2D and 3D Shape Descriptors

Department of Computer Science

University of Exeter

Submitted by Carlos Martinez-Ortiz, to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science, September 2010.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Carlos Martinez-Ortiz

Abstract

The field of computer vision studies the computational tools and methods required for computers to be able to process visual information, for example images and video. Shape descriptors are one of the tools commonly used in image processing applications. Shape descriptors are mathematical functions which are applied to an image and produce numerical values which are representative of a particular characteristic of the image. These numerical values can then be processed in order to provide some information about the image. For example, these values can be fed to a classifier in order to assign a class label to the image.

There are a number of shape descriptors already existing in the literature for 2D and 3D images. The aim of this thesis is to develop additional shape descriptors which provide an improvement over (or an alternative to) those already existing in the literature.

A large majority of the existing 2D shape descriptors use surface information to produce a measure. However, in some applications surface information is not present and only partially extracted contours are available. In such cases, boundary based shape descriptors must be used. A new boundary based shape descriptor called Linearity is introduced. This measure can be applied to open or closed curve segments.

In general the availability of 3D images is comparatively smaller than that of 2D images. As a consequence, the number of existing 3D shape descriptors is also relatively smaller. However, there is an increasing interest in the development of 3D descriptors. In this thesis we present two basic 3D measures which afterwards are modified to produce a range of new shape descriptors. All of these descriptors are similar in their behaviour, however they can be combined and applied in different image processing applications such as image retrieval and classification. This simple fact is demonstrated through several examples.

Contents

List of Figures

List of Tables

Acknowledgements

This work was only possible thanks to the funding provided by the Mexican Science Council (Consejo Nacional de Ciencia y Tecnología, CONACyT) through the scholarship number 205662.

I want to thank to all the people who have supported me in so many ways to complete this work: to my supervisor Dr. Joviša Žunić for his extreme patience throughout my PhD; to all my friends and colleagues for their academic support; to my family and friends in the UK, in Mexico and to those scattered around the globe for their love and constant support. And above all, thanks to my parents for their guidance, support and unconditional love through all these years: porque sin ustedes nada de esto sería posible, muchas gracias.

Special thanks to all my volunteer proof readers: Jacqueline Christmas, Annick Pijnenburg, Adji Sjadzali, Marta Söffker and Nivritti Tandan, who willingly gave up their time to read this work and provide valuable comments.

Permission has been granted by Quipos s.r.l. to use the *La Linea* images shown on Figure 3.5 (c) and (f) in Chapter 3.

Chapter 1

Introduction

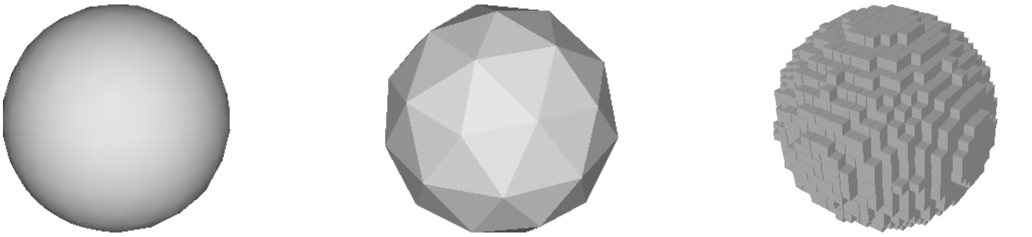
It is a common saying that *a picture is worth a thousand words*. It is true that visual information is very useful in communicating ideas and certainly humans rely heavily on sight to perceive information regarding the physical world surrounding us. The way in which the human brain processes visual information has been a topic of study for many years. Indeed the way in which humans are able to recognise an object or identify people in a fraction of a second is remarkable. It would be desirable for computers and robotic applications to be able to mimic these capacities.

The field of computer vision investigates the algorithms required to eventually be able imitate human vision. Such algorithms would provide computers with the means to collect information and learn about the world around them. In this way computers could eventually understand their surroundings in a very similar way as humans do. The fields of application for computer vision are diverse: any human activity could potentially benefit from it.

1.1 Background

An image is a two dimensional function $f(x, y)$ of colour intensity (or grey level) on a plane. For a digital computer to be able to process an image, the (x, y) coordinates of any given point in such image must be discrete values. The image is then a discrete grid of picture elements (pixels) which also have discrete values. This is known as a digital image. Regardless of the method used to acquire a digital image (e.g. digital camera, scanner, etc.) images must be quantised (i.e. continuous values must be converted to integer values) which causes an inherent loss of information. Higher resolution acquisition instruments (e.g. high resolution cameras) may help reduce the error induced by the quantisation process, but will never fully eliminate it. The loss of information during digitisation is a well known problem and has been widely studied in the literature. There are other alternatives to pixel based representation, like vector graphics, which represent images as points, lines and curves that have well defined mathematical equations.

When dealing with 3D objects, there are again several ways in which these may be represented. One possible representation, analogous to pixel representation, is a voxel (volume element) representation. In this case a 3D binary object would be represented by a 3D matrix of zeroes and ones which indicate the presence or absence of the object in a particular location of a regular discrete grid. The voxel representation of an object has similar limitations to the pixel representation of an object, being susceptible to quantisation error and dependant on the resolution of the voxel grid. Another way of representing a 3D object is by using a polygonal mesh representation. A polygonal mesh is composed of a set of vertices, edges and faces joined to form a 3D object. Figure 1.1 shows a sphere with 2 different possible representations.



(a) A sphere (b) Polygonal mesh representation (c) Voxel representation

Figure 1.1: A *real* sphere (a) with two different representations: (b) a polygonal mesh representation and (c) a voxel representation. Notice that there are infinitely many *real* 3D objects which would have the same voxel representation.

There is a variety of representations which can be used for 2D and 3D images. Some information may be easier to obtain depending on the representation used, for example computing the volume of a voxelized 3D object is simpler than using a polygonal mesh. However it is worth emphasising the importance of independence between the object and its representation. The *true nature* of an object's shape should not be constrained by its representation. We should always be concerned with the shape of the object we are dealing with and not with the limitations that its current representation imposes on our understanding of the shape. For example, the spheres in Figure 1.1 have different representation but they are, in essence, the same shape: a sphere. The fact that they are expressed in different formats is completely unrelated with the shape they represent. *What's in a name? That which we call a rose by any other name would smell as sweet*[1]. Likewise, that which we call a sphere in any other representation is still a sphere.

The several image representations methods are ways of expressing visual data. This data still needs to be processed in a way which is useful for a specific purpose. This is what we call image processing application. Generally speaking most image processing applications would have the following elements:

- Acquisition – The first thing that is required is to capture a *real* image and transform it into a digital representation. This is commonly done by means of digital cameras, but other acquisition techniques can also be used (e.g. scanners, other types of sensors) or even be constructed artificially (e.g. using graphics editing software).

- Pre-processing – The digital image must be prepared to be used at a later stage. At this stage, the representation of the image may be modified in order to discard unimportant information and preserve only information which may be of use at later stages. Tasks involved in the pre-processing stage are diverse, including but not limited to: image enhancement, grey scale conversion, object segmentation, binarization, boundary detection, polygonal approximation, normalisation, etc.

- Feature extraction – This is the process of obtaining information from the object in question, highlighting its features of interest. These features or descriptors may be obtained from the object's boundary, surface, texture, or any other characteristic. The features of an object are typically represented in a quantitative form which allows for comparison between objects. Such features can be extracted by using computational methods called *shape descriptors.*

- Further processing – Once the relevant information has been extracted from the image we may use this information to perform the task which the application aims to achieve, for example, recognition, classification, similarity search, etc.

When dealing with 3D imagery, the processing stages involved are very similar. However acquisition of 3D models is not yet as well developed as it is for 2D images. Several 3D model acquition methods are currently available in the literature. One alternative is scanning real objects using laser scanners, which normally produce a

[1] From Shakespeare's *Romeo and Juliet*

voxel representation of the object. Another technique is construction of 3D models from 2D images, using 2D sections of a 3D object (Bors et al., 2002). Yet another method is construction of 3D models from stereo images, however this is a topic of ongoing research (Grum and Bors, 2008; Kim and Sohn, 2005). Probably the most common practise at the moment is to create 3D models using 3D modelling software, like Blender[2]. There is a wide range of free and commercial CAD/CAM software packages available which makes 3D modelling generally available to end users.

1.2 Shape descriptors

Shape descriptors are computational tools used for analysing image shape information. Shape descriptors consist of mathematical functions which when applied to an image produce numerical values. These numerical values are representative of a specific characteristic of the given shape. The nature and meaning of such values depends on the definition of the shape descriptor. After shape features have been extracted, they can be used as input features for an image processing application. For example, for a shape classification application, a vector of shape descriptors can be fed to a classifier to determine the class membership of the given image. More details on shape descriptor applications are given later in this section (see page 24).

Generally speaking we could classify shape descriptors depending on the information they take into account to compute their measures. In 2D, shape area and boundary (perimeter) are the two major sources of information, so we could speak of area based descriptors and boundary based descriptors. However it is important to point out that some descriptors may use a combination of boundary and interior points to compute their measures.

Both boundary based and area based approaches have particular characteristics which could be considered to be advantages or disadvantages depending on the particular needs of each application.

Boundary based methods are generally sensitive to small changes on the shape; this usually causes these methods to produce different results when the shape boundary changes slightly. This means that boundary based methods can perceive small changes in the shape. However this also means that such methods can be affected by the presence of noise in the shape, producing undesirable results.

On the contrary, area based methods are not sensitive to such small changes on the shape; even large changes on the boundary have a small effect on the shape area.

[2] http://www.blender.org/

This means that area based descriptors are not greatly affected by the presence of noise, thus they are robust with respect of noise. However, this also means that they are unable to perceive small variations on the shape, which implies that shape details are ignored.

In 3D, volume and surface area are the two major sources of information, so we could speak of volume based and surface based descriptors, also pointing out some descriptors may use a combination of the two.

Surface area is analogous to boundary, implying that surface based descriptors are more sensitive to small changes but more susceptible to noise. Likewise volume based descriptors are analogous to area based descriptors, implying they are robust to noise, but insensitive to small shape details.

There are some properties which are desirable for shape descriptors in general, although they are not, strictly speaking, requirements for all shape descriptors. The following is a list of some of these desirable properties which by no means intends to be extensive:

- Rotation invariance – The rotation of an object does not affect its shape; therefore it would be expected that a shape descriptor should produce the same measure for a shape S and for the same shape rotated by θ degrees, $R(S, \theta)$,

- Translation invariance – The shape of an object is independent of the coordinate axes used; therefore it would be expected that a shape descriptor should produce the same measure for the shape S regardless of its location in the coordinate plane.

- Scale invariance – Because the shape of an object is independent of its representation, the scale of an object should not affect the measure produced by a shape descriptor.

- Well defined range – Having an idea of the range of values produced by a shape descriptor can be important when interpreting the meaning of the values produced by the descriptor. Also it may be useful to know the range produced by a descriptor in particular when designing an application (in case normalisation is required).

1.2.1 Existing 2D descriptors

Over the years there has been a continuous interest in shape descriptors. Many different approaches have been taken and there is a large existing body of available literature. This section provides the reader with a brief survey of some of the methods considered most relevant to this thesis.

One of the most well known shape descriptors are moment invariants introduced by Hu (1962) which have been used extensively in the literature on their own and as basis to construct other shape descriptors. The basic definition of 2D moments of $(p+q)$-th order is as follows:

$$m_{p,q}(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy, \tag{1.1}$$

where $p, q \in \mathbb{Z}^+$ and $f(x,y)$ is the intensity function of the image. If the given image is a binary image representing a shape S, we can simplify the definition of moments to the following expression:

$$m_{p,q}(S) = \iint_S x^p y^q dxdy, \tag{1.2}$$

In order to attain translation invariance, moments are translated to the centroid of the shape. These are called central moments $\mu_{p,q}$ and are defined as:

$$\mu_{p,q}(S) = \iint_S (x-\bar{x})^p (y-\bar{y})^q dxdy, \tag{1.3}$$

where $(\bar{x}, \bar{y})$ is the centroid of the shape:

$$C = (\bar{x}, \bar{y}) = \left(\frac{m_{1,0}(S)}{m_{0,0}(S)}, \frac{m_{0,1}(S)}{m_{0,0}(S)}\right). \tag{1.4}$$

From these, Hu derives a set of invariants which are independent of position, size and orientation which have been widely used in the literature.

Moments as defined in equation (1.2) are computed from shape area information. However moments can also be computed from shape boundary information (Lambert and Gao, 2006). These are referred to as line moments. In order to compute line moments, the boundary of any given shape S has to be expressed in parametric form.

First lets introduce the parametric representation of a given shape's contour. Let B be the boundary of a given shape S and let us start by assuming that B is an open curve. Also let P be the length of B. We then can refer to the coordinates of any point of $B(s) = (x(s), y(s))$ for $0 \leq s \leq P$. Of course if B is a closed curve it only means that $B(0) = B(P)$.

An easy way to understand contour parametric representation is to think about an athletics track: runners run a distance of 400 meters ($0 \leq s \leq 400mts$) although the distance is not measured in a straight line, but rather the distance travelled on the track; in hurdle races, the first hurdle is placed after 45 meters (i.e. $B(45mts)$) and then every 35 meters after (i.e. $B(80mts), B(115mts), \ldots$). For further detail on contour parametric representation the reader is referred to Kindratenko (1997).

Line moments can be calculated for open curve segments and partially extracted boundaries. Line moments are defined as follows:

$$m_{p,q}^{(l)}(B) = \int_B x(s)^p y(s)^q ds \tag{1.5}$$

where the shape boundary B is given in parametric form and $s \in [0, length(B)]$.

Central moments can also be derived from the boundary information:

$$\mu_{p,q}^{(l)}(B) = \int_B (x(s) - \bar{x}^{(l)})^p (y(s) - \bar{y}^{(l)})^q ds, \tag{1.6}$$

In this case $(\bar{x}^{(l)}, \bar{y}^{(l)})$ the centroid is obtained from boundary information:

$$(\bar{x}^{(l)}, \bar{y}^{(l)}) = \left(\frac{m_{1,0}^{(l)}(B)}{m_{0,0}^{(l)}(B)}, \frac{m_{0,1}^{(l)}(B)}{m_{0,0}^{(l)}(B)} \right). \tag{1.7}$$

Moments are widely used in literature, for different 2D and 3D applications and for the creation of shape descriptors. Moments can be used, for example, to detect the principal axes of given shapes (Tsai and Chou, 1991) (this will be discussed in more detail in Chapter 2) or for object watermarking (Bors, 2006).

An example of shape descriptors derived from moment invariants is the circularity descriptor developed by Žunić et al. (2010b). They use second order centralised moments to derive their measure, by showing that the quantity

$$\frac{\mu_{2,0}(S) + \mu_{0,2}(S)}{\mu_{0,0}(S)^2}$$

reaches its minimum if and only if the given shape S is a circle. This shape descriptor gives a measure of similarity between a given shape and a perfect circle – this is also referred to as compactness. In the literature there is a common definition of compactness as a ratio between the perimeter and area of a shape:

$$\mathcal{C}_{Std}(S) = \frac{4\pi \cdot Area(S)}{Perimeter(S)}. \tag{1.8}$$

The circularity measure introduced by Žunić et al. is defined as follows:

$$\mathcal{C}(S) = \frac{1}{2\pi} \cdot \frac{\mu_{0,0}(S)^2}{\mu_{2,0}(S) + \mu_{0,2}(S)}. \tag{1.9}$$

The descriptor is modified by an additional tuning parameter β which modifies the behaviour of the descriptor. The modified circularity measure is defined as follows:

$$\mathcal{C}_\beta(S) = \begin{cases} \dfrac{1}{(\beta+1)\pi^\beta} \cdot \dfrac{\mu_{0,0}(S)^{\beta+1}}{\iint\limits_S (x^2+y^2)^\beta dxdy} & \beta > 0, \\ \dfrac{(\beta+1)\pi^\beta}{1} \cdot \dfrac{\iint\limits_S (x^2+y^2)^\beta dxdy}{\mu_{0,0}(S)^{\beta+1}} & \beta \in (-1,0). \end{cases} \tag{1.10}$$

An image can be thought of as a particular type of signal, and as such image processing is a branch of the broader field of signal processing. In general, signals can be decomposed into a linear combination of simpler signals, for instance wavelets. Wavelets are localised oscillation signals which typically look like *small waves*. Wavelets provide a powerful tool for image decomposition and analysis. Such analysis can be performed at various resolution levels. This is usually called *multiresolution analysis* and has the additional advantage that some details which are overlooked at low level resolutions, might become more evident in higher levels of resolution (Meyer, 1993).

Wavelets are a very popular tool in image processing. Several extensions to wavelets have been developed over the years. An example of these is the contourlet transform developed by Do and Vetterli (2005). The contourlet transform uses directional filters to decompose the image contours. It was used by Chen and Kgl (2010) for palmprint classification.

Other examples of shape descriptors are: the sigmoidality descriptors introduced by Rosin (2004), the ellipticity, rectangularity, and triangularity descriptors introduced by Rosin (2003) and the elongation descriptor introduced by Stojmenović and Žunić (2008).

1.2.2 Existing 3D descriptors

In the past, the use of 3D media was very restricted due to the computational power required to process this type of media. With the continuous increase of computational power available to end users, this restriction seems to be fading away. As a consequence of this, in recent years there has been an increase in the use of 3D media for different purposes: medical applications, CAD/CAM applications, museum catalogues, entertainment, etc. This appears to be an ongoing trend and the use of 3D media is likely to continue growing in the foreseeable future.

With this growth in the use of 3D media comes an increasing need for 3D search, matching and classification applications. 3D shape descriptors provide a powerful and useful tool for such applications. Thus, in recent years there has been a growing interest in developing new and more efficient 3D shape descriptors; some of them come as an extension of existing 2D descriptors while others have been specifically developed for the 3D domain.

Moment invariants are one of the most popular shape descriptors in 2D and extending them to 3D is straight forward (Sadjadi and Hall, 1980). In fact it has been shown by Mamistvalov (1998) that moment invariants can be extended to n dimensions. The basic definition of 3D moments of $(p+q+r)$-th order is as follows:

$$m_{p,q,r}(f) = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} x^p y^q z^r f(x,y,z)dxdydz, \tag{1.11}$$

where $p,q,r \in \mathbb{Z}^+$ and $f(x,y,z)$ is the intensity function of the 3D image. Notice that this is analogous to the definition of 2D moments in equation (1.1). Again we can assume the 3D image to be a binary image representing a shape S and simplify the definition to:

$$m_{p,q,r}(S) = \iiint\limits_{S} x^p y^q z^r dxdydz, \tag{1.12}$$

Also analogously to equation (1.3), central moments $\mu_{p,q,r}$ are defined as:

$$\mu_{p,q,r}(S) = \iiint\limits_{S} (x-\bar{x})^p (y-\bar{y})^q (z-\bar{z})^r dxdydz, \tag{1.13}$$

where $(\bar{x},\bar{y},\bar{z})$ is the centroid of the shape is defined analogous to equation (1.4):

$$(\bar{x},\bar{y},\bar{z}) = \left(\frac{m_{1,0,0}(S)}{m_{0,0,0}(S)}, \frac{m_{0,1,0}(S)}{m_{0,0,0}(S)}, \frac{m_{0,0,1}(S)}{m_{0,0,0}(S)}\right). \tag{1.14}$$

In order to reduce the computational complexity involved with computing moments, some algorithms for efficiently computing 3D moments have been developed. For 3D shapes represented as a set of voxels Yang et al. (1997) have developed an algorithm that computes moments as a summation over the discrete surface enclosing the region. Furthermore, their algorithm can be applied to n-dimensional discrete regions. Sheynin and Tuzikov (2001) presented an explicit formulae for computing moments of 3D polyhedra, which is useful for 3D models represented as a polygon mesh.

One 2D descriptor which is easily extended to 3D is compactness (circularity). In a similar manner as the ratio between area and perimeter is used to compute compactness in 2D (see equation (1.8)), a ratio between surface area and volume is commonly used in the literature to compute compactness of 3D shapes. This measure of compactness is defined as follows:

$$\mathcal{C}_{Std}^{3D}(S) = \frac{36\pi \cdot Volume(S)^2}{Surface(S)^3}. \tag{1.15}$$

Another approach is the measure of discrete compactness introduced by Bribiesca (2000, 2008). This descriptor takes into account the volume of the object (number of voxels) and the area of the enclosing surface, i.e. the number of voxel faces which are not directly in contact with another voxel. This compactness measure is defined as follows:

$$\mathcal{C}_d = \frac{n - A/6}{n - \sqrt[3]{n^2}} \tag{1.16}$$

where n is the total number of voxels and A is the area of the enclosing surface.

In 2D Fourier analysis can be applied to extract features of a shape. Similarly in 3D Fourier analysis can be used for shape description. A 3D Discrete Fourier Transform (3D-DFT) was used by Vranić and Saupe (2001) as a shape descriptor: the 3D-DFT was applied to a voxelized 3D model to produce feature vectors which were later used to search in a database for similar objects. The developed descriptor is invariant with respect to translation, rotation, scaling and reflection. It is also robust with respect to level of detail.

1.3 Shape descriptor applications

The applications where visual information needs to be processed are as complex as they are diverse. As it has been mentioned before, shape descriptors are useful tools for extracting features from visual information and can be of aid for this kind of applications. Shape descriptors have been successfully used in a wide range of applications from medical applications to search engines. In this section I mention a small collection of examples where shape descriptors have been implemented.

1.3.1 2D applications

In biology, the study of animal movement in their geographical space is of great importance. The differences in the paths animals follow gives specialists information on animal behaviour (Nams, 2006). In the study conducted by Benhamou (2004) the author utilises 3 shape descriptors (straightness, sinuosity and fractal dimension) to determine the behaviour of an animal. A higher number of changes in the direction of the path followed by the animal (i.e. a less straight path) indicates that the animal might be performing a random search. Imre (2009) used fractal analysis to study the tortuosity of animal paths highlighting the importance of chronological information being recorded as part of the path.

Several example classification applications for the circularity measure are given by Žunić et al. (2010b). The first example uses circularity measure to classify mammograms achieving up to 90.74% accuracy for classification of circumscribed/spiculated classes. The second example uses circularity to classify galaxies in two groups: spiral and elliptical. Despite not being able to reach the same accuracy as using image texture the results obtained are good, taking into account that only one circularity measure was used; the combination of more measures may increase the classification rates obtained. The third and final example uses circularity to measure print quality: a linear combination of nine circularity measures $\mathcal{C}_\beta(S)$ (see equation 1.10) is used and the average correlation coefficient obtained is 0.77 which is favourable according to Žunić et al.

The work presented by Gouet-Brunet and Lameyre (2008) uses a set of local and global descriptors for object tracking in video sequences. Local descriptors are used to identify points in the images which have interesting properties. Global descriptors provide high level description of the object shape. As a result, global descriptors can be used without previous image segmentation. In this way object segmentation in video sequences is attained, even if the object in question is partially occluded.

1.3.2 3D applications

Some applications of 3D shape descriptors can be found in medicine. The object classification framework presented by Atmosukarto et al. (2010) has proved its usefulness for medical applications. The frameworks extracts low level features from a given 3D shape and identifies points of interest on the object's surface. The classifier learns the features of these points of interest and maps them to a 2D plane. The 2D is then used to train a classifier which is used on craniofacial data to detect 22q11.2 deletion syndrome[3] and deformational plagiocephaly[4], obtaining classification accuracy comparable to that of medical experts.

Probably the most common application of 3D shape descriptors is constructing

[3] Also known as DiGeorge Syndrome, this is a disorder caused by the deletion of a small piece of chromosome 22. Symptoms may include mild differences in facial features. http://www.medic8.com/genetics/

[4] Also known as Flathead Syndrome, this is a condition characterised by a flattening on one side of the back of the head. http://www.londonorthotics.co.uk/plagiocephaly/deformational.html

3D model retrieval applications (i.e. 3D search engines). A comprehensive survey and comparison of various feature based methods for 3D retrieval was presented by Bustos et al. (2005). The authors highlight the importance of rotation, translation, reflection and scale invariance for the shape descriptors used to create feature vectors. The need for robustness with respect to level of detail and geometry is also highlighted. The authors also point out two criteria to be considered to evaluate 3D search engines: effectiveness, i.e. retrieve models which are most relevant to the given search; and efficiency, i.e. each query to the database should be responded as quickly as possible.

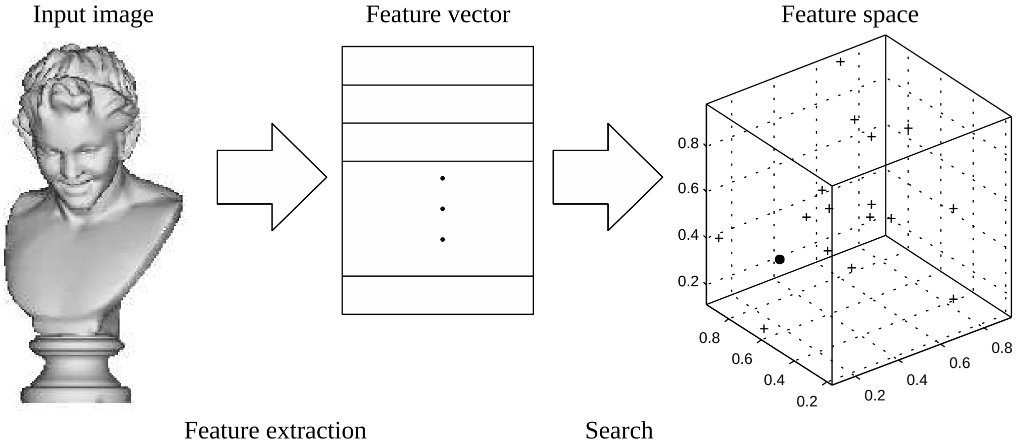


Figure 1.2: Structure of a typical image retrieval system. An image or 3D model is taken as input; relevant features are extracted using shape descriptors and collected in a feature vector; neighbouring objects in the feature space are considered to be similar to the input image.

Figure 1.2 shows the basic design of typical 3D retrieval applications: an image (3D model) is taken as an input to the system; a feature extraction process is applied to the image; a feature vector is formed using quantitative descriptors which are representative of the shape; the feature vector is used to search for other shapes in the feature space which are *close by*, under the assumption that they will be similar to the input image.

A 3D model retrieval based on spatial structure was developed by Gao et al. (2010). This 3D model retrieval system uses 2D images to describe spatial information of 3D models. In this way every 3D model is represented by a set of 2D images, which are called spatial structure circular descriptor images (SSCD images). Histogram information is then used to compare SSCDs and find similarities between 3D models.

Another 3D model retrieval which is based on 2D view point images was developed by Li et al. (2010). This retrieval system uses a set of 2D view point images to represent each 3D model. Different number of view point images and camera settings are used to create the image set for each 3D model. A quasi-histogram from each of the view is converted into a sequence of states and a Markov Chain is utilised for modeling the views from each 3D model. In order to convert the quasi-histogram into a sequence of states, each bin of the histogram is treated as an independent random variable with exponential distribution. Once the 3D objects have been characterised by the Markov Chain, Li et al. used the Kullback-Leibled divergence to determine the distance between two objects. These distances are then used to retrieve the 3D objects from the database.

The 3D model retrieval system developed by Vranić and Saupe (2001) is based on a descriptor which uses 3D Fourier Transform. The 3D model is characterised by feature vectors extracted using the 3D DFT. Similar objects are represented by neighbouring points in the feature space. The method proposed improves the performance of the retrieval process over previous approaches.

A descriptor based on Fourier series (FSD) was presented by Lmaati et al. (2010) and used to develop a 3D search engine. 3D models are represented by means of 3D closed curves and feature vectors are extracted using the Fourier series. The FSD are invariant with respect of translation, rotation, flipping and scaling, and also robust to noise and level of detail. Similar to the model retrieval presented by Vranić and Saupe, the feature vectors are used to search the database for similar 3D objects.

The 3D search engine developed by Funkhouser et al. (2003) (a Google for 3D models, as the authors call it) provides several methods for performing searches: 3D sketches, 2D sketches, 3D models and text keywords. Spherical harmonics are used to measure similarity between objects when a 3D model is provided as a query. The search engine is fast enough to be used for web based searches. The results presented by the authors suggest that a combination of shape and relevant text can provide positive results for 3D object searches.

When the canonical orientation of an object is unknown, principal component analysis (PCA) is normally used to determine the orientation of an object; this is known as pose normalisation. The shape descriptor presented by Lian et al. (2009) can be used for pose normalisation as well, in many instances performing better that PCA. Both methods are combined into an algorithm which achieves better accuracy.

One area where 3D applications have a great potential is in medicine. The shape descriptor presented by Feng et al. (2008) uses the relative angle distribution at each vertex point of a 3D surface (represented in a mesh format) and establishes a point correspondence between two models. The relative angle distribution is measured with respect to a reference frame, calculated from principal components analysis. This descriptor can be used as a similarity measure for 3D models of symmetric organs such as the liver, stomach and head. Shen and Makedon (2006) also mentions the use of high quality 3D images for diagnosis in medical applications.

Another example where 3D shape descriptors have been successfully applied is for detecting shape similarities in 3D video sequences (Huang et al., 2010). Video frames from the 3D video sequence with similar shape and motion are detected. The shapes detected include people with loose clothing performing complex motions, and despite the difficulty of such scenario good results are achieved.

1.4 Thesis context

As it can be seen from Section 1.3, shape descriptors have a wide range of applications. Chapters 2 to 6 of this thesis introduce some novel shape descriptors which can be applied to various applications in the same way as described above.

The main contribution of this thesis is the development of 2D and 3D shape descriptors suitable for implementation in 2D and 3D image processing applications. Each chapter provides some examples to illustrate how the developed descriptors can be applied. Notice that the examples provided in each chapter are meant as an illustration of the possible applications and not as a definitive guide of application. As with all shape descriptors, these can be applied to a variety of applications, on their own or in combination with other descriptors.

1.4.1 Thesis structure

The rest of the thesis is structured as follows: Chapter 2 introduces two shape orientation methods which use different shape features to calculate shape orientation. Specific features, like high curvature points, may be made more or less relevant to the computed orientation depending on a weighting parameter. Chapter 3 introduces a linearity measure: a 2D shape descriptor which provides an indication of how similar an open curve segment is to a straight line segment. This method can also be applied to closed curves. Chapter 4 introduces two 3D shape descriptors: compactness and cubeness. These descriptors work as a similarity measure with a sphere and a cube respectively. Chapter 5 modifies the compactness and cubeness measures to include different weighting parameters. The behaviour of the 3D descriptors can be modified by tuning these weighting parameters. Chapter 6 introduces another 3D shape descriptor: octahedroness measures. This descriptor is derived in a similar way as those presented in Chapter 4, but a different distance metric (city block distance) is used to derive this measure. Chapter 7 concludes with a summary of the thesis and outlining some lines of future research.

1.4.2 List of publications

The work in this thesis has been submitted to various journals. The following is a list of conference and journal articles already published or under review:

- Martinez-Ortiz, C. and Žunić, J. (2008). Points position weighted shape orientation. *Electronics Letters*, 44(10):623–625
- Žunić, J. and Martinez-Ortiz, C. (2009). Linearity measure for curve segments. *Applied Mathematics and Computation*, 215(8):3098–3105
- Martinez-Ortiz, C. and Žunić, J. (2009). Measuring cubeness of 3d shapes. In *CIARP '09: Proceedings of the 14th Iberoamerican Conference on Pattern Recognition*, volume 5856/2009 of *LNCS*, pages 716–723, Berlin, Heidelberg. Springer-Verlag
- Martinez-Ortiz, C. and Žunić, J. (2010a). Curvature weighted gradient based shape orientation. *Pattern Recognition*, 43(9):3035–3041
- Žunić, J., Hirota, K., and Martinez-Ortiz, C. (2010a). Compactness measure for 3d shapes. *Submitted*
- Martinez-Ortiz, C. and Žunić, J. (2010b). A family of cubeness measures. *Submitted*
- Martinez-Ortiz, C. and Žunić, J. (2010c). Measuring cubeness in the limit cases. *Submitted*
- Martinez-Ortiz, C. and Žunić, J. (2010d). Tuneable cubeness measures for 3d shapes. *Submitted*

Chapter 2

Shape Orientation Methods

This chapter includes material from:

> *Martinez-Ortiz, C. and Žunić, J. (2008). Points position weighted shape orientation.* Electronics Letters*, 44(10):623–625*

and

> *Martinez-Ortiz, C. and Žunić, J. (2010a). Curvature weighted gradient based shape orientation.* Pattern Recognition*, 43(9):3035–3041*

2.1 Introduction

Image normalisation is a common task during the pre-processing stage of image processing applications. Image normalisation helps us define a reference frame for images, so they can be processed with respect of such reference frame. Some of the tasks which take place during image normalisation process are: positioning, scale normalisation and *orientation normalisation* (a.k.a. pose normalisation or simply shape orientation).

Shape orientation consists of rotating the shape by a given angle so it is oriented in a *standard* or *canonical* orientation. It is important to have shapes in a canonical orientation, for example, for matching applications; a given shape S and a rotated copy $R(S, \theta)$ can only be identified as being identical once they have been rotated to their canonical orientation.

Some shapes have a clear and easily distinguishable axis of orientation, for example it would be easy to orient a rectangle or an ellipse using their longest axis as axis of orientation. However some other shapes may not have such a clear axis of orientation. One example of such shapes are rotationally symmetric shapes, which have multiple possible axes of orientation. A shape is said to be rotationally symmetric when it is identical to itself when rotated by an angle θ. More specifically, a shape S is n-fold rotational symmetric when it is identical to $R(S, 2\pi/n)$ for $n > 1$ and $n \in \mathbb{Z}^+$.

There are several methods for computing the orientation of a shape, each of them with its own strengths and weaknesses; a method which performs outstandingly for one application may have a poor performance for a different application.

This chapter introduces two alternative methods for computing shape orientation. The first method is an area based method which computes the orientation modifying the impact of each individual shape point depending on its position with respect of the shape centroid. The second method is a boundary based method which gives different impact to boundary points depending on the boundary curvature at that point.

2.2 Existing orientation methods

This section reviews some of the orientation methods which are relevant to the developed methods presented later in this chapter. An understanding of the existing methods is necessary to be able to understand the motivation for developing new methods and, to some extent, because they are the basis used to develop the new methods.

2.2.1 Standard method

The most standard definition of orientation is the axis of least second moment of inertia (Jain et al., 1995). This method of orientation says that the orientation of an object is given by the line which minimises the sum of squared distance of points belonging to the shape to the line. More formally the standard orientation method is defined as follows:

Definition 2.1. *The orientation of a given shape S is defined by the line which minimises the integral:*

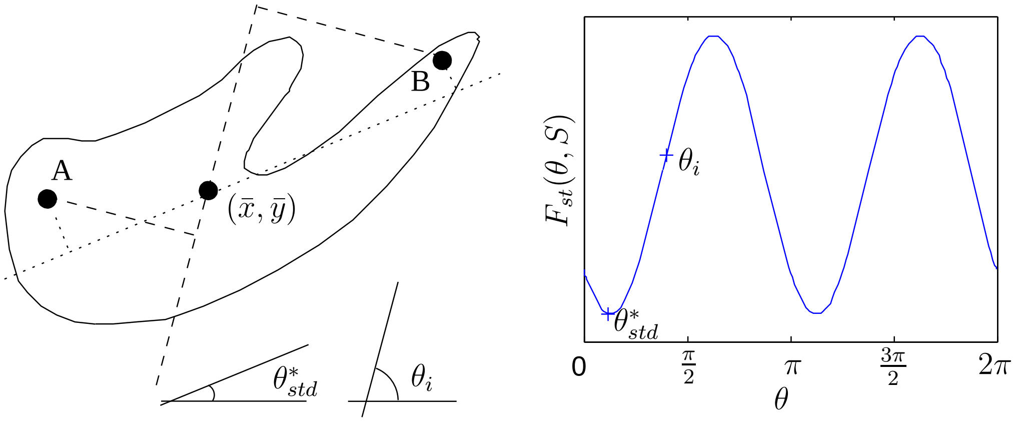
$$I(\theta, S, \rho) = \iint_S r^2(x, y, \theta, \rho)\,dxdy \tag{2.1}$$

where $r(x, y, \theta, \rho)$ is the perpendicular distance from the point (x, y) to the line given in the form $x \cdot \sin\theta - y \cdot \cos\theta = \rho$.

