Surface-Area-Based Attribute Filtering in 3D

Fred N. Kiwanuka[1,2], Georgios K. Ouzounis[3], and Michael H.F. Wilkinson[1]

[1] Institute for Mathematics and Computing Science, University of Groningen,
P.O. Box 407, 9700 AK Groningen, The Netherlands
[2] Faculty of Computing and Information Technology, Makerere University,
P.O. Box 7062 Kampala, Uganda
[3] School of Medicine, Democritus University of Thrace,
University General Hospital of Alexandroupoli, 68100 Alexandroupoli, Greece
F.N.Kiwanuka@rug.nl, gouzoun@med.duth.gr, m.h.f.wilkinson@rug.nl

Abstract. In this paper we describe a rotation-invariant attribute filter based on estimating the sphericity or roundness of objects by efficiently computing surface area and volume of connected components. The method is based on an efficient algorithm to compute all iso-surfaces of all nodes in a Max-Tree. With similar properties to moment-based attributes like sparseness, non-compactness, and elongation, our sphericity attribute can supplement these in finding blood-vessels in time-of-flight MR angiograms. We compare the method to a discrete surface area method based on adjacency, which has been used for urinary stone detection. Though the latter is faster, it is less accurate, and lacks rotation invariance.

1 Introduction

Connected filters [1,2] are members of the larger family of morphological operators, that act on the flat zone level of gray-scale images. Connected filters have the capacity to precisely identify and extract connected components in their entirety and without distorting their boundaries. This property, critical in many applications such as medical imaging, increases their popularity and makes them a suitable tool for problems in which accurate shape analysis is of importance. Connected components can either be removed or remain intact but new ones cannot emerge. Emergence and distortion of components is an existing problem in many other filtering methods.

Attribute filters are a subset of connected operators [3,4], that access connected components of threshold sets based on their attributes. Examples are attribute openings, closings, thinnings and thickenings [3,4,5]. Attribute filters often rely on either size or shape criteria such as volume, simplicity, complexity, moment of inertia, non-compactness, etc. [3,4,6,7]. Attributes are called *shape descriptors* provided they satisfy three key properties: translation, scale and rotation invariance [8]. Including scale invariance comes at the expense of increasingness [8], and rotation invariance can come at the cost of a high computational overhead, and reducing this is the topic of ongoing research. Exceptions are moment invariants [7,8], which are rotation invariant by definition, and can

M.H.F. Wilkinson and J.B.T.M. Roerdink (Eds.): ISMM 2009, LNCS 5720, pp. 70–81, 2009.

be computed efficiently. Description by moment invariants is limited, hence the need for new shape descriptors. In this paper we present a non-increasing 3D shape descriptor to measure how spherical (round) objects are, emphasizing rotation invariance. The *sphericity* attribute relies on accurate computation of surface area for which a new method is developed. It is compared to a simpler method based on the 3D equivalent of the city-block perimeter length. The performance of both methods is evaluated in terms of speed and accuracy.

Outline. The remainder of this article is organized as follows. A short introduction on attribute filters is given in Section 2 followed by a brief description of the Max-Tree algorithm in Section 3, used for computing the sphericity attribute. The latter is described in further detail in Section 4. Finally, Section 5 presents the results of the sphericity attribute used for 3D medical image enhancement, comparing it to other methods.

2 Theoretical Background

In the following, binary images X are subsets of some non-empty, universal set E. The set of all subsets of E is denoted $\mathcal{P}(E)$.

2.1 Attribute Filters

Attribute filters are based on connectivity openings. A connectivity opening $\Gamma_x(X)$ yields the connected component containing the point $x \in X$, and $\emptyset$ otherwise. A connectivity opening is anti-extensive i.e. $\Gamma_x(X) \subseteq X$, increasing i.e. $X \subseteq Y \Rightarrow \Gamma_x(X) \subseteq \Gamma_x(Y)$ and idempotent i.e.$\Gamma_x(\Gamma_x(X)) = \Gamma_x(X)$. Furthermore, for all, $X \subseteq E$, $x, y \in E, \Gamma_x(X)$ and $\Gamma_x(Y)$ are equal or disjoint. Attribute filters are defined using a family of connectivity openings, by imposing constraints on the connected components they return. Such constraints are expressed in the form of binary criteria which decide to accept or to reject components based on some attribute measure. Let $\Delta : \mathcal{P}(E) \to \{\text{false}, \text{true}\}$ be an attribute criterion; then Γ_Δ is a trivial opening returning the connected component C if $\Delta(C)$ is true and $\emptyset$ otherwise. Moreover, $\Gamma_\Delta(\emptyset) = \emptyset$. Attribute criteria can be represented as: $\Delta(C) = \mathit{Attr}(C) \geq \lambda$ where $\mathit{Attr}(C)$ is some real-value attribute of C and λ is an attribute threshold. A binary attribute opening Γ^Δ of a set X with an increasing criterion can be defined as

