procedures are more useful when combined with image segmentation in image and video editing and composition applications. For example, to enhance a selected object or area, we need automated boundary detection of the object. Automated detection in a complex scene requires sophisticated image edge detection, segmentation, and data clustering algorithms. That is, we must integrate low- and high-level image analysis techniques to improve the overall system performance.

Image restoration's task is to correct distortions in the image, such as noise and artifact reduction and de-blurring. To solve a restoration problem, we must build a mathematical model describing the image formation process and then find the inverse of the model to determine the source image. Unfortunately, the problem is usually ill-posed and we often have to

Table 1. Three levels—low, intermediate, and high—of basic image analysis methods and their applications. This classification is approximate, as one method can apply to different task levels.

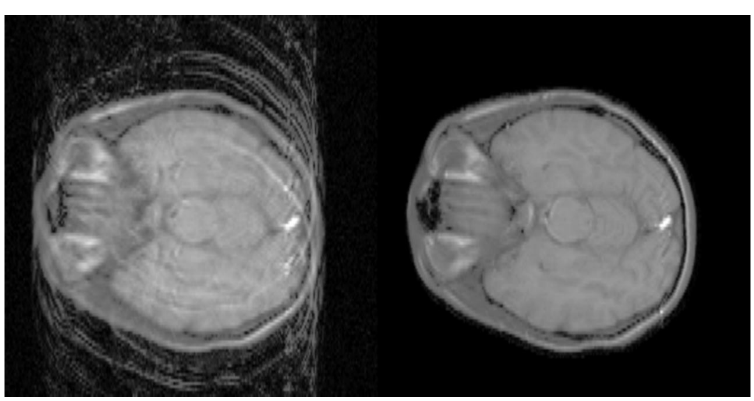
Type	Techniques and Applications
Image processing (low-level analysis)	*Transformations:* filtering, feature extraction, enhancement, compression[1]
	Maximum entropy method (MEM): de-convolution, super resolution, reconstruction[2]
	Projection onto convex sets (POCS): reconstruction, de-convolution, filter design[3]
	Fractals: compression, object matching[4,5]
Feature extraction (intermediate-level analysis)	*Thresholding:* object extraction from background[6]
	Edge detection: boundary detection[1]
	Thinning: skeletonization[1,7]
	Morphological operations: noise removal, object extraction[1]
	Snakes: boundary detection, object tracking[8]
	Self-organizing maps (SOMs): segmentation, clustering[9]
	Fuzzy c-means algorithms (FCMs): clustering[9]
	Morphing: animation through shape deformation[10]
Object recognition and matching (high-level analysis)	*Bayes theory:* classification[11]
	Neural net classifiers: object recognition, segmentation[11]
	Fuzzy classifiers: object recognition, rule-based systems[9]
	Hidden Markov models (HMMs): speech and handwriting recognition[12]
	Graph matching: structural matching[1]
	Hough transform: known shape detection[1]
	Shape from shading: finding 3D shapes from 2D images
	Relaxation labeling: object matching[13]

(a) (b)

1 Example of image enhancement: (a) the original color image and (b) the enhanced image. Fine features of the trees and flowers become more visible after the enhancement.

reconstruct the image from incomplete information.

Maximum entropy[2] and projection onto convex sets (POCS)[3,15] are two effective algorithms for solving restoration problems. Figure 2 shows an example of magnetic resonance image (MRI) restoration from motion-corrupted data. Image restoration is application dependent, and existing methods are usually iterative. It remains an active research problem to find robust, stable, and fast algorithms to solve problems in this area.



2 An example of magnetic resonance image (MRI) reconstructed from data corrupted by motion. Left: the image with so-called ghost artifacts from motion. Right: image restored using the projection onto convex sets (POCS) method.

3 **Blocking artifact removal from highly compressed images. (a) A compressed color image using the block discrete cosine transform (BDCT). It has a compression ratio of 77 and peak signal-to-noise ratio (PSNR) of 23.98 dB. (b) Result of blocking artifact removal using wavelet transforms. This image has a PSNR of 25.43 dB.**

4 **Example of image compression using fractals: (a) the original image and (b) the compressed image. The compression ratio is 20.07 and the peak signal-to-noise ratio (PSNR) is 31.21 dB.**

Image and video compression

Researchers have extensively studied the compression of sound, image, and video data in signal processing. JPEG and MPEG standards recommend the discrete cosine transform (DCT) and wavelet transform for image and video compression. These methods divide an image into small blocks and transform each block to another domain to reduce spatial correlation, then quantize the transform coefficients to minimize the amount of data required for image reconstruction.[1] When the compression ratio is high, boundaries between the blocks become visible. Researchers have investigated several methods to solve this problem. Figure 3 shows the result of blocking artifact reduction using a wavelet-transform-based method.