It is well known that the axis of least second moment of inertia passes through the centroid $(\bar{x}, \bar{y})$ of the shape. By assuming that the centroid of the shape coincides with the origin (i.e. $(\bar{x}, \bar{y}) = (0, 0)$), the axis of least second moment of inertia is given by $x \cdot \sin\theta - y \cdot \cos\theta = 0$. The squared distance from any given point (x, y) to such line is then $(x \cdot \sin\theta - y \cdot \cos\theta)^2$ and so in order to compute the orientation of S we have to minimise the function:

$$F_{st}(\theta, S) = \iint\limits_{S} (x \cdot \sin\theta - y \cdot \cos\theta)^2 dxdy \tag{2.2}$$

The orientation of shape S, computed by the standard method, is then given by the angle θ^*_{std} which minimises $F_{st}(\theta, S)$. Figure 2.1 illustrates this concept: the orientation of shape S is given by the line with slope θ^*_{std} which minimises $F_{st}(\theta, S)$ as it can be seen in the graph on (b).



(a) Shape S and lines with angles θ_i and θ^*_{std} going through its centroid $(\bar{x}, \bar{y})$.

(b) Function $F_{st}(\theta, S)$ for $\theta \in (0, 2\pi)$.

Figure 2.1: A shape S and its function $F_{st}(\theta, S)$ for $\theta \in (0, 2\pi)$. In (a), the distances from points A and B to the lines with angles θ_i and θ^*_{std} which goes through its centroid are marked with dashed lines. The orientation of S is given by θ^*_{std} which minimises $F_{st}(\theta, S)$; the point $F_{st}(\theta_i, S)$ for the line with slope θ_i in (a) is also marked.

In order to find the angle θ^*_{std} which minimises the function $F_{st}(\theta, S)$, we set the first derivative $dF_{st}(\theta, S)/d\theta = 0$. This allows for the derivation of a simple closed formula as shown in the following Lemma:

Lemma 2.1. *The angle θ^*_{std} then satisfies the following equation:*

$$\tan(2\theta^*_{std}) = \frac{2 \cdot \mu_{1,1}(S)}{\mu_{2,0}(S) - \mu_{0,2}(S)} \tag{2.3}$$

Proof. The first derivative of $F_{st}(\theta, S)$ is:

$$\begin{aligned}
&\frac{dF_{st}(\theta,S)}{d\theta} = \\
&= \frac{d}{d\theta}\left[\iint\limits_S (x\cdot\sin\theta - y\cdot\cos\theta)^2 dxdy\right] \\
&= \frac{d}{d\theta}\left[\iint\limits_S (x^2\cdot\sin^2\theta - xy\cdot\sin(2\theta) + y^2\cdot\cos^2\theta)dxdy\right] \\
&= \iint\limits_S (2x^2\cdot\sin\theta\cos\theta - 2xy\cdot\cos(2\theta) - 2y^2\cdot\cos\theta\sin\theta)dxdy \\
&= \sin(2\theta)\iint\limits_S x^2dxdy - \cos(2\theta)\iint\limits_S 2xydxdy - \sin(2\theta)\iint\limits_S y^2dxdy
\end{aligned}$$

The point in which function $F_{st}(\theta, S)$ reaches its minimum (i.e. θ^*_{std}) satisfies $dF_{st}(\theta, S)/d\theta = 0$ so we have:

$$\begin{aligned}
\frac{dF_{st}(\theta,S)}{d\theta} &= 0 \\
\sin(2\theta^*_{std})\left[\iint\limits_S x^2dxdy - \iint\limits_S y^2dxdy\right] &= \cos(2\theta^*_{std})\iint\limits_S 2xydxdy \\
\frac{\sin(2\theta^*_{std})}{\cos(2\theta^*_{std})} &= \frac{2\cdot\iint\limits_S xydxdy}{\iint\limits_S x^2dxdy - \iint\limits_S y^2dxdy} \\
\tan(2\theta^*_{std}) &= \frac{2\cdot\mu_{1,1}(S)}{\mu_{2,0}(S)-\mu_{0,2}(S)}
\end{aligned}$$

□

2.2.1.1 Alternative definition

From Definition 2.1 it appears to be that the contribution of any point $P = (x, y)$ to $F_{st}(\theta, S)$ is independent of the distance from that point to the shape centroid $C = (\bar{x}, \bar{y})$. Looking at equation (2.2) each point of the shape contributes by $(x\cdot\sin\theta - y\cdot\cos\theta)^2$ to the function $F_{st}(\theta, S)$. In order to clarify the impact that the position of any point has on the contribution of that point to the computed orientation, the following alternative definition is given:

Definition 2.2. *The orientation of a given shape S is defined by the angle θ^*_{alt} which maximises the integral of the squared length of projections of all lines from shape points to the centroid of the shape C, onto a line having the slope θ.*

More formally, $\mathbf{pr}_\theta \overline{CP}$ is the projection of the line segment $\overline{CP}$ onto a line with slope θ and $|\mathbf{pr}_\theta \overline{CP}|$ is the length of such projection. Then, the orientation of S is determined by the angle θ^*_{alt} which maximises:

$$F_{alt}(\theta, S) = \iint\limits_{P \in S} |\mathbf{pr}_\theta \overline{CP}|^2 dxdy. \tag{2.4}$$

This alternative definition gives lower influence to points which are close to the shape centroid while giving higher influence to points which are far away. The following lemma shows that $F_{st}(\theta, S)$ and $F_{alt}(\theta, S)$ are equivalent.

Lemma 2.2. *Definition 2.1 and Definition 2.2 are equivalent and lead to the same computed orientation.*

Proof. Given a point $P = (x, y) \in S$, the square length of the projection of the line $\overline{CP}$ onto a line with slope θ is $|\mathbf{pr}_\theta \overline{CP}|^2 = (x \cdot \cos\theta + y \cdot \sin\theta)^2$. Thus, $F_{alt}(\theta, S)$ can be re-expressed as:

$$F_{alt}(\theta, S) = \iint\limits_{S} (x \cdot \cos\theta + y \cdot \sin\theta)^2 dxdy \tag{2.5}$$

Adding equations (2.2) and (2.5) together gives:

$$\begin{aligned} & F_{st}(\theta, S) + F_{alt}(\theta, S) = \\ & = \iint\limits_{S} (x \cdot \sin\theta - y \cdot \cos\theta)^2 dxdy + \iint\limits_{S} (x \cdot \cos\theta + y \cdot \sin\theta)^2 dxdy \\ & = \iint\limits_{S} x^2 dxdy + \iint\limits_{S} y^2 dxdy \end{aligned} \tag{2.6}$$

which is constant and does not depend on θ. It can be concluded that $F_{st}(\theta, S)$ reaches its minimum at the same point where $F_{alt}(\theta, S)$ reaches its maximum, and this completes the proof. □

As it has been mentioned before, rotationally symmetric shapes can not be oriented by the method given in Definition 2.1. For this reason an alternative method of orientation was developed, which deals with rotationally symmetric shapes (Tsai and Chou, 1991): The line that minimises the N-th order moment of inertia I_N can be used to orient an n-fold rotationally symmetric shapes for $N \geq n$.

2.2.2 Boundary based standard method

The orientation method given in Definition 2.1 and Definition 2.2 compute the shape orientation from surface information. An alternative method, which uses boundary information only, can be defined analogously:

Definition 2.3. *The orientation of a given shape boundary ∂S is defined by the line which minimises the integral:*

$$I^{(l)}(\theta, \partial S, \rho) = \int_{\partial S} r^2(x(s), y(s), \theta, \rho) ds \tag{2.7}$$

of square distances of the boundary points to a line with slope θ and which goes through the shape centroid.

In this case, the orientation of the shape boundary ∂S can be determined by the angle θ^*_{bb} which minimises:

$$F^{(l)}_{st}(\theta, \partial S) = \int_{\partial S} (x(s) \cdot \sin\theta - y(s) \cdot \cos\theta)^2 ds \tag{2.8}$$

Similar to equation (2.3), the angle θ^*_{bb} which minimises such function satisfies the following equation:

$$\begin{aligned} \tan(2\theta^*_{bb}) &= \frac{2 \cdot \int_{\partial S} xyds}{\int_{\partial S} x^2 ds - \int_{\partial S} y^2 ds} \\ &= \frac{2 \cdot \mu^{(l)}_{1,1}(\partial S)}{\mu^{(l)}_{2,0}(\partial S) - \mu^{(l)}_{0,2}(\partial S)} \end{aligned} \tag{2.9}$$

The proof is analogous to that of Lemma 2.1 and thus its omitted.

2.2.3 Orientation from boundary edge projections

An alternative boundary based method was introduced by Žunić and Stojmenović (2008) which defines the orientation of a polygonal boundary ∂S as the line with slope θ^*_{pr} which maximises the total sum of squared lengths of projections of shape edges onto such line. More formally:

Definition 2.4. *(Žunić and Stojmenović, 2008) Let ∂S be the polygonal boundary of a given shape. The orientation of ∂S is defined by the angle θ^*_{pr} which maximises:*

$$F_{pr}(\theta, \partial S) = \sum_{i}^{N} |\mathbf{pr}_\theta e_i|^2 \tag{2.10}$$

where e_i are each of the N edges of ∂S.

Then angle θ^*_{pr} for which function $F_{pr}(\theta, \partial S)$ reaches its maximum satisfies the following equation:

$$\tan(2\theta^*_{pr}) = \frac{\sum_{i=1}^{N} |e_i|^2 \cdot \sin(2\theta_i)}{\sum_{i=1}^{N} |e_i|^2 \cdot \cos(2\theta_i)} \tag{2.11}$$

where θ_i is the angle between the edge e_i and the x-axis.

The method in given by Definition 2.4 could not be applied to shapes with curved boundaries. A modification is also given by Žunić and Stojmenović which can be applied to shapes with smooth boundaries:

$$\tan(2\theta^*_{sb}) = \frac{2 \cdot \displaystyle\int_a^b \frac{x'(s)y'(s)}{\sqrt{x'(s)^2 y'(s)^2}} ds}{\displaystyle\int_a^b \frac{x'(s)^2 - y'(s)^2}{\sqrt{x'(s)^2 y'(s)^2}} ds} \tag{2.12}$$

where $\overrightarrow{(x'(s), y'(s))}$ is tangent vector at point $(x(s), y(s))$ of the boundary of ∂S given in parametric form. Figure 2.2 illustrates the projection of the vector tangent to a boundary point onto a line.

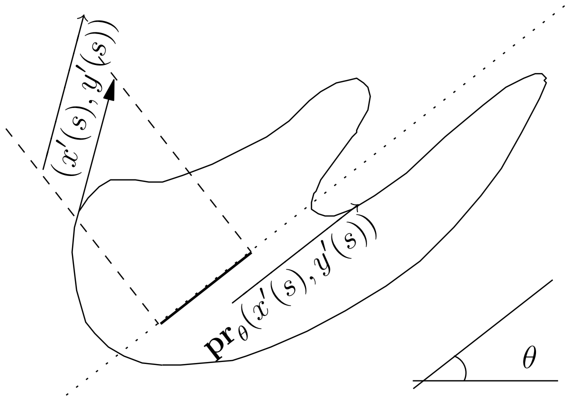

$\overrightarrow{(x'(s), y'(s))}$
$\mathbf{pr}_\theta\overrightarrow{(x'(s), y'(s))}$
θ

Figure 2.2: The vector $\overrightarrow{(x'(s), y'(s))}$ is tangent to the boundary of S at point $(x(s), y(s))$ and $\mathbf{pr}_\theta\overrightarrow{(x'(s), y'(s))}$ is its projection onto the line with slope θ.

2.3 Weighted shape orientation methods

This section introduces two new methods which compute the orientation of a shape taking into account specific shape characteristics. The first method is an area based method which takes into account the position of individual points with respect to the centroid and modifies its contribution to the computed orientation accordingly. The second method is a boundary based method which takes into account the boundary curvature at every boundary point and modifies its contribution to the computed orientation accordingly.

The behaviour of both methods can be adjusted depending on the parameters given. Their behaviour can highlight or suppress specific features of the shapes. This may be desirable in certain cases, but might not be suitable in all cases. Some previous knowledge of the data to which the orientation method will be applied is assumed. The orientation method must then be carefully selected to fulfil the specific needs of the application.

2.3.1 Position weighted orientation

As it has been mentioned in section 2.2.1.1, the impact that each individual point has on the orientation computed by the standard method depends on their distance to the origin. In this section Definition 2.1 is modified to enable control on the impact each point has on the orientation computed. The modified point position dependant orientation method is defined as follows:

Definition 2.5. *Let a given shape S and fixed $\delta < 3/2$. The orientation of S is defined by the angle θ^*_{pw} that maximises the function:*

$$\begin{aligned} F_\delta(\theta, S) &= \iint\limits_{P\in S} \frac{|\mathbf{pr}_\theta \overline{CP}|^2}{(x^2+y^2)^\delta} dxdy \\ &= \iint\limits_{S} \frac{(x\cdot\cos\theta + y\cdot\sin\theta)^2}{(x^2+y^2)^\delta} dxdy \qquad (2.13) \end{aligned}$$

The value of $\delta < 3/2$ is assumed to preserve the existence of the integral in equation (2.13).

Under the new definition, any given point $P = (x, y) \in S$ contributes to the function $F_\delta(\theta, S)$ by $\dfrac{(x\cdot\cos\theta + y\cdot\sin\theta)^2}{(x^2+y^2)^\delta}$. Similar to equation (2.3), the angle θ^*_{pw} which maximises $F_\delta(\theta, S)$ can be found by setting the first derivative to zero $(dF_\delta(\theta, S)/d\theta = 0)$ and such angle satisfies the following equation:

$$\tan(2\theta^*_{pw}) = \frac{2\cdot \displaystyle\iint\limits_{S} \frac{xy}{(x^2+y^2)^\delta} dxdy}{\displaystyle\iint\limits_{S} \frac{x^2-y^2}{(x^2+y^2)^\delta} dxdy} \qquad (2.14)$$

The proof is analogous to that of Lemma 2.1 and thus its omitted.

Obviously for $\delta = 0$ equation (2.14) is equivalent to equation (2.3). In other words, the standard orientation method is a particular case of the point position dependant method with $\delta = 0$. Table 2.1 summarises the effect that different values of δ have on the influence given to points, depending on their distance from the centroid.

	Effect on influence given to each point.
$\delta = 0$	Higher influence is given to points which are far away from the centroid (standard method).
$\delta < 1$	Higher influence is given to points which are far away from the centroid.
$\delta = 1$	All points are given equal influence, regardless of their distance from the centroid.
$\delta \in (1, 3/2)$	Higher influence is given to points which are close to the centroid.

Table 2.1: The value of δ will have an effect on the influence given to each point of S over the computed orientation. The value of δ should be carefully selected depending on the desired effect.

Table 2.1 provides some indication of how the choice of values for δ may modify the behaviour of the computed orientation. The selection of δ should be made depending on the behaviour required by a particular application, based on the nature of the shapes it will be dealing with.

For example, assume that the shape to be oriented is an airplane and that the most important features are the wings, which are located away from the main body (centroid) of the object. In this case $\delta < 0$ could be selected. On the other hand, assume that the shape to be oriented is a watch, where the most important feature is the (shorter) hour hand; in this case a value of $\delta \in (1, 3/2)$ might be more suitable.

As it is mentioned in Definition 2.5 it is assumed that $\delta < 3/2$. This is done to preserve the integration in equation (2.13). However, it could also be interpreted from its geometric meaning: for $\delta \in (1, 3/2)$, higher influence is placed on points closer to the centroid, getting closer to the centroid as $\delta \to 3/2$; δ can never reach $\delta = 3/2$, because this would mean highest influence is given to a centroid. This would be meaningless, because the orientation of a single point can not be determined. Thus δ should remain bound to keep geometric meaning.

2.3.2 Curvature weighted orientation

The method presented in this section considers the tangent vectors at every point of the shape boundary and their projections onto a line. In order to control the impact that each boundary point has on the computed orientation a function of curvature is proposed as a weighting factor. It is reasonable to assume that in some situations the orientation should be more influenced by long flat boundary segments (i.e. low curvature points) and in other situations higher influence should be given to points where the line changes direction rapidly (i.e. high curvature points). Notice that this orientation method does not require the shape boundary to be a closed curve, so it can also be applied to open curve segments.

First let us note that for all appearing boundaries, it is assumed that the length of tangent vector $\overrightarrow{(x(s)', y(s)')}$ and that the boundary length are normalised, such that $|\overrightarrow{(x(s)', y(s)')}| = 1$ and $s \in [0, 1]$.

Also note that for a given shape boundary ∂S given in parametric form, the curvature at point $(x(s), y(s))$ is given by:

$$\kappa(s) = \frac{|x'(s)y''(s) - y'(s)x''(s)|}{(x'(s)^2 + y'(s)^2)^{3/2}}. \tag{2.15}$$

Such equation for computing the curvature at a given point, assumes the boundary to be given in parametric form and also that the first and second derivative of such boundary are known, which may not always be the case. Alternative algorithms based on Digital Straight Segments (DSS) are used instead to estimate the curvature (Hermann and Klette, 2007).

Now, the curvature weighted orientation method is defined as follows:

Definition 2.6. *Given the boundary of a shape ∂S given in parametric form, lets assume a function of curvature $f(\kappa(s))$. The orientation of ∂S is given by the line with slope θ_κ^* that maximises the integral:*

$$F_\kappa(\theta, \partial S) = \int_{\partial S} f(\kappa(s)) \cdot |\mathbf{pr}_\theta \overrightarrow{(x(s), y(s))}|^2 ds \tag{2.16}$$

of squared length of projection of tangent vectors of ∂S on the line, weighted by the curvature at the point corresponding to the tangent vector.

Again similar to equation (2.3), the angle θ_κ^* which maximises $F_\kappa(\theta, \partial S)$ can be found by setting $dF_\kappa(\theta, \partial S)/d\theta = 0$ and such angle satisfies the following equation:

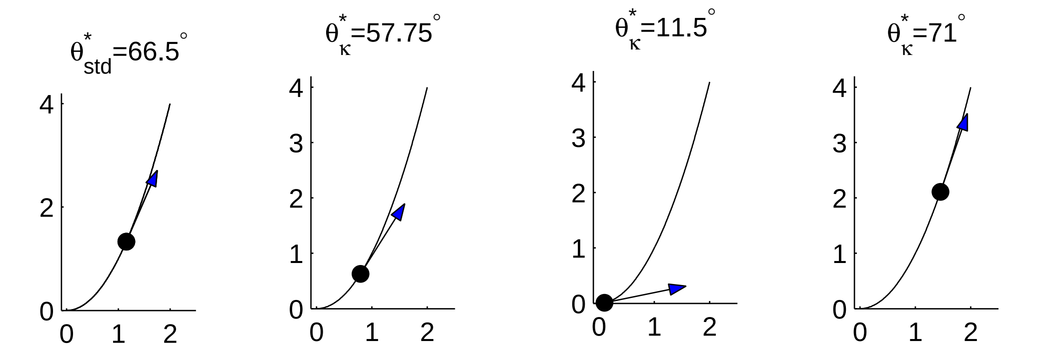
$$\tan(2\theta_\kappa^*) = \frac{2 \cdot \int_{\partial S} f(\kappa(s)) \cdot x'(s)y'(s) ds}{\int_{\partial S} f(\kappa(s)) \cdot (x'(s)^2 - y'(s)^2) ds}. \tag{2.17}$$

The proof is analogous to that of Lemma 2.1 and thus its omitted.

Although any function of curvature can be used as weighting function $f(\kappa(s))$, for the scope of our experiments the 3 curvature weighting functions shown in Table 2.2 are used. A short comment is provided as a brief explanation of the effect each weighting function has, depending on the curvature. Notice that two of the weighting functions have additional parameters, α and β, which allows enhancing or diminishing the produced effect. Also notice that weighting function $f(\kappa(s)) = 1$ causes equation (2.17) to be equivalent to equation (2.12), given that $|\overrightarrow{(x(s)', y(s)')}| = 1$ was assumed.

Function	Comment
$f(\kappa(s)) = 1$	All points are weighted equally.
$f(\kappa(s)) = \kappa(s)^\alpha$	Gives high weight to high curvature points when $\alpha > 0$.
$f(\kappa(s)) = e^{-\beta \cdot \kappa(s)}$	Gives high weight to low curvature points when $\beta > 0$.

Table 2.2: Curvature weighting functions used for the experiments in this chapter. Functions of curvature could be specially designed to produce different behaviours.



(a) Standard method (b) $f(\kappa(s)) = 1$ (c) $f(\kappa(s)) = \kappa(s)^2$ (d) $f(\kappa(s)) = e^{-2 \cdot \kappa(s)}$

Figure 2.3: Computed orientation of parabola segment using different curvature weighting functions $f(\kappa(s))$ to illustrate the effect of each function.

In order to illustrate the effect different curvature weighting functions have, lets take the parabola segment $(x, x^2), x \in [0, 2]$ and compute its orientation using different curvature weighting functions. Figure 2.3 shows the computed orientation values for the different weighting functions: using the standard boundary based method given by equation (2.9), we get an orientation of 66.5°; using weighting function $f(\kappa(s)) = 1$, we get an orientation of 57.75°; using weighting function $f(\kappa(s)) = \kappa(s)^2$, a higher influence is given to high curvature points which are found close to the base of the parabola and thus an orientation of 11.5° is given; finally, using weighting function $f(\kappa(s)) = e^{-2\cdot\kappa(s)}$, a higher influence is given to low curvature points which are found towards the top of the parabola segment and thus the given orientation is 71.0°.

2.3.2.1 Modification for rotationally symmetric shapes

By looking at equation (2.9) it is easy to see that the standard boundary based method given by Definition 2.3 can not give an orientation for shapes which satisfy the following:

$$\mu_{1,1}^{(l)}(\partial S) = 0 \qquad \text{and} \qquad \mu_{2,0}^{(l)}(\partial S) - \mu_{0,2}^{(l)}(\partial S) = 0 \tag{2.18}$$

Such is the case for n-fold rotationally symmetric shapes (for $n > 2$), as shown in the following Lemma:

Lemma 2.3. *Assume an n-fold rotationally symmetric shape, with $n > 2$. Then function $F_\kappa(\theta, \partial S)$ is constant, for any weighting function $f(\kappa(s))$ and independent of θ.*

Proof. If we assume that $F_\kappa(\theta, \partial S)$ is not a constant function then, because of the symmetry assumed , it should have at least $2n$ strict maxima and $2n$ strict minima. Because:

$$\begin{aligned} &\frac{dF_\kappa(\theta, \partial S)}{d\theta} = \\ &= \; 2 \cdot \int_{\partial S} f(\kappa(s)) \cdot (x'(s)\cos\theta + y'(s)\sin\theta)(-x'(s)\sin\theta + y'(s)\cos\theta) ds \\ &= \; 2\cos(2\theta) \cdot \int_{\partial S} f(\kappa(s)) \cdot x'(s)y'(s) ds - 2\sin(2\theta) \cdot \int_{\partial S} f(\kappa(s)) \cdot (x'(s)^2 - y'(s)^2) ds \end{aligned}$$

can not have more than two strict maxima and two strict minima (differing by 180°). Then $F_\kappa(\theta, \partial S)$ not being constant would be a contradiction. □

Following the idea from Tsai and Chou (1991), the orientation method can be modified to overcome the problem of non-orientability of symmetric shapes. By increasing the exponent of $F_\kappa(\theta, \partial S)$, we arrive to the following definition:

Definition 2.7. *The orientation of a shape boundary ∂S is given by the angle $\theta^*_{\kappa M}$ which maximises the function:*

$$F_{\kappa,2M}(\theta, \partial S) = \int_{\partial S} f(\kappa(s)) \cdot |\mathbf{pr}_\theta \overrightarrow{(x(s), y(s))}|^{2M} ds \tag{2.19}$$

with $M > 1$.

Definition 2.7 gives a boundary based, curvature weighted, orientation method which can be used to calculate the orientation of rotationally symmetric shapes. Increasing the order of $F_{\kappa,2M}(\theta, \partial S)$ allows to compute the orientation of n-fold rotationally symmetric shapes. This method uses M-th order central moments of inertia to compute such orientation. Tsai and Chou (1991) showed that for an n-fold rotationally symmetric shape, its M-th order central moment of inertia is constant about any line going through the centroid of such shape for any M less than n. Therefore, $M \geq n$ is required for $F_{\kappa,2M}(\theta, \partial S)$ to be able to compute orientation. Unfortunately it is not possible to derive a closed formula for his method so numerical computation is required, which is more computationally expensive.

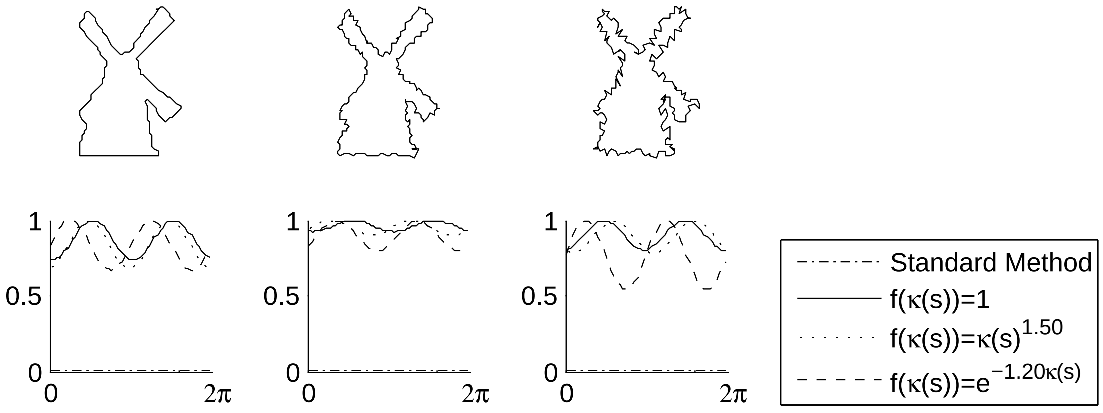
2.3.2.2 Reliability of computed orientation

Using the orientation methods presented in this chapter, the computed orientation of a given shape is said not to be reliable if no clear maxima (minima) can be found for the optimisation function used in that method. For the shapes in Figure 2.4 the standard boundary based method gives an orientation which is not reliable. The optimisation function $F_{st}^{(l)}(\theta, \partial S)$ (from equation (2.8)) varies between 0.0026 and 0.0060, therefore the corresponding graph (see Figure 2.4(a)) is an almost flat line. Since the standard method does not give a distinct orientation, small deviations on the shape boundary (caused by the noise or the digitisation process applied) could lead to an essential deviation of the computed orientation.

	Standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = \kappa(s)^{1.5}$	$f(\kappa(s)) = e^{-1.5\kappa(s)}$
(a)	–	97.40°	89.29°	45.84°
(b)	–	103.13°	51.57°	66.37°
(c)	–	85.94°	108.86°	45.84°

Table 2.3: Curvature weighted orientation of shapes in Figure 2.4. Notice that orientation given by the standard method is not given because no clear orientation can be given.

As mentioned in section 2.3.2.1, second order moments (i.e. $\mu_{1,1}^{(l)}$, $\mu_{2,0}^{(l)}$ and $\mu_{0,2}^{(l)}$) are crucial in computing orientation by the standard boundary based method. Table 2.4 shows the second order moments for the shapes in Figure 2.4. These values imply that the optimisation function $F_{st}^{(l)}(\theta, \partial S)$ is *almost* flat (i.e. constant), and thus it



(a) 0% noise (b) 3% noise (c) 7% noise

Figure 2.4: Windmill with different levels of noise. The graph below each windmill is the optimisation function $F_\kappa(\theta, \partial S)$ maximised to find the orientation of the shape. Notice that all optimisation function have been normalised.

is expected that the standard boundary based method will not perform well on this example.

	$\mu_{1,1}^{(l)}$	$\lvert \mu_{2,0}^{(l)} - \mu_{0,2}^{(l)} \rvert$
Figure 2.4(a)	0.000135	−0.003466
Figure 2.4(b)	0.000095	−0.004299
Figure 2.4(c)	−0.000025	−0.005772

Table 2.4: Second order moments of shapes in Figure 2.4. Notice that these values are very close to zero and thus no maxima can be reliably found for function $F_{st}^{(l)}(\theta, \partial S)$.

However, the curvature weighted orientation method can produce a reliable orientation. Using the weighting function $f(\kappa(s)) = e^{-1.5\kappa(s)}$ produces reliable orientation, although not completely unaffected by noise. As it can be seen on Figure 2.4 the graphs of function $F_\kappa(\theta, \partial S)$ using such weighting function (see dashed line) have easily distinguishable maxima which implies that the computed orientation is reliable. In this way the noise effects are minimised.

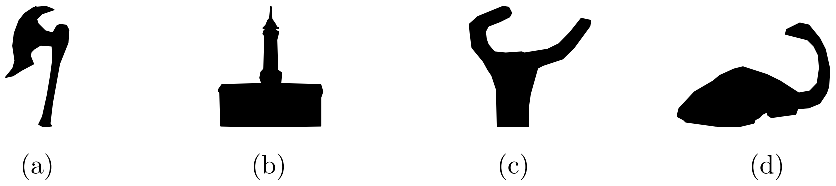
On the other hand, if we use weighting functions which give higher weight to high curvature points, then the computed orientations are expected to be highly dependent on the noise. This is exactly what happens when using $f(\kappa(s)) = \kappa(s)^{1.5}$: the computed orientation changes dramatically when noise is added.

2.4 Examples

This section features a compilation of examples which illustrate the different orientation methods presented in this chapter. Each one of the methods has special characteristics and may be better suited for specific task. Therefore, in order to better illustrate the type of scenario in which each method might be preferred, specific examples have been selected to highlight the strengths of each orientation method.

2.4.1 Position weighted orientation

The orientation method given by Definition 2.5 gives different weight to shape points depending on their position with respect of the shape centroid. For this example we use some images which have part of their mass (i.e. a large number of points) away from the centroid. These images are shown in Figure 2.5 and their orientation for different values of δ is summarised in Table 2.5.



(a) (b) (c) (d)

Figure 2.5: Shapes with part of their mass located away from the centroid.

	(a)	(b)	(c)	(d)
$\delta = -1.4$	104.2°	92.6°	44.8°	34.8°
$\delta = -1.2$	104.6°	93.4°	46.4°	34.2°
$\delta = -1.0$	104.8°	95.2°	48.6°	33.6°
$\delta = -0.5$	106.0°	170.2°	58.4°	31.6°
$\delta = 0.0$	107.8°	177.8°	75.8°	29.0°
$\delta = 0.5$	110.8°	178.8°	90.2°	25.4°
$\delta = 1.0$	118.4°	179.2°	95.0°	20.4°
$\delta = 1.2$	126.4°	179.2°	93.6°	18.0°
$\delta = 1.4$	141.2°	178.4°	84.2°	15.4°

Table 2.5: Orientation of shapes in Figure 2.5 for different values of δ.

Take for example the axe in (a): for lower values of δ (e.g. $\delta = -1.4$) its orientation is more influenced by the handle because the points in the handle are further away from the centroid; on the other hand larger values of δ (e.g. $\delta = 1.4$) give higher impact to points belonging to the blade because they are closer to the centroid.

Similarly happens with the building in (b): orientation computed using low values of δ are influenced by the tower, giving a near vertical orientation, while using larger values of δ gives a nearly horizontal orientation due to the influence of the building's main body.

The human torso in (c) has its orientation influenced by the arms for low values of δ, being oriented at roughly 45° while, for larger values of δ, it is the waist that has highest influence on the computed orientation. Probably the most intuitive orientation is achieved for $\delta = 0.5$ with a computed orientation of 90.2°.

Finally, the dinosaur in (d) has its orientation influenced by its neck for low values of δ, while the back is probably most influential for larger values of δ.

2.4.2 Curvature weighted orientation

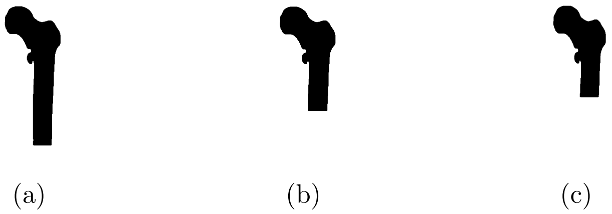
The orientation method given by Definition 2.6 gives different weight to boundary points depending on the curvature of the boundary. The examples in this section show how different weighting functions can be applied in order to give more influence to particular shape characteristics.

Note: Since the equation of the shapes used for these experiments is unknown, curvature can not be estimated using the formula in equation (2.15). Instead, curvature is estimated using the method suggested by Hermann and Klette (2007).

2.4.2.1 Human femurs

In this example the orientation of a human femur is computed, showing how the straight segments of the bone can be used to orient the shape. The standard boundary based method gives inconsistent orientations: 107.0°, 120.4° and 131.0°. Giving equal weight to all boundary points (using $f(\kappa(s)) = 1$), which produces better results but still unsatisfactory.

Due to the straight nature of the femur shape, it could be expected that straight segments (i.e. segments of low curvature) could be more influential to determine the orientation. The curvature function $f(\kappa(s)) = e^{-\beta \cdot \kappa(s)}$ does exactly that: it gives higher influence to low curvature points (see Table 2.2). By setting $\beta = 0.05$ an orientation of 91.7° is achieved for all 3 shapes which matches our expectation.



(a) (b) (c)

Figure 2.6: The bone from (a) is cut to a shorter length in (b) and then cut even shorter in (c). In this case, it would be preferable for the bone socket to have low influence on the computed orientation, and have the orientation being mainly determined by the straight bone segment.

	standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = e^{-0.01\cdot\kappa(s)}$	$f(\kappa(s)) = e^{-0.05\cdot\kappa(s)}$
(a)	107.0°	91.7°	91.7°	91.7°
(b)	120.4°	95.0°	95.0°	91.7°
(c)	131.0°	97.4°	97.4°	91.7°

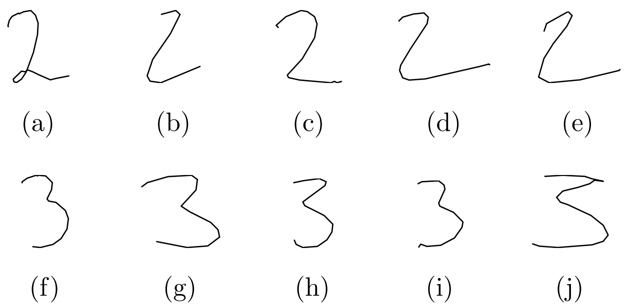
Table 2.6: Curvature weighted orientation of shapes in Figure 2.6. The change in length greatly affects the computed orientation.