$$\Gamma^\Delta(X) = \bigcup_{x \in X} \Gamma_\Delta(\Gamma_x(X)) \qquad (1)$$

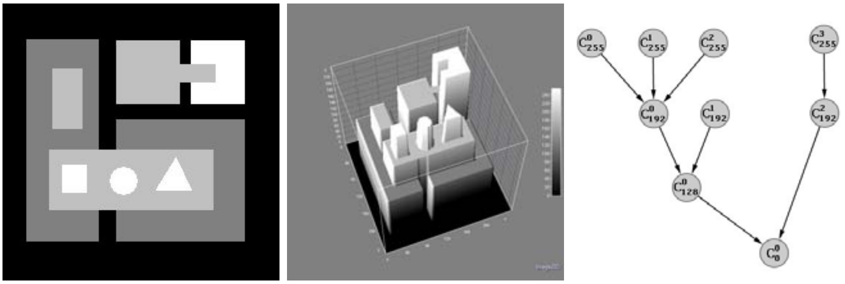
For non-increasing criteria Δ we obtain an attribute thinning. The sphericity attribute $S(C)$ is an example and is represented in terms of surface area($A(C)$) and volume($V(C)$) of each component as

$$S(C) = \frac{\pi^{\frac{1}{3}}(6V(C))^{\frac{2}{3}}}{A(C)} \qquad (2)$$

3 The Max-Tree

The Max-Tree [4] data structure was designed for morphological attribute filtering in image processing. The nodes (C_h^k, k is the node index, h the gray level) of the Max-Tree represent connected components for all threshold levels in a data set (see Fig. 1). These components are referred to as *peak components* and are denoted as P_h^k. Each node C_h^k contains only those pixels of peak component P_h^k which have gray level h. The root node represents the set of pixels belonging to the background, that is the set of pixels with the lowest intensity in the image and each node has a pointer to its parent. The nodes corresponding to the components with the highest intensity are the leaves. The filtering process is separated into three stages: construction, filtering and restitution. During the construction phase, the Max-Tree is built from the flat zones of the image, collecting auxiliary data used for computing the node attributes at a later stage. The auxiliary data can be used to compute one or more attributes, that describe certain properties of the peak components represented by those nodes. The filtering process is based on certain rules like the *Direct*, *Min*, *Max*, and *Viterbi* rules [3,4], and more recently the *Subtractive* rule [8], and *Branches* rule [9,10]. These filtering rules are all designed to deal with *non-increasing* attributes, in which accept and reject decisions might alternate along any path from leaf to root. In the following we will only use the Subtractive rule, which works best for blood-vessel enhancement and image decomposition based on shape [6,7,8].

Whatever filtering rule used, a key problem when computing Max-Trees lies in efficient computation of the attributes. Following the approach in Breen and Jones [3], most methods use a recursive procedure in which data are collected during the construction phase, and passed down to the parents. This approach is fine for any attribute which can be computed incrementally, such as those based on moments, histograms, or on the minimum and maximum coordinate values. It has however tended to limit the attributes used. In [7] several augmentations to the Max-Tree were made for interactive visualization purposes, including fast iso-surface rendering. We will use these extensions to compute surface area efficiently.



(a) (b) (c)

Fig. 1. Example of a Max-Tree: (a) original image; (b) viewed as 3D surface; (c) corresponding Max-Tree

4 Sphericity Attribute Computation

A simple algorithm (called adjacency method in the following) to compute an approximate surface area is based on 6 connectivity in 3D [11]. Surface area computation begins by detecting all surface voxels, i.e. object voxels that are 6-connected to background voxels. For a three-dimensional connected component that is represented as a set of voxels, one can easily identify the voxels that are 6-connected to the background and view them as the border of the object. The boundary of the object is the set of voxel faces that separates the object from the background. Now the concept of 6-connectedness stems from the fact that, for any two voxels (x_1, x_2, x_3) and (y_1, y_2, y_3) are called 6 face adjacent if $\Sigma_{i=1}^{3}(x_i - y_i)^2 = 1$ [12]. To solve the problem of boundary extraction, one approach is to visit each voxel in each object and determine whether it is 6-connected to a background voxel or not.

Once the set of voxel faces that separate the object from the background are identified their areas are estimated separately and added up to constitute the surface area. In short, for each voxel in a connected component, simply compute the number of 6-connected neighbors *outside* the component. This is equal to the number of faces of each voxel on the boundary. The sum of these values over the component is the surface area of the discrete representation of the object.

In gray scale, things are more complicated, as the voxels cannot readily be classified as object and background. Using the depth-first construction algorithm for Max-Tree construction from [4] (or for that matter the union-find approach of [13]), the surface area of the current node C_h^k is initialized at zero, and the surface areas of any child node at higher grey level is added to it. For each voxel at level h within C_h^k we compute the number of adjacent voxels of *lower* gray level, and add this to the surface area of the component. We also compute the number of adjacent voxels with gray level *higher* than h. This number is subtracted from the surface area of the current node C_h^k, because it represents part of the boundary between the current node and one of its children. This part has previously been added to the surface area of the node, but is *not* part of the boundary of the peak component P_h^k represented by node C_h^k, and must therefore be subtracted.