The DCT and wavelet transforms don't consider high-level image contents, such as the shapes of image objects, so they provide only moderate compression. Fractal- and segmentation-based compression methods overcome this problem. Fractal methods derive from the collage theorem, which states that an image can be reconstructed using the iterated function systems (IFS), which map one area to another starting from a random initialization.[4,5] Unfortunately, arbitrary shapes cannot be matched to generate the IFS codes automatically. Instead, areas of regular shapes, such as squares and triangles, are used for matching in an image.[4] Figure 4 shows a compressed color image using the IFS of square areas. Geometric mappings, such as nonlinear transforms, improve the fractal-based method's compression efficiency.[5]

Segmentation-based image compression (or second-generation) methods provide a very high compression ratio by dividing an image into small homogeneous regions that can be compressed efficiently.[16] However, applying this method to a wide class of natural images requires more research. MPEG-4 introduces image segmentation to separate the objects and the scene in an image, which allows user interactions with different video data components.[17]

MPEG-4 also accepts synthetic images. For example, to show a face in a video, we need only face definition parameters (FDPs) to describe the face shape and face animation parameters (FAPs) to describe facial expressions. Because the data needed are the parameters only, not the actual image frame sequence, we can minimize the bit rate. One way to generate personalized FDPs is to analyze the person's face in photographs.[18] This requires key feature point extraction and matching as well as texture mapping. Making the procedure automatic is a challenge.

Media creation and data retrieval

Two relatively new applications of image analysis are computer animation and media data retrieval. In both cases, unsolved problems are accurate image-content analysis and pattern matching, recognition, and modeling.

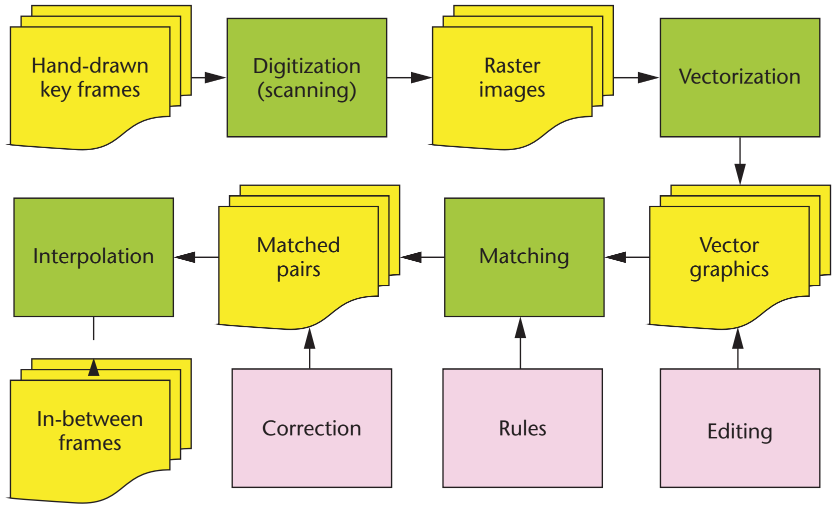
Computer animation

Image analysis is increasingly useful for computer-generated scene images, character animation, and virtual reality applications. The analysis of hand drawings, photographs, video, and motion data is useful in digital media applications for two reasons: to let the computer perform labor-intensive tasks and to develop graphics models of natural objects based on the features extracted from the images of real objects.

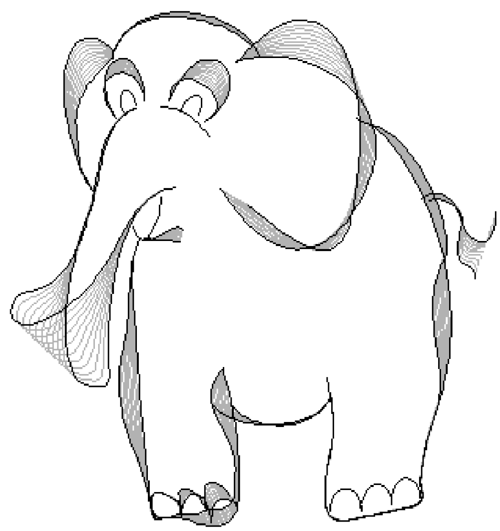
2D animation. In cartoon movie production, animators draw key frames and then draw many *in-betweens*—frames that interpolate the shapes in the key frames to show moving objects, a tedious and time-consuming process. The computer can generate in-betweens automatically or semiautomatically by analyzing key frames.

