Child's Play

Image Processing Basics: Shape Detection

Human visual perception is excellent in detecting shapes of objects. Even small children will easily differentiate between circles, squares or triangles when wooden toys or sweets are at stake. Similarly, machine vision applications on the factory floor or in the open environment sometimes call for quantitative parameters to characterize a shape. Traffic signs, e.g., may well be pre-classified by their shape, and for some pills the integrity of their shape is an important issue in quality control. This article describes some simple, well-known methods of shape detection.

A method for shape-detection has to provide a quantitative parameter as a unique means for characterizing the shape of an object, independent of its orientation, position and dimensions. Let us look at the simple, two-dimensional case: the objects are on a flat surface and are flat themselves. Optical imaging is supposed to have no influence on the shape, that is perspective, optical distortions, inhomogeneous lighting, the spatial variation of the efficiency of the lens and compression or expansion due to image acquisition are negligible or have been corrected. To keep our approach simple, let us for the time being deal with objects without holes and with good contrast to the background only. An image which might result under these conditions is shown in figure 1. A human observer will immediately be able to describe the objects by naming their shapes: rectangle, square, circle, triangle, moon, heart, ellipse. A quantitative measure for these shapes is the compactness. The basic idea behind this feature is the fact that the circle is the geometrical figure with the smallest circumference for a given area. A square is less compact than a circle with the same area, since its circumference is larger. For a circle with radius r the ratio between circumference U and area A is $2\pi r/\pi r^2$, that is $2/r$; for the square with edge length a the ratio is $4a/a^2$, that is $4/a$. This parameter is invariant under rotation, but depends upon the linear dimensions. A better choice is the ratio U^2/A, where the linear dimensions r and a, respectively, are eliminated. For the

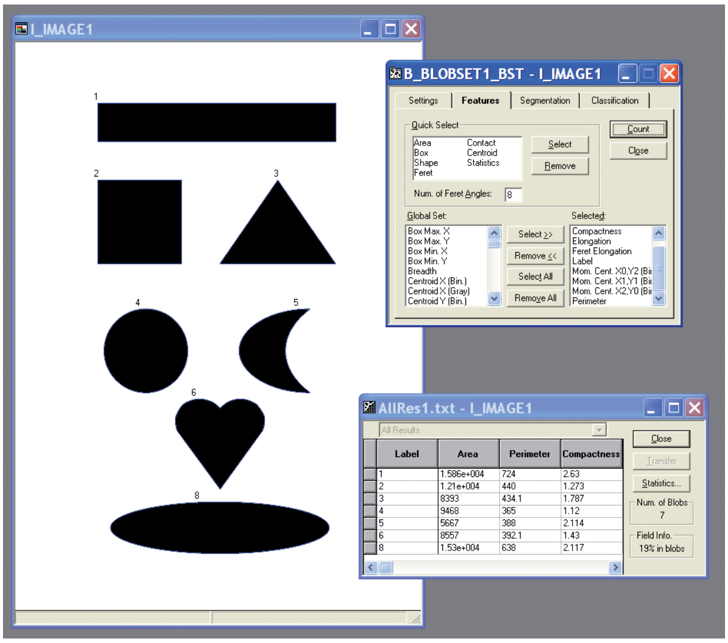


Fig. 1: Some simple shapes in a binary image and results of the blob-analysis

think big

Grasshopper®: high performance, compact size, big value

Every little Grasshopper® is full of big features. Big resolutions (**11 CCD models,** up to 5MP). Big sensitivity (**EXview HAD CCD**™). And big performance (**2MP** at **30FPS** over FireWire, 14-bit A/D). Get a big bite of value (**USD $1195 to $3195**)