2.4.2.2 Hand written numbers

The next example illustrates how the method can be applied to open end curves, like hand-written numbers. Notice that these objects are linear in their nature (i.e. they have no surface) and thus area based methods may not be applied for computing their orientation.

When using weighting function $f(\kappa(s)) = \kappa(s)^2$ most numbers displayed in Figure 2.7 are given an almost horizontal orientation, which is actually our preference. The use of other weighting functions ($f(\kappa(s)) = 1$ and $f(\kappa(s)) = e^{-0.1\cdot\kappa(s)}$), as well as the use of the standard method, leads to a number of distinctive orientations. For example, using weighting function $f(\kappa(s)) = e^{-0.1\cdot\kappa(s)}$, the computed orientation of number '2' is in the range [20.5°, 63.0°] and the computed orientation of number '3' is in the range [−4.8°, 20.5°].

This small example illustrates how different weighting functions can be applied for different tasks. For example, $\kappa(s)^2$ could be used to align all numbers, while by additionally considering $e^{-0.1\cdot\kappa(s)}$ we could differentiate between numbers '2' and '3'.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 2.7: Five samples of hand-written numbers 2 and 3 are presented with different hand writing styles.

	standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = \kappa(s)^2$	$f(\kappa(s)) = e^{-0.1\cdot\kappa(s)}$
(a)	115.1°	54.9°	3.4°	63.0°
(b)	65.1°	37.7°	0.0°	43.5°
(c)	118.3°	20.5°	−2.4°	32.0°
(d)	127.4°	17.2°	0.0°	20.5°
(e)	−14.4°	28.7°	0.0°	28.7°
(f)	119.5°	17.2°	9.1°	20.5°
(g)	−18.5°	0.0°	0.0°	0.0°
(h)	95.9°	5.7°	0.0°	9.1°
(i)	111.2°	5.7°	3.4°	9.1°
(j)	84.3°	−2.4°	−2.4°	−4.8°

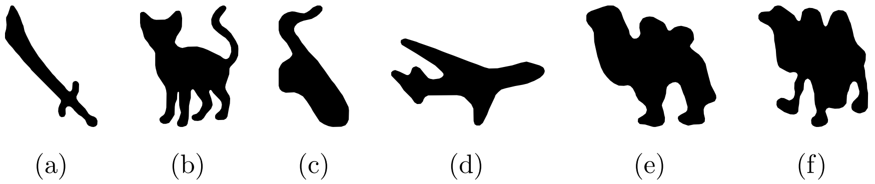
Table 2.7: Curvature weighted orientation of shapes in Figure 2.7. Different orientation is given depending on the curvature weighting function applied.

2.4.2.3 Real world objects

The aim of this example is to illustrate how the curvature weighted orientation method behaves for “real world” objects. For this example, a selection of images of real world objects from the Kimia database[1]. As in previous examples, the different weighting functions used produce different orientations caused by specific features of each shape.

For example, Figures (e) and (f) have very different orientation depending on the weighting function used. Function $f(\kappa(s)) = \kappa(s)^2$ gives more influence to the curved sections on the image (e.g. the hump of the camels). Function $f(\kappa(s)) = e^{-0.1\cdot\kappa(s)}$

[1] http://www.lems.brown.edu/~dmc/



(a) (b) (c) (d) (e) (f)

Figure 2.8: Some images of real objects from the Kimia database and their computed orientations.

	standard method	$f(\kappa(s)) = 1$	$f(\kappa(s)) = \kappa(s)^2$	$f(\kappa(s)) = e^{-0.1\cdot\kappa(s)}$
(a)	−39.2°	118.0°	17.2°	120.3°
(b)	96.5°	91.7°	14.8°	97.4°
(c)	135.5°	123.7°	−2.4°	126.1°
(d)	−12.6°	−10.5°	22.9°	−13.8°
(e)	135.1°	112.2°	17.2°	106.5°
(f)	−15.3°	95.0°	3.4°	95.0°

Table 2.8: Curvature weighted orientation of real world shapes in Figure 2.8.

gives a higher influence to straight segments (like the legs and neck of the camels), and therefore the computed orientation is different (in this case, close to vertical).

2.5 Conclusions

As mentioned before, shape orientation is commonly applied during the early stages of image processing applications. This is still an area of interest in the field and new orientation methods continue to be developed (Saleem et al., 2007; Žunić and Stojmenović, 2008). This chapter provided an overview of some of the existing area based and boundary based shape orientation methods. The difficulties encountered when dealing with rotationally symmetric shapes were analysed (Tsai and Chou, 1991; Žunić et al., 2006).

This chapter introduced novel modifications to the existing orientation methods. The modified methods introduce two different weighting factors: a point position weighting factor for the area based method and a curvature weighting factor for the boundary based method.

The position weighted orientation method uses a tuning parameter, δ, which adjusts the importance each point has, relative to its distance to the centroid. As the examples from Figure 2.5 show, lower values of δ give greater influence to points which are away from the centroid, while higher values of δ give greater influence to points which are close to the centroid. Setting $\delta = 0$ causes the position weighted method to be equivalent to the standard method. The effect of different values of δ is summarised in Table 2.1.

The curvature weighted orientation method uses a function of curvature, $f(\kappa(s))$ as weighting factor. Depending on the choice of $f(\kappa(s))$, greater influence can be given to high curvature or low curvature points. Table 2.2 provides some possible curvature functions which can be used, but other functions of curvature can be used.

The orientation methods introduced in this chapter can be applied for automatically generating properly oriented views from large image repositories, as it was done by Saleem et al. (2007).

Chapter 3

2D Shape Descriptors

This chapter includes material from:

> *Žunić, J. and Martinez-Ortiz, C. (2009). Linearity measure for curve segments.* Applied Mathematics and Computation, *215(8):3098–3105*

3.1 Introduction

Most image processing applications work with 2D images and the majority of shape descriptors developed are built for such images. As it was mentioned in section 1.2.1 a number of 2D shape descriptors have been developed which can be generally divided in two groups: boundary and area based. So far, there has been a greater interest in area based descriptors, because of their robustness with respect to noise and because they can be reliably estimated. For example, the area of a shape can be estimated simply by counting the pixels whose centre falls inside of the digitised shape (Klette and Žunić, 1999), however to estimate the perimeter of a shape more complex methods are required, like DSS estimation (Coeurjolly and Klette, 2004).

Recently there has been an increased interest on boundary based descriptors (Liu et al., 2008; Manay et al., 2006; Mio et al., 2007; Stojmenović and Žunić, 2008) because of their sensitivity to fine detail. Another reason of interest is that boundary based descriptors can be applied to objects of linear nature (e.g. human signatures) where area based descriptors can not be used.

This chapter introduces a boundary based descriptor: *linearity*. The linearity measure indicates the degree to which a given curve differs from a perfect straight line segment. The linearity measure ranges over the interval $(0, 1]$ and reaches its maximum value 1 if and only if the measured line is a perfect straight line segment. The measure is invariant with respect to translations, rotations and scaling transformations. It is initially defined as a measure for open curve segments, and a simple modification allows for the measure to be applied to closed curves. Other comparable measures have been developed by Benhamou (2004) and Stojmenović et al. (2008).

3.2 Linearity

This section introduces the shape linearity measure. First, the linearity measured for open curve segments will be introduced. Just to mention, the straightness index and sinuosity measure presented by Benhamou (2004) are comparable to the linearity measure introduced here. The straightness index is referred to as *a measure of discrepancy between the path actually followed and a perfectly oriented straight segment* (Benhamou, 2004), which is essentially the same idea behind the linearity measure introduced here. The straightness index is defined as the ratio:

$$\mathcal{S}_{idx}(\partial S) = \frac{D}{L} \tag{3.1}$$

where D is the length of the open curve segment ∂S and L is the distance between its end points.

The sinuosity measure is defined as a ratio between the standard deviation of the turning angle of a path (σ) and the step length of such path (p), as expressed by the following formula:

$$\mathcal{S}_b(\partial S) = \frac{\sigma}{\sqrt{p}} \tag{3.2}$$

However this requires the path to be sampled in regular intervals of identical length.

3.2.1 Open curve segments

As it has been mentioned before, the linearity measure should provide an indication of how much an open curve segment differs from a straight line segment. To define the new linearity measure, we start with a Lemma that describes some facts about the integral of distance from line points to the line end points. These facts are then used as a basis to define the new linearity measure.

Lemma 3.1. *Let an open curve segment ∂S given in the arc-length parametric form*

$(x = x(s), y = y(s)), s \in [0, 1]$. *Then:*

$$\int_{\partial S} (x(s) - x_0)^2 + (y(s) - y_0)^2 ds \leq \frac{1}{3}, \tag{3.3}$$

$$\int_{\partial S} (x(s) - x_1)^2 + (y(s) - y_1)^2 ds \leq \frac{1}{3}, \tag{3.4}$$

$$\int_{\partial S} (x(s) - x_0)^2 + (y(s) - y_0)^2 ds = \frac{1}{3} \Leftrightarrow \partial S \textit{is a straight line segment}, \tag{3.5}$$

$$\int_{\partial S} (x(s) - x_1)^2 + (y(s) - y_1)^2 ds = \frac{1}{3} \Leftrightarrow \partial S \textit{is a straight line segment}, \tag{3.6}$$

where (x_0, y_0) *and* (x_1, y_1) *are the end points of* ∂S.

Proof. Since s is the distance from (x_0, y_0) to the point $(x(s), y(s))$ along the curve ∂S (see Figure 3.1):

$$(x(s) - x_0)^2 + (y(s) - y_0)^2 \leq s^2 \tag{3.7}$$

Now, (3.7) gives:

$$\int_{\partial S} ((x(s) - x_0)^2 + (y(s) - y_0)^2) ds \leq \int_{\partial S} s^2 ds = \int_{s=0}^{s=1} s^2 ds = \frac{1}{3},$$

which proves (3.3). The proof of (3.4) is analogous. Trivially, the squared distance from the endpoint (x_0, y_0) to a given point $(x(s), y(s))$ is s^2 (i.e. $(x(s) - x_0)^2 + (y(s) - y_0)^2 = s^2$) only when the segment of ∂S between (x_0, y_0) and $(x(s), y(s))$ is a straight line segment. So:

$$\int_{\partial S} ((x(s) - x_0)^2 + (y(s) - y_0)^2) ds = \int_{\partial S} s^2 ds = \int_{s=0}^{s=1} s^2 ds = \frac{1}{3},$$

which proves (3.5). Again, the proof of (3.6) again is analogous. □

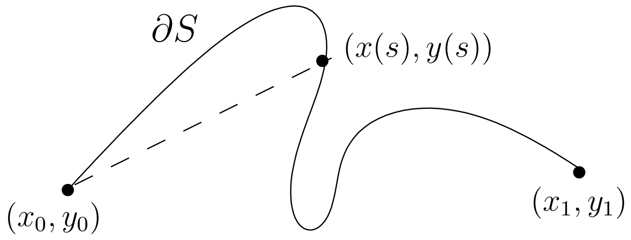


Figure 3.1: The distance (dashed line) between the points (x_0, y_0) and $(x(s), y(x))$ is not bigger than s, which is the length of the ∂S between those points.

For simplicity we will use the following two equalities

$$\begin{aligned}
&\int_{\partial S}((x(s)-x_0)^2+(y(s)-y_0)^2)ds = \\
&= \int_{\partial S}(x(s)^2-2x(s)x_0+x_0^2+y(s)^2-2y(s)y_0+y_0^2)ds \\
&= \mu_{2,0}^{(l)}(\partial S)+\mu_{0,2}^{(l)}(\partial S)+x_0^2+y_0^2 \qquad (3.8)
\end{aligned}$$

and

$$\begin{aligned}
&\int_{\partial S}((x(s)-x_1)^2+(y(s)-y_1)^2)ds = \\
&= \int_{\partial S}(x(s)^2-2x(s)x_1+x_1^2+y(s)^2-2y(s)y_1+y_1^2)ds \\
&= \mu_{2,0}^{(l)}(\partial S)+\mu_{0,2}^{(l)}(\partial S)+x_1^2+y_1^2 \qquad (3.9)
\end{aligned}$$

Notice that $\int_{\partial S} x(s)ds = 0$ and $\int_{\partial S} y(s)ds = 0$ because ∂S is assumed to be centred in the origin. Also notice that the identity $\mu_{p,q}^{(l)}(B) = \int_B (x(s)-\bar{x}^{(l)})^p(y(s)-\bar{y}^{(l)})^q ds$ from equation (1.6) is used in these equations.

Now, the new linearity measure for open curve segments is defined as follows:

Definition 3.1. *Let an open curve segment ∂S having the arc-length parametrisation $(x(s), y(s))$ and $s \in [0,1]$. Then, the linearity measure $\mathcal{L}(\partial S)$ of ∂S is defined as:*

$$\mathcal{L}(\partial S) = 3\cdot\mu_{2,0}^{l}(\partial S)+3\cdot\mu_{0,2}^{l}(\partial S)+\frac{3}{2}\cdot(x_0^2+y_0^2+x_1^2+y_1^2) \qquad (3.10)$$

Theorem 3.1. *The new measure $\mathcal{L}(\partial S)$ has the following properties:*

(a) $\mathcal{L}(\partial S) \in (0,1]$ for all open curve segments ∂S;

(b) $\mathcal{L}(\partial S) = 1 \quad \Leftrightarrow \quad \partial S$ is a straight line segment;

(c) $\mathcal{L}(\partial S)$ is invariant with respect to similarity transformations;

(d) For each $\delta > 0$ there is an open curve segment ∂S such that $0 < \mathcal{L}(\partial S) < \delta$.

Proof. **(a):** $\mathcal{L}(\partial S) > 0$ is trivial since $\mu_{2,0}^{l}(\partial S)+\mu_{0,2}^{l}(\partial S) > 0$ for all curve segments ∂S. $\mathcal{L}(\partial S) \le 1$ follows directly from equations (3.3), (3.4), (3.8) and (3.9).

(b): The proof of $\mathcal{L}(\partial S) = 1$ follows from adding together equations (3.5) and (3.6), and multiplying by 3/2.

(c): To prove this item it is enough to notice that $\mu_{2,0}^{l}(\partial S)+\mu_{0,2}^{l}(\partial S)$ is a well known similarity transformations invariant (see Chen (1993); Hu (1962)). Under assumptions that ∂S is normalised so it has length 1 and that the centroid of ∂S coincides with the origin (i.e. $(\bar{x}^{(l)}, \bar{y}^{(l)}) = (0,0)$), it is easy to deduce that $(x_0^2 + y_0^2) + (x_1^2 + y_1^2)$ is also invariant with respect to similarity transformations.

(d): It is easy to construct a curve segment ∂S such that 0< $\mathcal{L}(\partial S) < \delta$ for any given δ. For example, let a circle $\mathcal{C}(r)$ with radius r, and let ∂S_r be an open segment, normalised so its length is 1, and that is inscribed in $\mathcal{C}(r)$. Then:

$$\begin{aligned} \mu_{2,0}^{l}(\partial S_r) + \mu_{0,2}^{l}(\partial S_r) &= \int_{\partial S_r} (x(s)^2 + y(s)^2) ds \\ &\leq \max\{x(s)^2 + y(s)^2 | (x(s), y(s)) \in \partial S_r\} \\ &\leq r^2 \end{aligned}$$

Also, the end points (x_0, y_0) and (x_1, y_1) of ∂S_r are within a distance of r from the origin. This, together with the previous inequality and equation (3.1) gives:

$$\begin{aligned} \mathcal{L}(\partial S_r) &\leq 3 \cdot r^2 + \frac{3}{2} \cdot (r^2 + r^2) \\ &= 6 \cdot r^2 \end{aligned}$$

The statement (d) follows by setting $r < \sqrt{\delta/6}$. □

3.2.2 Closed curve segments

This section introduces a modification of Definition 3.1 which allows the linearity measure to be applied to closed curves. Generally speaking, a closed curve can be understood as an open curve segment whose ending points coincide, i.e. $(x_0, y_0) = (x_1, y_1)$. However when a method designed for open curves (which is the case for $\mathcal{L}(\partial S)$) is extended for closed curves, the main problem is how to select a particular point which will play the role of the *start-and-finish-point*, i.e. $P_{sf} = (x_0, y_0) = (x_1, y_1)$. Selecting such point is not trivial because the method applied could lead to different results depending on the choice of P_{sf}.

The measure $\mathcal{L}(\partial S)$ is susceptible to such dependency on the choice of P_{sf}. For example, consider a closed curve ∂S and select a start-and-finish-point P_{sf}, so ∂S can then be considered an open curve segment (let us denote it ∂S_{sf}). Also let us denote the centroid of ∂S as C. Then $\mathcal{L}(\partial S_{sf})$ would depend on the choice of P_{sf}. Furthermore, it is possible that for two different choices of start-and-finish-points, P_{sf1} and P_{sf2}, the measure $\mathcal{L}(\partial S)$ produces different results, i.e. $\mathcal{L}(\partial S_{sf1}) \neq \mathcal{L}(\partial S_{sf2})$.

More precisely, equation (3.10) says that $\mathcal{L}(\partial S)$ depends only on the distance from P_{sf} to the centroid C: because $\mu_{2,0}^{l}(\partial S) = \mu_{2,0}^{l}(\partial S_{sf})$ and $\mu_{0,2}^{l}(\partial S) = \mu_{0,2}^{l}(\partial S_{sf})$

for any $P_{sf} \in \partial S$, then the moments $\mu^l_{2,0}(\partial S)$ and $\mu^l_{0,2}(\partial S)$ do not depend on the choice of P_{sf}. Thus, $\mathcal{L}(\partial S_{sf})$ depends only on the choice of P_{sf}. So if the distance from P_{sf1} to the centroid C is different from the distance from P_{sf2} to C their measured linearities will also be different, i.e.

$$|\overline{CP_{sf1}}| \neq |\overline{CP_{sf2}}| \Longrightarrow \mathcal{L}(\partial S_{sf1}) \neq \mathcal{L}(\partial S_{sf2})$$

From equation (3.10) it can be easily deduced that the linearity measure will increase as the distance from point P_{sf} to the centroid (i.e. $|\overline{CP_{sf}}|$) increases:

$$|\overline{CP_{sf1}}| < |\overline{CP_{sf2}}| \Longrightarrow \mathcal{L}(\partial S_{sf1}) < \mathcal{L}(\partial S_{sf2})$$

There are two logical approaches for selecting P_{sf}:

- Select a $P_{sf} = P_{max}$ which is furthest away from the centroid in order to maximise $\mathcal{L}(\partial S_{sf})$, or
- Select a $P_{sf} = P_{min}$ which is closest to the centroid in order to minimise $\mathcal{L}(\partial S_{sf})$.

Taking this into account, the following definition is given for two linearity measures for closed curves, as an extension of the linearity measure give in Definition 3.1:

Definition 3.2. *Let a closed curve ∂S given in an arc length parametrisation and placed so that the centroid of ∂S is located at the origin. And let $\mathbf{d}^2_{max}(\partial S)$ and $\mathbf{d}^2_{min}(\partial S)$ denote the maximum and minimum possible squared distance of points from ∂S to the centroid C:*

$$\mathbf{d}^2_{max}(\partial S) = \max\{x^2 + y^2 | (x,y) \in \partial S\},$$
$$\mathbf{d}^2_{min}(\partial S) = \min\{x^2 + y^2 | (x,y) \in \partial S\}.$$

Then, the linearity measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ are defined as follows:

$$\mathcal{L}_{max}(\partial S) = 3 \cdot \left(\mu^l_{2,0}(\partial S) + \mu^l_{0,2}(\partial S) + \mathbf{d}^2_{max}(\partial S)\right) \tag{3.11}$$

and

$$\mathcal{L}_{min}(\partial S) = 3 \cdot \left(\mu^l_{2,0}(\partial S) + \mu^l_{0,2}(\partial S) + \mathbf{d}^2_{min}(\partial S)\right) \tag{3.12}$$

Note. There can be many points on ∂S which are at $\mathbf{d^2}_{max}(\partial S)$ distance (or $\mathbf{d^2}_{min}(\partial S)$ distance) from the centroid. This does not interferes with the computation of $\mathcal{L}_{max}(\partial S)$ (or $\mathcal{L}_{min}(\partial S)$) because Definition 3.2 uses the distance $\mathbf{d}^2_{max}(\partial S)$ (or $\mathbf{d}^2_{min}(\partial S)$) and not the points coordinates of any point located as such distance.

Because closed curves have the additional restriction of starting and ending at the same point there is an upper bound to $\mathbf{d}^2_{max}(\partial S)$ and $\mathbf{d}^2_{min}(\partial S)$. The furthest away a given point can be from the centroid of a closed curve ∂S of length 1 is 1/4. Such is the case of the infinitely thin rectangle shown in Figure 3.2: starting from point P_{sf} at $s = 0$, the curve ∂S moves away (to the left of C in Figure 3.2) from the centroid in a straight line until it reaches $s = 0.25$. At this point the curve ∂S must go back and return to the centroid where it reaches $s = 0.5$. Notice that in Figure 3.2 half of the line (i.e. $s \in [0.5, 1]$) must be to the right of the centroid. From equation (3.11), it can be shown that for such a shape $\mathcal{L}_{max}(\partial S) = 0.25$.

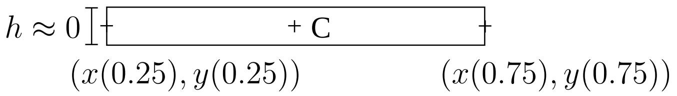


Figure 3.2: Infinitely thin rectangle centred in the origin. If P_{sf} is selected at the origin, the most remote points of the curve ∂S_{sf} are located at $(x(0.25), y(0.25))$ and $(x(0.75), y(0.75))$.

The upper bound of $\mathbf{d}^2_{min}(\partial S)$ would be reached for a shape which maximises $\mathbf{d}^2_{min}(\partial S)$. Such a shape would be a circle of unitary perimeter, having radius $r = 1/2\pi$. By definition, all points of the circle are at the same distance of its centroid, thus $\mathbf{d}^2_{min}(\partial S) = 1/2\pi$. From equation (3.12), it can be shown that for such a shape $\mathcal{L}_{min}(\partial S) = 3/2\pi^2$.

3.2.3 Other comparable measure

During the review process of Žunić and Martinez-Ortiz (2009) it was suggested that the linearity measures for open curve segments would be in a sense comparable to the well known standard compactness measure (also known as shape circularity, see equation (1.8)).

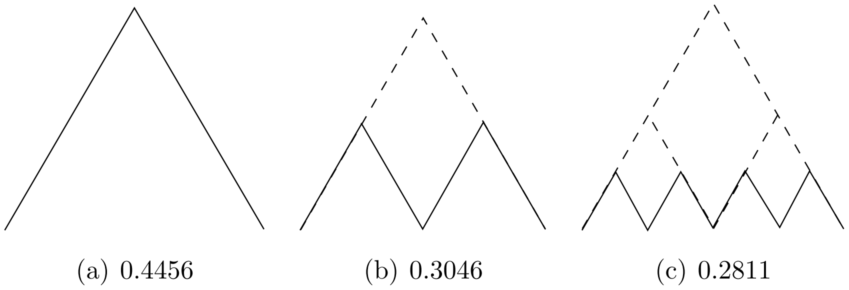
The compactness measure $\mathcal{C}_{Std}(S)$ provides an indication of how much given shape differs from a circle, producing its maximal possible value of 1 if and only if the given shape is a circle. At the same time, the compactness measure $\mathcal{C}_{Std}(S)$ produces low values if the given shape is an elongated rectangle approximating a straight line. It could be expected that a higher circularity $\mathcal{C}_{Std}(S)$ implies a lower linearity, and vice-versa.

3.3 Experiments

This section provides examples which illustrate the behaviour of linearity measures $\mathcal{L}(\partial S)$, $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$. This section is divided in two parts: Subsection 3.3.1 provides some examples for the linearity measure for open curve segments given in Definition 3.1. Subsection 3.3.2 presents some experiments to illustrate the behaviour of linearity measures for closed curves given in Definition 3.2.

3.3.1 Linearity of open curve segments

The first example in this section shows some simple cases where the linearity measure $\mathcal{L}(\partial S)$ might be preferable over the straightness index $\mathcal{S}_{idx}(\partial S)$. The three polygonal lines in Figure 3.3 preserve the same length and the same distance between their end points, thus their straightness indices would be equal. However this is clearly not the preferred case. The linearity measure $\mathcal{L}(\partial S)$ gives different measures to each of these lines.



(a) 0.4456 (b) 0.3046 (c) 0.2811

Figure 3.3: Three displayed curves have different linearities measured by $\mathcal{L}(\partial S)$. The straightness index $\mathcal{S}_{idx}(\partial S)$ gives the same value for all three curves.

The original motivation to develop the straightness index was for analysing the path that an animal follows while moving towards its goal (Benhamou, 2004). A straighter path usually means that the animal has more efficient orientation, while a path with constant changes in direction could indicate that the animal is wandering (random search) (Nams, 2006). Measure $\mathcal{L}(\partial S)$ may be more suitable than $\mathcal{S}_{idx}(\partial S)$ for identifying such behaviours.

The next synthetic example illustrates how the measured $\mathcal{L}(\partial S)$ of a circular arc segment decreases as the arc segment approaches a full circle. Different arc segments on the form $\partial S_\alpha = (x = \sin\theta, y = \cos\theta), \theta \in [0, \alpha]$ are constructed for $\alpha \in (0, 2\pi)$. Figure 3.4 shows the graph of the measured linearities $\mathcal{L}(\partial S_\alpha)$ for different values of α.

Figure 3.4 also shows three circular arc segments. Since all measured curve segments are assumed to be of unit length, all appearing arcs should be scaled for a certain factor. The left hand arc displayed in Figure 3.4 is the shortest from the displayed circles, so the scaling factor applied to it is larger than in the case of two arcs.

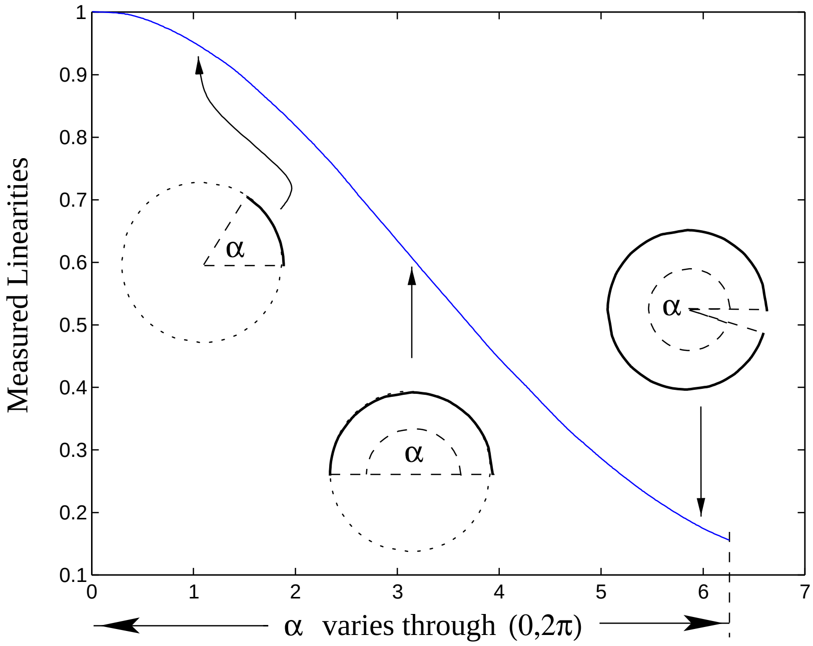
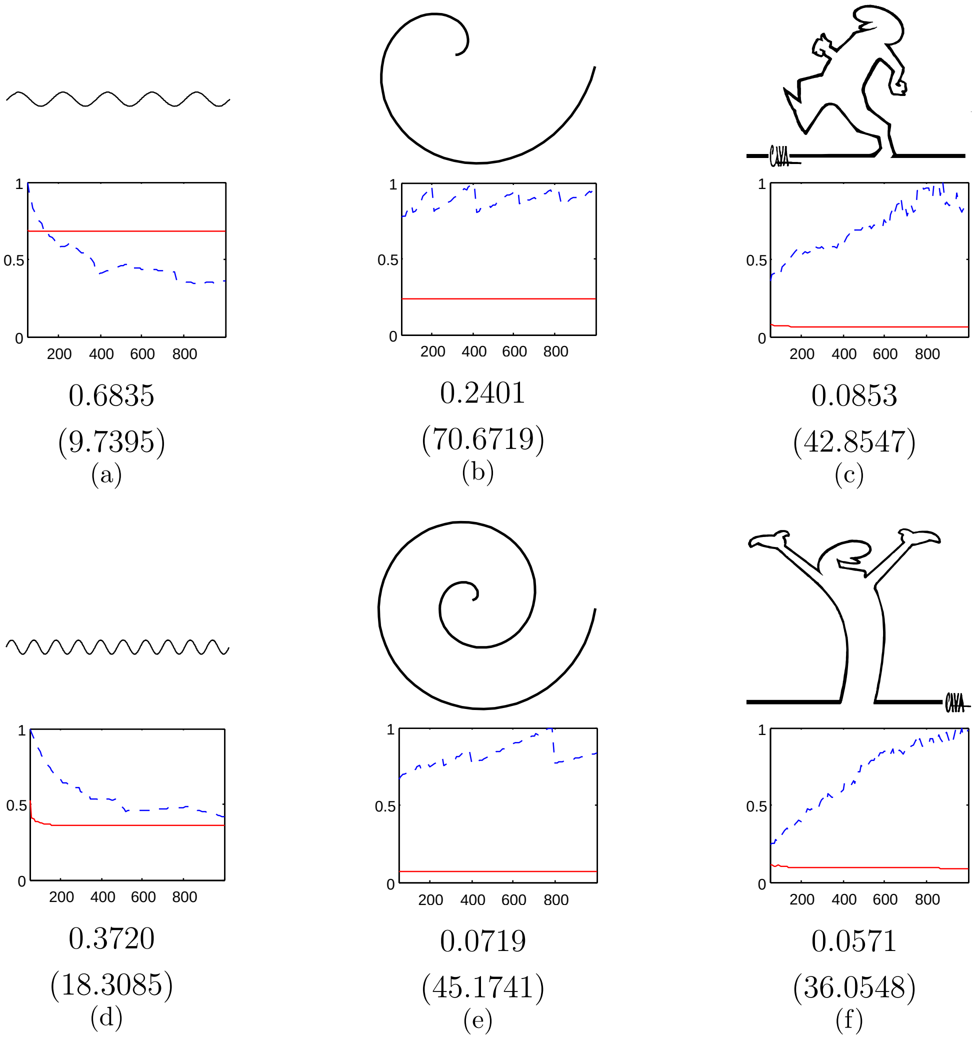


Figure 3.4: The graph presents the measured linearities $\mathcal{L}(\partial S)$ of the circular arcs: $(x = \sin\theta, y = \cos\theta), \theta \in [0, \alpha]$, while different arcs correspond to a different $\alpha \in (0, 2\pi)$. Three particular arcs are displayed (bold lines) and their measured linearities (0.95169, 0.6079 and 0.17439, respectively) are pointed out by the corresponding arrows.

Using equation (3.10) it is easy to see that the measured linearity of the circular arc in the middle of Figure 3.4 (a "half-circle") is $6/\pi^2 \approx 0.6079$. Let us mention that in the two limit cases, when $\alpha \to 0$ and when $\alpha \to 2\pi$ the graph shows the expected results: if $\alpha \to 0$ we get the arc would tend to a straight line segment and its measured linearity would tend to 1; if $\alpha \to 2\pi$ then the measured linearity decreases to a limit value of $3/2\pi^2 \approx 0.1520$ (easily computed from equation (3.10)).

Figure 3.5 shows the linearity measure applied to a variety of open curves. The measured linearities of the curves in figures (a) and (d) seem reasonable: a higher linearity for shape in (a) matches with our perception. Similarly happens for the curves in (b) and (e): a lower linearity for (e) seems reasonable since it is more bent than (b).



0.6835
(9.7395)
(a)

0.2401
(70.6719)
(b)

0.0853
(42.8547)
(c)

0.3720
(18.3085)
(d)

0.0719
(45.1741)
(e)

0.0571
(36.0548)
(f)

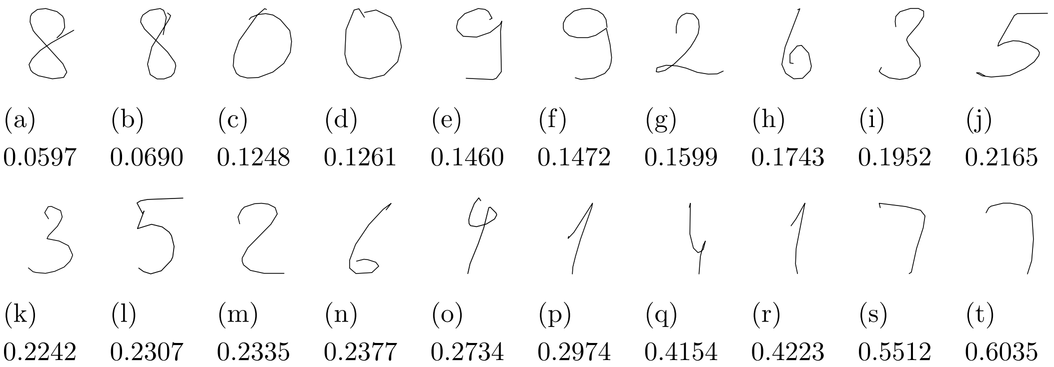
Figure 3.5: Open curve segments and their measured linearity $\mathcal{L}(\partial S)$ and sinuosity $\mathcal{S}_b(\partial S)$ (in parenthesis) re-sampled to 1,000 steps. The graphs below each open curve shows the change in linearity and sinuosity when the curves are re-sampled to different number of steps. Sinuosity has been normalised.

The number in parenthesis shown underneath each figure is the sinuosity measure $\mathcal{S}_b(\partial S)$ obtained for these curves, re-sampling each curve into 1,000 steps of uniform length.

As it has been mentioned before, the sinuosity measure requires the path to be re-sampled into uniform length steps. It is mentioned by Benhamou (2004) that the sinuosity measure depends on the way the path is discretized. The graphs under each shape in Figure 3.5 shows the linearity $\mathcal{L}(\partial S)$ and sinuosity $\mathcal{S}_b(\partial S)$ measured for each shape re-sampled to different number of steps ranging from 50 to 1,000 steps. *Note:* on each graph, sinuosity has been normalised so its produced values are in the range $[0, 1]$.

These graphs illustrate to what extent the sinuosity measure $\mathcal{S}_b(\partial S)$ is affected by the re-sampling process. This shows that the linearity measure $\mathcal{L}(\partial S)$ is more robust in this respect, making the results produced by linearity measure be more reliable.

The last example in this section illustrates uses the linearity measure on its own to group together small data set of handwritten digits. Figure 3.6 shows the handwritten digits, approximated by polygonal lines, arranged in ascending order of linearity. $\mathcal{L}(\partial S)$ successfully groups together the pairs of digits: “0”, “7”, “8” and “9”. Additionally digits “1”, “3”, “4”, and “5” are ranked very closely – i.e. only one digit is ranked in between pairs of those digits: the pair of digit “3” overlaps with the pair of digit “5”, and the pair of digit “1” overlaps with the pair of digit “4”. $\mathcal{L}(\partial S)$ gives its worst matching for the digits “2” and “6” – i.e. five other digits are ranked in between both instances of digit “2” (and in between both instances of digit “6”).



