MODERN DIGITAL IMAGE ANALYSIS

Ian T. Young

Faculty of Applied Physics
Delft University of Technology
Delft, The Netherlands

Abstract

While image processing is concerned with the transformation of images into *images*, image analysis is primarily concerned with the transformation of images into *data*. In many applications this ability to extract quantitative data from two-dimensional signals is the primary goal. These data may then be used to characterize the signals or to aid in a decision-making process.

To achieve this end a number of significant techniques have been developed in the past decade. These include filters which transform either binary or grey-level images into other "images" where the numerical values in the output matrix represent measures. In classical signal analysis the Fourier transform would be such an example. In the newer techniques the values may represent topological, size, texture, or distance measures applied to the original image.

The issue of measurement accuracy, itself, has begun to receive considerable attention. Traditional concepts such as Nyquist frequency have been shown to be inadequate when the goal of processing is measurement rather than filtering. The relationship between measurement accuracy, the sampling frequency, and the formulas for parameter estimation have been studied. The levels of accuracy required in industrial inspection, for example, frequently require sampling frequencies far in excess of the Nyquist frequency.

Introduction

In 1989 we celebrate the 150th anniversary of the technical (as opposed to artistic) recording of images by L.J.M. Daguerre [1]. Image processing – that is, the processing of images with the aid of digital computers – is just thirty years old [2,3] and yet in that time there has been an impressive series of developments.

Thirty years ago digital image processing was in its infancy. Computers were located in inaccessible computation centers. Frame grabbers did not exist. Computer languages were limited to assembly language. The first laboratory-oriented computer was introduced in the early 1960's with the PDP-1 and the first collection of languages appeared at about the same time. As of this writing (1989) the computational power, memory, and storage capacity available 30 years ago in computation centers has been substantially surpassed by the power available in a desk-top personal computer that is priced at a level that makes it affordable to individual researchers *and their families*. The software sophistication available on all fronts – operating systems, user interfaces, languages, and image processing algorithms – has witnessed a comparable improvement [4-8].

Image analysis – the extraction of data from images – is approximately 25 years old. Some of the first work in this field was the attempt to classify automatically the 46 chromosomes in the human cell [9]. This work and the many efforts that have followed it have led us to a situation in 1989 where we now understand what are the proper questions to ask in the context of image analysis and how to apply the answers where they are known. Some of these questions are:

• How can the verbal description of some property in an "analog" image be translated into a quantitative measure?

• What is the proper representation of the digital image in order to compute efficiently the desired measure?

• What is the proper formula for estimating the analog measure given the digital data?

• What is the sampling density required to obtain an arbitrary accuracy and precision in the analog measure given the digital data?

• How can the desired measure be tested as to efficacy in a controlled way?

• Can image transformations be used in such a way as to implement the measurement process or aid in the implementation of the measurement process?

I will return to specific examples to illustrate the implications of these questions but first it is important to understand the central distinction that exists between image *processing* and image *analysis*. (This distinction will play an important role when I discuss the issue of sampling frequency.)

In image *processing* one is concerned with "image in → image out" and the theory surrounding this model is constructed accordingly. In image *analysis* one is concerned with "image in → data out". The output is not an image but rather data derived from the image. Finally there is the model "data in → image out" and this represents the increasingly important field of image *graphics*. The first two of these quite distinct fields of study are illustrated in Figure 1.

While I have painted a very black/white picture of the split between processing and analysis the delineation is not quite as stark. Many measures can, in fact, be derived through a series of image transformations or filters. Later in this manuscript I will present several such examples.

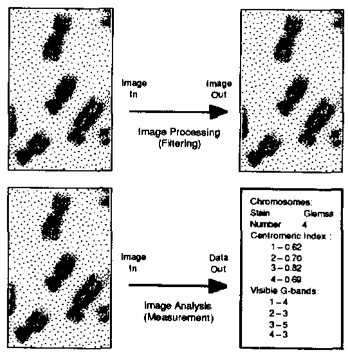

Figure 1: Illustration of the distinction between image processing and image analysis.

Analysis Through Measurement

The basic questions raised in the previous section are best explained through an example. Consider the digital image shown in Figure 2. The problem is to estimate the area of the original analog object that gave rise to this set of digital values.

It is straightforward to justify at least three different formulas for estimating the area of the original object. Let us assume that the distance between the centers of the pixels is 1 mm. Then:

Estimate #1 – The area is simply the number of pixels belonging to the object. Thus $A = 10\ mm^2$.