SAVE $50 NOW!
Enter promo code **GRAS-03MH15** at **www.ptgrey.com/thinkbig**

POINT GREY
RESEARCH

Innovation in Imaging

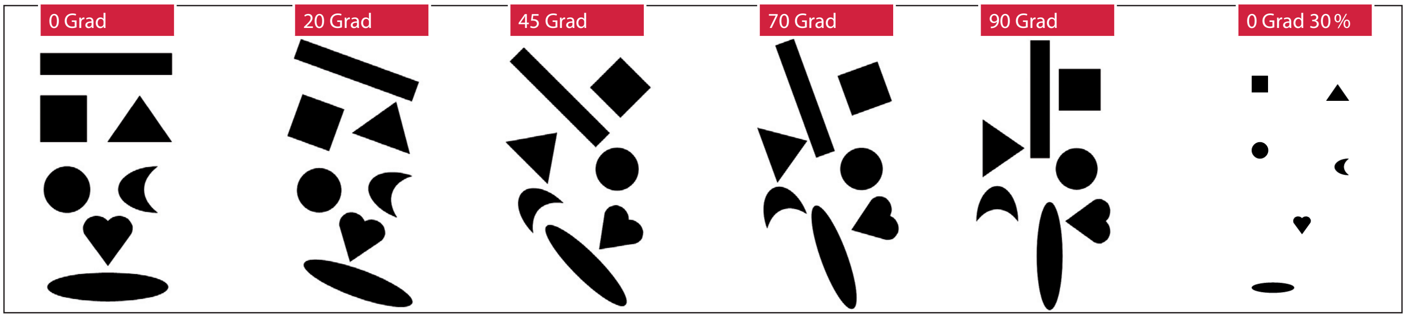


Fig. 2: The binary image from figure 1 with different orientations and scaled-down to 30%

circle, this feature has the value 4π, for the square the value 16, independent of the dimension of the object, the orientation or the position in the image.

Compactness

Most image processing libraries provide the compactness or a similar feature in their blob-analysis. Often, this shape-parameter is normalized to get the value 1 for the circle. Within the tool which was used for the blob-analysis in figure 1 the "compactness" is calculated as $U^2/4\pi A$. Other shapes will thus have a "compactness" larger than 1. Some other tools prefer the reciprocal value. Closer inspection of the result-listing in figure 1 shows that rectangle, square, triangle, circle and heart can well be differentiated by their compactness. The value for the circle, however, is significantly different from the theoretical value 1. There are two reasons for this strange effect. First, there exist different methods to calculate the circumference of an object. Some methods just take the number of pixels on the contour, some use the factor √2 to account for diagonal steps, others calculate the circumference from the chain-code as the total length of the lines connecting the centers of the contour pixels. Similar differences exist for the calculation of the area. An extreme example shows the problem: for an object which consists of a single pixel only, one method may yield the value 1 for the area and 4 for the circumference, another method may result in 0 for the area and 0 for the circumference. Second, every object in the digital plane is made up of pixels, it is discrete. Circles are always approximated as polygons. Therefore, features describing the area and the circumference of an object may show deviations from the ideal values. Both effects will have an influence when the compactness is calculated. The discrete nature of the digital plane will usually be more important for smaller than for larger blobs or relevant structures. An-

Table 1: Compactness and normalized moment of inertia for the objects in figure 2