(a) 0.0597 (b) 0.0690 (c) 0.1248 (d) 0.1261 (e) 0.1460 (f) 0.1472 (g) 0.1599 (h) 0.1743 (i) 0.1952 (j) 0.2165

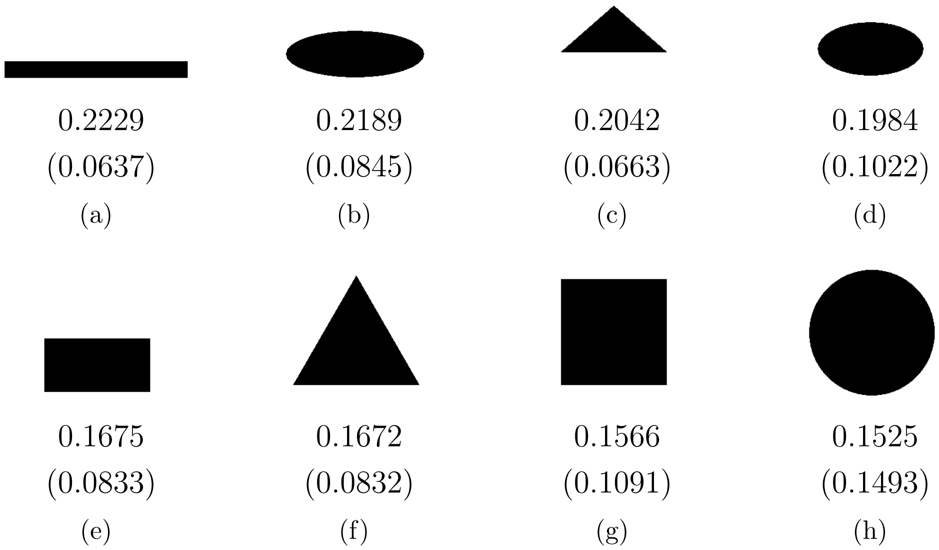
(k) 0.2242 (l) 0.2307 (m) 0.2335 (n) 0.2377 (o) 0.2734 (p) 0.2974 (q) 0.4154 (r) 0.4223 (s) 0.5512 (t) 0.6035

Figure 3.6: Hand written digits ordered by their linearity $\mathcal{L}(\partial S)$.

3.3.2 Linearity of closed curve segments

The first experiment in this section applies the linearity measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ to a set of geometric shapes. Figure 3.7 shows the geometric shapes and their measured linearities. It can be observed that the values obtained for $\mathcal{L}_{min}(\partial S)$ for all shapes are quite low – i.e. all values are in the range $[0.0637, 0.1091]$ for shapes (a)-(g). This implies that the scores given by the linearity measure $\mathcal{L}_{min}(\partial S)$ may not be very informative. Also notice that from equations (3.11) and (3.12) it is expected that for the circle in (h), linearity measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ should be the same – this is not the case due to digitisation errors.

The shapes used in Figure 3.7 are sampled to approximately 1,000 points, sampled images of 200x200 pixels. Using a higher resolution images will reduce the error induced due to approximation errors. For example, for the circle in (h), using an image of 1000x1000 the obtained values are $\mathcal{L}_{max}(\partial S) = 0.1521$ and $\mathcal{L}_{min}(\partial S) = 0.1516$. These values are less affected by digitisation errors, being closer to the theoretical value of $\mathcal{L}_{max}(\partial S) = \mathcal{L}_{min}(\partial S) = 3/2\pi^2$ which should ideally be obtained for a circle (easily computed from equation (3.10)).



0.2229 (0.0637) (a)

0.2189 (0.0845) (b)

0.2042 (0.0663) (c)

0.1984 (0.1022) (d)

0.1675 (0.0833) (e)

0.1672 (0.0832) (f)

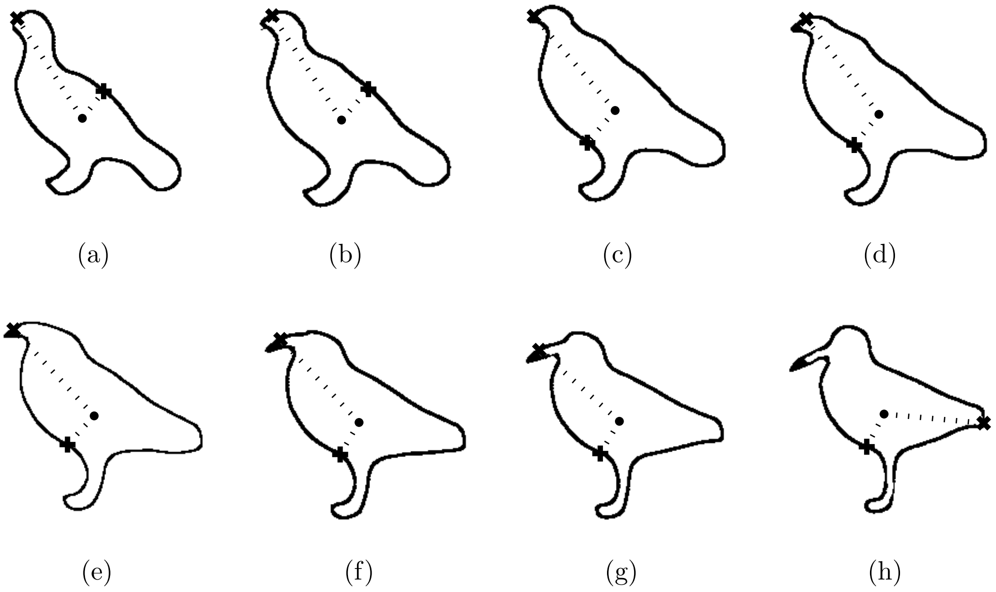
0.1566 (0.1091) (g)

0.1525 (0.1493) (h)

Figure 3.7: Geometric shapes with their corresponding linearities $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ in brackets.

Figure 3.8 shows a sequence of shapes obtained from the elastic deformation between two bird shapes[1]: shape (a) is deformed to become shape (h), with shapes (b)-(g) being the intermediate stages. The linearity measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ and standard circularity measure $\mathcal{C}_{Std}(S)$ are applied to these shapes (see Table 3.1).

[1] Images taken from Srivastava et al. (2009)



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.8: Elastic deformation of shapes: Shape (a) is deformed to become (g). The distance from the centroid to the maximal and minimal distance points are marked for each shape.

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
$\mathcal{L}_{max}(\partial S)$	0.1601	0.1551	0.1531	0.1491	0.1431	0.1361	0.1251	0.1151
$\mathcal{L}_{min}(\partial S)$	0.0571	0.0601	0.0611	0.0611	0.0581	0.0561	0.0531	0.0511
$\mathcal{C}_{Std}(S)$	0.4070	0.4317	0.4577	0.4612	0.4498	0.4355	0.4129	0.3861

Table 3.1: Linearity measures $\mathcal{L}_{max}(\partial S)$, $\mathcal{L}_{min}(\partial S)$ and circularity measure for the closed curves in Figure 3.8.

In this case again it seems to be that the linearity measure $\mathcal{L}_{min}(\partial S)$ is less informative, giving very similar scores for all shapes in the sequence. On the other hand, the linearity measure $\mathcal{L}_{max}(\partial S)$ decreases in each stage of the deformation process. This seems reasonable since the latter shapes have a more elaborated boundary, which leads to the observed decrease in $\mathcal{L}_{max}(\partial S)$ linearity.

Notice that the linearity measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ vary consistently although the maximal and minimal distance points (P_{max} and P_{min}) change their relative position (denoted by "+" and "x" in the images in Figure 3.8), e.g. P_{max} is located near the beak of the bird in shape (a) but it changes to be near the tail of the bird in shape (h). This is in accordance with Definition 3.2 because $\mathcal{L}_{max}(\partial S)$

is computed using the distance $\mathbf{d}^2_{max}(\partial S)$ and not the coordinates of the point P_{max} itself.

By looking at Table 3.1 it would be difficult to tell whether the circularity measure $\mathcal{C}_{Std}(S)$ or the linearity measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ would be preferable in every case: Linearity measure $\mathcal{L}_{max}(\partial S)$ decreases continuously on each step of the deformation process; mean while circularity measure $\mathcal{C}_{Std}(S)$ increases on the first steps of the deformation process only to decrease on the later steps. Which of these behaviours might be preferable is a completely application dependant choice.

3.4 Conclusions

This chapter introduced the linearity measure, which can be applied to open and closed curves. The linearity of a curve segment ∂S provides an indication of similarity between ∂S and a straight line segment.

When applied to closed curves, a start-and-finish point must be selected; this can be selected as the point closest to the centroid or the point furthest away from the centroid. The choice of the start-and-finish point will have an impact on the computed linearity: as mentioned before $\mu^l_{2,0}(\partial S)$ and $\mu^l_{0,2}(\partial S)$ are unaffected by the choice of start-and-finish point, so the difference between $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$ will be determined by $\mathbf{d}^2_{max}(\partial S)$ and $\mathbf{d}^2_{min}(\partial S)$ only. When the point closest to the centroid is selected, the linearity values will generally be quite low, so generally speaking $\mathcal{L}_{min}(\partial S)$ will have smaller values than $\mathcal{L}_{max}(\partial S)$.

When the linearity measure is applied to closed curves, a feature selection algorithm can be used to decide whether to use $\mathcal{L}_{max}(\partial S)$, $\mathcal{L}_{min}(\partial S)$ or both. However, the selection of one linearity measure over another is dependant on the goal to be achieved. For example, with the shapes in Figure 3.8, if our aim is to classify all birds as being of the same class, perhaps $\mathcal{L}_{min}(\partial S)$ would be a more suitable feature for this task, since all values produced by $\mathcal{L}_{min}(\partial S)$ are very close together. On the other hand, if our aim is to classify them in two different classes, perhaps $\mathcal{L}_{max}(\partial S)$ would be a more suitable feature, since it provides a better separation of the shapes.

The linearity measure introduced here can be applied to objects which, due to the nature of the information they represent, are best represented as curve segments. One example is the path followed by animals when exploring their surroundings as described by Benhamou (2004); another example could be the path described by an individual being tracked by a CCTV camera as described by Han et al. (2004).

Chapter 4

3D Shape Descriptors

This chapter includes material from:

> *Žunić, J., Hirota, K., and Martinez-Ortiz, C. (2010a). Compactness measure for 3d shapes.* Submitted

and

> *Martinez-Ortiz, C. and Žunić, J. (2009). Measuring cubeness of 3d shapes. In* CIARP '09: Proceedings of the 14th Iberoamerican Conference on Pattern Recognition*, volume 5856/2009 of* LNCS*, pages 716–723, Berlin, Heidelberg. Springer-Verlag*

4.1 Introduction

As it has been mentioned before, in recent years there has been an increased interest in development of new 3D shape descriptors. Plenty of research has been done for 2D shape descriptors, however, less research has been made on 3D shape descriptors as the use of 3D images is relatively new. It is only recently that the use of 3D media is becoming more easily available and thus increasingly common in their use. Some 3D shape descriptors have already been developed as extension of 2D shape descriptors, some of which are mentioned in section 1.2.2.

In this chapter we introduce two 3D shape descriptors: compactness and cubeness. Both descriptors are derived in a very similar manner, following the same process but using different distance metrics. These descriptors work, in fact, as a measure of compactness under their given distance metric: using euclidean distance, the most compact shape is a sphere; using chessboard distance, the most compact shape is a cube. The next section explains these distance metrics and why the sphere and the cube are *most compact* under these metrics.

4.2 Distance metrics

In general a measure of distance is a number which indicates how far two objects are from each other. The physical distance between two points is normally understood as the length of a straight line segment joining these points. However, there exist some alternative sets of rules for measuring the space between two points. These are known as *distance metrics*. Although there is a wide variety of distance metrics available in the literature, in this section we will only briefly review two of these metrics which are relevant for the development of the 3D shape descriptors which are the subject of our study.

4.2.1 Euclidean distance

Euclidean distance is probably the most commonly used distance metric. It is defined, as mentioned above, as the length of straight line segment joining two points. It can be calculated by means of the Pythagoras' theorem. The euclidean distance between two n-dimensional points $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$ is given by:

$$l_2(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} \tag{4.1}$$

4.2.2 Chessboard distance

One alternative distance metric is the *chessboard distance* metric (also known as Chebyshev distance or l_∞-distance). The distance between two points is defined as the greatest of their differences along any coordinate axis. More formally, the l_∞-distance between two points $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$ is given by:

$$l_\infty(P, Q) = \max(|p_1 - q_1|, |p_2 - q_2|, ..., |p_n - q_n|) \tag{4.2}$$

4.2.3 Unit disks

Let us notice that compactness (in 2D) is generally understood as the degree to which a given shape differs from a region bounded by a circle (compactness in 2D is sometimes also called the circularity). For this reason before introducing the 3D compactness measure it is important to review how a disk is defined.

Generally speakings, a disk of radius r is the collection of all points which are (at most) at a distance r from the centre of such disk. For simplicity, lets assume that the centre of the disk is situated at the origin O. So, if euclidean distance is used, it is clear that a disk of radius r is a circle and that such circle can be expressed as:

$$c_2(r) = \{P = (x, y, z) : l_2(P, O) \leq r\} \tag{4.3}$$

Of course this definition can easily be extended to 3D in which case the resulting shape is a sphere. However, if the distance metric used is the chessboard distance instead of the euclidean distance, the resulting disk is not a circle but a square.

$$c_\infty(r) = \{P = (x, y, z) : l_\infty(P, O) \leq r\} \tag{4.4}$$

This definition can also be extended to 3D in which case the resulting shape in this case is a cube. Figure 4.1 shows the $c_2(r)$ and $c_\infty(r)$ with $r = 1$, $r = 2$ and $r = 3$ in 2D and 3D space.

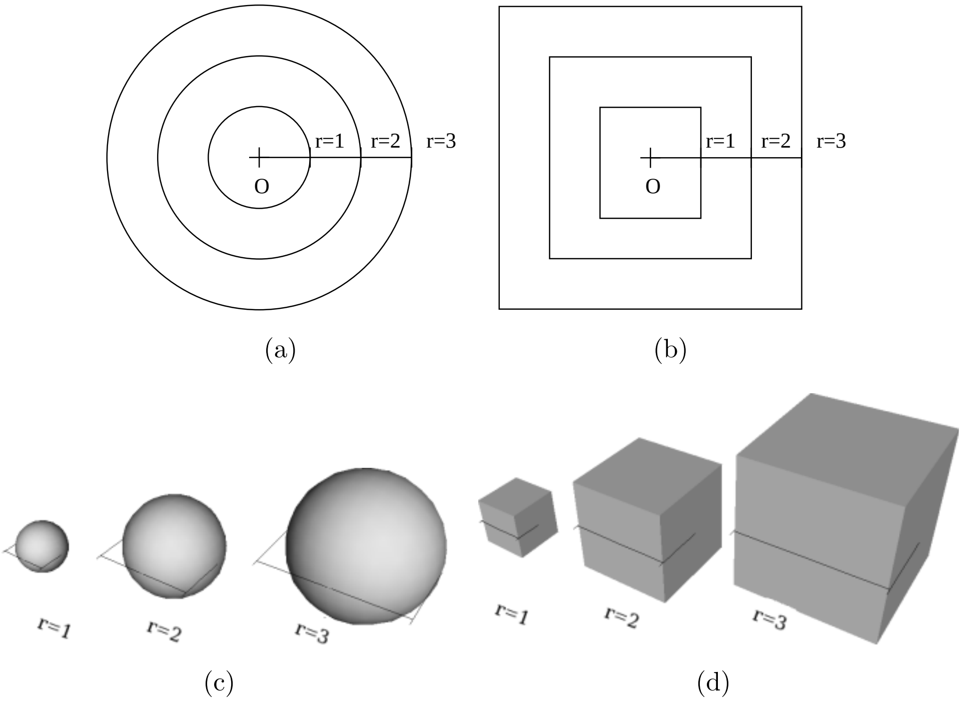


Figure 4.1: Most compact shapes using euclidean and chessboard distances. (a) and (b) show the most compact shapes in 2D (circles and squares) while (c) and (d) shows the most compact shape in 3D (spheres and cubes).

After seeing how the different distance metrics produce different shapes as their *most compact shapes*, it is obvious that the definition of a compactness measure will change depending on the distance metric used for each definition.

4.3 3D Compactness

As it was mentioned in the previous section, when euclidean distance is used, the most compact shape in 3D is a sphere. Following this idea the 3D shape compactness measure should describe the degree to which a given 3D shape differs form a region bounded by a perfect sphere.

For the compactness measure defined in this section central moments $\mu_{p,q,r}$ are used (see equation (1.13)). Notice that for any given shape S, the moment $\mu_{0,0,0}(S)$ equals the volume of such shape.

In order to to keep our proofs mathematically rigorous, it will be assumed that two given shapes S_1 and S_2 are considered to be equal if the symmetric set difference $(S_1 \setminus S_2) \cup (S_2 \setminus S_1)$ has the volume zero, i.e.

$$\iiint\limits_{(S_1 \setminus S_2) \cup (S_2 \setminus S_1)} dxdydz = 0$$

Under this assumption an open ball $\{P = (x, y, z) : l_2(P, O) < r\}$ is equal to the closed ball $\{P = (x, y, z) : l_2(P, O) \leq r\}$ even if they differ for a spherical surface of $\{P = (x, y, z) : l_2(P, O) = r\}$. This condition is not a restriction for image processing applications where volume based shape descriptors are used. It is also assumed that all appearing shapes have a nonempty interior, i.e. they have a strictly positive volume.

We begin with the expression:

$$\frac{\mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,2}(S)}{\mu_{0,0,0}(S)^{5/3}}$$

and prove that it reaches its minimum value of $\dfrac{3^{5/3}}{5(4\pi)^{2/3}}$ only when the given shape S is a sphere.

Theorem 4.1. *Let S be a given 3D shape, then:*

$$\frac{\mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,2}(S)}{\mu_{0,0,0}(S)^{5/3}} \geq \frac{3^{5/3}}{5(4\pi)^{2/3}} \tag{4.5}$$

$$\frac{\mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,3}(S)}{\mu_{0,0,0}(S)^{5/3}} = \frac{3^{5/3}}{5(4\pi)^{2/3}} \Leftrightarrow Sis\ a\ sphere \tag{4.6}$$

Proof. Let S be a given shape whose centroid coincides with the origin. Also, let B_{fit} be the sphere with radius $r = \left(\frac{3}{4\pi} \cdot \mu_{0,0,0}(S)\right)^{1/3}$ and centred at the origin. Trivially $\mu_{0,0,0}(B_{fit}) = \mu_{0,0,0}(S)$ and also:

(a) The set differences $S \setminus B_{fit}$ and $B_{fit} \setminus S$ have equal volume;

(b) The points from $B_{fit} \setminus S$ are closer to the origin than the points from $S \setminus B_{fit}$. More formally:

$$(u, v, w) \in (S \setminus B_{fit}) \text{ and } (x, y, z) \in (B_{fit} \setminus S) \Rightarrow u^2 + v^2 + w^2 > x^2 + y^2 + z^2$$

(see Figure 4.2).

Further, (a) and (b) give:

$$\iiint\limits_{S \setminus B_{fit}} (x^2 + y^2 + z^2) dxdydz \geq \iiint\limits_{B_{fit} \setminus S} (x^2 + y^2 + z^2) dxdydz \tag{4.7}$$

Now, we derive:

$$\begin{aligned}
&\mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,2}(S) = \\
&= \iiint\limits_{S} (x^2 + y^2 + z^2) dxdydz \\
&= \iiint\limits_{S \setminus B_{fit}} (x^2 + y^2 + z^2) dxdydz + \iiint\limits_{S \cap B_{fit}} (x^2 + y^2 + z^2) dxdydz \\
&\geq \iiint\limits_{B_{fit} \setminus S} (x^2 + y^2 + z^2) dxdydz + \iiint\limits_{S \cap B_{fit}} (x^2 + y^2 + z^2) dxdydz \\
&= \iiint\limits_{B_{fit}} (x^2 + y^2 + z^2) dxdydz \\
&= \int\limits_{\theta=0}^{2\pi} \int\limits_{\phi=0}^{\pi} \int\limits_{\rho=0}^{(\frac{3}{4\pi}\mu_{0,0,0}(S))^{1/3}} ((\rho \cdot \cos\theta \sin\phi)^2 + (\rho \cdot \sin\theta \sin\phi)^2 \\
&\quad +(\rho \cdot \cos\phi)^2)\rho^2 \cdot \sin\phi d\theta d\phi d\rho \\
&= \frac{3^{5/3}}{5(4\pi)^{2/3}} \cdot \mu_{0,0,0}(S)^{5/3}
\end{aligned}$$

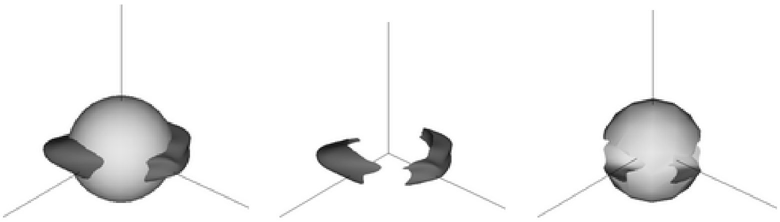
This proves equation (4.5).

To prove equation (4.6), notice that if S is not a sphere then the inequality (4.7) is strict. For any shape S different from a sphere we also have:

$$\mu_{2,0,0}(S)+\mu_{0,2,0}(S)+\mu_{0,0,2}(S) >$$
$$\mu_{2,0,0}(B_{fit})+\mu_{0,2,0}(B_{fit})+\mu_{0,0,2}(B_{fit}) = \frac{3^{5/3}}{5(4\pi)^{2/3}}\cdot\mu_{0,0,0}(S)^{5/3}$$

which establishes the proof.

□



(a) B_{fit} and S (b) $S \setminus B_{fit}$ (c) $B_{fit} \setminus S$

Figure 4.2: (a) shows a given shape S and the sphere B_{fit} which has the same volume as S and both shapes are centred at the origin. (b) and (c) show $S \setminus B_{fit}$ and $B_{fit} \setminus S$ respectively. Any point from $B_{fit} \setminus S$ is closer to the origin than any point from $S \setminus B_{fit}$.

Notice that the quantity

$$\frac{\mu_{2,0,0}(S)+\mu_{0,2,0}(S)+\mu_{0,0,2}(S)}{(\mu_{0,0,0}(S))^{5/3}}$$

is well known from the literature: this quantity appears as one of the geometric invariants derived in Mamistvalov (1998) and Xu and Li (2008). The 3D compactness measure is defined as follows:

Definition 4.1. *Let S be a given 3D shape, the compactness measure $\mathcal{C}_2^{3D}(S)$ is defined as:*

$$\mathcal{C}_2^{3D}(S) = \frac{3^{5/3}}{5(4\pi)^{2/3}}\cdot\frac{\mu_{0,0,0}(S)^{5/3}}{\mu_{2,0,0}(S)+\mu_{0,2,0}(S)+\mu_{0,0,2}(S)} \tag{4.8}$$

The following theorem summarises the desirable properties of $\mathcal{C}_2^{3D}(S)$.

Theorem 4.2. *The compactness measure $\mathcal{C}_2^{3D}(S)$ has the following properties:*

(a) $\mathcal{C}_2^{3D}(S) \in (0,1]$ *for any shape* S*;*

(b) $\mathcal{C}_2^{3D}(S) = 1 \quad \Leftrightarrow \quad S$ *is a sphere;*

(c) $\mathcal{C}_2^{3D}(S)$ *is invariant with respect to the similarity transformations;*

(d) For any $\delta > 0$ *there is a shape* S *such that* $0 < \mathcal{C}_2^{3D}(S) < \delta$ *(i.e. there are shapes whose measured compactness is arbitrarily close to 0, and consequently the best possible lower bound for* $\mathcal{C}_2^{3D}(S)$ *is 0).*

Proof. The items (a) and (b) follow directly from Theorem 4.1. The proof of item (c) is also straightforward: $\mathcal{C}_2^{3D}(S)$ is invariant to similarity transformations because

$$\frac{\mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,2}(S)}{\mu_{0,0,0}(S)^{5/3}}$$

is such an invariant (Mamistvalov, 1998).

To prove (d), let us consider a cuboid $Q(t)$ whose edges are aligned with the coordinate axes and have the length $1, 1$ and t respectively, with t being an arbitrary positive number. It is easy to verify that $\mu_{0,0,0}(Q(t)) = t$, $\mu_{2,0,0}(Q(t)) = t/3$, $\mu_{0,2,0}(Q(t)) = t/3$ and $\mu_{0,0,2}(Q(t)) = t^3/3$. Thus,

$$\begin{aligned} \mathcal{C}_2^{3D}(Q(t)) &= \frac{3^{5/3}}{5(4\pi)^{2/3}} \cdot \frac{\mu_{0,0,0}(Q(t))^{5/3}}{\mu_{2,0,0}(Q(t)) + \mu_{0,2,0}(Q(t)) + \mu_{0,0,2}(Q(t))} \\ &= \frac{3^{8/3}}{5(4\pi)^{2/3}} \cdot \frac{t^{2/3}}{2+t^2} \end{aligned} \tag{4.9}$$

It can be observed that $\lim_{t\to\infty} \mathcal{C}_2^{3D}(Q(t)) = 0$ as $Q(t)$ approximates an infinitely elongated stick; and $\lim_{t\to 0} \mathcal{C}_2^{3D}(Q(t)) = 0$ as $Q(t)$ approximates a planar square. This proves (d). □

4.4 3D Cubeness

As it was mentioned in section 4.2, when chessboard distance is used, the most compact shape in 3D is a cube. So, the 3D cubeness measure presented in this section describes a degree to which a given 3D shape differs from a region bounded by a perfect cube.

For the cubeness measure and other subsequent measures, let $R(S, \theta, \phi)$ denote a shape S rotated first along the x-axis by an angle θ and then along the y-axis by an angle ϕ.

Following the same idea as in section 4.3, we begin to define the cubeness measure by showing that the quantity

$$\min_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz$$

reaches its minimum value of $\frac{3}{8}$ only when the given shape is a cube.

Theorem 4.3. *Let S be a given 3D shape, and let $R(S,\theta,\phi)$ denote the rotation of S along the x-axis by an angle θ and along the y-axis by an angle ϕ. Then,*

$$\frac{\iiint\limits_{S} \max\{|x|,|y|,|z|\}dxdydz}{\mu_{0,0,0}(S)^{4/3}} \geq \frac{3}{8} \tag{4.10}$$

$$\frac{\iiint\limits_{S} \max\{|x|,|y|,|z|\}dxdydz}{\mu_{0,0,0}(S)^{4/3}} = \frac{3}{8} \iff S = Q\left(\frac{\mu_{0,0,0}(S)^{1/3}}{2}\right) \tag{4.11}$$

$$\frac{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz}{\mu_{0,0,0}(S)^{4/3}} = \frac{3}{8} \iff S \textit{ is a cube.} \tag{4.12}$$

Proof. Let S be given 3D shape centred in the origin and let Q_{fit} denote a cube centred in the origin, aligned with the coordinate axes and with sides of length $a = \mu_{0,0,0}(S)^{1/3}$ (so Q_{fit} has the same volume as S). The faces of Q_{fit} intersect the axes at points: $(a/2,0,0)$, $(-a/2,0,0)$, $(0,a/2,0)$, $(0,-a/2,0)$, $(0,0,a/2)$ and $(0,0,-a/2)$. Thus we have that:

(a) The set differences $S \setminus Q_{fit}$ and $Q_{fit} \setminus S$ have equal volume;

(b) The points from $Q_{fit} \setminus S$ are closer to the origin than the points from $S \setminus Q_{fit}$ (note that chessboard distance is considered). More formally:

$$(u,v,w) \in (S \backslash Q_{fit}) \text{ and } (x,y,z) \in (Q_{fit} \backslash S) \Rightarrow \max\{|u|,|v|,|w|\} > \max\{|x|,|y|,|z|\}$$

(see. Figure 4.3).

Further (a) and (b) give:

$$\iiint\limits_{S\backslash Q_{fit}} \max\{|x|,|y|,|z|\}dxdydz \geq \iiint\limits_{Q_{fit}\backslash S} \max\{|x|,|y|,|z|\}dxdydz. \tag{4.13}$$

Now, we derive:

$$\iiint\limits_{S} \max\{|x|,|y|,|z|\}dxdydz = 8 \cdot \iiint\limits_{\substack{(x,y,z)\in S\\ x,y,z\geq 0}} \max\{x,y,z\}dxdydz$$

$$= \quad 48 \cdot \iiint\limits_{\substack{(x,y,z)\in S\\ x\geq y\geq z\geq 0}} xdxdydz = 48 \cdot \int\limits_0^{a/2}\int\limits_0^{x}\int\limits_0^{y} xdxdydz = \frac{3}{8}\cdot a^4,$$

which proves equation (4.10) since $a = \mu_{0,0,0}(S)^{1/3}$.

To prove equation (4.11) notice that equation (4.13) holds only when the shapes S and Q_{fit} are identical, i.e. when $\mu_{0,0,0}(S \setminus Q) = \mu_{0,0,0}(Q \setminus S) = 0$.

To prove equation (4.12) let θ_0 and ϕ_0 be the angles which minimise the quantity $\iiint\limits_{R(S,\theta_0,\phi_0)} \max\{|x|,|y|,|z|\}dxdydz$:

$$\iiint\limits_{R(S,\theta_0,\phi_0)} \max\{|x|,|y|,|z|\}dxdydz = \min_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz \qquad (4.14)$$

Since the volume of S remains is not affected by its orientation (i.e. $\mu_{0,0,0}(S) = \mu_{0,0,0}(R(S,\theta,\phi)) = \mu_{0,0,0}(R(S,\theta_0,\phi_0))$, then (see equation (4.11)):

$$\frac{\iiint\limits_{R(S,\theta_0,\phi_0)} \max\{|x|,|y|,|z|\}dxdydz}{\mu_{0,0,0}(R(S,\theta_0,\phi_0))^{4/3}} = \frac{3}{8}$$

would imply that $R(S,\theta_0,\phi_0)$ must be equal to Q_{fit} and thus S is a cube. □

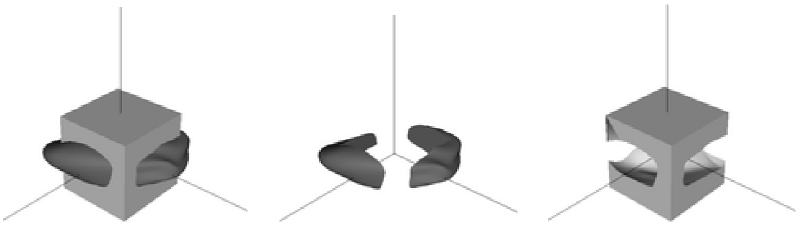

(a) Q_{fit} and S (b) $S \setminus Q_{fit}$ (c) $Q_{fit} \setminus S$

Figure 4.3: (a) shows a given shape S and a cube Q_{fit} which has the same volume as S and both shapes are centred at the origin. (b) and (c) show $S \setminus Q_{fit}$ and $Q_{fit} \setminus S$ respectively. Any point from $Q_{fit} \setminus S$ is closer to the origin than any point from $S \setminus Q_{fit}$ (in terms of chessboard distance).

The 3D cubeness measure is defined as follows:

Definition 4.2. *Given a shape S, the cubeness measure $\mathcal{C}_{\infty}^{3D}(S)$ is defined as:*

$$\mathcal{C}_{\infty}^{3D}(S) = \frac{3}{8} \cdot \frac{\mu_{0,0,0}(S)^{4/3}}{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz} \tag{4.15}$$

The following theorem summarises the desirable properties of $\mathcal{C}_{\infty}^{3D}(S)$.

Theorem 4.4. *The cubeness measure $\mathcal{C}_{\infty}^{3D}(S)$ has the following properties:*

(a) $\mathcal{C}_{\infty}^{3D}(S) \in (0,1]$ *for any shape* S*;*

(b) $\mathcal{C}_{\infty}^{3D}(S) = 1 \quad \Leftrightarrow \quad S$ *is a cube;*

(c) $\mathcal{C}_{\infty}^{3D}(S)$ *is invariant with respect to the similarity transformations;*

Proof. The items (a) and (b) follow directly from Theorem 4.3.

The proof of item (c) is also straight forward: both $\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz$ and $\mu_{0,0,0}(S)$ are rotation invariant, which makes $\mathcal{C}_{\infty}^{3D}(S)$ rotation invariant; because it is assumed for the definition that any given shape S has its centroid located at the origin, then $\mathcal{C}_{\infty}^{3D}(S)$ is translation invariant; finally $\mathcal{C}_{\infty}^{3D}(S)$ is scale invariant because if the shape S is scaled by a factor $\mathbf{r}$ we have that:

$$\begin{aligned} & \min_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(\mathbf{r}\cdot S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz \\ = \mathbf{r}^4 \quad \cdot \quad & \min_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz \end{aligned}$$

and

$$\mu_{0,0,0}(\mathbf{r}\cdot S) = \mathbf{r}^3 \cdot \mu_{0,0,0}(S).$$

These two equalities give:

$$\begin{aligned} \mathcal{C}_{\infty}^{3D}(\mathbf{r}\cdot S) &= \frac{3}{8} \cdot \frac{\mu_{0,0,0}(\mathbf{r}\cdot S)^{4/3}}{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{\mathbf{r}\cdot R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz} \\ &= \frac{3}{8} \cdot \frac{(\mathbf{r}^3 \cdot \mu_{0,0,0}(S))^{4/3}}{\mathbf{r}^4 \cdot \min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}dxdydz} = \mathcal{C}_{\infty}^{3D}(S) \end{aligned}$$

which means that $\mathcal{C}_{\infty}^{3D}(S)$ is scale invariant. □

4.5 Comparable 3D descriptors

In the existing literature, there are a few shape descriptors which have been developed with the aim of providing an index of compactness for shapes, just like the $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$ descriptors in this chapter. It is worth comparing our proposed descriptors with the already existing ones. Some of these descriptors have already been mentioned in section 1.2.2 so this section presents a closer look at these descriptors.

4.5.1 Standard compactness

One such measure is the standard measure of 3D compactness which is an extension of the standard compactness measure (also known as circularity). This measure was introduced in Chapter 1 (see equation (1.15)). The behaviour of the standard 3D compactness measure is comparable to that of the measure $\mathcal{C}_2^{3D}(S)$, reaching its maximum value of 1 only for a sphere. The measure $\mathcal{C}_{Std}^{3D}(S)$ also shares the properties defined in Theorem 4.2.

Although this measure has a simple definition, computing it may be complicated depending on the representation used for the 3D model: approximating the surface area of a voxelised shape is not a straight forward process and requires some additional computation. For this reason the measure $\mathcal{C}_{Std}^{3D}(S)$ was only used in a limited number of the examples presented in section 4.6.