CH2673-2/89/0000-1699 $1.00 © 1989 IEEE

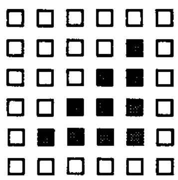

Figure 2: Pixels belonging to the object are dark. Pixels belonging to the object are white.

Estimate #2 – The collection of pixels form a triangle. The base of the triangle and the height of the triangle are both 3 mm. Thus $A = 3\cdot 3/2 = 4.5\ mm^2$.

Estimate #3 – The collection of pixels form a triangle. The base of the triangle and the height of the triangle are both 4 mm. Thus $A = 8\ mm^2$.

Each of these arguments – by itself – is quite reasonable. Taken together they represent three very different estimates for what should be a simple problem. It has been shown [10] that in fact the first formula is the correct one for achieving an unbiased, consistent estimate of the area of the original *analog* object.

This same, fundamental question – what is the proper formula – can be posed for every property that one wishes to measure from a digital image. Dorst and Smeulders [11] have studied this problem extensively in the context of the measurement of the length of straight lines and – to a more limited extent – in the estimate of the perimeter length of closed curves. They find that three factors determine the accuracy and precision of any given length estimator:

1. The representation of the digitized line;
2. The formula used to estimate the length given the representation, and;
3. The sampling density.

As an example, given a Freeman chain code representation of a straight line [12], there are several possible formulas in the literature for estimating the line length:

$L_S = N_e + N_o$	[Ref. 13]
$L_F = 1.000\cdot N_e + \sqrt{2}\cdot N_o$	[Ref. 12]
$L_G = 1.059\cdot N_e + 1.183\cdot N_o$	[Ref. 14]
$L_P = 0.984\cdot N_e + 1.340\cdot N_o$	[Ref. 15]
$L_C = 0.980\cdot N_e + 1.406\cdot N_o - 0.091\cdot N_c$	[Ref. 16]

where N_e is the number of even chain codes, N_o the number of odd codes, and N_c the number of corners (see [16]).

Each of these formulas leads to a different accuracy in the estimation of the line length given a certain sampling density. This is illustrated in Figure 3.

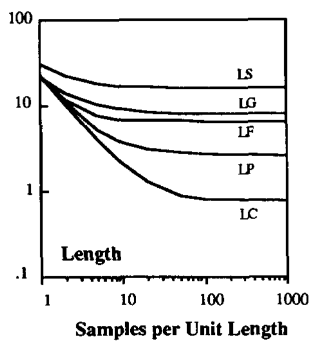

Figure 3: Percentage error associated with five different formulas for estimating the length of a straight line given the Freeman chain code representation.

The asymptotic behavior of these formulas varies considerably. Only the estimator L_C from [16] is capable of obtaining accuracies of 1%. And this occurs only when the sampling density exceeds about 50 samples per unit line length. As shown in [11], this sample result holds for the perimeter length of circular contours as well. If one wishes to measure length, perimeter, distance and so forth to accuracies better than 1% – as can easily occur in the inspection of industrial components – then the Freeman code representation is not appropriate. Fortunately, other techniques are available [11] that offer essentially unlimited accuracy at the cost of sampling density and computational complexity.

Sampling Density and Measurement

A perhaps surprising aspect of this result is the role played by the sampling density. One might imagine – based upon classical signal analysis – that given a bandlimited analog image the sampling frequency would only play an important role up to the Nyquist frequency. Such is not the case. Sampling densities significantly higher than what one one expect from the sampling theorem are needed for accurate measurement in images [17]. The reasons for this are two-fold:

1. The assumption that the collection of numbers in a computer memory faithfully represent the samples of a bandlimited, two-dimensional signal cannot be true, and;

2. There is such a profound difference between image filtering and image analysis that the sampling theorem is not the proper reference point for choosing a sampling density. At best the sampling theorem can be said to offer a starting point for the discussion.