0 Grad	U^2/A	Iz/A^2	$100Iz/A^2$
Rechteck	33,05	0,5217	52,17
Quadrat	16,00	0,1667	16,67
Dreieck	22,45	0,1956	19,56
Kreis	14,07	0,1592	15,92
Mond	26,57	0,2168	21,68
Herz	17,97	0,1747	17,47
Ellipse	26,60	0,3523	35,23
20 Grad			
Rechteck	38,40	0,5199	51,99
Quadrat	18,54	0,1665	16,65
Dreieck	23,69	0,1955	19,55
Kreis	14,18	0,1592	15,92
Mond	26,46	0,2168	21,68
Herz	18,16	0,1747	17,47
Ellipse	27,58	0,3536	35,36
45 Grad			
Rechteck	33,13	0,5229	52,29
Quadrat	15,96	0,1666	16,66
Dreieck	22,32	0,1956	19,56
Kreis	14,07	0,1592	15,92
Mond	26,46	0,2165	21,65
Herz	17,91	0,1746	17,46
Ellipse	26,75	0,3550	35,50
70 Grad			
Rechteck	38,46	0,5195	51,95
Quadrat	18,64	0,1667	16,67
Dreieck	23,90	0,1956	19,56
Kreis	14,06	0,1592	15,92
Mond	26,57	0,2163	21,63
Herz	18,18	0,1747	17,47
Ellipse	27,72	0,3538	35,38
90 Grad			
Rechteck	33,05	0,5217	52,17
Quadrat	16,00	0,1667	16,67
Dreieck	22,27	0,1955	19,55
Kreis	14,07	0,1592	15,92
Mond	26,55	0,2163	21,63
Herz	18,02	0,1748	17,48
Ellipse	26,71	0,3529	35,29
0 Grad 30 %			
Rechteck	32,18	0,5035	50,35
Quadrat	16,00	0,1666	16,66
Dreieck	21,93	0,1954	19,54
Kreis	14,41	0,1591	15,91
Mond	26,15	0,2178	21,78
Herz	18,32	0,1741	17,41
Ellipse	26,29	0,3433	34,33

other interesting observation in the result-listing of figure 1 is the striking similarity between the compactness of the moon-like object and the ellipse. For an application in industrial image processing or a machine-vision task the stability of the features is always an issue and has to be carefully checked to meet the requirements of the task. Since the circumference is sensitive for noise in the grey-level signal, we may well make the educated guess that the compactness will not provide a safe basis to distinguish between these two shapes in an application on the factory floor.

Stability

The sequence of images in figure 2 and the values for the features in the corresponding table 1 gives a first impression of the stability of the parameters. The shapes in figure 1 have been rotated by an angle of 20°, 45°, 70° and 90° with respect to their original orientation. A further image shows the same objects scaled down to 30 % of their original size. In figure 3 two of the shapes, heart and ellipse, are shown in arbitrary orientations and dimensions. The corresponding features are listed in table 2. As a measure for compactness the term U^2/A has been calculated. The values for U and A have been taken from the blob-analysis provided by the tool used in figure 1. For some shapes the orientation seems to have a large effect on the value for the compactness. The compactness of the square, e.g., varies between 16.0 and 18.6, which amounts to a relative range of 15 %. The same range appears for the rectangle. The results also confirm that the shapes "moon" and "ellipse" can not be properly distinguished by compactness. Values for the ellipses range between 26.60 and 27.72, for the moon-like object between 26.46 and 26.57. In addition, for the image where the objects have been scaled down to 30 %, ellipse and moon have compactness-values of 26.29 and 26.15, respectively. Thus, when

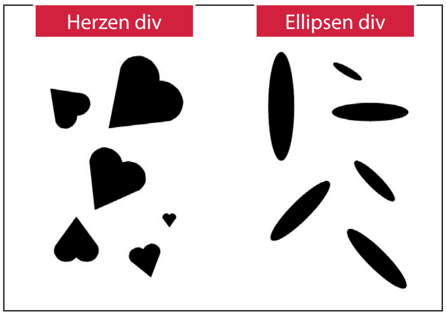


Fig. 3: Hearts and ellipses with arbitrary orientations and dimensions

Table 2: Compactness and normalized moment of inertia for the objects in figure 3