4.5.2 Fitting shape methods

A very natural and simple approach to develop measure of similarity with a sphere (or a cube) could be derived by comparing the given shape S with a sphere (or cube) with the same volume as shape S. The measure using a fit sphere is defined as follows:

Definition 4.3. *Let S be a given 3D shape and B_{fit} be a sphere such that $Volume(B_{fit}) = Volume(S)$ and centred in the origin. The measure $\mathcal{C}_{fit-2}^{3D}(S)$ is defined as:*

$$\mathcal{C}_{fit-2}^{3D}(S) = \frac{Volume(S \cap B_{fit})}{Volume(S \cup B_{fit})} \tag{4.16}$$

It is important to point out that the matching between a given shape S and a cube of equal volume may vary depending on the orientation of S. Therefore, in order to achieve rotational invariability, it is necessary to test for every possible angle. The measure using a fit cube is defined as follows:

Definition 4.4. *Let S be a given 3D shape and Q_{fit} be a cube such that $Volume(Q_{fit}) = Volume(S)$; also, Q_{fit} is assumed to be centred in the origin and aligned with the coordinate axes. The measure $\mathcal{C}^{3D}_{fit-\infty}(S)$ is defined as:*

$$\mathcal{C}^{3D}_{fit-\infty}(S) = \max_{\theta,\phi\in[0,2\pi]} \frac{Volume(S \cap R(Q_{fit},\theta,\phi))}{Volume(S \cup R(Q_{fit},\theta,\phi))} \tag{4.17}$$

The measures $\mathcal{C}^{3D}_{fit-2}(S)$ and $\mathcal{C}^{3D}_{fit-\infty}(S)$ also shares the properties defined in Theorems 4.2 and 4.4.

4.5.3 Digital compactness

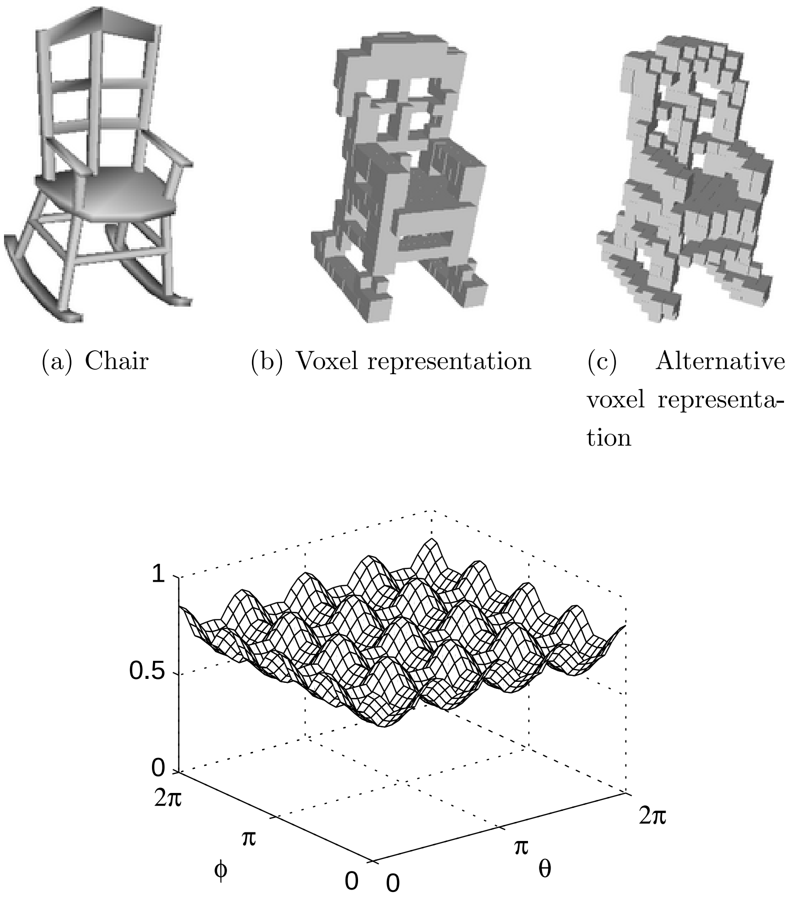
Another of the descriptors briefly presented in section 1.2.2 which could be comparable with the descriptors introduced in this chapter is the measure of discrete compactness $\mathcal{C}_d$ (see equation (1.16)).

Although this descriptor is defined as a measure of discrete compactness (and thus should be comparable to measure $\mathcal{C}^{3D}_2(S)$), its behaviour is more similar to that of measure $\mathcal{C}^{3D}_\infty(S)$ considering the cube as the most compact shape in 3D space.

Ideally, shape descriptors should provide information a given object regardless of their representation. However, measure $\mathcal{C}_d(S)$ is defined in discrete space and therefore shapes must be converted to a voxel representation before the measure can be computed. This additional translation step may induce undesirable errors to the measure $\mathcal{C}_d(S)$, e.g. a coarse voxelisation grid may induce error in the measure.

The measure $\mathcal{C}_d(S)$ shares most of the properties defined in Theorem 4.4, except being rotationally invariant. In fact, the measure $\mathcal{C}_d(S)$ produces different results repending on the orientation of the given shape S. For example, take a 3D shape S in continuous space (like the chair in Figure 4.4(a)), voxelise it and compute its compactness $\mathcal{C}_d(S)$; then rotate S by θ and ϕ voxelise it and compute its compactness $\mathcal{C}_d(R(S,\theta,\phi))$. The two measures are not (necessarily) equal, i.e. it may be the case that $\mathcal{C}_d(S) \neq \mathcal{C}_d(R(S,\theta,\phi))$. Figure 4.4(d) shows a plot of the values produced by $\mathcal{C}_d(R(S,\theta,\phi))$ for $\theta \in (0,2\pi), \phi \in (0,2\pi)$.

This is a major drawback of measure $\mathcal{C}_d(S)$ because it may not produce the results which would be expected. For example, if used to build a 3D search engine it would be unable to properly identify two objects as being equal if they are identical but have been voxelised in different orientations.



(a) Chair

(b) Voxel representation

(c) Alternative voxel representation

(d) Compactness plot

Figure 4.4: (a), (b) and (c) show a chair and two alternative voxel representations. (d) shows the $\mathcal{C}_d(S)$ as a function of $\theta \in [0, 2\pi], \phi \in [0, 2\pi]$. Notice that $\mathcal{C}_d(S)$ is periodic every 90° since $R(S, \theta, \phi)$ repeats itself with this periodicity.

Both chairs in (b) and (c) are sampled to the same voxel resolution, and each chair is made of approximately 500 voxels. The values in figure (d) were calculated using a higher resolution (chairs sampled to around 25,000 voxels).

4.6 Experiments

The first experiment in this section illustrates the behaviour of the measures introduced earlier in this section applied to some geometrical shapes. These geometrical shapes are shown in Figure 4.5 and the computed descriptors for these shapes are listed in Table 4.1.

In accordance with their definition, the measures $\mathcal{C}^{3D}_{Std}(S)$, $\mathcal{C}^{3D}_{2}(S)$, and $\mathcal{C}^{3D}_{fit-2}(S)$

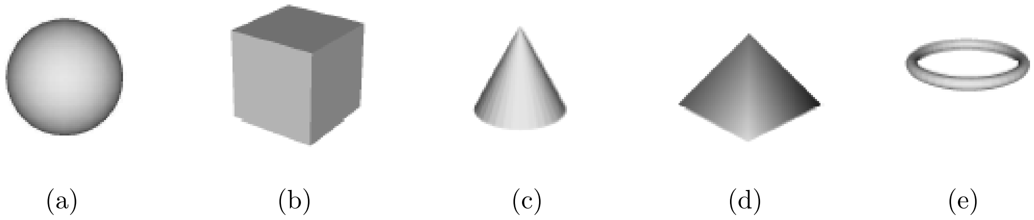


Figure 4.5: Sample basic geometric shapes: (b) a cube, (a) a sphere, (c) a cone, (d) a squared pyramid and (e) a torus. The expected compactness and cubeness values for the cube and the sphere are intuitively clear but not so much for the other geometric shapes.

reach their maximum value of 1 for the sphere in (a). Notice that measure $\mathcal{C}^{3D}_{Std}(S)$ theoretically should give a value of 1 for a perfect sphere; however, the reported value is 0.997. This deviation from the expected value is assumed to be caused by digitisation error. Similarly, the measures $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ reach their maximum value of 1 for the cube in (b).

Also notice that for the torus in (e) measures $\mathcal{C}^{3D}_{fit-2}(S)$ and $\mathcal{C}^{3D}_{fit-\infty}(S)$ give a score of 0.000. This is because the intersection between the torus and the fitting sphere/cube are an empty set, i.e. the sphere/cube which has volume equal to that of the torus fits within the inner hole of the torus without touching the torus at all. This would also be the case for any other shape which has a hole large enough (around its centroid) to encircle the fit sphere/cube. Therefore the measures $\mathcal{C}^{3D}_{fit-2}(S)$ and $\mathcal{C}^{3D}_{fit-\infty}(S)$ would be unable to make any distinction between two shapes with such characteristic.

	$\mathcal{C}^{3D}_{Std}(S)$	$\mathcal{C}^{3D}_{2}(S)$	$\mathcal{C}^{3D}_{\infty}(S)$	$\mathcal{C}^{3D}_{fit-2}(S)$	$\mathcal{C}^{3D}_{fit-\infty}(S)$	$\mathcal{C}_d(S)$
(a)	0.997	1.000	0.971	1.000	0.723	0.991
(b)	0.524	0.925	1.000	0.726	1.000	1.000
(c)	0.471	0.845	0.812	0.618	0.549	0.734
(d)	0.308	0.747	0.795	0.532	0.443	0.754
(e)	0.069	0.116	0.340	0.000	0.000	0.785

Table 4.1: Different compactness and cubeness values for the geometric shapes from Figure 4.5. Notice that a small deviation from the expected values might be induced due to digitisation error.

It may be considered advantageous that the scores given by measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$ span a wider range than the scores given by $\mathcal{C}_d(S)$, i.e. the scores obtained for the shapes in Figure 4.5 by measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$ are in the range $[0.116, 1.000]$ and $[0.340, 1.000]$ respectively, while the scores obtained by $\mathcal{C}_d(S)$ are in the range $[0.734, 1.000]$. This indicates that $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$ assign more distinctive values than $\mathcal{C}_d(S)$, thus making it easier to differentiate between these shapes.

Figure 4.6 illustrates the behaviour of the compactness measures with respect to erosion. Starting from a cube, on each step of the erosion process pieces of the 3D object are progressively cut out. As it would be expected the scores produced by all measures, with exception of $\mathcal{C}_d(S)$, decrease as the level of erosion increases.

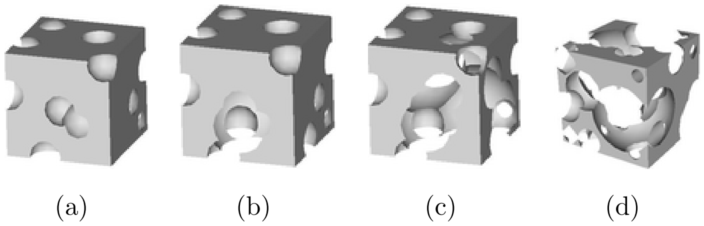


Figure 4.6: Cube under different levels of erosion. On each step of the erosion process, a portion of the original shape is carved out reducing the total volume of the shape and altering its enclosing surface area.

In this example, the measure $\mathcal{C}_d(S)$ remains almost unchanged regardless of the level of erosion. This may be due to the fact that measure $\mathcal{C}_d(S)$ is a voxel based approach and thus it is sensitive to errors induced by the voxelisation process. It is well known that the resolution level used when converting an object to voxel representation may be critical in subsequent processing. In this case, insufficient resolution may cause that small levels of erosion go unnoticed: removing a very small part of the shape may not cause any voxels to be removed from the voxel representation and therefore the score produced by the measure $\mathcal{C}_d(S)$ would remain the same.

The cubes in Figure 4.6 have a resolution of 60 voxels per side, with a perfect cube being made up of 216,000 voxels. At the first stage of erosion shown in (a), around 10% of the total number of voxels has been removed; for the following stages of erosion in (b), (c) and (d), around 20% of the initial volume is removed in each stage.

The standard compactness measure $\mathcal{C}_{Std}^{3D}(S)$ decays very rapidly, producing values

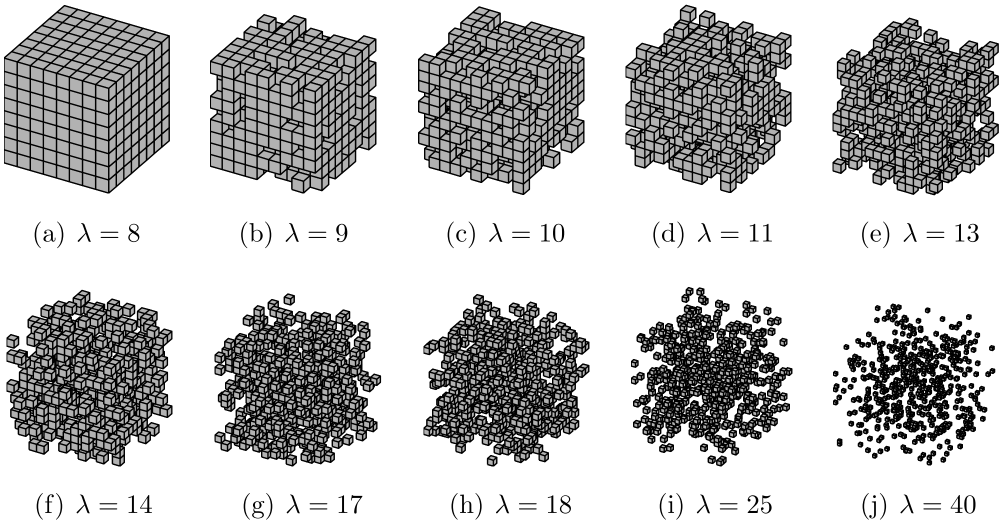
	$\mathcal{C}^{3D}_{Std}(S)$	$\mathcal{C}^{3D}_{2}(S)$	$\mathcal{C}^{3D}_{\infty}(S)$	$\mathcal{C}^{3D}_{fit-2}(S)$	$\mathcal{C}^{3D}_{fit-\infty}(S)$	$\mathcal{C}_{d}(S)$
(a)	0.174	0.855	0.961	0.488	0.784	0.684
(b)	0.092	0.668	0.850	0.342	0.517	0.672
(c)	0.052	0.452	0.708	0.148	0.272	0.681
(d)	0.012	0.239	0.507	0.004	0.026	0.642

Table 4.2: Different compactness and cubeness values for the shapes in Figure 4.6.

very close to zero even for the lowest level of erosion. Such behaviour is expected since the volume of the shape decreases while the surface area increases as it becomes more intricate.

Notice also that both shape fitting methods, $\mathcal{C}^{3D}_{fit-2}(S)$ and $\mathcal{C}^{3D}_{fit-\infty}(S)$, decay more rapidly than the measures $\mathcal{C}^{3D}_{2}(S)$ and $\mathcal{C}^{3D}_{\infty}(S)$. This suggests the robustness with respect to noise of the measures $\mathcal{C}^{3D}_{2}(S)$ and $\mathcal{C}^{3D}_{\infty}(S)$. These measures are able to perceive small variations in the shape while still producing reasonable results in the presence of noise. Such robustness is generally expected from volume based measures such as these.

Figure 4.7 illustrates a scenario where measures $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_{d}(S)$ decay more rapidly than measure $\mathcal{C}^{3D}_{\infty}(S)$. This also shows that measure $\mathcal{C}^{3D}_{\infty}(S)$ is more suitable than the other two, for objects which are composed of several disconnected parts.


(a) $\lambda = 8$ (b) $\lambda = 9$ (c) $\lambda = 10$ (d) $\lambda = 11$ (e) $\lambda = 13$

(f) $\lambda = 14$ (g) $\lambda = 17$ (h) $\lambda = 18$ (i) $\lambda = 25$ (j) $\lambda = 40$

Figure 4.7: Objects composed of 512 voxels randomly distributed over an increasingly larger bounding box. As the size of the bounding box increases, the voxels are more spaced out while still roughly resembling a cube.

Each of the objects shown in Figure 4.7 is composed of an equal number of voxels (512 voxels) randomly distributed inside a cubical bounding box having side length λ. Table 4.3 shows the scores of $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ for each of these objects.

From their definitions, all three measures give a score of 1.000 to the cube in (a), however, as it can be seen from Table 4.3 measures $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ approach zero quicker than $\mathcal{C}^{3D}_{\infty}(S)$. In fact, both measures $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ can give a value of zero for any shape given certain circumstances: measure $\mathcal{C}^{3D}_{fit-\infty}(S)$ gives zero for any shape where there is an empty space around the centroid of the shape large enough to contain the fitting cube; measure $\mathcal{C}_d(S)$ gives zero for any shape where all voxels which compose the shape are disconnected from each other, regardless of their relative distribution.

	$\mathcal{C}^{3D}_{\infty}(S)$	$\mathcal{C}^{3D}_{fit-\infty}(S)$	$\mathcal{C}_d(S)$
(a)	1.000	1.000	1.000
(b)	0.904	0.702	0.833
(c)	0.806	0.428	0.656
(d)	0.743	0.319	0.531
(e)	0.617	0.164	0.308
(f)	0.564	0.112	0.253
(g)	0.469	0.078	0.138
(h)	0.443	0.060	0.128
(i)	0.326	0.027	0.034
(j)	0.200	0.004	0.010

Table 4.3: Different cubeness values for shapes in Figure 4.7.

In accordance with its definition, measure $\mathcal{C}^{3D}_{\infty}(S)$ continues to produce values which are arbitrarily close to zero as the size λ of the bounding box continues to increase. For any given size of the bounding box, measure $\mathcal{C}^{3D}_{\infty}(S)$ would still produce different non-zero values depending on the voxel's relative positions.

Figure 4.8 shows a graph of the obtained measures $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ as a function of λ. Notice how both $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ measures already have values very close to zero for $\lambda \geq 20$ while measure $\mathcal{C}^{3D}_{\infty}(S)$ continues to decrease, asymptotically approaching zero as the values of λ continue to increase. This smooth decay of the values of $\mathcal{C}^{3D}_{\infty}(S)$ is preferable to the more abrupt decay exhibited by the other two measures.

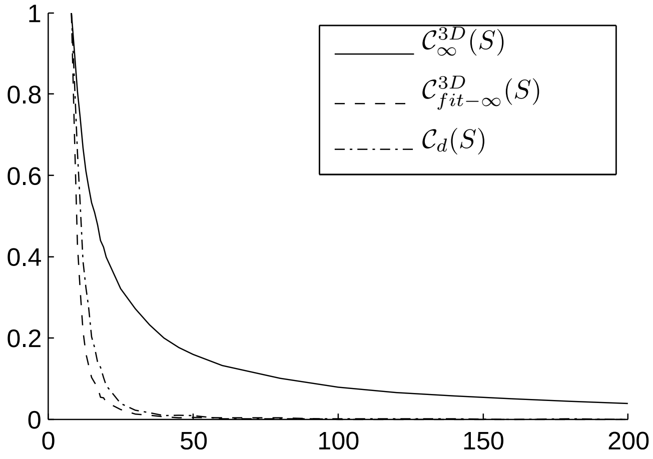


Figure 4.8: Measures $\mathcal{C}_{\infty}^{3D}(S)$, $\mathcal{C}_{fit-\infty}^{3D}(S)$ and $\mathcal{C}_d(S)$ over a range of λ bounding box sizes.

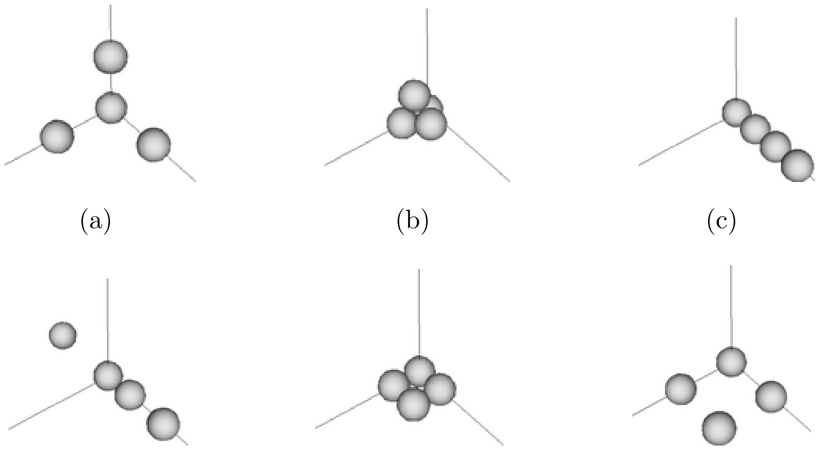
Figure 4.9 shows some objects which are composed of multiple disconnected objects. This example illustrates the influence that the relative position of the spheres has on the computed measures. The relative position of elements in a group may be relevant when in describing the group as a whole as it could be the case with a flocks of birds or a school of fish.

Each object in Figure 4.9 is composed of four isometric spheres (S_1, S_2, S_3, S_4) located in different positions in space and which never intersect with each other. The four spheres are considered to be part of a single object: $(S = S_1 \cup S_2 \cup S_3 \cup S_4)$.

It is clear that all shapes in Figure 4.9 have the same surface area and volume. Because the measure $\mathcal{C}_{Std}^{3D}(S)$ is computed from these two quantities, it remains constant for all objects (a)-(f) (see Table 4.4). This is also the case for measure $\mathcal{C}_d(S)$. This inability to detect changes in the relative position of the elements of a group suggests that the measures $\mathcal{C}_{Std}^{3D}(S)$ and $\mathcal{C}_d(S)$ are not suitable for analysis of multi-component objects. Notice that for measure $\mathcal{C}_d(S)$ there is a slight variation on the values in Table 4.4 induced by the voxelisation process.

Measures $\mathcal{C}_{fit-2}^{3D}(S)$ and $\mathcal{C}_{fit-\infty}^{3D}(S)$ can detect changes in relative position of the individual elements of a group, producing different values for each shape in Figure 4.9. However, as it has been mentioned before, they will produce a zero for any shape S where the fitting sphere/cube does not intersect with the shape S. This is the case of the object (f), where the spheres are arranged in a square pattern; the fitting sphere/cube would be located in the middle of the four spheres and it would not intersect them.

Measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_{\infty}^{3D}(S)$ are capable of detecting changes in relative position



(a) (b) (c)

(d) (e) (f)

Figure 4.9: Multi-component shapes composed of fours spheres of equal volume in different relative positions. Notice that the surface area and volume of these shapes remains constant.

	$\mathcal{C}^{3D}_{Std}(S)$	$\mathcal{C}^{3D}_{2}(S)$	$\mathcal{C}^{3D}_{\infty}(S)$	$\mathcal{C}^{3D}_{fit-2}(S)$	$\mathcal{C}^{3D}_{fit-\infty}(S)$	$\mathcal{C}_{d}(S)$
(a)	0.250	0.226	0.494	0.076	0.069	0.963
(b)	0.250	0.818	0.915	0.578	0.529	0.966
(c)	0.250	0.299	0.591	0.268	0.234	0.965
(d)	0.250	0.105	0.382	0.193	0.180	0.961
(e)	0.250	0.632	0.812	0.393	0.396	0.967
(f)	0.250	0.102	0.352	0.000	0.000	0.963

Table 4.4: Different compactness and cubeness values for the shapes in Figure 4.9.

of the individual elements of a group and will produce a non-zero value for any given shape. The behaviour of these measures also matches our natural preception, giving a higher measure to the shapes when the spheres are closer together than when they are spread out. This can be appreciated in the shapes (c) and (d) where all four spheres are co-linear but the distance between them changes; similarly happens with shapes (e) and (f) where spheres are coplanar but the distance between them again changes.

Figure 4.10 shows some voxelised 3D models of real world objects[1]. Table 4.5 summarises the compactness measures for these objects. As it has been mentioned

[1] Some of these shapes are taken from the McGill database: http://www.cim.mcgill.ca/~shape/benchMark/

before, computing measure $\mathcal{C}^{3D}_{Std}(S)$ from the voxel representation of an object would require additional computations and due to the complexity of this process it is avoided in this experiment.

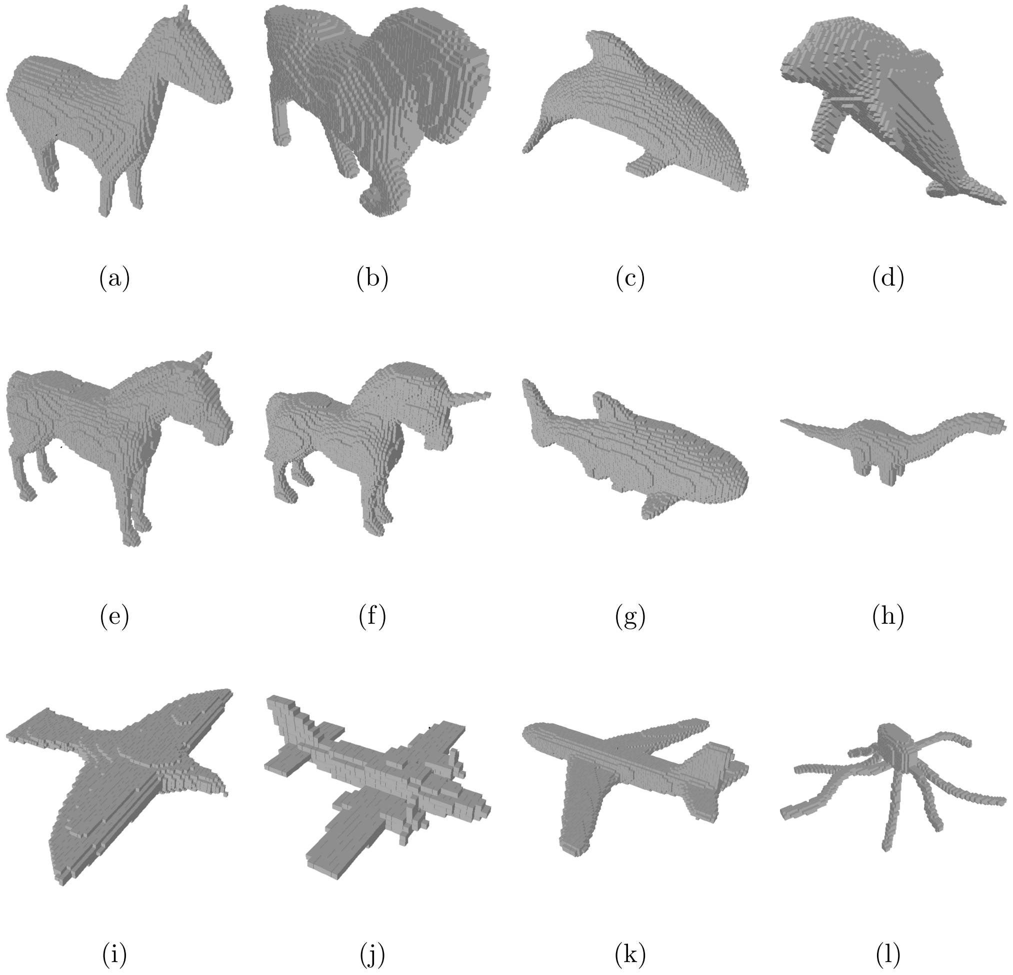


Figure 4.10: Voxelised 3D model of real world objects: different species of animals, two airplanes and a mushroom.

From a human perception point of view, it is obvious that objects (j) and (k) are similar because they are both airplanes. So it seems reasonable that measures $\mathcal{C}^{3D}_{2}(S)$, $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-2}(S)$ and $\mathcal{C}^{3D}_{fit-\infty}(S)$ give very similar scores to these two objects as it can be seen in Table 4.5. It also seems reasonable that the object in (i) has a similar ranking according to measure $\mathcal{C}^{3D}_{2}(S)$: the bird has a score of 0.237, while the airplanes have 0.224 and 0.205 respectively. This could be considered to be in accordance with human perception since airplanes and birds have similar shapes.

It also seems reasonable that measure $\mathcal{C}^{3D}_{2}(S)$ gives the lowest compactness

	$\mathcal{C}_2^{3D}(S)$	$\mathcal{C}_\infty^{3D}(S)$	$\mathcal{C}_{fit-2}^{3D}(S)$	$\mathcal{C}_{fit-\infty}^{3D}(S)$	$\mathcal{C}_d(S)$
(a)	0.435	0.745	0.347	0.333	0.976
(b)	0.421	0.726	0.325	0.306	0.973
(c)	0.408	0.724	0.346	0.305	0.976
(d)	0.382	0.730	0.342	0.301	0.973
(e)	0.380	0.702	0.313	0.291	0.967
(f)	0.364	0.684	0.316	0.290	0.962
(g)	0.377	0.739	0.372	0.347	0.972
(h)	0.281	0.677	0.409	0.366	0.944
(i)	0.237	0.582	0.260	0.241	0.932
(j)	0.224	0.533	0.186	0.180	0.847
(k)	0.205	0.522	0.188	0.179	0.946
(l)	0.139	0.460	0.220	0.205	0.859

Table 4.5: Compactness measures for shapes in Figure 4.10. Notice that measure $\mathcal{C}_{Std}^{3D}(S)$ is not computed because the objects are in voxel representation.

amongst these object to the octopus shape in (l): the octopus could be perceived as being the least compact object from the shapes in Figure 4.10 because it has its arms stretching out of away from the main mass of its body.

Notice that the range of values covered by the measure $\mathcal{C}_d(S)$ (i.e. $[0.847, 0.976]$) is narrower than the range of any other measure (e.g. $[0.139, 0.612]$ for measure $\mathcal{C}_2^{3D}(S)$). Such a short range of values produced by measure $\mathcal{C}_d(S)$ implies that it is difficult to differentiate between the objects in Figure 4.10 using this measure alone.

It is also in accordance with our expectations that both dolphins, (c) and (d), and the fish (g) have similar scores with respect of measure $\mathcal{C}_\infty^{3D}(S)$, having their cubeness scores being in the range $[0.724, 0.739]$.

Looking at the shapes in Figure 4.10 and at the values in Table 4.5 it seems to be that there is an overlap between the values of the horses ((a), (b), (e) and (f)) and the marine creatures ((c), (d) and (g)). It would be desirable to be able to separate these two groups of shapes.

Using measure $\mathcal{C}_2^{3D}(S)$, shapes the horses ((a), (b), (e) and (f)) are spread in a wide range, which completely overlaps with that of the marine creatures ((c), (d) and (g)); a similar thing happens for descriptor $\mathcal{C}_\infty^{3D}(S)$. Although it is assumed that members of the same class should have similar scores for the same descriptor, some classes may have more variability within members of the same class. In this

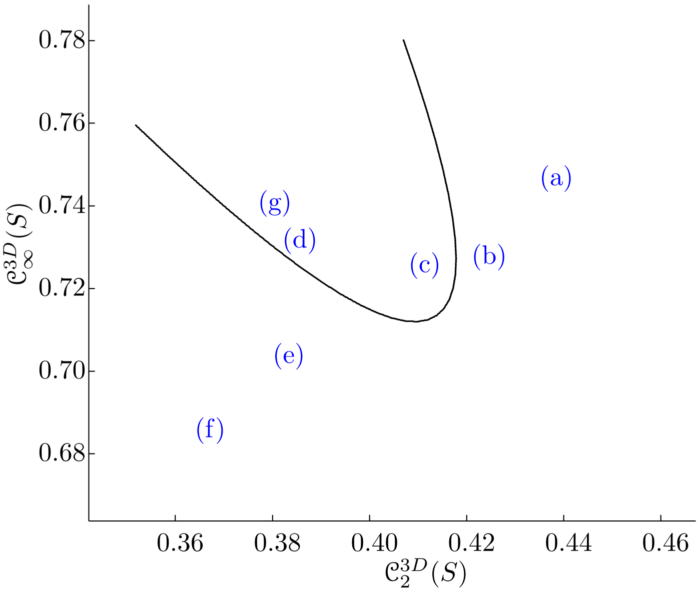


Figure 4.11: Plot of features $\mathcal{C}_2^{3D}(S)$ vs. $\mathcal{C}_\infty^{3D}(S)$. Some objects from Figure 4.10 are located in this feature space.

case, horses span a wider range than marine creatures.

Even though neither of the shape descriptors shown in Table 4.5 can achieve such a separation, it is possible to separate these two groups of shapes by combining the measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$.

When both descriptors $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$ are combined, a two dimensional feature space is created. Figure 4.11 shows a plot of the feature space created by combining these two measures and locates the horses and marine creatures ((a), (b), (c), (d), (e), (f) and (g)) in this feature space. As it can be seen in Figure 4.11, a simple curve can easily split these shapes in two groups in this feature space.

However, there are still some limitations: (c) is closer to (b) than it is to (d) and (g). Also, class overlap would still exist if elements from other classes were to be included; furthermore, there is no guarantee that more items of the same classes would be properly classified by this decision boundary. One way to work around this problem would be to include more descriptors, thus creating a larger feature space where a better class separation may be achieved.

4.7 Conclusions

In this chapter, the Euclidean and chessboard distance metrics are used to derive the 3D compactness and cubeness descriptors. Both descriptors are derived in a similar way, and both provide an indication of similarity with the most compact shape according to the metric used: using the Euclidean distance, the most compact shape is a sphere; using the chessboard distance, the most compact shape is a cube.

The new measures are compared with some similar existing descriptors from the literature. Examples illustrate the behaviour of the new measures, their robustness to noise and their suitability for classification applications.

As mentioned before, the development of 3D shape descriptors is still a topic of interest in the field of image processing. Applications like those presented by Funkhouser et al. (2003), Tangelder and Veltkamp (2004) and Bustos et al. (2005), provide excellent examples where these methods could be applied. The compactness and cubeness measures can be incorporated to the feature sets used in these applications. An increase in retrieval performance would be expected as a consequence of the use of an extended descriptor set.

In the next chapter, these descriptors are modified to incorporate different weighting parameters which allows tuning of their behaviour.

Chapter 5

Tuneable 3D Descriptors

This chapter includes material from:

> *Žunić, J., Hirota, K., and Martinez-Ortiz, C. (2010a). Compactness measure for 3d shapes.* Submitted
>
> *Martinez-Ortiz, C. and Žunić, J. (2010b). A family of cubeness measures.* Submitted
>
> *Martinez-Ortiz, C. and Žunić, J. (2010c). Measuring cubeness in the limit cases.* Submitted

and

> *Martinez-Ortiz, C. and Žunić, J. (2010d). Tuneable cubeness measures for 3d shapes.* Submitted

5.1 Introduction

Chapter 2 introduced two weighted orientation methods for 2D shapes: section 2.3.1 introduced the idea of modifying a shape orientation method by taking into account the influence that each individual point has on the computed orientation; similarly, the modified method introduced in section 2.3.2 computes the orientation by taking into account the influence of boundary points, weighted using a function of the curvature. These methods include additional parameters which allow for certain shape features to be highlighted or ignored, thus tuning the behaviour of the orientation methods. This may be desirable to suit specific needs in a particular application.

Following the idea of tuneable orientation measures introduced in Chapter 2, this chapter introduces tuneable modifications of the 3D shape descriptors introduced in Chapter 4. These modifications incorporate additional parameters which allow tuning of the behaviour of the 3D shape descriptors.

5.2 Position-dependent compactness

This section introduces a modification of the compactness measure $\mathcal{C}_2^{3D}(S)$ introduced in Chapter 4 which takes into account the impact that each individual point has on the computed measure.

From equation (4.8) it can be seen that each individual point $(x, y, z) \in S$ contributes to $\mathcal{C}_2^{3D}(S)$ by $x^2+y^2+z^2$. This indicates that points which are closer to the origin have a lower influence on the computed compactness and points which are far away have a greater influence. However, this may not necessarily be desirable in every situation and it may be preferred to have some control on the impact that the distance to the origin has over the computed compactness measure.