The first point is easy to see in that we can only store a finite number of samples of an image (e.g. 512 x 512). This means that the (perhaps) bandlimited analog signal is being multiplied by a two-dimensional window function and thus the final result cannot be bandlimited. One could argue that perhaps the effect of the window was negligible and that the values stored in the computer contained "almost all" of the information necessary to reconstruct the original analog image. The key here is the word "reconstruct". Once again the goal is not reproducing images (image processing) but acquiring measurements (image analysis). This goal requires access to all of the image values and the reconstruction (or as it is more commonly known interpolation) formula says:

$$f(x) = \sum_{n=-\infty}^{+\infty} f[nx_0] \frac{\sin(\omega_s(x - nx_0))}{\omega_s(x - nx_0)} \tag{1}$$

A formula for extracting an exact measure from an image requires *all* of the samples and thus an infinite number of terms. Further, if that formula also requires interpolated values of $f(x)$, that is values of $f(x \neq nx_0)$, then an infinite number of sinc() functions must be evaluated to provide the correct value. Neither step can be performed in a finite number of computations.

We must therefore conclude that it is not possible to write a computer program that executes in a finite number of steps (or amount of time) that can extract an exact measure (e.g. area, perimeter) from a sampled image. *And this would be true even if the integrity of the data had not been compromised by limiting the data to a finite number of samples.*

The end result is compromise. We cannot have an exact measure *and* finite computation time. The central problem of digital image measurement then becomes one of choosing the method that offers the maximum accuracy with the minimum amount of computation.

The Area of a Circle

To illustrate this point let us consider the accurate measurement of the area of ideal circles of known diameter. The circles are digitized using a sampling density that yields S samples per diameter. For each image we assign a pixel as belonging to a circle if $(x - e_x)^2 + (y - e_y)^2 \leq D^2/4$. For each circle generated the values of e_x and e_y are allowed to vary so as to represent random placement of the circles on the digitizing grid. For each value of the sampling density S we generate 16 circles each circle differing by the choice of the random center coordinates (e_x, e_y). The distribution of the independent random variables e_x and e_y is uniform over the interval (0,1). The percentage error of the area estimate – based upon counting the number of pixels assigned to the circle – is given by:

$$\varepsilon(\%) = \frac{|\,\text{Estimated Area} - \text{True Area}\,|}{\text{True Area}} \times 100\%$$

If we then plot $\varepsilon(\%)$ as a function of the sampling density S, expressed in samples per diameter, we have the result shown in Figure 4.

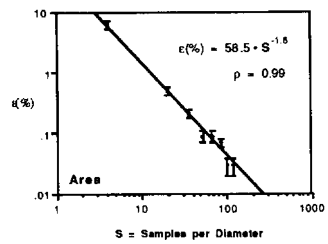

Figure 4: An ideal circle is digitized at various sampling densities. The **area** is computed by counting the number of pixels assigned to the circle. The average percentage absolute error is graphed. The error brackets indicate the standard deviation spread of the average percentage error (the standard error of the estimate).

We see from Figure 4 that a minimum of 10 samples per diameter is needed before an accuracy of 1% can be reached in the measurement of the area. In this model, of course, there are no lenses, cameras, amplifiers, etc. The image can be generated entirely within the computer. While the image and object are both of finite extent and thus not bandlimited, this does not detract from the correctness of the observation that the only way to achieve an arbitrarily high accuracy (low percentage error) is to increase the sampling density.

If we were to start with samples of a true, bandlimited image, then the act of saying which pixels belong to the object and which to the background would be equivalent to this simulation. If we were to attempt to use the information contained in the transition grey values between the object and background, then we would be led inevitably back to the (impossible) evaluation of equation (1).

Analysis Through Transformations

The image analysis techniques described above are – in an important sense – based upon the concepts of parameter estimation. An alternative approach is based upon the concept of performing transformations upon an image; transformations whose very nature lead to analysis. A simple example can help to illustrate this idea.

Nuts and Bolts

We wish to analyze the image shown in Figure 5 in such a way as to identify and separate the nuts from the bolts.

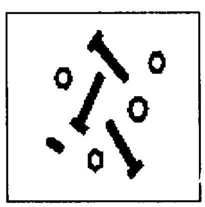

Figure 5: Image of a combination of nuts and bolts

There are several properties of these objects that we can list that suggest how this analysis can be implemented:

Nuts: small objects that may contain a hole;
Bolts: larger objects that do not contain a hole.

The *size* of an object can be estimated by the techniques described earlier or another approach can be used. We form a binary image of these high-contrast objects and then apply the binary neighborhood (Minkowski) operators. The *erosion* operation offers us an image transformation that is linked to the size of the individual binary objects in the image. One erosion iteration removes from an object the single layer of pixels connected to the background. Any pixels that remain in the image after N iterations of the erosion operation are derived from objects whose original "diameters" were greater than 2N. By "propagating" these surviving pixels back into the original image, it is easy to isolate the large objects. By "exclusive–oring" the result with the original, the small objects can be isolated.