Figure 5 shows a cartoon image processing system we recently developed. Two image analysis tasks are involved. *Vectorization*, explained later, converts scanned bitmap images to lines, which can then be edited, stored, and re-used easily. The second task implements *pattern matching*, which identifies and matches the same parts of an object in different frames, then interpolates them to generate the in-betweens. In a method we have developed recently, relaxation labeling and graph matching algorithms solve this problem.

Figure 6 shows an example of in-between frames generated from two key frames. The matching procedure can make mistakes if one frame is substantially different from the other. For example, an open mouth can have many more features than a closed mouth. Animators solve these problems based on rules, library patterns, and knowledge. Programming a computer to perform the same operations will be a challenge.



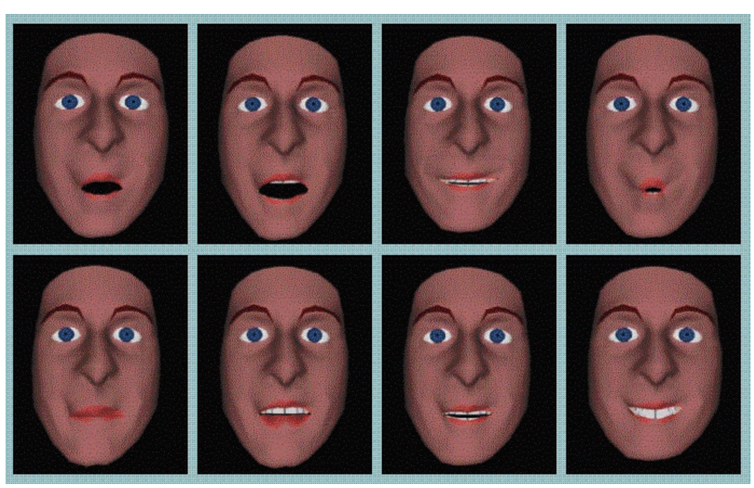
5 A cartoon image-processing system. Image analysis techniques solve hand-drawing vectorization and pattern-matching problems.



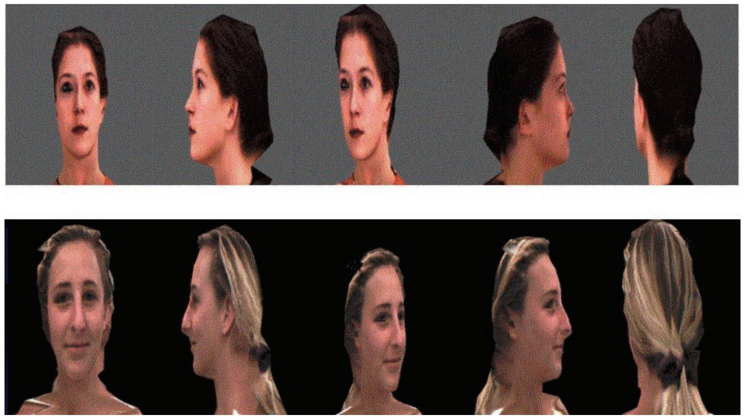
6 Key frames (dark lines) and in-betweens (gray lines) generated using our method. The matching between the two key frames is done automatically.

3D animation. Computer animation's most difficult and exciting task is probably 3D shape and motion modeling of humans.[19,20] Current computer modeling techniques always result in a stiff human image, even when the person is smiling. It's long been the objective of computer animation research to make the human facial expressions look natural. One solution is to model the human head's physical structure and muscle movement, as Figure 7 shows. Another promising method to obtain more natural-looking facial appearances is to identify clues from a real object's image or video and use the features in graphics models.

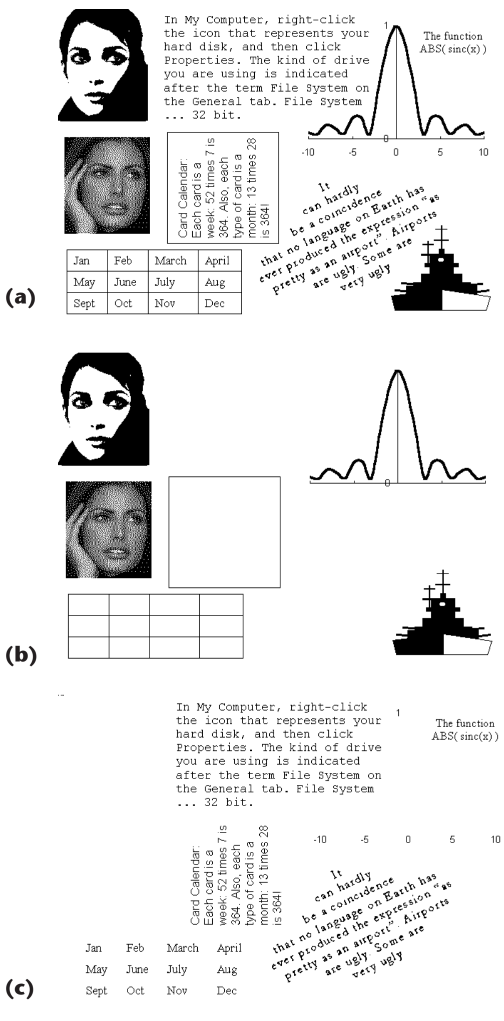
Image analysis techniques aid in processing data from motion-capturing devices, 3D scanners, and video images. For example, one way to create human face models is by analyzing two orthogonal head pictures of a person, then matching corresponding key points and mapping them to a generic mesh.[18] We can make the head model more