To this end, the definition of measure $\mathcal{C}_2^{3D}(S)$ is modified to include an additional parameter, β, which provides the means to control the influence that the point's position has on the measured compactness.

Before the modified measure of compactness can be defined, the following lemma is given:

Lemma 5.1. *Given a shape S whose centroid coincides with the origin, and a constant $\beta > 0$. Then:*

$$\frac{\iiint_S (x^2+y^2+z^2)^\beta dxdydz}{\mu_{0,0,0}(S)^{(2\beta+3)/3}} \geq \frac{3}{2\beta+3}\left(\frac{3}{4\pi}\right)^{2\beta/3} \tag{5.1}$$

$$\frac{\iiint_S (x^2+y^2+z^2)^\beta dxdydz}{\mu_{0,0,0}(S)^{(2\beta+3)/3}} = \frac{3}{2\beta+3}\left(\frac{3}{4\pi}\right)^{2\beta/3} \Leftrightarrow S is\ a\ sphere \tag{5.2}$$

Lemma 5.1 can be proven in a similar manner as Theorem 4.1:

Proof. Let S be a given shape whose centroid coincides with the origin and B_{fit} be the sphere with radius $r = \left(\frac{3}{4\pi}\cdot\mu_{0,0,0}(S)\right)^{1/3}$ and centred at the origin. Since β is assumed to be a positive number:

$$(u, v, w) \in (S \setminus B_{fit}) \text{ and } (x, y, z) \in (B_{fit} \setminus S) \Rightarrow (u^2+v^2+w^2)^\beta > (x^2+y^2+z^2)^\beta$$

which gives immediately:

$$\iiint\limits_{S\setminus B_{fit}} (x^2+y^2+z^2)^\beta dxdydz \geq \iiint\limits_{B_{fit}\setminus S} (x^2+y^2+z^2)^\beta dxdydz \tag{5.3}$$

and further:

$$\begin{aligned}
&\iiint\limits_{S} (x^2+y^2+z^2)^\beta dxdydz = \\
&= \iiint\limits_{S\setminus B_{fit}} (x^2+y^2+z^2)^\beta dxdydz + \iiint\limits_{S\cap B_{fit}} (x^2+y^2+z^2)^\beta dxdydz \\
&\geq \iiint\limits_{B_{fit}\setminus S} (x^2+y^2+z^2)^\beta dxdydz + \iiint\limits_{S\cap B_{fit}} (x^2+y^2+z^2)^\beta dxdydz \\
&= \iiint\limits_{B_{fit}} (x^2+y^2+z^2)^\beta dxdydz \\
&= \int\limits_{\theta=0}^{2\pi}\int\limits_{\phi=0}^{\pi}\int\limits_{\rho=0}^{\left(\frac{3\cdot\mu_{0,0,0}(S)}{4\pi}\right)^{1/3}} ((\rho\cdot\cos\theta\sin\phi)^2+(\rho\cdot\sin\theta\sin\phi)^2 \\
&\quad +(\rho\cdot\cos\phi)^2)^\beta\rho^2\cdot\sin\phi d\theta d\phi d\rho \\
&= \frac{3}{2\beta+3}\left(\frac{3}{4\pi}\right)^{2\beta/3}\cdot\mu_{0,0,0}(S)^{(2\beta+3)/3}
\end{aligned}$$

This proves equation (5.1).

To prove equation (5.2) is enough to notice that if S is not a sphere then the inequality (5.3) is strict. Consequently,

$$\iiint\limits_{S} (x^2+y^2+z^2)^\beta dxdydz > \iiint\limits_{B_{fit}} (x^2+y^2+z^2)^\beta dxdydz$$

for any shape different from a sphere. □

In Lemma 5.1 the values of β were assumed to be positive. However, this is not a restriction and negative values can be considered as well. The following lemma considers $-1<\beta<0$. Notice that $\beta>-1$ is assumed in order to preserve the convergence of the integrals. Also notice that, according to this lemma, for $-1<\beta<0$ the expression:

$$\frac{\iiint\limits_{S}(x^2+y^2+z^2)^\beta dxdydz}{\mu_{0,0,0}(S)^{(2\beta+3)/3}}$$

reaches its maximum value of $\dfrac{3}{2\beta+3}\left(\dfrac{3}{4\pi}\right)^{2\beta/3}$ only when the given shape S is a sphere.

Lemma 5.2. *Given a shape S whose centroid coincides with the origin and a constant β such that $-1 < \beta < 0$. Then:*

$$\frac{\iiint\limits_S (x^2+y^2+z^2)^\beta dxdydz}{\mu_{0,0,0}(S)^{(2\beta+3)/3}} \leq \frac{3}{2\beta+3}\left(\frac{3}{4\pi}\right)^{2\beta/3} \tag{5.4}$$

$$\frac{\iiint\limits_S (x^2+y^2+z^2)^\beta dxdydz}{\mu_{0,0,0}(S)^{(2\beta+3)/3}} = \frac{3}{2\beta+3}\left(\frac{3}{4\pi}\right)^{2\beta/3} \Leftrightarrow S \textit{is a sphere} \tag{5.5}$$

The proof of Lemma 5.2 is analogous to the proof of Lemma 5.1 and thus it is omitted.

Based on the two lemmas above, the following definition is given for the position-dependent compactness measure:

Definition 5.1. *Given a shape S with its centroid located at the origin and a real number β such that $-1 < \beta$ and $\beta \neq 0$, the compactness measure $\mathcal{C}_2^{3D}(S,\beta)$ is defined as:*

$$\mathcal{C}_2^{3D}(S,\beta) = \begin{cases} \dfrac{3}{2\beta+3}\left(\dfrac{3}{4\pi}\right)^{2\beta/3} \cdot \dfrac{\mu_{0,0,0}(S)^{(2\beta+3)/3}}{\iiint\limits_S (x^2+y^2+z^2)^\beta dxdydz} & \beta > 0 \\ \dfrac{2\beta+3}{3}\left(\dfrac{4\pi}{3}\right)^{2\beta/3} \cdot \dfrac{\iiint\limits_S (x^2+y^2+z^2)^\beta dxdydz}{\mu_{0,0,0}(S)^{(2\beta+3)/3}} & -1 < \beta < 0 \end{cases} \tag{5.6}$$

The modified measure $\mathcal{C}_2^{3D}(S,\beta)$ is a generalisation of the previously defined measure $\mathcal{C}_2^{3D}(S)$. It can be said that measure $\mathcal{C}_2^{3D}(S)$ is a particular case of $\mathcal{C}_2^{3D}(S,\beta)$ with $\beta = 1$. The measure $\mathcal{C}_2^{3D}(S,\beta)$ still complies with the desirable properties established by Theorem 4.2.

For lower values of β, the measure $\mathcal{C}_2^{3D}(S,\beta)$ gives higher impact to points which are closer to the shape centroid; on the other hand for higher values of β, a higher impact is given to the points which are further away from the centroid. This means that $\mathcal{C}_2^{3D}(S,\beta)$ penalises deviations from a perfect sphere for large values of β. This property could be used to detect small deviations from a perfect sphere, provided that the selected value of β is large enough. Trivially, if S is a perfect sphere, $\mathcal{C}_2^{3D}(S,\beta) = 1$ for any given value of β.

5.3 Position-dependent cubeness

In the previous section, measure $\mathcal{C}_2^{3D}(S)$ is modified to include an additional parameter β to modify its behaviour. In a similar way, in this section measure $\mathcal{C}_\infty^{3D}(S)$ is also modified to include a tuning parameter β which controls the impact that each point has on the computed cubeness.

Before the modified cubeness measure can be defined, the following lemma is given:

Lemma 5.3. *Given a shape S whose centroid coincides with the origin, let $R(S, \theta, \phi)$ denote the shape S rotated along the x-axis by an angle θ and along the y-axis by an angle ϕ. Then:*

$$\frac{\iiint\limits_S \max\{|x|,|y|,|z|\}^\beta dxdydz}{\mu_{0,0,0}(S)^{(\beta+3)/3}} \geq \frac{3}{2^\beta(\beta+3)} \tag{5.7}$$

$$\frac{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}^\beta dxdydz}{\mu_{0,0,0}(S)^{(\beta+3)/3}} = \frac{3}{2^\beta(\beta+3)} \Leftrightarrow S \textit{is a cube.} \tag{5.8}$$

Proof. Let S be a given shape centred in the origin and let Q_{fit} be a cube centred in the origin, aligned with the coordinate axes and whose volume is equal to that of S. Also, for $\beta > 0$:

$$(u,v,w) \in (S \backslash Q_{fit}) \text{ and } (x,y,z) \in (Q_{fit} \backslash S) \Rightarrow \max\{|u|,|v|,|w|\}^\beta > \max\{|x|,|y|,|z|\}^\beta$$

which gives immediately:

$$\iiint\limits_{S\backslash Q_{fit}} \max\{|x|,|y|,|z|\}^\beta dxdydz \geq \iiint\limits_{Q_{fit}\backslash S} \max\{|x|,|y|,|z|\}^\beta dxdydz \tag{5.9}$$

and further:

$$\begin{aligned}
&\iiint\limits_S \max\{|x|,|y|,|z|\}^\beta dxdydz \\
&= \iiint\limits_{S\backslash Q_{fit}} \max\{|x|,|y|,|z|\}^\beta dxdydz + \iiint\limits_{S\cap Q_{fit}} \max\{|x|,|y|,|z|\}^\beta dxdydz \\
&\geq \iiint\limits_{Q_{fit}\backslash S} \max\{|x|,|y|,|z|\}^\beta dxdydz + \iiint\limits_{S\cap Q_{fit}} \max\{|x|,|y|,|z|\}^\beta dxdydz \\
&= \iiint\limits_{Q_{fit}} \max\{|x|,|y|,|z|\}^\beta dxdydz
\end{aligned}$$

$$= 48 \cdot \int_0^{1/2} \int_0^{x} \int_0^{y} x^\beta dx dy dz$$
$$= \frac{3}{2^\beta(\beta+3)}$$

This proves (5.7).

To prove (5.8) is enough to notice that inequality (5.9) is strict if shapes S and Q_{fit} are different, and therefore,

$$\iiint_S \max\{|x|,|y|,|z|\}^\beta dx dy dz > \frac{3}{2^\beta(\beta+3)}$$

for any shape different from the cube. □

Notice that according to Lemma 5.3 the expression:

$$\frac{\iiint_S \max\{|x|,|y|,|z|\}^\beta dx dy dz}{\mu_{0,0,0}(S)^{(\beta+3)/3}}$$

reaches its minimum possible value of $\frac{3}{2^\beta(\beta+3)}$ only when the given shape S is a cube centred in the origin and aligned with the ordinate axes.

The following lemma considers negative values of β. Notice that according to this lemma, for $-1 < \beta < 0$ the expression

$$\iiint_S \max\{|x|,|y|,|z|\}^\beta dx dy dz$$

reaches its maximum value of $\frac{3}{2^\beta(\beta+3)}$ only when the given shape S is a suitably oriented cube.

Lemma 5.4. *Given a shape S whose centroid coincides with the origin and a constant β such that $-1 < \beta < 0$, then:*

$$\frac{\iiint_S \max\{|x|,|y|,|z|\}^\beta dx dy dz}{\mu_{0,0,0}(S)^{(\beta+3)/3}} \leq \frac{3}{2^\beta(\beta+3)} \tag{5.10}$$

$$\frac{\min_{\theta,\phi\in[0,2\pi]} \iiint_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}^\beta dx dy dz}{\mu_{0,0,0}(S)^{(\beta+3)/3}} = \frac{3}{2^\beta(\beta+3)} \Leftrightarrow Sis\ a\ cube. \tag{5.11}$$

The proof of the Lemma 5.4 is analogous to the proof of Lemma 5.3 and thus it is omitted.

Based on the two lemmas above, the following definition is given for the position-dependent cubeness measure:

Definition 5.2. *Given a shape S whose centroid coincides with the origin and a real valued β such that $-1 < \beta$ and $\beta \neq 0$, the cubeness measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ is defined as:*

$$\mathcal{C}_{\infty}^{3D}(S,\beta) = \begin{cases} \dfrac{3}{2^{\beta}(\beta+3)} \cdot \dfrac{\mu_{0,0,0}(S)^{(\beta+3)/3}}{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}^{\beta} dxdydz} & \beta > 0 \\ \dfrac{2^{\beta}(\beta+3)}{3} \cdot \dfrac{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|,|y|,|z|\}^{\beta} dxdydz}{\mu_{0,0,0}(S)^{(\beta+3)/3}} & -1 < \beta < 0 \end{cases} \tag{5.12}$$

Again, similarly to $\mathcal{C}_{2}^{3D}(S,\beta)$, the modified measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ is a generalisation of the previously defined measure $\mathcal{C}_{\infty}^{3D}(S)$ and it can be said that $\mathcal{C}_{\infty}^{3D}(S) = \mathcal{C}_{\infty}^{3D}(S,\beta)$ for $\beta = 1$. This measure still fulfils the desirable properties established in Theorem 4.4.

For lower values of β, the measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ gives higher impact to points which are close to the shape centroid; for higher values of β the measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ gives higher impact to points which are away from the centroid. Trivially, when S is a perfect cube, $\mathcal{C}_{\infty}^{3D}(S,\beta) = 1$ for any given value of β.

5.3.1 Convergence of measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$

This section takes a closer look at the behaviour of measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ when it approaches its limit cases. These limit cases are $\beta \to 0$ and $\beta \to \infty$. From equation (5.12) it is easy to see that:

$$\lim_{\beta\to 0} \mathcal{C}_{\infty}^{3D}(S,\beta) = 1 \tag{5.13}$$

However the other limit case is not as straightforward. The following theorem shows that for $\beta \to \infty$, the measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ also has a consistent behaviour.

Lemma 5.5. *Given a shape S different from a cube:*

$$\lim_{\beta\to\infty} \mathcal{C}_{\infty}^{3D}(S,\beta) = 0 \tag{5.14}$$

Proof. Let S be a given shape different from a cube, with unit volume. It is easy to see that:

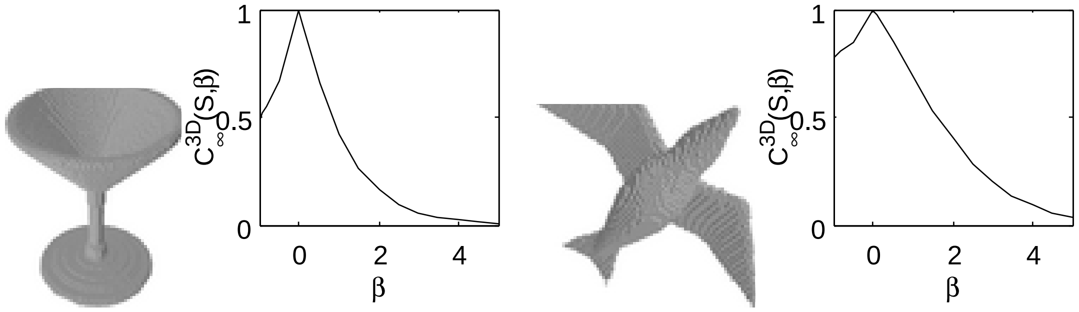
$$\lim_{\beta\to\infty} \frac{3}{2^\beta(\beta+3)} = 0$$

Since the shape S is different from a cube, the inequality (5.7) is strict. And because

$$\iiint\limits_S \max\{|x|,|y|,|z|\}^\beta dxdydz > \frac{3}{2^\beta(\beta+3)}$$

then the quantity $\mathcal{C}_\infty^{3D}(S,\beta) \to 0$ as $\beta \to \infty$. □

Figure 5.1 shows two shapes and the corresponding graphs which display all the computed cubeness measures $\mathcal{C}_\infty^{3D}(S,\beta)$ as a function of β, as β varies through $(-1,0)\cup(0,5)$. These graphs show how $\mathcal{C}_\infty^{3D}(S,\beta)$ approaches zero as $\beta \to \infty$, as it has been mentioned before. This is in accordance with equation (5.14).



(a)

(b)

Figure 5.1: Two shapes and their corresponding measured $\mathcal{C}_\infty^{3D}(S,\beta)$ as β varies through $(-1,0)\cup(0,5)$.

5.3.2 Analysis of $\mathcal{C}_\infty^{3D}(S,\beta)$ graphs

As it is shown in Figure 5.1, for a given object S the measure $\mathcal{C}_\infty^{3D}(S,\beta)$ can be seen as a function of β, as shown in the graph to the right of each one of the objects in Figure 5.1. It is interesting to analyse the graph of these functions. For simplicity, we denote these as $\mathcal{G}_S(\beta)$, so for a fixed shape S:

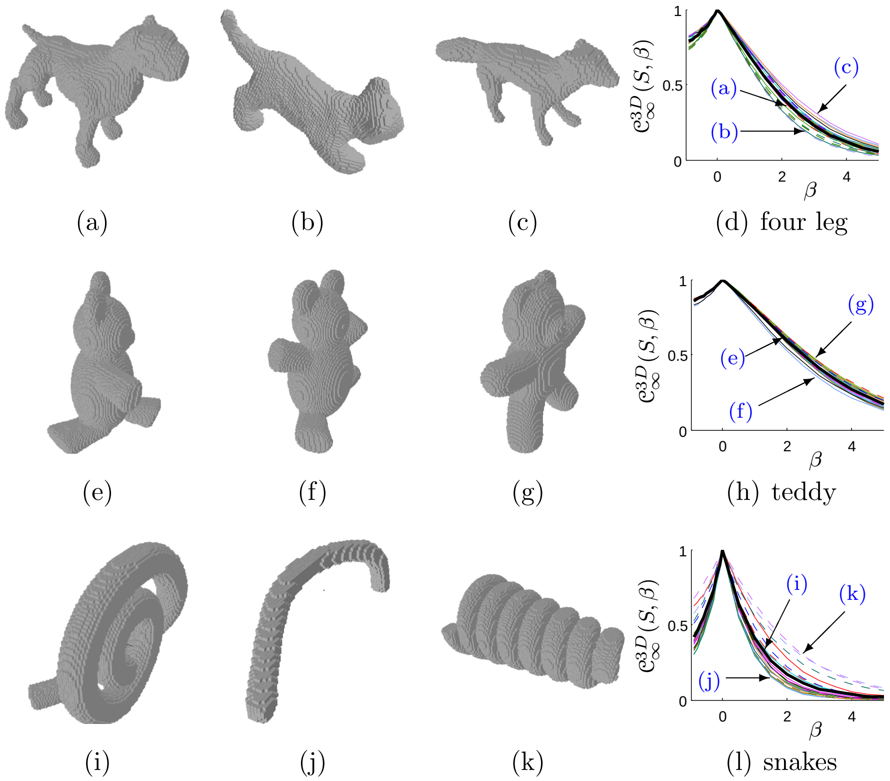
$$\mathcal{G}_S(\beta) = \mathcal{C}_\infty^{3D}(S,\beta)$$

Because measure $\mathcal{C}_\infty^{3D}(S,\beta)$ gives similar scores to objects which are similar in shape (e.g. objects of the same class), it is reasonable to assume that the graphs

$\mathcal{G}_S(\beta)$ for two objects of the same class will be very similar. Under this assumption a graph which is representative of a class can be computed as:

$$\overline{\mathcal{G}}_C(\beta) = \frac{\sum_{n=1}^{N} \mathcal{G}_{S_n}(\beta)}{N} \tag{5.15}$$

where S_n are the N elements of a given class C. We call this the *mean-graph* of the objects in class C (this is not to be confused with the mean graphs from graph theory (Vaidya and Bijukumar, 2010)). Figure 5.2 illustrates how the mean-graph of a class follows the trend of the $\mathcal{G}_S(\beta)$ graphs for objects in that class.



(a) (b) (c) (d) four leg

(e) (f) (g) (h) teddy

(i) (j) (k) (l) snakes

Figure 5.2: Objects from 3 different classes from the McGill database. The right column shows the $\mathcal{G}_S(\beta)$ graph for all objects of the class shown and the mean-graph for that class.

Figure 5.2 uses objects from 3 classes from the McGill database: (a) - (d) belong to class *four leg* (31 shapes); (e) - (h) belong to class *teddy* (20 shapes); (i) - (l) belong to class *snake* (25 shapes). The graphs to the right of each row ((d), (h) and (l)) show the mean-graph of their class (bold line) and the $\mathcal{G}_S(\beta)$ of each object in that class.

The images to the left of each graph show the most representative object for each class ((a), (e) and (i)) as well as the top extreme ((b), (f) and (j)) and bottom extreme ((c), (g) and (k)) objects in each class. The most representative object of each class is the object whose $\mathcal{G}_S(\beta)$ is closest to the mean graph $\overline{\mathcal{G}}_C(\beta)$ for that class; likewise, the extreme top/bottom object of each class is the object whose $\mathcal{G}_S(\beta)$ is furthest above/below the mean graph $\overline{\mathcal{G}}_C(\beta)$ for that class.

It can be seen from (l) that the $\mathcal{G}_S(\beta)$ graphs of the objects from class *snakes* deviate more from their $\overline{\mathcal{G}}_C(\beta)$ than the objects from the other two classes. This suggests that there is a higher intra-class variability, i.e. objects in the same class are more distinct from each other. This is confirmed by looking at objects (j) and (k), which correspond to the extreme top and bottom $\mathcal{G}_S(\beta)$ graphs in (l): the snake in (j) is almost completely stretched, forming an arch, while the snake in (k) is coiled in a very compact shape.

The area between the $\mathcal{G}_S(\beta)$ graphs of the extreme top and extreme bottom objects provides a numerical indication of the intra-class variability. For example, the area between the $\mathcal{G}_S(\beta)$ graphs of (b) and (c) is 0.865, while the area between between the $\mathcal{G}_S(\beta)$ graphs of (f) and (g) is 0.545. As mentioned before, this gives an indication that class *four leg* has a higher intra-class variability compared to that of class *teddy*.

The area between the $\mathcal{G}_S(\beta)$ graphs of (j) and (k) is 1.834, showing that class *snake* has a much larger intra-class variability than the other two classes. Such a large variation leads to question if objects in this class should be grouped together, or whether it would be more meaningful to group them in two separate classes. If the $\overline{\mathcal{G}}_C(\beta)$ graph is used as a division boundary, two resulting sub-classes would be formed: $snake_A$ consisting of 10 objects with their $\mathcal{G}_S(\beta)$ graphs between the $\mathcal{G}_S(\beta)$ graph of (j) and the $\overline{\mathcal{G}}_C(\beta)$ graph, covering an area of area of 0.890; and $snake_B$, consisting of 15 objects with their $\mathcal{G}_S(\beta)$ graphs between the $\overline{\mathcal{G}}_C(\beta)$ graph and the $\mathcal{G}_S(\beta)$ graph of (k), covering an area of 0.9436.

These $\overline{\mathcal{G}}_C(\beta)$ graphs can be used as a tool for selecting the value of β which gives a better class separation. Additionally, the graphs themselves can also be used as features for shape classification applications. Section 5.5.4 gives some examples of how $\overline{\mathcal{G}}_C(\beta)$ graphs can be useful for classification applications.

5.4 Cubeness with different axis weighting

The modified measure introduced in section 5.3 takes into account the distance from each point to the centroid of the shape. Even though this may have some applications, the modified measure is still very similar in its essence to the original measure. In this section an additional modification to measure $\mathcal{C}_{\infty}^{3D}(S)$ is introduced. This measure includes two additional parameters, γ and δ, which modify the behaviour of the measure: these parameters cause the new measure to function as a similarity measure not with a cube, but with a cuboid $\mathcal{W}(\gamma, \delta)$ whose edges are in proportion $1 : \gamma : \delta$. For simplicity, these cuboids are assumed to have unit volume. Such cuboid can be given in the form:

$$\mathcal{W}(\gamma, \delta) = \left\{ (x, y, z) | \max\{|x|, \gamma \cdot |y|, \delta \cdot |z| \leq \frac{1}{\gamma \cdot \delta} \right\} \qquad (5.16)$$

Figure 5.3 illustrates such a cuboid.

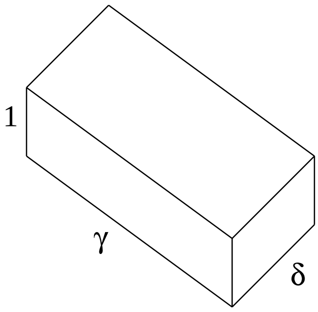


Figure 5.3: Cuboid given by the expression $\mathcal{W}(\gamma, \delta)$. The edges of the cuboid are in proportion $1 : \gamma : \delta$.

The new measure is derived in a similar way to the previous introduced measures, beginning from the fact that expression

$$\iiint\limits_{S} \max\{|x|, \gamma \cdot |y|, \delta \cdot |z|\} dxdydz$$

reaches its maximum value only when the given shape S is a suitably oriented cuboid $\mathcal{W}(\gamma, \delta)$ (not necessarily a cube). Notice that *suitably oriented* refers to the cuboid $\mathcal{W}(\gamma, \delta)$ being aligned with the ordinate axes.

As for the previous measures, first of all the following lemma is given:

Lemma 5.6. *Given a shape S with its centroid located at the origin, let $R(S, \theta, \phi)$ denote the shape S rotated along the x-axis by an angle θ and along the y-axis by* *an angle ϕ. Then:*

$$\frac{\iiint\limits_S \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz}{\mu_{0,0,0}(S)^{4/3}} \geq \frac{3}{8}\cdot(\gamma\cdot\delta)^{1/3} \tag{5.17}$$

$$\frac{\min\limits_{\theta,\phi\in[0,2\pi]}\iiint\limits_{R(S,\theta,\phi)} \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz}{\mu_{0,0,0}(S)^{4/3}} = \frac{3}{8}\cdot(\gamma\cdot\delta)^{1/3} \tag{5.18}$$

$$\Leftrightarrow \quad S \text{ is a cuboid } \mathcal{W}(\gamma,\delta)$$

Proof. Given a shape S with unit volume and centred in the origin and $\mathcal{W}(\gamma,\delta)$ as given by equation (5.16), we derive:

$$\iiint\limits_{S\setminus\mathcal{W}(\gamma,\delta)} \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz \geq \iiint\limits_{\mathcal{W}(\gamma,\delta)\setminus S} \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz$$

and consequently,

$$\iiint\limits_{S} \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz \geq \iiint\limits_{\mathcal{W}(\gamma,\delta)} \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz$$

Notice that both S and $\mathcal{W}(\gamma,\delta)$ have unit volume. Also notice that the faces of the cuboid $\mathcal{W}(\gamma,\delta)$ intersect with the axes at points $(a,0,0)$, $(-a,0,0)$, $(0,a/\gamma,0)$, $(0,-a/\gamma,0)$, $(0,0,a/\delta)$ and $(0,0,-a/\delta)$, where $a=(\gamma\cdot\delta)^{1/3}/2$. Then we have:

$$\begin{aligned}
&\iiint\limits_{\mathcal{W}(\gamma,\delta)} \max\{|x|,\gamma\cdot|y|,\delta\cdot|z|\}dxdydz = \\
&= 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ x,y,z\geq 0}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz \\
&= 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ 0\leq x\leq\gamma\cdot y\leq\delta\cdot z}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz + 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ 0\leq x\leq\delta\cdot z\leq\gamma\cdot y}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz + \\
&\quad 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ 0\leq\gamma\cdot y\leq x\leq\delta\cdot z}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz + 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ 0\leq\gamma\cdot y\leq\delta\cdot z\leq x}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz + \\
&\quad 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ 0\leq\delta\cdot z\leq x\leq\gamma\cdot y}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz + 8\cdot\iiint\limits_{\substack{(x,y,z)\in\mathcal{W}(\gamma,\delta)\\ 0\leq\delta\cdot z\leq\gamma\cdot y\leq x}} \max\{x,\gamma\cdot y,\delta\cdot z\}dxdydz
\end{aligned}$$

$$\begin{aligned}
= \quad & 8 \cdot \int\limits_{z=0}^{a/\delta} \left(\int\limits_{y=0}^{\delta \cdot z/\gamma} \left(\int\limits_{x=0}^{\gamma \cdot y} \delta \cdot z dx \right) dy \right) dz + 8 \cdot \int\limits_{y=0}^{a/\gamma} \left(\int\limits_{z=0}^{\gamma \cdot y/\delta} \left(\int\limits_{x=0}^{\delta \cdot z} \gamma \cdot y dx \right) dz \right) dy + \\
& 8 \cdot \int\limits_{z=0}^{a/\delta} \left(\int\limits_{x=0}^{\delta \cdot z} \left(\int\limits_{y=0}^{x/\gamma} \delta \cdot z dy \right) dx \right) dz + 8 \cdot \int\limits_{x=0}^{a} \left(\int\limits_{z=0}^{x/\delta} \left(\int\limits_{y=0}^{\delta \cdot z/\gamma} x dy \right) dz \right) dx + \\
& 8 \cdot \int\limits_{y=0}^{a/\gamma} \left(\int\limits_{x=0}^{\gamma \cdot y} \left(\int\limits_{z=0}^{x/\delta} \gamma \cdot y dz \right) dx \right) dy + 8 \cdot \int\limits_{x=0}^{a} \left(\int\limits_{y=0}^{x/\gamma} \left(\int\limits_{z=0}^{\gamma \cdot y/\delta} x dz \right) dy \right) dx \\
= \quad & \frac{6}{\gamma \cdot \delta} \cdot a^4
\end{aligned}$$

which proves equation (5.17).

To prove equation (5.18) lets assume θ_0 and ϕ_0 to be the angles such that:

$$\iiint\limits_{R(S,\theta_0,\phi_0)} \max\{|x|, \gamma \cdot |y|, \delta \cdot |z|\} dxdydz = \min_{\theta,\phi \in [0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|, \gamma \cdot |y|, \delta \cdot |z|\} dxdydz \tag{5.19}$$

Obviously the volume of S remains the same regardless of its orientation, therefore $R(S(\theta_0, \phi_0))$ must be equal to $\mathcal{W}(\gamma, \delta)$, which implies that S differs of $\mathcal{W}(\gamma, \delta)$ only in their orientation. □

Based on the above lemma, the following definition is given for the cuboid similarity measure:

Definition 5.3. *Given a shape S centred in the origin and two real positive values γ and δ, the cuboidness measure $\mathcal{C}_\infty^{3D}(S, \gamma, \delta)$ is defined as follows:*

$$\mathcal{C}_\infty^{3D}(S, \gamma, \delta) = \frac{3}{8} \cdot \frac{(\gamma \cdot \delta)^{1/3} \cdot \mu_{0,0,0}(S)^{4/3}}{\min\limits_{\theta,\phi \in [0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} \max\{|x|, \gamma \cdot |y|, \delta \cdot |z|\} dxdydz} \tag{5.20}$$

The measure $\mathcal{C}_\infty^{3D}(S, \gamma, \delta)$ has very similar properties to those established in Theorem 4.4, however it differs slightly reaching its maximum value for a cuboid $\mathcal{W}(\gamma, \delta)$ instead of a cube. The following theorem summarises these desirable properties:

Theorem 5.1. *The measure $\mathcal{C}_\infty^{3D}(S, \gamma, \delta)$ has the following properties:*

(a) $\mathcal{C}_\infty^{3D}(S, \gamma, \delta) \in (0, 1]$, *for any shapes S with non-empty interior;*

(b) $\mathcal{C}_\infty^{3D}(S, \gamma, \delta) = 1 \quad \Leftrightarrow \quad$ *S and $\mathcal{W}(\gamma, \theta)$ are isometric (i.e. they differ only on their orientation);*

(c) $\mathcal{C}_\infty^{3D}(S,\gamma,\delta)$ *is invariant with respect to similarity transformations;*

The proof of Theorem 5.1 is omitted because it is analogous to the proof of Theorem 4.4.

The values of (γ,δ) need to be carefully chosen for measure $\mathcal{C}_\infty^{3D}(S,\gamma,\delta)$. It must be considered that there are different pairs of (γ,δ) which describe equivalent $\mathcal{W}(\gamma,\delta)$ cuboids. Obviously, given two numbers p and q, the cuboid $\mathcal{W}(\gamma=p,\delta=q)$ is equivalent to the cuboid $\mathcal{W}(\gamma=q,\delta=p)$. Furthermore, the length of the sides of a cuboid $\mathcal{W}(\gamma,\delta)$ is in proportion $1:\gamma:\delta$ and the measure $\mathcal{C}_\infty^{3D}(S,\gamma,\delta)$ is scale invariant. From this it can be deduced that a two parameter pairs (γ_1,δ_1) and (γ_2,δ_2) can produce identical behaviours from measure $\mathcal{C}_\infty^{3D}(S,\gamma,\delta)$.

Take for example the cuboids $\mathcal{W}(p,q)$ and $\mathcal{W}(1/p,q/p)$, these two cuboids have equivalent proportions. The sides of cuboids $\mathcal{W}(p,q)$ are in proportion $1:p:q$, while the sides of cuboid $\mathcal{W}(1/p,q/p)$ are in proportion $1/p:1:q/p=(1:p:q)/p$, thus the cuboid $\mathcal{W}(p,q)$ is equivalent to the cuboid $\mathcal{W}(1/p,q/p)$ scaled by a factor of $1/p$. From this and because measure $\mathcal{C}_\infty^{3D}(S,\gamma,\delta)$ is scale invariant, it can be seen that the measures $\mathcal{C}_\infty^{3D}(S,\gamma=p,\delta=q)$ and $\mathcal{C}_\infty^{3D}(S,\gamma=1/p,\delta=q/p)$ will have the same behaviour.

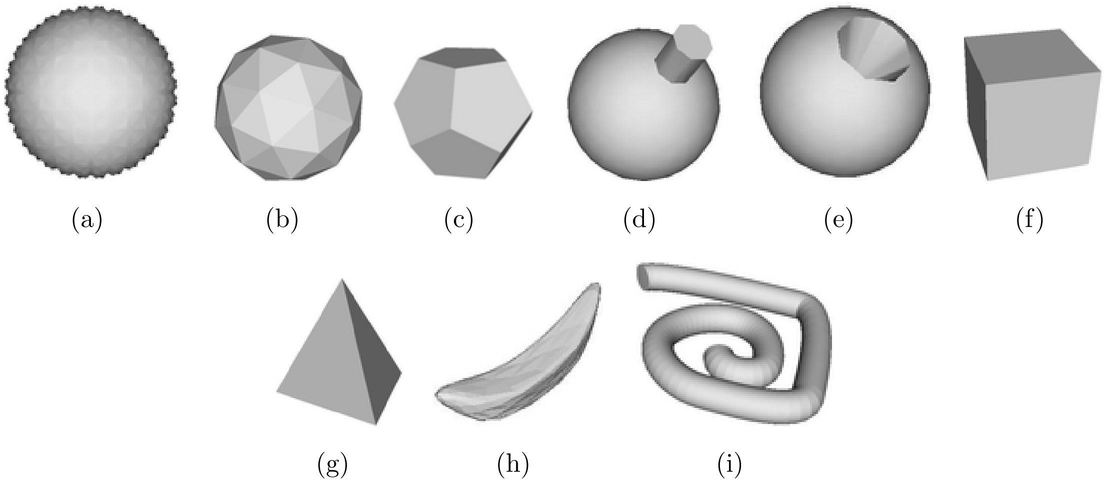
5.5 Experiments

This section features several experiments which demonstrate the behaviour of the measures $\mathcal{C}_2^{3D}(S,\beta)$, $\mathcal{C}_\infty^{3D}(S,\beta)$ and $\mathcal{C}_\infty^{3D}(S,\gamma,\delta)$ introduced in this chapter. These experiments are illustrative of their behaviour in specific scenarios. A comparative study between these measures has not been carried out because it is not possible to establish one measure as being better than the rest in every case; each measure will perform better than the rest under specific circumstances. Instead of a comparative study, the experiments have been designed to show cases in which each measure could be suitably applied.