Herzen div	Fläche A	Umfang U	U^2/A	Iz/A^2	$100Iz/A^2$
1	24970	668,7	17,91	0,1748	17,48
2	7403	362,9	17,79	0,1745	17,45
3	15690	532,3	18,06	0,1747	17,47
4	801	118,7	17,59	0,1740	17,40
5	8549	392,9	18,06	0,1747	17,47
6	4405	281,8	18,03	0,1745	17,45
Ellipsen div					
1	15300	636,9	26,51	0,3512	35,12
2	1417	196,0	27,11	0,3456	34,56
3	7586	449,4	26,62	0,3510	35,10
4	4044	328,4	26,67	0,3540	35,40
5	8667	481,6	26,76	0,3541	35,41
6	8673	482,1	26,80	0,3536	35,36

orientation and dimension are changed independent of each other, the intervals for the compactness of ellipse and moon will overlap. Figure 3 and the corresponding table 2 show that the situation for hearts and ellipses is much better. The relative range for the compactness is about 2.5 % only and will add up to about 5 %, when the values from figure 2 are also taken into account, but without any reasonable risk of overlap.

Moment of Inertia

Ellipse and moon may well be distinguished from each other by means of another, well-known feature for shape detection: the moment of inertia. Moments are statistical parameters. In general, the moment m_{pq} of a cloud of pixels with coordinates (x, y) is calculated as $\Sigma x^p y^q$; p and q are integers. The moment m_{00}, e.g., is equal to the number of pixels in the foreground and usually is taken as a measure for the area. The moments m_{02} and m_{20} have the structure Σy^2 und Σx^2; they are similar to the moment of inertia with respect to the x-axis and the y-axis, respectively. For shape-detection, these moments are first calculated with respect to the centre of mass (x_s, y_s) of an object („centered"), thus having the form $\Sigma(x\text{-}x_s)^2$ and $\Sigma(y\text{-}y_s)^2$, respectively. These are the moments of inertia I_y and I_x, respectively, with respect to an axis parallel to the y- and x-axis, respectively, through the centre of mass of the object. A circle has the same moment of inertia for both directions. An elongated ellipse with the semi-major axis parallel to the x-axis like in figure 1 has a small moment of inertia with respect to the x-axis and a large moment of inertia with respect to the y-axis. The sum of both moments is invariant under rotation and corresponds to the moment of inertia I_z of the object with respect to an axis perpendicular to the image plane. Normalizing the moments, in this case to the square of the area, yields a feature independent of the dimensions of the object. Such parameters are called normalized, centered moments. The feature I_z/A^2 is listed in table 1 and 2. The data show that the moon-like shape and the ellipse can be distinguished by this parameter, even when variations of orientation and dimensions occur. For the other shapes this feature also is quite stable in comparison with the compactness. The absolute numerical differences between the values seem to be small for some shapes, but the stability of this feature against rotation and scaling is good enough to take it into account for shape-detection even in these cases. The combination of several different centered, normalized moments leads to the construction of further features suited for shape-detection [1]. By weighting with the grey-level of an object-pixel the concept of moments may even be applied to grey-level images. And another advantage of moments compared to the features generated by blob-analysis may be of importance: moments may be calculated for arbitrary groups of pixels, whether they are connected or not, whereas blob-analysis always needs binary objects made up of closely connected pixels.

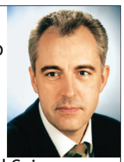
Conclusion

Shape-features have to be invariant under rotation, translation and scaling. Shape detection is not a simple task, even with two-dimensional objects and optimum conditions for image acquisition. The discrete nature of the image plane may lead to significant deviations from the ideal values for simple shape indicators such as the compactness. These features, however, may be well suited when only a few, well-defined shapes can appear in the application and when the stability of the shape-parameters has been carefully evaluated. For shape features directly taken from image processing libraries it might be a good idea to look at the details of the methods used. Combinations of normalized, centered moments may show very good performance as shape-parameters. In addition, these features may be calculated for arbitrary clouds of pixels and are not restricted to binary objects as in blob-analysis.

Reference

[1] R. C. Gonzales, R. E. Woods, Digital Image Processing, Addison-Wesley, 1993, p. 516

Author
Prof. Dr. Christoph Heckenkamp



Contact
Darmstadt University of Applied Sciences
Darmstadt, Germany
Department of Technical Optics and Machine Vision
heckenkamp@h-da.de
www.fbmn.h-da.de