7 Facial expressions generated using an abstract muscle model.



8 Examples of 3D head models created based on laser-scanned range and reflectance data.



9 Document layout analysis: (a) the original document image, (b) images and diagrams extracted from the document image, and (c) text in different orientations extracted from the document image.

realistic by mapping texture images to the mesh. We can also build a 3D face model with data from laser scanners.

In all these methods, to automate the process, we must use sophisticated feature extraction algorithms to locate and match the positions of nose, eyes, and ears—for example, in the images.[21] Figure 8 shows 3D head images reconstructed from 3D laser-scanned range and reflectance data.

Multimedia data retrieval

Because of the rapidly swelling volume of digital media data, the indexing and retrieval of database items assumes greater importance.[22] Traditionally, database queries are textual, which is somewhat useful for image retrieval. For example, to buy a coffeemaker with a specific color, shape, and size, we can search through a database of coffeemakers using a particular set of parameters. One problem is how to specify in words the exact shape and size, and match the user's requirement with the shapes in the database. If we are given only the coffeemaker-component images, we must analyze these images and extract the shapes to do the matching.

For image retrieval, we can also submit an object image or sketch as a database query to find similar objects. Such retrievals require us to match the input image or drawing with the database images. In some cases, we can match them based on statistical information, such as color histograms and image transform coefficients. However, when structural image information is required, we must analyze the image contents and deal with complicated pattern classification and matching problems.[9,11,23]

Document processing and character recognition

Printed and handwritten materials, such as newspapers, books, and office documents, have historically served as one of the most useful forms of media information exchange. The aim of document imaging systems is to convert paper documents to an electronic format for more efficient document storage, indexing, and distribution. Although many new documents can now be directly generated as computer files, we must digitize and analyze a huge number of existing paper documents, in addition to new ones being produced every day.

Document imaging

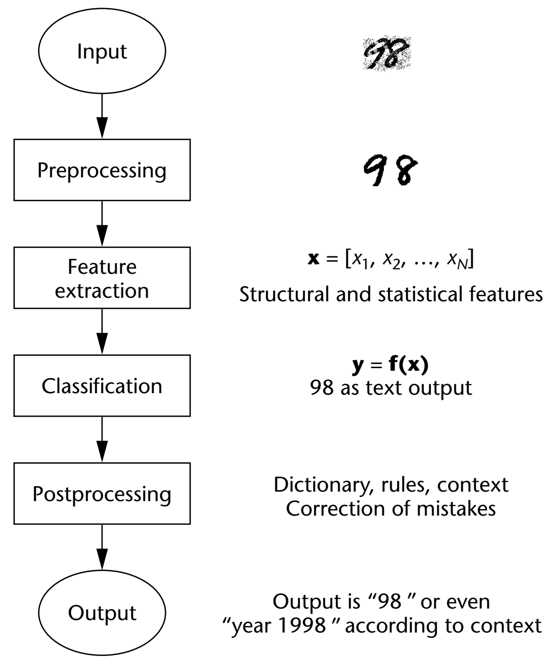
Although it's a simple procedure to scan a document, it can be complex to analyze its contents, as we explain here.

Image binarization. Most document images are stored in binary (black-and-white) format, which we obtain by thresholding a gray-scale image.[6] If the gray-scale image has a high contrast, a global threshold is sufficient to separate the image background from foreground. However, an image containing a varying, textured background requires more sophisticated binarization, in which case we must determine the threshold value locally, even at each individual pixel.[6]

Layout analysis. Layout analysis, which separates image text, diagrams, and pictures, must precede document content recognition and analysis because text and graphics require different analysis techniques.[12,24] If text lines are horizontally oriented, and the text and diagram blocks are rectangular, we can use pixel projection histograms to extract text and diagram areas. This method works only if the document can be deskewed first.[25] A more complicated document structure requires additional analysis features, such as pixel run lengths, connected components, thinned white space, texture, and k-nearest neighbors.