5.5.1 Experiments for $\mathcal{C}_2^{3D}(S,\beta)$ measure

This example illustrates how the measure $\mathcal{C}_2^{3D}(S,\beta)$ changes its behaviour for different values of β. Figure 5.4 features some 3D objects. Their $\mathcal{C}_2^{3D}(S,\beta)$ values (using different values of β) are summarised in Table 5.1.

Notice how for $\mathcal{C}_2^{3D}(S,\beta=1)$ all objects (a)-(e) give compactness scores which are close to 1. This is to be expected because these objects are roughly spherical, or


(a) (b) (c) (d) (e) (f)

(g) (h) (i)

Figure 5.4: Several 3D objects ordered in descending order of compactness $\mathcal{C}_2^{3D}(S, \beta = 1)$.

it could be said that they are highly similar to a sphere. However, when the value of β is set to $\beta = 10$, the small deviations from the sphere become more important and it is possible to differentiate between them.

	$\beta = -0.9$	$\beta = -0.5$	$\beta = 0.5$	$\beta = 1.0$	$\beta = 2.0$	$\beta = 5.0$	$\beta = 10.0$
(a)	0.9630	0.9962	0.9999	0.9998	0.9995	0.9977	0.9916
(b)	0.9663	0.9968	0.9977	0.9950	0.9885	0.9646	0.9124
(c)	0.9579	0.9933	0.9948	0.9871	0.9642	0.8447	0.5635
(d)	0.9974	0.9968	0.9914	0.9761	0.9182	0.4688	0.0248
(e)	0.9270	0.9787	0.9848	0.9707	0.9433	0.8616	0.7261
(f)	0.9566	0.9825	0.9695	0.9254	0.8057	0.3875	0.0555
(g)	0.9066	0.9438	0.8951	0.7569	0.4612	0.0425	0.0002
(h)	0.8028	0.8292	0.6876	0.4038	0.1014	0.0006	0.0000
(i)	0.5792	0.6312	0.5388	0.2727	0.0629	0.0004	0.0000

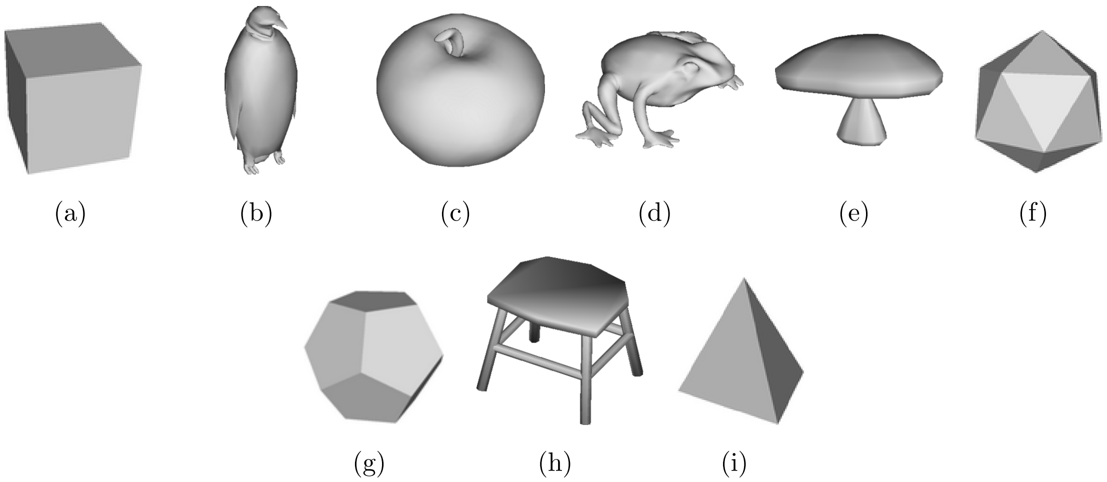
Table 5.1: Measures $\mathcal{C}_2^{3D}(S, \beta)$ for shapes in Figure 5.4 for different values of β.

As it can be seen from Table 5.1, object (d) is more seriously penalised than the others; this is because object (d) is the only object to have a protrusion: as it was mentioned before, measure $\mathcal{C}_2^{3D}(S, \beta)$ gives a higher impact to the points which are away from the centroid, as it is the case of the protrusion on object (d), for higher values of β. The points which are away from the centroid greatly affect the compactness of the object, making it significantly less compact as the values of β are increased.

The values of β used must be carefully selected in order to avoid undesirable results: for $\beta \geq 5$, objects (g), (h) and (i) are impossible to differentiate using $\mathcal{C}^{3D}_{2}(S, \beta)$, even though they are perfectly distinguishable when $\beta = 1$ is used. The value of β to be used is dependent on the particular application in which the measure will be used and on the kind of shapes which it will be dealing with.

5.5.2 Experiments for $\mathcal{C}^{3D}_{\infty}(S, \beta)$ measure

This example illustrates how the behaviour of measure $\mathcal{C}^{3D}_{\infty}(S, \beta)$ changes for different values of β. Figure 5.5 shows a set of 3D objects and their $\mathcal{C}^{3D}_{\infty}(S, \beta)$ measures (for different values of β) are summarised in Table 5.2.


(a) (b) (c) (d) (e) (f)

(g) (h) (i)

Figure 5.5: Several 3D objects ordered in descending order of compactness $\mathcal{C}^{3D}_{\infty}(S, \beta = 1)$.

Notice that the cube in (a) is given a score $\mathcal{C}^{3D}_{\infty}(S, \beta) = 1$ for any value of β. This is in accordance with the definition of $\mathcal{C}^{3D}_{\infty}(S, \beta)$. Also notice that for any other shape in Figure 5.5, the computed values of $\mathcal{C}^{3D}_{\infty}(S, \beta)$ decrease as the values of β increase. For $\beta = 5.0$, objects (b) and (c) still have relatively high $\mathcal{C}^{3D}_{\infty}(S, \beta)$ scores however, higher values of β get lower scores. In fact, for (b) $\mathcal{C}_{\beta=20.0}(S) = 0.065$ and for (c) $\mathcal{C}_{\beta=20.0}(S) = 0.090$. These values are in accordance with Lemma 5.5; given an arbitrarily high value of β, the values of $\mathcal{C}_{\beta}(S)$ will tend to zero.

From the values in Table 5.2 it can be seen that for any shape (except the cube) the values produced by measure $\mathcal{C}^{3D}_{\infty}(S, \beta)$ decrease as the values of β increase.

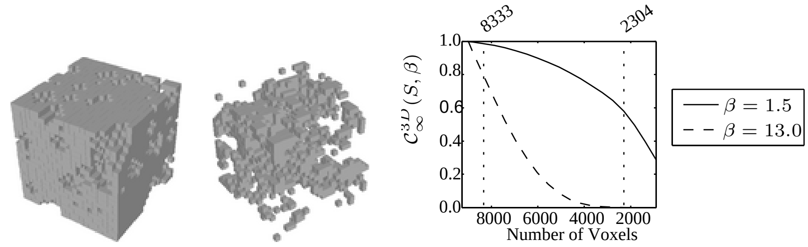
	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1.0$	$\beta = 2.5$	$\beta = 5.0$
(a)	1.000	1.000	1.000	1.000	1.000
(b)	0.999	0.996	0.976	0.882	0.665
(c)	0.997	0.986	0.969	0.899	0.745
(d)	0.985	0.917	0.817	0.481	0.128
(e)	0.981	0.905	0.810	0.550	0.258
(f)	0.998	0.969	0.745	0.434	0.025
(g)	0.998	0.968	0.715	0.381	0.009
(h)	0.931	0.697	0.482	0.153	0.020
(i)	0.979	0.792	0.193	0.016	0.000

Table 5.2: Measures $\mathcal{C}^{3D}_{\infty}(S, \beta)$ for shapes in Figure 5.5 for different values of β.

However, the ordering of the shapes does not necessarily remains the same. For example, using $\beta = 1.0$, (b) has a higher score than (c), but when $\beta = 5.0$ it is (c) that has the higher score. Something similar happens for objects (h) and (i): using $\beta = 1.0$, (h) has a higher score, but when $\beta = 0.1$ is used, (i) has a higher score.

5.5.3 Detecting small defects using $\mathcal{C}^{3D}_{\infty}(S, \beta)$ measure

This experiment shows how measure $\mathcal{C}^{3D}_{\infty}(S, \beta)$ can be used to detect small deviations from a perfect cube. It is required that a suitable value of β is selected, and the value of β depends on the size and number of defects that can be tolerated.



(a) Erosion: 10%

(b) Erosion: 75%

(c)

Figure 5.6: Figures (a) and (b) show cube eroded 10% and 75% respectively. The graph on the right show the rate of decay of measures $\mathcal{C}^{3D}_{\infty}(S, \beta = 1.5)$ and $\mathcal{C}^{3D}_{\infty}(S, \beta = 13.0)$ as more voxels are removed.

For this experiment a voxelised cube was considered, consisting of 21 voxels per side (9261 voxels in total). A process of erosion was applied to the cube, removing 5% of its initial volume on each step of erosion. For each step, measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ was computed and recorded for the eroded cube, using $\beta = 1.5$ and $\beta = 13.0$. Figure 5.6 shows the cube at two stages of the erosion process. A plot of the $\mathcal{C}_{\infty}^{3D}(S,\beta)$ measures is also shown. Table 5.3 shows the computed $\mathcal{C}_{\infty}^{3D}(S,\beta)$ scores.

As it would be expected, any given value of β gives a score $\mathcal{C}_{\infty}^{3D}(S,\beta) = 1$ when no erosion has been applied because the given shape is still a perfect cube. As soon as some level of erosion is applied, the values of $\mathcal{C}_{\infty}^{3D}(S,\beta)$ will decrease but the rate of decay depends on the value of β selected.

No. Voxels	% removed	$\beta = 1.5$	$\beta = 13.0$
9261	0%	1.000	1.000
8797	5%	0.995	0.935
8333	10%	0.986	0.793
7406	20%	0.959	0.521
6477	30%	0.922	0.296
5086	45%	0.845	0.087
3695	60%	0.734	0.012
2304	75%	0.579	0.001

Table 5.3: Measures $\mathcal{C}_{\infty}^{3D}(S,\beta = 1.5)$ and $\mathcal{C}_{\infty}^{3D}(S,\beta = 13.0)$ of a cube being eroded. The scores obtained for these measures when a percentage of the cube has been eroded are shown here.

Using $\beta = 1.5$, measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ is more tolerant to small defects and does not change too quickly. Even with 20% of erosion $\mathcal{C}_{\infty}^{3D}(S,\beta = 1.5)$ is still relatively close to 1, reflecting a behaviour tolerant to small defects on the shape.

Using $\beta = 13.0$ causes measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ to be very sensitive to small deviations from a perfect cube. With only 10% of erosion $\mathcal{C}_{\infty}^{3D}(S,\beta = 13.0) = 0.793$ and it continues to decrease rapidly while more voxels are eroded away from the cube. This behaviour may be preferred if the objective is to detect defects on a cube.

5.5.4 Classification using $\overline{\mathcal{G}}_C(\beta)$ graphs

The experiments in this section illustrate how the mean-graphs introduced in section 5.3.2 can be used for classification purposes. Two different methods were used, each of them illustrated by a simple classification example using a subset of 5 classes from the McGill database: C_1 : *chair* (23 shapes); C_2 : *four leg* (31 shapes); C_3 : *human* (29 shapes); C_4 : *snake* (25 shapes); and C_5 : *teddy* (20 shapes).

The first experiment in this section used the $\overline{\mathcal{G}}_C(\beta)$ to select the value of β which provided better separation between classes and subsequently the measure $\mathcal{C}^{3D}_{\infty}(S,\beta)$ with the selected value of β was used as a feature in a k-nearest neighbours (knn) classifier (with $k = 5$). Values of $k \in [1, 10]$ were tested and $k = 5$ was selected because it produced the best results.

Additionally to the $\mathcal{C}^{3D}_{\infty}(S,\beta)$ measures, the knn classifier used the following 3 moment invariants (Mamistvalov, 1998):

$$\begin{aligned}
J_1(S) &= \mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,2}(S)\\
J_2(S) &= \mu_{2,0,0}(S)\cdot\mu_{0,2,0}(S)\cdot\mu_{0,0,2}(S) - \mu_{1,1,0}(S)^2 - \mu_{1,0,1}(S)^2 - \mu_{0,1,1}(S)^2\\
J_3(S) &= \mu_{2,0,0}(S)\cdot\mu_{0,2,0}(S)\cdot\mu_{0,0,2}(S) + 2\cdot\mu_{1,1,0}(S)\cdot\mu_{1,0,1}(S)\cdot\mu_{0,1,1}(S)\\
&\quad -\mu_{2,0,0}(S)\cdot\mu_{0,1,1}(S)^2 - \mu_{0,2,0}(S)\cdot\mu_{1,0,1}(S)^2 - \mu_{0,0,2}(S)\cdot\mu_{1,1,0}(S)^2
\end{aligned}$$

The 5 classes were divided into a training and test sets, using 85% for training and 15% for test respectively. The knn classifier was trained using the training set and tested over the test set and the classification accuracy was recorded.

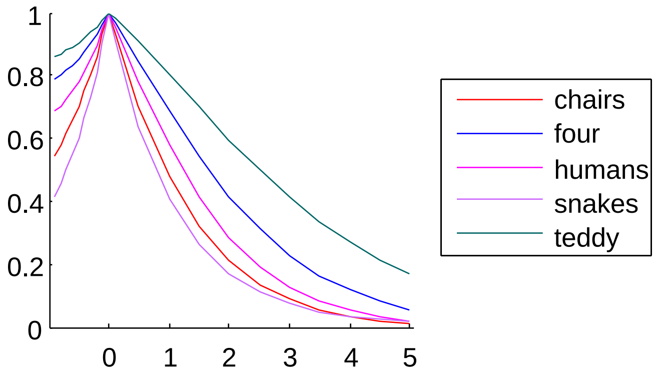


Figure 5.7: Mean-graphs $\overline{\mathcal{G}}_C(\beta)$ for five classes used in this experiments.

Figure 5.7 shows the mean-graphs for the 5 classes used in this experiment. From these graphs, the values of β were selected so that the distance between these graphs was maximised, i.e. a β_0 was selected such that the distance between $\overline{\mathcal{G}}_{C_1}(\beta_0)$, $\overline{\mathcal{G}}_{C_2}(\beta_0)$, $\overline{\mathcal{G}}_{C_3}(\beta_0)$, $\overline{\mathcal{G}}_{C_4}(\beta_0)$ and $\overline{\mathcal{G}}_{C_5}(\beta_0)$ was maximised. The selected values [1] of β were: $\beta = -0.9$, $\beta = -0.8$, $\beta = -0.6$ and $\beta = 1.5$. Table 5.4 shows the classification accuracy rates obtained by using the knn classifier with different number of

[1] The values of β were selected manually, but this process could be automated

descriptors used as input. As it can be seen from the table, using moment invariants only, yields a classification accuracy of 66.67% but this classification accuracy was improved to 80.95% when four $\mathcal{C}_{\infty}^{3D}(S,\beta)$ measures were used in conjunction with the moment invariants.

Descriptors	Accuracy
$J_1(S)$, $J_2(S)$, $J_3(S)$	66.67%
$J_1(S)$, $J_2(S)$, $J_3(S)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=1.5)$	71.42%
$J_1(S)$, $J_2(S)$, $J_3(S)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=-0.9)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=1.5)$	76.19%
$J_1(S)$, $J_2(S)$, $J_3(S)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=-0.9)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=-0.8)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=-0.6)$, $\mathcal{C}_{\infty}^{3D}(S,\beta=1.5)$	80.95%

Table 5.4: Classification accuracy using measure $\mathcal{C}_{\infty}^{3D}(S,\beta)$ with values of β selected from the mean-graphs in Figure 5.7.

The mean-graphs themselves can be used as features for classification purposes. As it was shown in Figure 5.2 the graphs of $\overline{\mathcal{G}}_C(\beta)$ are taken as the $\mathcal{G}_S(\beta)$ graph which is most representative of a certain class. The graph $\mathcal{G}_S(\beta)$ for objects belonging to a given class C would not differ very much from the $\overline{\mathcal{G}}_C(\beta)$ graph of this class.

For the purpose of this experiment, the difference between a graph $\mathcal{G}_S(\beta)$ of a given object and a graph $\overline{\mathcal{G}}_C(\beta)$ of a given class is measured as the area between these graphs. This is illustrated in Figure 5.8. The area between two graphs which are similar will be smaller than the area between two graphs which greatly differ.

Notice that in accordance with equation (5.13) the $\mathcal{G}_S(\beta)$ graphs will tend to 1 as $\beta \rightarrow 0$. This means that in the proximity of $\beta = 0$, the $\mathcal{G}_S(\beta)$ graphs will be very similar for any given shape S. Similarly happens for $\beta \rightarrow \infty$, where in accordance with equation (5.14) the $\mathcal{G}_S(\beta)$ graphs tend 0. Therefore these values have been ignored and the graphs $\mathcal{G}_S(\beta)$ and $\overline{\mathcal{G}}_C(\beta)$ are only considered in the interval $\beta \in [-0.9,-0.1] \cup [2.5,5.0]$. These intervals were selected by preliminary experimentation, having tried a number of different intervals. However a more exhaustive search for optimum intervals might yield even better results.

Following these considerations, the difference used between the graph $\mathcal{G}_S(\beta)$ of a given object and the graph $\overline{\mathcal{G}}_C(\beta)$ of a given class is computed as follows:

$$d_{area} = \sqrt{(d_{[-0.9,-01]})^2 + (d_{[2.5,5.0]})^2}$$

where $d_{[a,b]}$ is the area between $\mathcal{G}_S(\beta)$ and $\overline{\mathcal{G}}_C(\beta)$ in the interval $[a,b]$ normalised by dividing by the length of the interval.

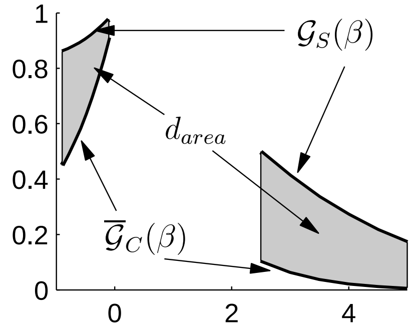


Figure 5.8: The difference (d_{area}) between the graph $\mathcal{G}_S(\beta)$ of a given object and the graph $\overline{\mathcal{G}}_C(\beta)$ of a given class is considered to be the area between these two graphs.

The same 5 classes as in the previous experiment (C_1 : *chair*; C_2 : *four leg*; C_3 : *human*; C_4 : *snake*; and C_5 : *teddy*) were used in this experiment. The mean-graphs for each class were generated from the training set, and then the shapes on the test set were classified using the mean-graphs: an object S was assigned to the class C which yielded the smallest d_{area} value. This experiment achieved **88.28%** classification accuracy. This is a remarkable result, especially considering that no additional descriptors were used for the classification process.

5.5.5 Experiments for $\mathcal{C}_\infty^{3D}(S, \gamma, \delta)$ measure

In this experiment the measure $\mathcal{C}_\infty^{3D}(S, \gamma, \delta)$ is applied to some geometrical and real world shapes, using different combinations of γ and β. Figure 5.9 shows the 3D shapes used in this experiment and their $\mathcal{C}_\infty^{3D}(S, \gamma, \delta)$ scores are summarised in Table 5.5.

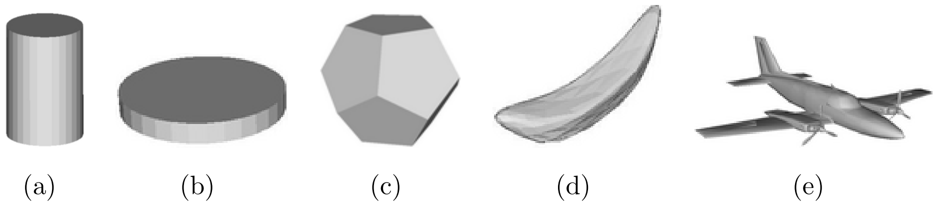


Figure 5.9: Several 3D geometric objects and 3D models of real world objects.

From equation (5.16) it can be seen that γ and δ influence the shape of the cuboid along the y and z axes respectively. A low value of γ or δ would stretch the cuboid $\mathcal{W}(\gamma, \delta)$ along the corresponding axis, and high values of γ or δ would compress it in the corresponding direction. Take for example the cuboid $\mathcal{W}(\gamma = 0.1, \delta = 0.1)$:

(γ, δ)	$(0.1, 0.1)$	$(0.1, 0.5)$	$(0.1, 1.5)$	$(0.5, 1.5)$	$(1.5, 2.0)$	$(2.0, 2.0)$
(a)	0.394	0.608	0.545	0.900	0.969	0.974
(b)	0.961	0.584	0.453	0.700	0.701	0.690
(c)	0.345	0.531	0.476	0.800	0.891	0.877
(d)	0.579	0.482	0.373	0.630	0.735	0.765
(e)	0.636	0.435	0.340	0.530	0.607	0.620

Table 5.5: Measure $\mathcal{C}_{\infty}^{3D}(S, \gamma, \delta)$ for shapes in Figure 5.9.

this cuboid would be equally stretched out along both axes resembling a tile or a pizza box. From this it seems reasonable that the object (b) in Figure 5.9 has the highest score for $\mathcal{C}_{\infty}^{3D}(S, \gamma = 0.1, \delta = 0.1)$ measure.

5.5.6 Shape retrieval using $\mathcal{C}_{\infty}^{3D}(S, \gamma, \delta)$ measure

The previous section illustrates how the measure $\mathcal{C}_{\infty}^{3D}(S, \gamma, \delta)$ changes its behaviour, producing different scores for different (γ, δ) pairs. These different scores provide additional information which can be used for particular applications such as image retrieval from a database.

The following example illustrates how measure $\mathcal{C}_{\infty}^{3D}(S, \gamma, \delta)$ can be used as an additional descriptor for an image retrieval application. Normally, having more descriptors can improve the performance of a retrieval application since the descriptor being added contributes more information to the system.

For this example, a 3D shape retrieval application was built. The retrieval system takes a query 3D shape and computes a feature vector using 3 shape descriptors described in Chapter 4: $\mathcal{C}_{\infty}^{3D}(S)$, $\mathcal{C}_{fit-\infty}^{3D}(S)$ and $\mathcal{C}_d(S)$. Using this feature vector, the shape is represented in a feature space. All shapes in the database are also represented in this feature space. The shapes which are closest to the query shape in the feature space are assumed to be similar in shape. The 5 closest shapes are retrieved and presented as the top 5 matching shapes. For an illustration of the structure of the retrieval system see Figure 1.2.

In order to demonstrate that the addition of measure $\mathcal{C}_{\infty}^{3D}(S, \gamma, \delta)$ actually improves the retrieval performance, the query is performed twice using the same query shape: the first time using only the 3 descriptors from Chapter 4; and the second time with an additional $\mathcal{C}_{\infty}^{3D}(S, \gamma, \delta)$ measure.

Figure 5.10 shows an example of the improved retrieval performance. The airplane in the box to the left is used as query shape. The shapes on the right are the

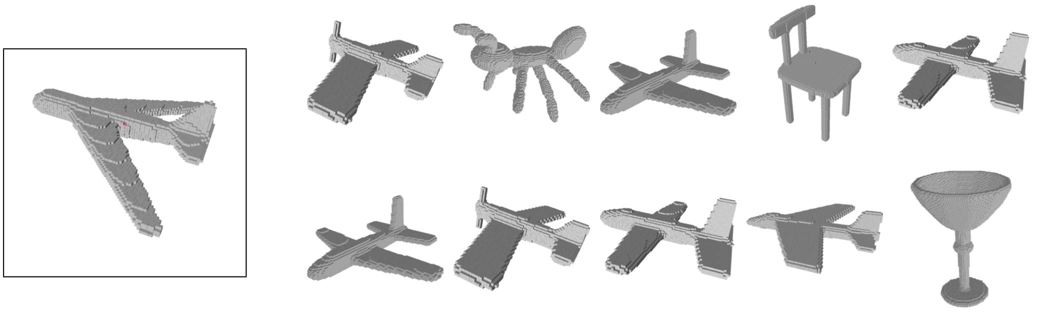


Figure 5.10: Retrieval example using airplane shape as query image. The first row uses descriptors: $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$. The second row uses descriptors: $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$, $\mathcal{C}_d(S)$ and $\mathcal{C}^{3D}_{\infty}(S, \gamma = 1.5, \delta = 1.5)$.

top 5 matches given by the retrieval system: the first row gives the results obtained using $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$. The second row gives the results obtained using $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$, $\mathcal{C}_d(S)$ and $\mathcal{C}^{3D}_{\infty}(S, \gamma = 1.5, \delta = 1.5)$. The selected values of $(\gamma = 1.5, \delta = 1.5)$ create a cuboid $\mathcal{W}(\gamma = 1.5, \delta = 1.5)$ which would be a square based prism. The measure $\mathcal{C}^{3D}_{\infty}(S, \gamma = 1.5, \delta = 1.5)$ seems to be taking into account the elongated shape of the airplane body (probably penalising the presence of the wings). The second query eliminates the ant and chair in the first query, but brings a cup in its place which is not necessarily desirable.

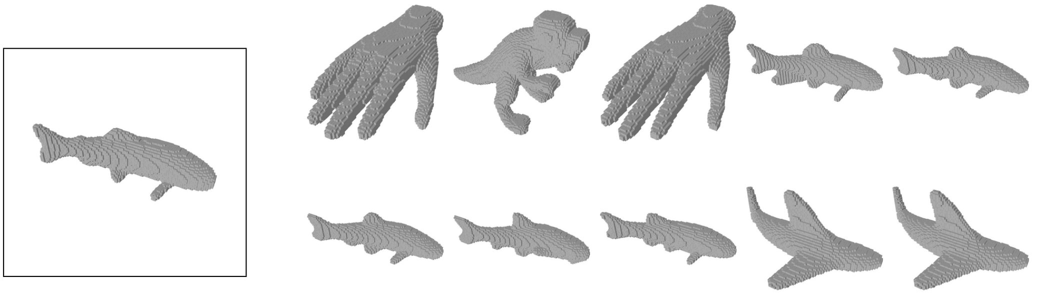


Figure 5.11: Retrieval example using fish shape as query image. The first row uses descriptors: $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$ and $\mathcal{C}_d(S)$ The second row uses descriptors: $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{fit-\infty}(S)$, $\mathcal{C}_d(S)$ and $\mathcal{C}^{3D}_{\infty}(S, \gamma = 0.1, \delta = 0.5)$.

Figure 5.11 shows another example of the retrieval system performance. For this experiment a fish is used as the query shape. In this case the selected values of $(\gamma = 0.1, \delta = 0.5)$ create a cuboid $\mathcal{W}(\gamma = 0.1, \delta = 0.5)$ which is a cuboid elongated along both y-axis and z-axis, but the elongation along the y-axis is greater. This matches with the flat and elongated nature of the fish, producing a better result when the $\mathcal{C}_{\infty}^{3D}(S, \gamma = 0.1, \delta = 0.5)$ measure is included.

5.6 Conclusions

In this chapter, the descriptors introduced in Chapter 4 were modified to incorporate tuning parameters. Position-dependant compactness and cubeness were developed in this way. Position-dependant cubeness was taken a step further by developing the idea of $\mathcal{C}_{\infty}^{3D}(S, \beta)$ graphs for each object and mean-graphs for each class. Although this concept was only explored for cubeness, it could easily be applied to compactness as well. Furthermore, different axis weighting was applied to cubeness as well, extending the concept from cube similarity to cuboid similarity.

Experiments in section 5.5 illustrate the behaviour of all the new descriptors. Different possible applications are analysed: defect detection, classification and retrieval.

Two different approaches are used for shape classification: for the first approach, several cubeness measures are combined (along with moment invariants) to create a feature vector (as described in Figure 1.2); a knn classifier uses this feature vector to classify the object. The second approach uses $\mathcal{G}_S(\beta)$ graphs to classify the objects; the distance between two objects is given by the area between the $\mathcal{G}_S(\beta)$ graphs of the objects.

The cuboid similarity measure is used in conjunction with other descriptors in a small shape retrieval application. Again, the descriptors are used to create a feature vector. The distance between feature vectors is calculated and the 5 closest vectors define the top 5 images retrieved.

As mentioned in Chapter 4, new shape descriptors can be incorporated to the descriptor set used by shape retrieval applications. The measures presented in this chapter could also be incorporated to such applications. Furthermore, the classification using $\mathcal{G}_S(\beta)$ graphs presents a new approach for shape classification and shape retrieval.

Chapter 6

Additional 3D Descriptor

6.1 Introduction

Chapter 4 and Chapter 5 introduced a range of different 3D shape descriptors. These chapters give some examples of specific scenarios where each of these descriptors could be applied. As was mentioned before, an increase in the number of descriptors used for any given application usually yields an improvement in performance. Section 5.5.6 shows an example where the descriptors from Chapter 4 and Chapter 5 are combined to achieve an improved retrieval performance.

Generally speaking, having a large range of shape descriptors is preferable in most situations. As is mentioned by Vranić and Saupe (2001), there is a trade-off between the computational cost of using additional descriptors and the resulting performance: using more descriptors requires more resources (e.g. processing time and storage) and normally gives better results.

This chapter introduces another 3D shape descriptor, similar to the ones introduced in Chapter 4. As before, the new descriptor is expected to outperform the already known descriptors in certain scenarios and may be outperformed in other scenarios. This descriptor could then be combined with other descriptors in an image processing application and achieve an overall increase in performance.

6.2 Basic concepts

Chapter 4 introduced two 3D shape descriptors, $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$. These two descriptors are constructed in a very similar way but using different distance metrics. As was mentioned in Chapter 4, a wide variety of distance metrics is available in the literature. Section 4.2.3 shows how different unit disks can be constructed when different metrics are used to measure the distance of points to the origin. Different distance metrics will create variously shaped unit disks and potentially different 2D and 3D shape descriptors could be derived from these metrics. This section looks at another distance metric and the unit disk constructed by this metric, leading to the definition of a new 3D shape descriptor.

6.2.1 City block distance

This popular distance metric is also known as l_1-distance, taxi cab distance or Manhattan distance. Using this metric, the distance between two points it given by the sum of the absolute differences of their coordinates. It is known as taxi cab distance because to get from point P to point Q the shortest path would be following straight horizontal and vertical lines, as a taxi would in the grid of a big city like New York. More formally, the distance between two n-dimensional points $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$ is given by:

$$l_1(P, Q) = |p_1 - q_1| + |p_2 - q_2| + ... + |p_n - q_n| \tag{6.1}$$

6.2.2 Unit disk under l_1-distance metric

Section 4.2.3 shows how unit disks are constructed using Euclidean (l_2) and chessboard (l_∞) distances. Equations (4.3) and (4.4) show how these unit disks can be expressed. Similarly, using l_1-distance metric, a unit disk situated in the origin O can be constructed as:

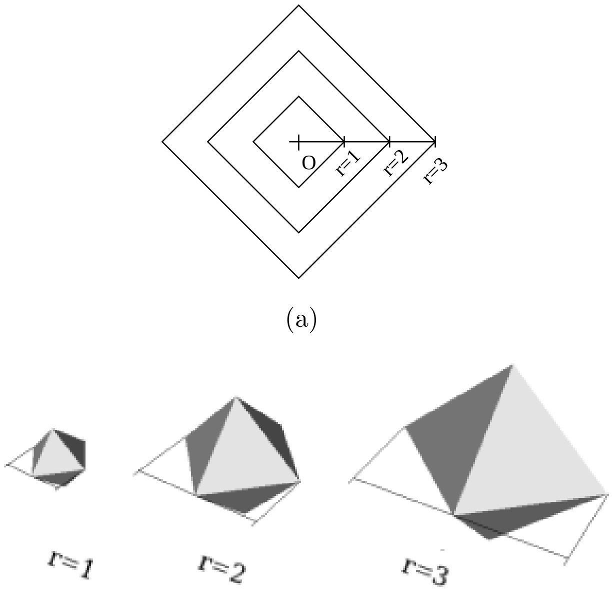
$$c_1(r) = \{P = (x, y, z) : l_1(P, O) \leq r\} \tag{6.2}$$

In 2D space, the disk $c_2(r)$ (using l_2-distance) is a circle of radius r while the disk $c_\infty(r)$ (using l_∞-distance) is a square of side $2r$, aligned with the coordinate axes. Similarly, the disk $c_1(r)$ would be a square, with its vertices intersecting the coordinate axes at the points $(r, 0)$, $(0, r)$, $(-r, 0)$ and $(0, -r)$ respectively (see Figure 6.1(a)). Figure 6.1 show $c_1(r)$ with $r = 1$, $r = 2$ and $r = 3$ in 2D and 3D space.

Assuming a scale invariant 2D shape descriptor was derived using l_1-distance, it would be equivalent to a 2D shape descriptor derived using l_∞-distance, since both distance metrics produce squares as their unit disks (although rotated 45°).

In 3D space, the unit disk $c_2(r)$ is a sphere and the unit disk $c_\infty(r)$ is a cube. However, instead of a cube, $c_1(r)$ is a octahedron with its vertices intersecting the coordinate axes at the points $(r, 0, 0)$, $(0, r, 0)$, $(0, 0, r)$, $(-r, 0, 0)$, $(0, -r, 0)$ and $(0, 0, -r)$ respectively (see Figure 6.1(b)).

A 3D shape descriptor derived using l_1-distance would function as a similarity measure between a given shape and a octahedron, unlike $\mathcal{C}_{\infty}^{3D}(S)$ derived using l_∞-distances which is a similarity measure between a given shape and a cube.



(a)

(b)

Figure 6.1: Most compact shapes using city block distance. (a) show the most compact shapes in 2D (a square) while (b) show the most compact shape in 3D (a octahedron). See also Figure 4.1.

6.3 3D octahedroness measure

In a similar way to how measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_{\infty}^{3D}(S)$ are derived in section 4.3 and section 4.4, in this section the octahedron similarity measure is derived. Following a similar methodology as in the previous cases, we begin by showing that the quantity

$$\min_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} (|x| + |y| + |z|)dxdydz$$

reaches its minimum value $\frac{7}{9}$ only when the given shape is an octahedron with its vertices intersecting the coordinate axes. Notice that, once again for simplicity, all shapes presented here are assumed to have unit volume. Just as a reminder, the volume of an octahedron is:

$$V = \frac{\sqrt{2}h^3}{3}$$

where h is the length of the edges of the octahedron. Alternatively, the volume of an octahedron may be calculated as:

$$V = \frac{4a^3}{3}$$

where a is the distance from the vertices to the centre of the octahedron.