The topological property – the object contains a hole – can be ascertained by use of the skeleton operation [18,19]. An object containing a hole will lead to a skeleton that is a closed contour. Otherwise the skeleton will not be closed. The closure, itself, can be detected by repeated application of the skeleton operation to the skeleton itself until either a single isolated pixel remains or a closed contour. Detection of this final state is simple.

The conclusion that we can draw is that a series of image transformations can lead to the analysis of an image. In this example the analysis is based upon size and topological properties. The basic concept, however, can be extended to other properties [20,21].

The Distance Transform

An important development of the past few years is the distance transform. This transform applied to binary images has proven to be an important tool in a variety of image analysis tasks. We can see this by examining the *erosion* operation defined in the previous section.

Each iteration of an erosion "peels away" a collection of pixels that have the common property that they are distance "1" from the background. The pixels that vanish in a third iteration are thus those pixels that were originally a distance of "3" from the background. This is illustrated in Figure 6.

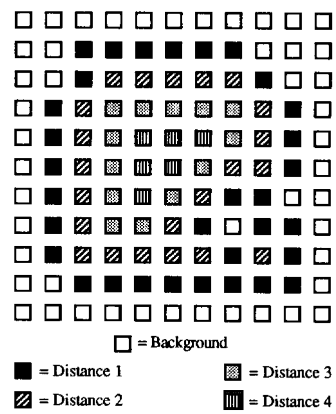

Figure 6: A binary object with the points labeled as to distance. The distance associated with a pixel may be interpreted as the number of iterations required to cause that pixel to vanish under the *erosion* operation.

In two passes through an image – the first from top–left to bottom–right and the second in the reverse direction – it is possible to transform a binary image into a distance image. Erosion may then be viewed as a simple thresholding operation on the distance–transformed image. (Note that since distances are non-negative the transform image may be viewed as a grey-value image.) Thus the distance transform provides a domain where the ordinarily neighborhood operation *erosion* can be implemented as a point operation – thresholding.

Further, the skeleton operation can be implemented on the distance transform image in an additional two passes. This is an important result in that skeletonizing is usually $O(N^3)$ where the image is N x N. Through the distance transform this becomes $O(N^2)$.

The distance transform illustrated in Figure 6 is based on specific definition of the connectivity of a point to its neighborhood: 4-connected. Had we chosen 8-connected instead the result would have been quite different. An alternative way to view this result is to specify the distance metric that is being used. Three such metrics are illustrated in Figure 7.

2	1	2
1	x	1
2	1	2

4-connected metric

1	1	1
1	x	1
1	1	1

8-connected metric

7	5	7
5	x	5
7	5	7

5,7 metric

Figure 7: Three metrics for assigning distance to points from a center point.

The (5,7) metric illustrated above has the advantage that it offers a better approximation to the Euclidean distance metric where the contour of constant distance is a circle, that is, rotation invariant. A binary image transformed through the (5,7) metric will thus have $7/5 = 1.400$ as an approximation to $\sqrt{2} \approx 1.414$. Further, the final result – whether the goal is erosion or skeletonization – will be considerably more rotation invariant as if the structuring element better approximated a true circle [22]. The net result will be that analyses – such as the Nuts and Bolts example above – will be more accurate when based upon the (5,7) metric.

In an earlier publication Young et. al. [23] showed that even at very low SNRs (approaching 0 dB), it is possible using the (5,7) distance metric to make accurate size measurements of the size of objects in an image. This is not possible, however, when the more common 4-connected (1,2) or 8-connected (1,1) metrics are used.

Summary

In this review of modern image analysis I have tried to show how image analysis differs substantially from image processing. Image analysis may be viewed as a direct measurement or parameter estimation problem or as a problem in image transformation. If the former is the case then great care must be taken in choosing the:

- Image representation
- Estimation formula
- Sampling density

If the latter is the case then care must be taken to understand and adapt the transformation for the quantization effects inherent in the digital representation of an image.

While a great many questions have now been properly formulated – and some properly answered – the challenges offered by two- and three-dimensional image analysis will continue for many years.