The layout analysis problem is essentially classifying each pixel or region. Figure 9 shows segmentation results of a document that contains pictures, diagrams, and text in different orientations.[26] Once these components are separated, they can be passed to the next document processing system for compression or recognition.

Layout analysis also aids document retrieval. For example, even one scanned newspaper page can hold too much data to download from the Internet. If developers can analyze the page structure, extract each article, and index the page using the article titles, users can browse the page and download articles only as needed. The solution to this complicated problem requires layout analysis combined with character recognition.



10 **Flow diagram of a pattern recognition system. The example shows connected handwritten character recognition.**

Character recognition. A well-defined technique with many applications, optical character recognition has long been one of the most actively researched topics in pattern recognition.[27] A related problem is signature verification,[28] in which we must identify the writer rather than the input sample contents. Figure 10 shows the components of a general OCR system. The preprocessing procedure usually isolates and normalizes the character image and removes noise. Then the feature extraction procedure computes a set of features $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ from the bitmap of the input character and passes them to the classifier. The classification results are modified according to additional information such as a word's spelling. The classifier is considered a function $f(\mathbf{x})$, which is produced from training samples. The features $\mathbf{x}$ represent statistical properties of the input character, such as character pixel distribution, and character structure, such as stroke positions, directions, and their relations.

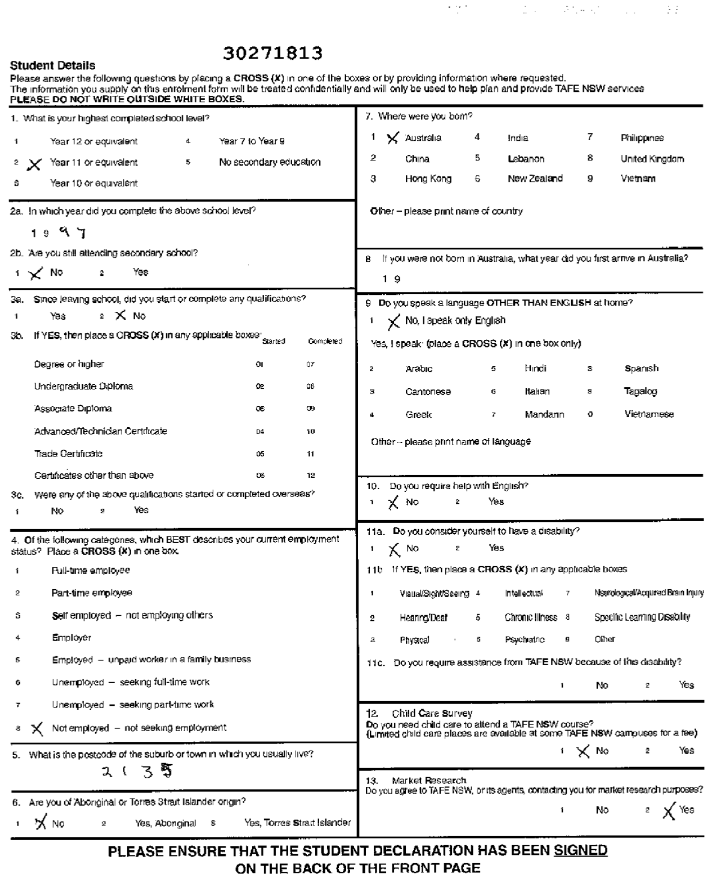
Researchers have studied many methods for classifier design, including nearest-neighbor classifiers, Bayes classifiers, neural nets, fuzzy classifiers, Markov models and classification trees, and combinations of these.[9,11] The training and classification times can differ significantly for different classifiers, but if designed correctly, classifiers can have a similar recognition performance for a given set of features because the classification error is bounded by the Bayes error.

Computers are humans' equal at recognizing isolated and clean-printed characters, but are much less able than humans in recognizing variable, degraded input. Research problems include recognition of connected or broken characters, of free handwriting or drawing, and of different classifier combinations.

Line vectorization. Usually called thinning, line vectorization is useful for extracting skeletons of characters and for analyzing mechanical drawings, maps, and cartoon images.[7] In traditional thinning algorithms, an "onion peeling" procedure removes redundant pixels in a line and makes it only one pixel thick. This method often produces artifacts across line cross sections and is sensitive to noise. In more sophisticated methods, we analyze the line boundary and the cross-section structure to generate a more accurate representation.[7] Heavy noise is an unsolved problem; a related problem is how to cluster scattered data that have a line shape. Clustering algorithms can be used to extract circles, ellipses, and straight lines, but more flexible techniques are needed to deal with more general shapes.