Theorem 6.1. *Given a 3D shape S, let $R(S, \theta, \phi)$ denote the rotation of S along the x-axis by an angle θ and along the y-axis by an angle ϕ. And let $\Omega(a)$ denote an octahedron centred in the origin and with its vertices intersecting the coordinate axes at points: $(a, 0, 0)$, $(0, a, 0)$, $(0, 0, a)$, $(-a, 0, 0)$, $(0, -a, 0)$ and $(0, 0, -a)$. Then,*

$$\frac{\iiint\limits_S (|x| + |y| + |z|)dxdydz}{\mu_{0,0,0}(S)^{4/3}} \geq \frac{7}{9} \tag{6.3}$$

$$\frac{\iiint\limits_S (|x| + |y| + |z|)dxdydz}{\mu_{0,0,0}(S)^{4/3}} = \frac{7}{9} \iff S = \Omega\left(\left(\frac{3 \cdot \mu_{0,0,0}(S)}{4}\right)^{1/3}\right) \tag{6.4}$$

$$\frac{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} (|x| + |y| + |z|)dxdydz}{\mu_{0,0,0}(S)^{4/3}} = \frac{7}{9} \iff S \text{ is an octahedron.} \tag{6.5}$$

Proof. Let S be a given 3D shape centred in the origin and let Ω_{fit} denote an octahedron as stated in the Theorem: $\Omega(a)$, where $a = (3 \cdot \mu_{0,0,0}(S)/4)^{1/3}$. Thus we have:

(a) The set differences $S \setminus \Omega_{fit}$ and $\Omega_{fit} \setminus S$ have equal volume;

(b) The points from $\Omega_{fit} \setminus S$ are closer to the origin (in terms of l_1-distance) than the points from $S \setminus \Omega_{fit}$. More formally:

$(u, v, w) \in (S \setminus \Omega_{fit})$ and $(x, y, z) \in (\Omega_{fit} \setminus S) \Rightarrow (|u|+|v|+|w|) > (|x|+|y|+|z|)$

(see. Figure 6.2).

From items (a) and (b) it is easily deduced that:

$$\iiint\limits_{S\setminus\Omega_{fit}} (|x| + |y| + |z|)dxdydz \geq \iiint\limits_{\Omega_{fit}\setminus S} (|x| + |y| + |z|)dxdydz. \tag{6.6}$$

To prove equation (6.3), we derive:

$$\iiint\limits_{S} (|x|+|y|+|z|)\ dxdydz = 8 \cdot \iiint\limits_{\substack{(x,y,z)\in S\\ x,y,z\geq 0}} (x+y+z)\ dxdydz$$

$$= \ 8 \cdot \int\limits_0^a \int\limits_0^{a-x} \int\limits_0^{a-x-y} (x+y+z)\ dxdydz = \frac{7}{9}\cdot a^4,$$

which proves equation (6.3) since $a = (3\mu_{0,0,0}(S)/4)^{1/3}$.

The proof of equation (6.4) comes from the fact that inequality (6.6) is strict except when S and Ω_{fit} are identical.

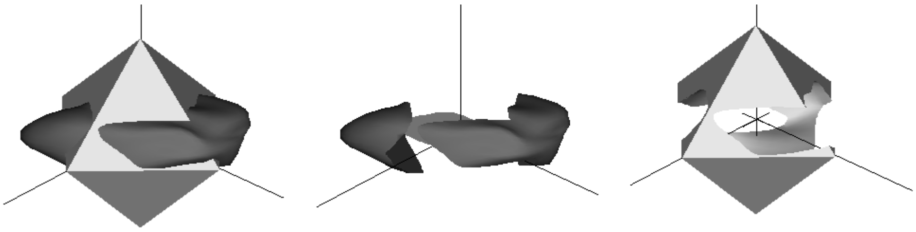
To prove equation (6.5) assume θ_0 and ϕ_0 to be the angles which minimise the quantity $\iiint\limits_{R(S,\theta,\phi)} (|x|+|y|+|z|)dxdydz$ such that:

$$\iiint\limits_{R(S,\theta_0,\phi_0)} (|x|+|y|+|z|)dxdydz = \min_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} (|x|+|y|+|z|)dxdydz$$

It is only possible for the quantity

$$\frac{\iiint\limits_{R(S,\theta_0,\phi_0)} (|x|+|y|+|z|)dxdydz}{\mu_{0,0,0}(S)^{4/3}}$$

to reach its minimum value of 7/9 if $R(S,\theta_0,\phi_0) = \Omega(a)$, which implies that S and $\Omega(a)$ may only differ in their orientation and therefore S is an octahedron. □



(a) Ω_{fit} and S (b) $S \setminus \Omega_{fit}$ (c) $\Omega_{fit} \setminus S$

Figure 6.2: (a) shows a given shape S and the octahedron Ω_{fit} which has the same volume as S and both shapes are centred at the origin. (b) and (c) show $S \setminus \Omega_{fit}$ and $\Omega_{fit} \setminus S$ respectively. Any point from $\Omega_{fit} \setminus S$ is closer to the origin than any point from $S \setminus \Omega_{fit}$.

The following definition is given for the octahedron similarity measure (octahedroness):

Definition 6.1. *Given a shape S, the octahedroness measure $\mathcal{C}_1^{3D}(S)$ is defined as:*

$$\mathcal{C}_1^{3D}(S) = \frac{7}{9}\left(\frac{3}{4}\right)^{4/3} \cdot \frac{\mu_{0,0,0}(S)^{4/3}}{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} (|x|+|y|+|z|)dxdydz} \tag{6.7}$$

The measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$ have a series of desirable properties. Likewise, measure $\mathcal{C}_1^{3D}(S)$ has very similar properties, summarised in the following Theorem:

Theorem 6.2. *The octahedroness measure $\mathcal{C}_1^{3D}(S)$ has the following properties:*

(a) $\mathcal{C}_1^{3D}(S) \in (0,1]$ *for any shape S;*

(b) $\mathcal{C}_1^{3D}(S) = 1 \quad \Leftrightarrow \quad$ *S is a octahedron;*

(c) $\mathcal{C}_\infty^{3D}(S)$ *is invariant with respect to the similarity transformations.*

Proof. The proof of items (a) and (b) follows directly from Theorem 6.1.

Item (c) is also easily verified: rotation invariance comes from the fact that both quantities $\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} (|x|+|y|+|z|)dxdydz$ and $\mu_{0,0,0}(S)$ are such invariants; translation invariant comes straight from the definition, since all shapes are assumed to be centred at the origin; scale invariance can also be easily verified by applying a scaling factor r to a given shape S:

$$\begin{aligned}
\mathcal{C}_1^{3D}(\mathbf{r}\cdot S) &= \frac{7}{9}\left(\frac{3}{4}\right)^{4/3} \cdot \frac{\mu_{0,0,0}(\mathbf{r}\cdot S)^{4/3}}{\min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{\mathbf{r}\cdot R(S,\theta,\phi)} (|x|+|y|+|z|)dxdydz} \\
&= \frac{7}{9}\left(\frac{3}{4}\right)^{4/3} \cdot \frac{(\mathbf{r}^3\cdot\mu_{0,0,0}(S))^{4/3}}{\mathbf{r}^4\cdot \min\limits_{\theta,\phi\in[0,2\pi]} \iiint\limits_{R(S,\theta,\phi)} (|x|+|y|+|z|)dxdydz} = \mathcal{C}_1^{3D}(S)
\end{aligned}$$

which proves that $\mathcal{C}_1^{3D}(S)$ is scale invariant. □

6.4 Comparable 3D descriptors

Spheres and cubes are very *natural* shapes as they are commonly found in art, architecture and nature. Octahedra on the other hand are not as commonly encountered, despite it being a well known *platonic solid*. This is probably the reason why, to the best of the author's knowledge, there are no existing octahedron similarity measures in the existing literature.

In order to have a reference measure to compare $\mathcal{C}_1^{3D}(S)$ with, the following octahedron fitting method is defined, in a similar way to the cube and sphere fitting methods previously defined in section 4.5.2:

Definition 6.2. *Let S be a given 3D shape and Ω_{fit} be a octahedron as in the statement of Theorem 6.1. The measure $\mathcal{C}_{fit-1}^{3D}(S)$ is defined as:*

$$\mathcal{C}_{fit-1}^{3D}(S) = \max_{\theta,\phi\in[0,2\pi]} \frac{Volume(S \cap R(\Omega_{fit},\theta,\phi))}{Volume(S \cup R(\Omega_{fit},\theta,\phi))} \tag{6.8}$$

6.5 Experiments

In this section some simple experiments are presented to illustrate the behaviour of measure $\mathcal{C}_1^{3D}(S)$. These experiments are similar to the experiments in Chapter 4 and Chapter 5 in that they portrait minimal scenarios to demonstrate the potential use of the presented measures in more complex scenarios.

6.5.1 Basic experiments

The first experiment in this section shows the values produced for a set of geometrical shapes and 3D models of real world objects. Figure 6.3 shows this set of 3D shapes and Table 6.1 shows the produced values for measures $\mathcal{C}_1^{3D}(S)$ and $\mathcal{C}_{fit-1}^{3D}(S)$ for these shapes.

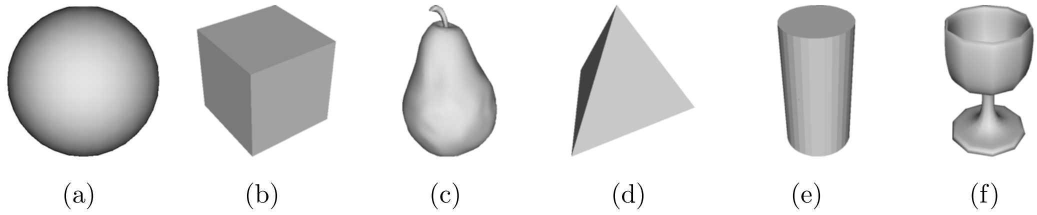

(a) (b) (c) (d) (e) (f)

Figure 6.3: 3D models of geometric and real world objects.

It seems reasonable that shapes (a) and (b) (the sphere and the cube) are given a high $\mathcal{C}_1^{3D}(S)$ score; these two shapes are most compact when l_2- and l_∞-distance metrics are used, so it seems reasonable that they also have a high compactness when l_1-distance metric is used.

	$\mathcal{C}_1^{3D}(S)$	$\mathcal{C}_{fit-1}^{3D}(S)$
(a)	0.978	0.731
(b)	0.966	0.440
(c)	0.959	0.670
(d)	0.883	0.400
(e)	0.865	0.331
(f)	0.200	0.016

Table 6.1: Measures $\mathcal{C}_1^{3D}(S)$ and $\mathcal{C}_{fit-1}^{3D}(S)$ for shapes in Figure 6.3. Objects are in descending order according to measure $\mathcal{C}_1^{3D}(S)$.

Notice also that the shape (f) is given a very low $\mathcal{C}_{fit-1}^{3D}(S)$ score. This indicates that the intersection between shape (f) and its fitting octahedron Ω_{fit} has a relatively low volume. The next example provides a more detailed explanation of a similar situation.

As mentioned before, shape fitting methods like $\mathcal{C}_{fit-1}^{3D}(S)$, and the measures $\mathcal{C}_{fit-2}^{3D}(S)$ and $\mathcal{C}_{fit-\infty}^{3D}(S)$ introduced in Chapter 4, will produce scores of zero for any shape where the shape and its fitting shape (Ω_{fit} in the case of $\mathcal{C}_{fit-1}^{3D}(S)$) do not intersect. When measure $\mathcal{C}_{fit-1}^{3D}(S)$ is applied to a multi-component shape this can easily be the case, thus making measure $\mathcal{C}_{fit-1}^{3D}(S)$ unsuitable for such a class of shapes. The measure $\mathcal{C}_1^{3D}(S)$ does not have this disadvantage and would still produce different values depending on the relative position of the different elements of the multi-component shape.

Figure 6.4 illustrates the behaviour of measure $\mathcal{C}_1^{3D}(S)$ when applied to multi-component shapes. These multi-component shapes are composed of 8 spheres of equal volume with their centroids located on the vertices of a cube[1]. The volume and relative position of these spheres remains constant throughout the experiment, but the distance between the spheres is increased.

Table 6.2 shows the values produced by measures $\mathcal{C}_1^{3D}(S)$ and $\mathcal{C}_{fit-1}^{3D}(S)$. The values produced by measure $\mathcal{C}_{fit-1}^{3D}(S)$ rapidly become zero; even for shape (b) the given score is already very close to zero. Contrary to this, measure $\mathcal{C}_1^{3D}(S)$ produces different scores for every shape in Figure 6.4. As it can be seen in Table 6.2 the scores given by measure $\mathcal{C}_1^{3D}(S)$ become lower as the distance between the spheres increases.

[1] A cube was used because it is the dual polyhedron of the octahedron, so the intersection with Ω_{fit} would disappear quickly.

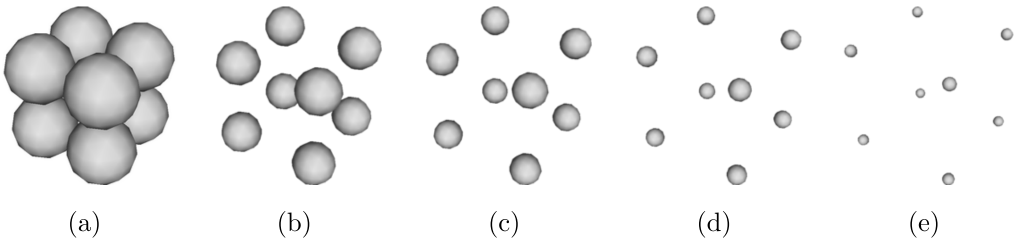


Figure 6.4: Multi-component shapes composed of 8 spheres located on the vertices of an imaginary cube. The distance between the spheres increases in each shape but the volume of the spheres and their relative position remain constant.

	$\mathcal{C}_1^{3D}(S)$	$\mathcal{C}_{fit-1}^{3D}(S)$
(a)	0.881	0.476
(b)	0.489	0.018
(c)	0.332	0.000
(d)	0.202	0.000
(e)	0.113	0.000

Table 6.2: Values produced by measures $\mathcal{C}_1^{3D}(S)$ and $\mathcal{C}_{fit-1}^{3D}(S)$ for multi-component shapes in Figure 6.4.

6.5.2 Classification experiment

The final experiment in this section shows how the measure $\mathcal{C}_1^{3D}(S)$ can be used in conjunction with other shape descriptors for classification tasks. The classification process used in this experiment is very similar to the one previously introduced in section 5.5.4.

The same classes as in the previous classification experiment were used: C_1 : *chair* (23 shapes); C_2 : *four leg* (31 shapes); C_3 : *human* (29 shapes); C_4 : *snake* (25 shapes); and C_5 : *teddy* (20 shapes). Again a knn classifier was used, with the value $k = 5$ as in the previous experiment. The data set was once again divided into training and test sets, using the first to train the knn classifier and the second to evaluate the classification accuracy. The value of k and the split in the data were kept the same in order to be consistent with the previous experiment.

The knn classifier was trained using a feature set produced by combining the two descriptors introduced in this chapter ($\mathcal{C}_1^{3D}(S)$ and $\mathcal{C}_{fit-1}^{3D}(S)$) and other shape descriptors previously introduced. These shape descriptors were the 3 moment invariants $J_1(S)$, $J_2(S)$, $J_3(S)$ and the measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$. Different combinations of these descriptors were used and their classification accuracy was recorded. Table 6.3 shows the classification accuracy obtained by different combinations.

Descriptors	Accuracy
$J_1(S)$, $J_2(S)$, $J_3(S)$	66.67%
$J_1(S)$, $J_2(S)$, $J_3(S)$, $\mathcal{C}^{3D}_{\infty}(S)$	71.43%
$J_1(S)$, $J_2(S)$, $J_3(S)$, $\mathcal{C}^{3D}_{2}(S)$, $\mathcal{C}^{3D}_{\infty}(S)$	71.43%
$\mathcal{C}^{3D}_{2}(S)$, $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{1}(S)$	76.19%
$\mathcal{C}^{3D}_{2}(S)$, $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{1}(S)$, $\mathcal{C}^{3D}_{fit-1}(S)$	80.95%

Table 6.3: Classification accuracy using measure $\mathcal{C}^{3D}_{1}(S)$ in conjunction with other shape descriptors.

As seen before, the combination of moment invariants produces a classification accuracy of 66.67%. This classification rate is improved to 71.43% when measure $\mathcal{C}^{3D}_{\infty}(S)$ is added to the mix. Adding measure $\mathcal{C}^{3D}_{2}(S)$ does not yield any improvement in performance of the classifier. However, when the moment invariants are replaced by measure $\mathcal{C}^{3D}_{1}(S)$ the classification accuracy improves to 76.19%. The best classification accuracy is achieved by the combination of $\mathcal{C}^{3D}_{2}(S)$, $\mathcal{C}^{3D}_{\infty}(S)$, $\mathcal{C}^{3D}_{1}(S)$ and $\mathcal{C}^{3D}_{fit-1}(S)$, reaching 80.95% accuracy, which is equivalent to the best classification rate obtained by the knn classifier in section 5.5.4.

6.6 Conclusion

This chapter introduced another shape descriptor, derived in a similar way as the compactness and cubeness measures introduced in Chapter 4. Instead of Euclidean or chessboard distance, city block distance is used to derive this measure. The most compact shape using this distance metric is an octahedron. The new descriptor works as a similarity measure with an octahedron and is named octahedroness.

The examples in section 6.5.1 illustrate the behaviour of the octahedroness measure and compares it with a simple fitting shape method. The octahedroness measure developed, overcomes some of the problems encountered with the fitting method: the shape fitting method fails when it is applied to shapes with deep concavities, like the goblet on Figure 6.3(f) or to multi-component shapes like those in Figure 6.4. The developed octahedroness measure overcomes this problem and still gives different scores to such shapes.

The examples in section 6.5.2 show how the octahedroness measure can be combined with other descriptors to improve classification accuracy. Different combinations of descriptors produce different classification accuracies. As it has been mentioned before, a feature selection algorithm can be implemented to determine which set of descriptors produces best results.

Different weighting parameters could be incorporated to the octahedroness descriptor in the same way as 3D compactness and cubeness in Chapter 5. However, none of these modifications were included in the scope of this thesis and are an area of future research.

In the same way as the measures in Chapter 4 and Chapter 5, the measures presented in this chapter can be used as additional features for image retrieval applications like those presented by Funkhouser et al. (2003), Tangelder and Veltkamp (2004) and Bustos et al. (2005). As with the measures from Chapter 4 and Chapter 5, the inclusion of additional features is expected to improve performance.

Chapter 7

Conclusions

The focus of the research presented in this thesis was to develop new 2D and 3D shape descriptors suitable for implementation of 2D and 3D image processing applications. This thesis is not a conclusion to that work, but a catalogue of the descriptors developed at the present moment. Also, these descriptors are not (and do no intend to be) a one-size-fits-all solution, but rather should be considered to be additional tools in the image processing tool box: the adequate combination of tools should be selected to meet the specific needs of each application.

The experiments in this thesis utilise only a handful of shape descriptors to illustrate the use of these descriptors. More realistic applications would be considerably more complex and may utilise dozens of shape descriptors to achieve specific tasks.

7.1 Thesis summary and future work

The following section outlines the basic structure of this thesis, highlighting the main points of interest in each of the previous chapters. Some suggestions for further work are also given.

Chapter 1 gives a general introduction to shape descriptors and applications. Some well known descriptors from the literature are presented in this chapter. Figure 1.2 gives the basic structure of image processing applications; this basic stucture is considered on the experiments in the rest of the chapters.

In Chapter 2 some existing shape orientation methods are modified by including additional weighting parameters. This modification allows certain degree of control on the behaviour of the orientation methods: specific features can be given higher or lower relevance depending on the desired effect.

Chapter 3 introduced the linearity 2D descriptor. A set of illustrative experiments is also given. The presented linearity measure provides a reasonable alternative to the existing methods found in the literature. This measure overcomes some of the shortcomings of the existing methods. One of the key features of the developed linearity measure is that the measure $\mathcal{L}(\partial S)$ is designed to be applied to open curve segments, but by carefully selecting a start-and-finish-point, it can also be applied to closed curves. The points closest and furthest away from the shape centroid are suggested as start-and-finish-points, thus giving the measures $\mathcal{L}_{max}(\partial S)$ and $\mathcal{L}_{min}(\partial S)$. Experimental results indicate that $\mathcal{L}_{max}(\partial S)$ is a more informative measure and might be preferable to use this measure for most applications.

The 3D compactness and cubeness measures were introduced in Chapter 4. These two measures and the octahedroness measure from Chapter 6 share a common process for their derivation: they all measure the compactness of shapes under different distance metrics. A line of future research is to follow the same approach, using yet another distance metric. One possibility is too use Mahalanobis distance (Mahalanobis, 1936), which would lead to having ellipsoids as its most compact shape. Because Mahalanobis distance uses the covariance of the data to calculate the distance, the distribution of the data would play an important role in the computed measure.

The measures $\mathcal{C}_{\infty}^{3D}(S)$ and $\mathcal{C}_{1}^{3D}(S)$ require an additional step for finding the angles θ and ϕ which provide optimal orientation to the shape. This additional step is very computationally intensive which makes it desirable to have alternative methods to compute such optimal orientation. One possibility which could be explored is to rotate the shape following a spherical spiral path; the orientation of the shape could be given by the a vector (x, y, z) given by:

$$(x, y, z) = \left(\frac{\cos t}{\sqrt{1 + a^2 t^2}}, \frac{\sin t}{\sqrt{1 + a^2 t^2}}, \frac{-at}{\sqrt{1 + a^2 t^2}} \right)$$

where a is a constant. The orientation vector depends on a single parameter t (as opposed to the angles θ and ϕ) which may lead to a reduction in complexity of the algorithms to compute $\mathcal{C}_{\infty}^{3D}(S)$ and $\mathcal{C}_{1}^{3D}(S)$.

Figure 4.11 shows a feature space given by measures $\mathcal{C}_{2}^{3D}(S)$ and $\mathcal{C}_{\infty}^{3D}(S)$. In this very simple example, two different kinds of objects are separated in two groups using a parabolic decision boundary. In this example the decision boundary was drawn manually, however it was done only to illustrate the idea of class separation. A similar decision boundary could be easily generated by a properly trained classifier such as a neural network or a support vector machine.

In Chapter 5 the $\mathcal{C}_{2}^{3D}(S)$ and $\mathcal{C}_{\infty}^{3D}(S)$ methods are modified by incorporating additional parameters. These modifications allow the measures to modify their behaviour, capturing different features depending on the given parameters. A larger set of shape features can then be used in the classification process to improve classification performance as it was shown in the experiments in section 5.5.4. The number of features in these examples is still quite limited and manually selecting features is still possible, however automatic feature selection algorithms could be applied for a larger number of features (Nguyen and de la Torre, 2010).

The modifications introduced to measure $\mathcal{C}_2^{3D}(S)$ are not the same as those introduced to measure $\mathcal{C}_\infty^{3D}(S)$: for example the creation of $\overline{\mathcal{G}}_C(\beta)$ graphs is only applied to measure $\mathcal{C}_\infty^{3D}(S,\beta)$. This does not imply that any kind of modification is exclusive to one measure or another: the same process of generating $\overline{\mathcal{G}}_C(\beta)$ graphs could be applied to measure $\mathcal{C}_2^{3D}(S,\beta)$; or the different axis weighting in section 5.4 could be applied to create a ellipsoid similarity measures. These are possible lines of research to be followed at a later stage and were unfortunately out of the scope of this thesis.

Finally Chapter 6 introduced the basic octahedroness measure. None of the modifications from Chapter 5 were applied to the octahedroness measure. However it is in fact possible to extend measure $\mathcal{C}_1^{3D}(S)$ by adding weighting parameters, creating mean-graphs, adding different axis weighting, etc. These modifications are also possible lines for future research.

The experiment in section 6.5.2 mixes measure $\mathcal{C}_1^{3D}(S)$ with 3D moment invariants and the basic measures $\mathcal{C}_2^{3D}(S)$ and $\mathcal{C}_\infty^{3D}(S)$. The basic definition of 3D compactness and cubeness were used in this experiment instead of their weighted versions for simplicity reasons: it was easier to select a subset of descriptors from an already small group of descriptors; using the weighted versions would imply the additional process of selecting the best weighting parameters. Because the data set used was also fairly simple, such a small subset of descriptors would be sufficient to classify it.

Once weighted and mean-graph versions of compactness, cubeness and octahedroness are available, a large set of features can be created. Such set of features could then be applied to a larger, more complex data set. It is expected that the classification accuracy achieved in this scenario will improve with the addition of more descriptors. Of course, other factors which influence the performance of the application as a whole will be the feature selection algorithm and the classifier used.

Another possible line of research is to decompose a given shape into more simple geometric shapes and apply the proposed shape descriptors to these individual components. Some approaches like pyramid decomposition (Svensson and di Baja, 2002) could be used to break down a complex object into more simple components. Using such methods, an object can be decomposed into increasingly complex levels.

Shape descriptors could then be applied and objects compared at each level; objects which are similar in the lower levels of resolution may be found to be different at higher levels of resolution. This could be used for shape matching.

References

Atmosukarto, I., Wilamowska, K., Heike, C., and Shapiro, L. G. (2010). 3d object classification using salient point patterns with application to craniofacial research. *Pattern Recognition*, 43(4):1502–1517. 25

Benhamou, S. (2004). How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension? *Journal of Theoretical Biology*, 229(2):209–220. 24, 52, 58, 61, 64

Bors, A. G. (2006). Watermarking mesh-based representations of 3-d objects using local moments. *IEEE Transactions on Image Processing*, 15(3):687–701. 21

Bors, A. G., Kechagias, L., and Pitas, I. (2002). Binary morphological shape-based interpolation applied to 3d tooth reconstruction. *IEEE Trans. Med. Imaging*, 21(2):100–108. 18

Bribiesca, E. (2000). A measure of compactness for 3d shapes. *Computers & Mathematics with Applications*, 40:1275–1284. 24

Bribiesca, E. (2008). An easy measure of compactness for 2d and 3d shapes. *Pattern Recognition*, 41(2):543–554. 24

Bustos, B., Keim, D. A., Saupe, D., Schreck, T., and Vranić, D. V. (2005). Feature-based similarity search in 3d object databases. *ACM Computing Surveys*, 37(4):345–387. 26, 87, 123

Chen, C.-C. (1993). Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686. 54

Chen, G. and Kgl, B. (2010). Invariant pattern recognition using contourlets and adaboost. *Pattern Recognition*, 43(3):579–583. 22

Coeurjolly, D. and Klette, R. (2004). A comparative evaluation of length estimators of digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):252–257. 51

Do, M. N. and Vetterli, M. (2005). The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106. 22

Feng, J., Ip, H. H., Lai, L. Y., and Linney, A. (2008). Robust point correspondence matching and similarity measuring for 3d models by relative angle-context distributions. *Image and Vision Computing*, 26(6):761–775. 27

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D. (2003). A search engine for 3d models. *ACM Transactions on Graphics*, 22(1):83–105. 27, 87, 123

Gao, Y., Dai, Q., and Zhang, N.-Y. (2010). 3d model comparison using spatial structure circular descriptor. *Pattern Recognition*, 43(3):1142–1151. 26

Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing*. Prentice Hall, Upper Saddl River, New Jersey 07458, 2nd edition edition.

Gouet-Brunet, V. and Lameyre, B. (2008). Object recognition and segmentation in videos by connecting heterogeneous visual features. *Computer Vision and Image Understanding*, 111(1):86–109. Special Issue on Intelligent Visual Surveillance (IEEE). 25

Grum, M. and Bors, A. G. (2008). Multiple image disparity correction for 3-d scene representation. In *ICIP*, pages 209–212, San Diego, CA. 18

Han, M., Xu, W., Tao, H., and Gong, Y. (2004). An algorithm for multiple object trajectory tracking. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–864 – I–871 Vol.1. 64

Hermann, S. and Klette, R. (2007). A comparative study on 2d curvature estimators. In *ICCTA '07: Proceedings of the International Conference on Computing: Theory and Applications*, pages 584–589, Washington, DC, USA. IEEE Computer Society. 40, 46

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187. 20, 54

Huang, P., Hilton, A., and Starck, J. (2010). Shape similarity for 3d video sequences of people. *International Journal of Computer Vision*. 28

Imre, A. R. (2009). Fractal dimension of time-indexed paths. *Applied Mathematics and Computation*, 207(1):90 – 94. Includes Special issue on Emergent Applications of Fractals and Wavelets in Biology and Biomedicine. 25

Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine vision*. Nueva York, EUA : McGraw-Hill. 32

Kim, H. and Sohn, K. (2005). 3d reconstruction from stereo images for interactions between real and virtual objects. *Signal Processing: Image Communication*, 20(1):61 – 75. 18

Kindratenko, V. (1997). *Development and Application of Image Analysis Techniques for Identification and Classification of Microscopic Particles*. Ph.d. thesis, University of Antwerp, Belgium. 21

Klette, R. and Žunić, J. (1999). Digital approximation of moments of convex regions. *Graphical Models and Image Processing*, 61(5):274 – 298. 51

Lambert, G. and Gao, H. (2006). Line moments and invariants for real time processing of vectorized contour data. In *Image Analysis and Processing*, volume 974 of *LNCS*, pages 347–352. Springer Berlin. 20

Li, F., Dai, Q., Xu, W., and Er, G. (2010). Statistical modeling and many-to-many matching for view-based 3d object retrieval. *Signal Processing: Image Communication*, 25(1):18 – 27. 27

Lian, Z., Rosin, P. L., and Sun, X. (2009). Rectilinearity of 3d meshes. *International Journal of Computer Vision*. 27

Liu, H., Latecki, L. J., and Liu, W. (2008). A unified curvature definition for regular, polygonal, and digital planar curves. *International Journal of Computer Vision*, 80(1):104–124. 51

Lmaati, E. A., Oirrak, A. E., Aboutajdine, D., Daoudi, M., and Kaddioui, M. N. (2010). A 3-d search engine based on fourier series. *Computer Vision and Image Understanding*, 114(1):1–7. 27

Mahalanobis, P. C. (1936). On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55. 126

Mamistvalov, A. G. (1998). n-dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):819–831. 23, 70, 71, 107

Manay, S., Cremers, D., Hong, B.-W., Yezzi Jr, A., and Soatto, S. (2006). Integral invariants and shape matching. In Krim, H. and Jr., A. Y., editors, *Statistics and Analysis of Shapes*, Modeling and Simulation in Science, Engineering and Technology, pages 137–166. Birkhuser Boston. 51

Martinez-Ortiz, C. and Žunić, J. (2008). Points position weighted shape orientation. *Electronics Letters*, 44(10):623–625.

Martinez-Ortiz, C. and Žunić, J. (2009). Measuring cubeness of 3d shapes. In *CIARP '09: Proceedings of the 14th Iberoamerican Conference on Pattern Recognition*, volume 5856/2009 of *LNCS*, pages 716–723, Berlin, Heidelberg. Springer-Verlag.

Martinez-Ortiz, C. and Žunić, J. (2010a). Curvature weighted gradient based shape orientation. *Pattern Recognition*, 43(9):3035–3041.

Martinez-Ortiz, C. and Žunić, J. (2010b). A family of cubeness measures. *Submitted*.

Martinez-Ortiz, C. and Žunić, J. (2010c). Measuring cubeness in the limit cases. *Submitted*.

Martinez-Ortiz, C. and Žunić, J. (2010d). Tuneable cubeness measures for 3d shapes. *Submitted*.

Meyer, Y. (1993). *Wavelets: Algorithms & Applications*. Society for Industrial and Applied Mathematics. Translated by Robert D. Ryan. 22

Mio, W., Srivastava, A., and Joshi, S. (2007). On shape of plane elastic curves. *International Journal of Computer Vision*, 73(3):307–324. 51

Nams, V. O. (2006). Detecting oriented movement of animals. *Animal Behaviour*, 72(5):1197–1203. 24, 58

Nguyen, M. H. and de la Torre, F. (2010). Optimal feature selection for support vector machines. *Pattern Recognition*, 43(3):584–591. 127

Rosin, P. L. (2003). Measuring shape: ellipticity, rectangularity, and triangularity. *Machine Vision and Applications*, 14(3):172–184. 22

Rosin, P. L. (2004). Measuring sigmoidality. *Pattern Recognition*, 37(8):1735–1744. 22

Sadjadi, F. A. and Hall, E. L. (1980). Three-dimensional moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(2):127–136. 23

Saleem, W., Wang, D., Belyaev, A. G., and Seidel, H.-P. (2007). Automatic 2d shape orientation by example. In *2007 International Conference on Shape Modeling and Applications (SMI 2007)*, pages 221–225, Lyon, France. IEEE. 49, 50

Shen, L. and Makedon, F. (2006). Spherical mapping for processing of 3d closed surfaces. *Image and Vision Computing*, 24(7):743–761. 28

Sheynin, S. A. and Tuzikov, A. V. (2001). Explicit formulae for polyhedra moments. *Pattern Recognition Letters*, 22(10):1103–1109. 23

Sonka, M., Hlavac, V., and Boyle, R. (1998). *Image Processing: Analysis and Machine Vision*. Thomson-Engineering.

Srivastava, A., Samir, C., Joshi, S. H., and Daoudi, M. (2009). Elastic shape models for face analysis using curvilinear coordinates. *Journal of Mathematical Imaging and Vision*, 33(2):253–265. 62

Stojmenović, M., Amiya, N., and Žunić Joviša (2008). Measuring linearity of planar point sets. *Pattern Recognition*, 41(8):2503–2511. 52

Stojmenović, M. and Žunić, J. (2008). Measuring elongation from shape boundary. *Journal of Mathematical Imaging and Vision*, 30(1):73–85. 22, 51

Svensson, S. and di Baja, G. S. (2002). Using distance transforms to decompose 3d discrete objects. *Image and Vision Computing*, 20(8):529 – 540. 127

Tangelder, J. W. H. and Veltkamp, R. C. (2004). A survey of content based 3d shape retrieval methods. In *SMI '04: Proceedings of the Shape Modeling International 2004*, pages 145–156, Washington, DC, USA. IEEE Computer Society. 87, 123

Tsai, W.-H. and Chou, S.-L. (1991). Detection of generalized principal axes in rotationally symmetric shapes. *Pattern Recognition*, 24(2):95 – 104. 21, 35, 42, 43, 49

Vaidya, S. and Bijukumar, L. (2010). Some new families of mean graphs. *Journal of Mathematics Research*, 2(3):169–176. 97

Vranić, D. V. and Saupe, D. (2001). 3d shape descriptor based on 3d fourier transform. In Fazekas, K., editor, *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001)*, pages 271–274, Budapest, Hungary. 24, 27, 113

Žunić, J., Hirota, K., and Martinez-Ortiz, C. (2010a). Compactness measure for 3d shapes. *Submitted*.

Žunić, J., Hirota, K., and Rosin, P. L. (2010b). A hu moment invariant as a shape circularity measure. *Pattern Recognition*, 43(1):47–57. 21, 25

Žunić, J., Kopanja, L., and Fieldsend, J. E. (2006). Notes on shape orientation where the standard method does not work. *Pattern Recognition*, 39(5):856 – 865. 49

Žunić, J. and Martinez-Ortiz, C. (2009). Linearity measure for curve segments. *Applied Mathematics and Computation*, 215(8):3098–3105. 57

Žunić, J. and Stojmenović, M. (2008). Boundary based shape orientation. *Pattern Recognition*, 41(5):1768 – 1781. 36, 37, 49

Xu, D. and Li, H. (2008). Geometric moment invariants. *Pattern Recognition*, 41(1):240–249. 70

Yang, L., Albregtsen, F., and Taxt, T. (1997). Fast computation of three-dimensional geometric moments using a discrete divergence theorem and a generalization to higher dimensions. *Graphical Models and Image Processing*, 59(2):97–108. 23