Acknowledgments

We would like to acknowledge the support of The Netherlands Foundation for Medical Research, MEDIGON, (Grant 900-538-016) and The Netherlands Foundation for Fundamental Research on Matter, FOM, (Grant DTN-95.0329).

References

[1] WF Bynum and JL Heilbron, "Eighteen eighty nine and all that", *Nature*, Vol. 337, pp. 29-32, 5 January 1989.

[2] WF Schreiber, "Third-order probability distribution of television signals", *IRE Transactions on Information Theory*, Vol. IT-2:3, pp. 94, 1956

[3] NG Deriugen NG, "The power spectrum and correlation function of the television signal", *Telecommunications*, Vol. 7, pp. 1-12, 1957.

[4] RPW Duin, "Interactive image processing in a multi-user environment", in: *Computing Structures for Image Processing*, Ed. MJB Duff, Academic Press, London, Chapter 8, 1983.

[5] K Preston, "Progress in image processing languages", in: *Computing Structures for Image Processing*, Ed. MJB Duff, Academic Press, London, Chapter 13, 1983.

[6] H Tamura, S Sakane, F Tomita, N Yokoya, M Kaneko, and K Sakaue, "Design and implementation of SPIDER – A transportable image processing software package", *Computer Vision, Graphics, and Image Processing*, Vol. 23, pp. 273-294, 1983.

[7] FCA Groen, RJ Ekkers, R De Vries, "Image processing with personal computers", *Signal Processing*, Vol. 15, pp. 279-291, 1988.

[8] IT Young and R Roos, "Acuity: Image analysis for the personal computer", in: *Pattern Recognition and Artificial Intelligence*, Eds. ES Gelsema and LN Kanal, Elsevier Science Publishers, Amsterdam, pp. 5-16, 1988.

[9] RS Ledley, "High speed automatic analysis of biomedical pictures", *Science*, Vol. 146, pp. 216-223, 1964.

[10] AWM Smeulders and L Dorst, "Measurement issues in morphometry", *Analytical and Quantitative Cytology and Histology*, Vol. 7:4, pp. 242-249, 1985.

[11] L Dorst and AWM Smeulders, "Length estimators for digitized contours", *Computer Vision, Graphics, and Image Processing*, Vol. 40, pp. 311-333, 1987.

[12] H Freeman, "Boundary encoding and processing", in: *Picture Processing and Psychopictorics*, (Eds. BS Lipkin and A Rosenfeld), Academic Press, New York, pp. 241-266, 1970.

[13] J Serra, *Image analysis and Mathematical morphology*, Academic Press, London, Chapter 7, 1982.

[14] FCA Groen and PW Verbeek, "Freeman code probabilities of object boundary quantized contours", *Computer Graphics and Image Processing*, Vol. 7, pp. 391-402, 1978.

[15] D Proffitt and D Rosen, "Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines of finite length", *Computer Graphics and Image Processing*, Vol. 10, pp. 318-322, 1979.

[16] AM Vossepoel and AWM Smeulders, "Vector code probabilities and metrication error in the representation of straight lines of finite length", *Computer Vision, Graphics, and Image Processing*, Vol. 20, pp. 347-364, 1982.

[17] IT Young, "Sampling density and quantitative microscopy", *Analytical and Quantitative Cytology and Histology*, Vol. 10:4, pp. 269-275, 1988.

[18] CJ Hilditch, "Linear Skeletons from square cupboards", *Machine Intelligence*, Vol. 4, pp. 403-420, 1969.

[19] C Arcelli, L Cordella, and S Levialdi, "Parallel thinning of binary pictures", *Electronic Letters*, Vol. 11, pp. 148-149, 1975.

[20] PW Verbeek, HA Vrooman, LJ van Vliet, "Low-level image processing by max-min filters", *Signal Processing*, Vol. 15, pp. 249-258, 1988.

[21] G Borgefors, "Distance transform in arbitrary dimensions", *Computer Vision, Graphics, and Image Processing*, Vol. 27, pp. 321-345, 1984.

[22] L Dorst, "Pseudo-Euclidean skeleton", Proceedings of the 8th International Conference on Pattern Recognition, Paris, France, IEEE Computer Society, pp. 286-288, 1986.

[23] IT Young, GL Beckers, L Dorst, and A Boerman, "Choosing filter parameters for non-linear image filtering", in: *Pattern Recognition in Practice II*, Eds. ES Gelsema and LN Kanal, Elsevier Science Publishers, Amsterdam, pp. 5-14, 1986.