Document image compression. Widely used for compressing binary images, the CCITT Group 4 method was developed based on run-length coding to minimize redundancy within a line and correlation to the previous line. This method provides lossless compression. A new standard, JBIG2, proposed by the Joint Bi-level Image Experts Group, is emerging that provides compression based on the symbol-matching concept.[29] For example, a document may contain the letter "A" in many places. To place the letter correctly in the document reconstructed after compression, we need only to store one bitmap of "A" and code its positions. The original bitmaps of the letter may be somewhat different in different places, but they can all be replaced by the same representative bitmap without changing its meaning.[30] As long as all symbols are matched correctly, the compression can be made content-lossless, although it's lossy at the pixel level.

Figure 11 (next page) shows a document image and the associated symbols needed to represent the printed and handwritten text in the document. A symbol-matching-based compression method provides a compression ratio several times higher than the CCITT Group 4



(a)

(b)

11 Document image compression: (a) the original document and (b) extracted symbols to represent printed and handwritten text in the document.

method and is useful for developing new fax machine, document scanner, and other document-imaging system applications. Current research problems include extracting and matching the symbols in an image efficiently, quickly, and accurately.

Electronic books

Electronic books have many advantages over paper books. For example, e-books can present multimedia data and can be interactive. Also, they are environmentally friendly; require much less storage; and are easily indexed, searched, and distributed.

An e-book usually consists of two parts: a data file that contains the book's contents and a reader that makes the presentation. The reader can usually be used to read many different books. Some e-books can be read on a general-purpose computer in which the reader is just a computer program. There are also specialized hardware e-book readers, ranging from small palm devices to special multiscreen notebook computers. Some readers allow user interactions and written and voice annotations.

Our work on e-books for educational purposes is inspired by the observation that many mathematical operations are easily explained with animation. The author's use of computer movies in undergraduate teaching has been very well received by students. For example, animation movies effectively show signal convolution, as Figure 12 shows, and the decomposition of a rotation around the axis in an arbitrary direction (see http://www.HyperAcademy.com).

In terms of image analysis, research is needed in two main areas. One is to improve the reader display, including fonts, layout, and user interface. Ordinary computer screens still have much lower resolution than laser printers, and the touch and feel of e-books are different from the paper books people are used to. The reading of e-books must be made more comfortable.

Research is also needed to create more intelligent, easy-to-use authoring tools, especially for educational material. For e-books to become popular, we need many authors to write them. Simple static material like scanned pages will attract few users. However, it's difficult and time-consuming to create e-books of interactive mathematical and technical material, which discourages authors. One solution would be authoring software that accepts handwriting. It should let authors write an equation and draw a diagram, and then have a computer recognize and change them to font symbols and well-formatted diagrams.

The recognition need not be perfect. For example, it could provide the user with several choices or templates. It should also allow the user to create animation to relate equations and variables to diagrams and show the change of equations so that a concept or procedure can be described clearly. Essentially, the authoring tool must be simple and intuitive to use, and be able to handle all the user's work.

Discussion

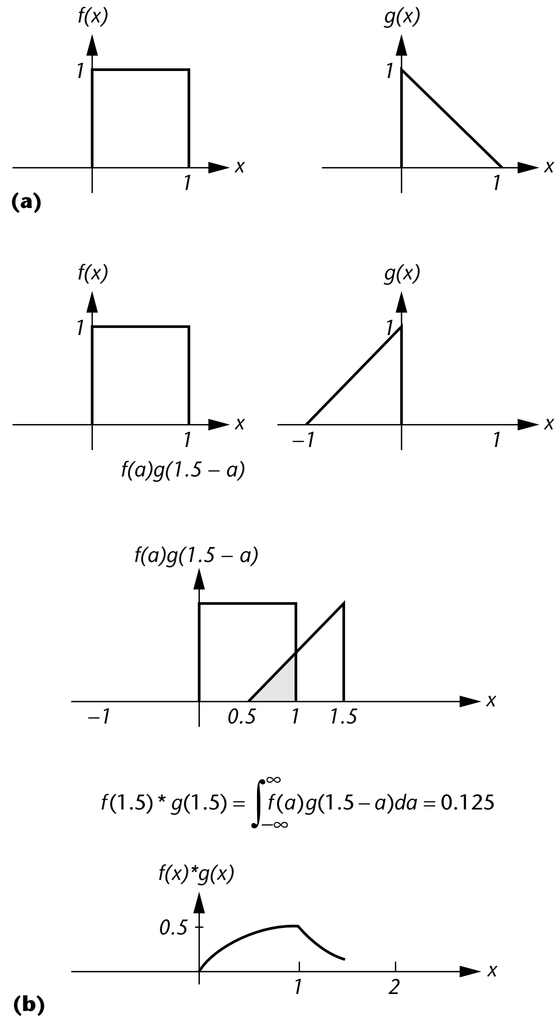
Image filtering, enhancement, transformation, compression, and restoration algorithms are relatively well understood and well developed. These techniques can have many useful applications in Internet and mobile communications systems. For example, vectorized cartoon images are well suited for mobile displays. Handwriting analysis and recognition in e-books can be used for processing handwritten notes and drawings. In the next few years, we shall be able to use more and more Internet- and mobile-based imaging and video services in education, business, and entertainment. Future research needed in these areas includes faster hardware implementation, for example, for image and video coding; efficient and secure data transmission through Internet and wireless systems; and development of robust and stable algorithms for image restoration.

