IMAGE PROCESSING ALGORITHMS FOR GRAPHICS

T. Pavlidis

Bell Laboratories
Murray Hill, N.J.

ABSTRACT

Computer graphics deals with the generation of images from descriptions while pattern recognition derives descriptions from images. It is natural that both disciplines rely on image processing and there are many problems where interaction among the three areas is promising. For example, image segmentation, a prerequisite for scene analysis, can benefit from the work on shading and illumination of surfaces in graphics. The talk will concentrate on geometric transformations, where methods from image processing and pattern recognition can be of benefit to graphics. Such methods suggest how to do these transformations in the display memory in terms of pixels rather than in main memory in terms of vectors. For example, contour filling may use region traversal algorithms, while scaling can be performed in terms of thinned images.

RÉSUMÉ

L'informatique graphique porte sur la production d'images à partir de descriptions, tandis que la reconnaissance des formes tire des descriptions des images. Comme ces disciplines reposent naturellement sur le traitement des images, il existe un grand nombre de cas où l'interaction entre ces trois techniques se révèle très avantageuse. Ainsi, le découpage des images, qui est indispensable à l'analyse des images, peut tirer profit des techniques utilisées pour ombrer ou éclairer les surfaces dans les représentations graphiques. Cette communication portera surtout sur les transformations géométriques, pour lesquelles les méthodes tirées du traitement des images et de la reconnaissance des formes peuvent être utiles à la représentation graphique. Ces méthodes indiquent de quelle façon réaliser ces transformations dans la mémoire de l'écran de visualisation, en termes de pixels, plutôt que dans la mémoire centrale, sous forme de vecteurs. Ainsi, le remplissage des contours peut faire appel aux algorithmes de parcours de région, tandis que la mise à l'échelle peut être exécutée en termes d'images amincies.

Computer graphics deals with the generation of images from descriptions while pattern recognition produces descriptions from images. In this sense, the fundamental operations in graphics are the inverse of pattern recognition operations. Both rely on image processing and there are many problems where interaction among the three areas is promising. For example, *image segmentation*, a prerequisite for scene analysis, can benefit from the work on shading and illumination of surfaces in graphics. This presentation will concentrate on problems where the applications of results from image processing or pattern recognition seems promising for graphics. Most of them are still open problems and my goal is not to offer solutions, but rather focus the attention to concrete research questions. The first three problems belong to *Discrete Geometry* which deals with geometrical operations over a discrete grid rather than on the Euclidean plane. The resulting problems are quite challenging.

1. *Display of Lines and Curves:* naive quantization of lines results in situations where if A is a point of the line segment BC, the displayed line segment AC is not a subset of the displayed segment BC. Also two "discrete lines" do not have a unique intersection. (Finding the intersection on the analog plane and then mapping it on the discrete does not eliminate the ambiguity.) Some of the commercially available displays do a poor job in drawing conics because of inadequate attention to these type of problems. A popular class of algorithms is based on defining a curve by a differential equation (a carry over from the days of analog computers) and then plotting the curve by integrating a quantized from of the equation. Such techniques have difficulty in satisfying certain types of end conditions so that the resulting displays may overshoot the desired endpoint. It is desirable to investigate discrete forms that approximate straight lines or conics but satisfy precisely end conditions and certain fundamental geometrical properties.

2. *Contour filling* may be performed in terms of the analog contour or it may be done using region traversal algorithms. Such discrete implementations have the advantage that, if properly implemented, require little communication between the host computer and the display processor. Also they impose few constraints on the form of the contour so that they can be used for interactive graphics. Presorted contours are appropriate for applications where the same object is displayed repeatedly: animation, phototypesetting, etc. Region traversal algorithms seem to be more appropriate for applications where the form of objects is not fixed. In addition to interactive graphics, these include image' processing where a contour may be found as a result of thresholding or edge detection.

3. *Scaling of discrete regions* is still an unsolved problem in spite of its relevance to font design for displays. For example, given a set of pixels P that is symmetric around some axis a, we would like to find another set R that is half the size of P and it is also symmetric around a, *without having to change the origin*. Some progress towards the solution of this class of problems may be made if we design algorithms operating on thinned versions of the sets. Thus P is first thinned to $T(P)$, then $T(P)$ is scaled, and R is found as the expansion of $T(P)$.

4. *"Aliasing:"* the meaning of this term, as used in the computer graphics literature, is different from its traditional use in image processing. There aliasing means the mapping of high analog frequencies into low ones because of undersampling. In graphics the term refers to the "staircase" or "jagged" appearance of the piecewise constant reconstructions. The artifact is not a low frequency, but a high frequency. It is possible to reduce this phenomenon by increasing the sampling frequency, a remedy that is also used to counteract aliasing. The similarity of the remedies has probably contributed to the confusion in the terminology. The question is not purely academic because if we have to live with a fixed sampling rate, then the remedies are different. To avoid aliasing in the original sense of the word we must *prefilter* the *analog input*. To avoid aliasing as the term is used in computer graphics we must *filter* the *digital output*. The source of the problem is that most if not all display devices use piecewise constant reconstructions of the signal (zero order holds in signal processing terminology). These are far from optimal and require sampling at rates far above the Shannon rate. Clearly, hardware implementation of higher order two-dimensional filter will provide better displays at low resolution.