The success of new industrial products in digital media will also depend on user acceptance. For example, e-books can potentially have a very large market, but it is not known whether the users can switch from paper books to e-books. As discussed above, authors' acceptance is also crucial. At the same time, the study of user requirements might also lead to many interesting, practical research problems.

Feature extraction and object recognition are key but largely unsolved image analysis problems. Current computer algorithms are far less reliable than humans. The problems include these: Computer models are too sensitive to noise and image distortions; learning algorithms are too slow and require too many training samples; and low- and high-level feature extraction and object classification procedures are mainly separate processes. Although researchers have tried to solve the problems based on structural and statistical methods, symbolic and numerical formulations, and on various clues from physiological findings of the brain, the recognition problem remains challenging.

Despite the difficulties in searching for a theoretical solution to the object recognition problem, it's still possible to build useful systems for practical applications. We can improve system performance by putting more restrictive conditions on input images, having well-defined objectives, using many expert rules, and combining different classifiers. All these strategies can work well in a handwriting recognition system. For example, we can achieve a high recognition rate for well-isolated characters by integrating several classifiers.[9]

In research, useful ideas can be learned from a different field. For example, the hidden Markov model (HMM), which has been used successfully for speech recognition, can also be useful to solve image recognition problems. Image analysis and computer graphics have till now been studied mainly in separate fields. The merging of techniques from other fields can lead to new algorithms and applications. ■



12 **Two frames of an animation movie show the operations of signal convolution. View this movie at http://www.HyperAcademy.com or http://www.hy8.com.**

Acknowledgment

Our work on cartoon image analysis, handwriting recognition, and document image compression is supported by several grants from the Australian Research Council. Our work on Electronic Books is supported by an information technology grant from the University of Sydney and a teaching development grant from the City University of Hong Kong. Many research staff and postgraduate students of the Image Processing Lab at the University of Sydney and the Signal Processing Lab at the City University of Hong Kong have worked on these projects and provided the images presented in this article.

References

1. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1992.
2. *Maximum Entropy and Bayesian Methods*, J. Skilling, ed., Kluwer Academic, Dordrecht, the Netherlands, 1989.
3. H. Stark and Y. Yang, *Vector Space Projections*, John Wiley and Sons, New York, 1998.
4. A.E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," *IEEE Trans. Image Processing*, vol. 1, no. 1, Jan. 1992, pp. 18-30.
5. D. Popescu and H. Yan, "A Non-Linear Model for Fractal Image Coding," *IEEE Trans. Image Processing*, vol. 6, no. 3, 1997, pp. 373-382.
6. Y. Yang and H. Yan, "An Adaptive Logical Method for Binarization of Degraded Document Images," *Pattern Recogni-*

tion, vol. 33, no. 5, 2000, pp. 787-807.
7. H.H. Chang and H. Yan, "Analysis of Stroke Structure of Handwritten Chinese Characters," *IEEE Trans. Systems, Man and Cybernetics*, vol. 29, no. 1, 1999, pp. 47-61.
8. Y. Zhu and H. Yan, "Computerized Tumor Boundary Detection Using a Hopfield Network," *IEEE Trans. Medical Imaging*, vol. 16, no. 1, 1997, pp. 55-67.
9. Z. Chi, H. Yan, and T.D. Pham, *Fuzzy Algorithms, with Applications to Image Processing and Pattern Recognition*, World Scientific Publishing, Singapore, 1996.
10. J. Gomes et al., *Warping and Morphing of Graphics Objects*, Morgan Kaufmann, San Francisco, 1999.
11. J. Schurmann, *Pattern Classification, A Unified View of Statistical and Neural Approaches*, John Wiley and Sons, New York, 1996.
12. *Document Image Analysis*, L. O'Gorman and R. Kasturi, eds., IEEE Computer Soc. Press, Los Alamitos, Calif., 1995.
13. R. Klette, K. Schluns, and A. Koschan, *Computer Vision, Three Dimensional Data from Images*, Springer-Verlag, Singapore, 1998.
14. N. Liu and H. Yan, "Improved Method for Color Image Enhancement Based on Luminance and Color Contrast," *J. Electronic Imaging*, vol. 3, no. 2, 1994, pp. 190-197.
15. M. Hedley, H. Yan, and D. Rosenfeld, "Motion Artifact Correction in MRI using Generalized Projections," *IEEE Trans. Medical Imaging*, vol. 10, no. 1, 1991, pp. 40-46.
16. T. Ebrahimi and M. Kunt, "Visual Data Compression for Multimedia Applications," *Proc. IEEE*, vol. 86, no. 6, June 1998, pp. 1109-1125.
17. G.A. Abrantes and F. Pereira, "MPEG-4 Facial Animation Technology: Survey, Implementation, and Results," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 9, no. 2, Mar. 1999, pp. 290-305.
18. W.-S. Lee and N. Magnenat-Thalmann, "Fast Head Modeling for Animation," *Image and Vision Computing*, vol. 18, 2000, pp. 355-364.
19. F.I. Parke and K. Waters, *Computer Facial Animation*, A.K. Peters, Wellesley, Mass., 1996.
20. S.K. Karunaratne and H. Yan, "An Emotional Viseme Compiler for Facial Animation," *Proc. Fifth Int'l Symp. Signal Processing and Its Applications*, Queensland University of Technology, Brisbane, Australia, Aug. 1999, pp. 459-462.
21. K.M. Lam and H. Yan, "An Analytic-to-Holistic Approach to Face Recognition Based on a Single Frontal View," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, 1998, pp. 673-686.
22. Y. Rui et al., "Relevance Feedback: A Power Tool for Interactive Content-based Image Retrieval," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 8, no. 5, Sept. 1998, pp. 644-655.
23. T. Tan and H. Yan, "Face Recognition by Fractal Transformations," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, IEEE Press, Piscataway, N.J., Mar. 1999, pp. 3537-3540.
24. A.K. Jain and B. Yu, "Document Representation and Its Application to Page Decomposition," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 20, no. 3, Mar. 1998, pp. 294-308.
25. H. Yan, "Skew Correction of Document Images Using Interline Cross-Correlation," *CVGIP: Graphical Models and Image Processing*, Vol. 55, no. 6, 1993, pp. 538-543.
26. P.E. Mitchell and H. Yan, "Document Page Segmentation Based on Pattern Spread Analysis," *Optical Eng.*, vol. 39, no. 3, 2000, pp. 724-734.
27. J. Hu and H. Yan, "A Model-Based Segmentation Method for Handwritten Numeral Strings," *Computer Vision and Image Understanding*, vol. 70, no. 3, 1998, pp. 383-403.
28. K. Huang and H. Yan, "Off-Line Signature Verification Based on Geometric Feature Extraction and Neural Network Classification," *Pattern Recognition*, vol. 30, no. 1, 1997, pp. 9-17.
29. P.G. Howard et al., "The Emerging JBIG2 Standard," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 8, no. 7, Nov. 1998, pp. 838-848.
30. Y. Yang, H. Yan, and D. Yu, "Content Lossless Compression of Document Images Based on Structural Analysis and Pattern Matching," *Pattern Recognition*, vol. 33, no. 8, 2000, pp. 1277-1293.

Hong Yan *is a professor of computer engineering in the Electronic Engineering Dept. at the City University of Hong Kong. His research interests include computer animation, signal and image processing, pattern recognition, and neural and fuzzy algorithms. Yan received a BE from Nanking Institute of Posts and Telecommunications in 1982, an MSE from the University of Michigan in 1984, and a PhD from Yale University in 1989, all in electrical engineering. He is a fellow of the International Association for Pattern Recognition; a fellow of the Institution of Engineers, Australia; a senior member of the IEEE; and a member of the International Society for Optical Engineers, the International Neural Network Society, and the International Society for Magnetic Resonance in Medicine.*

Yan is currently on leave from the School of Electrical and Information Engineering, University of Sydney, NSW 2006, Australia.

Contact him at Dept. of Electronic Eng., City Univ. of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong; h.yan@cityu.edu.hk.