The adjacency method can be implemented in the usual recursive way, using auxiliary data as proposed by [3,4]. The computation of the fast sphericity from the image data is performed as follows:

- Compute Max-Tree according to the algorithm in [4].
- As the Max-Tree is built compute the Volume($V(C_h^k)$) of each node using the existing voxel-based algorithm
- For each node (C_h^k)
  - initialize surface area $A(C_h^k)$ to zero,
  - add surface areas of children (components at higher levels) to $A(C_h^k)$,
  - add number of 6-adjacent voxels to C_h^k with gray level $< h$ to $A(C_h^k)$

- subtract number of 6-adjacent voxels to C_h^k with gray level $> h$ from $A(C_h^k)$
- compute sphericity $\pi^{\frac{1}{3}}(6V(C_h^k))^{\frac{2}{3}}/A(C_h^k)$.

The above algorithm assumes isotropic voxels with unit surface area faces. To deal with anisotropic voxels, we compute separate sums of neighbours in x, y and z directions, multiply each sum with the appropriate surface area, and add the individual results together to obtain the final surface area of each node. The time complexity of this algorithm (disregarding Max-Tree construction) is $O(N)$, with N the number of voxels. This is because for each voxel, we need to inspect 6 neighbors, perform at most one addition and one subtraction. The surface area of each node is added only once to its parent, and because the number of nodes is bounded by N this part is also $O(N)$.

The problem, as many have noted, is that this does not yield an accurate measure of surface area [14]. Worse still, as resolution increases, the estimate does not improve. We therefore adapt a method for fuzzy sets [14], based on iso-surface estimation to grey-scale volumes and attribute filtering.

Rendering of an object often requires iso-surface detecting algorithms such as marching cubes [15] which produce a list of triangles approximating the surface. Iso-surfaces are formed from each level set of the function f for which points (x, y) have a constant intensity. To find these triangle meshes we need to process all cells of the volume. A cell is a cube with the voxel centers at its corners. These triangle meshes are quickly obtained from *active cells* that intersect the surface during visualization process, using either range based search or seed set generation methods [16], but also using an augmented Max-Tree [7]. There can be up to 5 triangles per each active cell.

In each active cell we have n triangles. Let $\boldsymbol{v}_i$ be vertices of the triangles, such that $\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{v}_{i+2}$ form a triangle. Then the sum of the areas of each triangle approximates the surface area $A(S)$ of the iso-surface patch S intersecting the cell. This is computed as

$$A(S) = \frac{1}{2}\Sigma_{i=0}^{n}|(\boldsymbol{v}_{3i+1} - \boldsymbol{v}_{3i}) \times (\boldsymbol{v}_{3i+2} - \boldsymbol{v}_{3i})| \tag{3}$$

where × denotes the vector cross product. We could recursively go through the Max-Tree, and for each node C_h^k obtain triangle meshes from the augmented Max Tree representation and compute their area $A(C_h^k)$. This would lead to repeated visits of cells in the volume leading to cash thrashing. Here we purpose a different approach, as explained below.

In our *iso-surface* method we use the augmented Max-Tree that adds visualization data to the Max-Tree [7]. For the purposes of this paper the most important additions are the `Dilated` and `Eroded` arrays. These contain the maximum and minimum node along the root path passing through each cell. Note that we use 26 connectivity so that all eight voxels at the corners of each cell are in the same root path. The aim is to compute the surface areas of all iso-surfaces at all levels in the Max-Tree corresponding to the nodes.

The surface area attribute computation takes the following procedure:

- Compute the augmented Max Tree.
- Compute the volume $V(C_h^k)$ of each node using the existing voxel-based algorithm.
- For each cell
  - find maximum and minimum nodes from `Dilated` and `Eroded` arrays
  - set current node to maximum
  - while current node is not minimum
    - compute grey level intermediate between current node and its parent
    - compute triangle mesh of iso-surface at that level for the current cell
    - calculate area of the triangles in the mesh and add to area of corresponding node.
    - set current node to its parent
- compute the surface area of the root node from the volume dimensions.
- For each node C_h^k compute sphericity $\pi^{\frac{1}{3}}(6V(C_h^k))^{\frac{2}{3}}/A(C_h^k)$.

Sphericity for a sphere is 1.0 and for a cube it is approximately 0.806. For our iso-surface method computation of surface area and volume for data objects should be as close as possible for specific objects' true values, and retain rotation invariance. Thus we compare our sphericity computation with the adjacency method in [11], which is the only other 3-D surface-area-based attribute filter implementation to date.

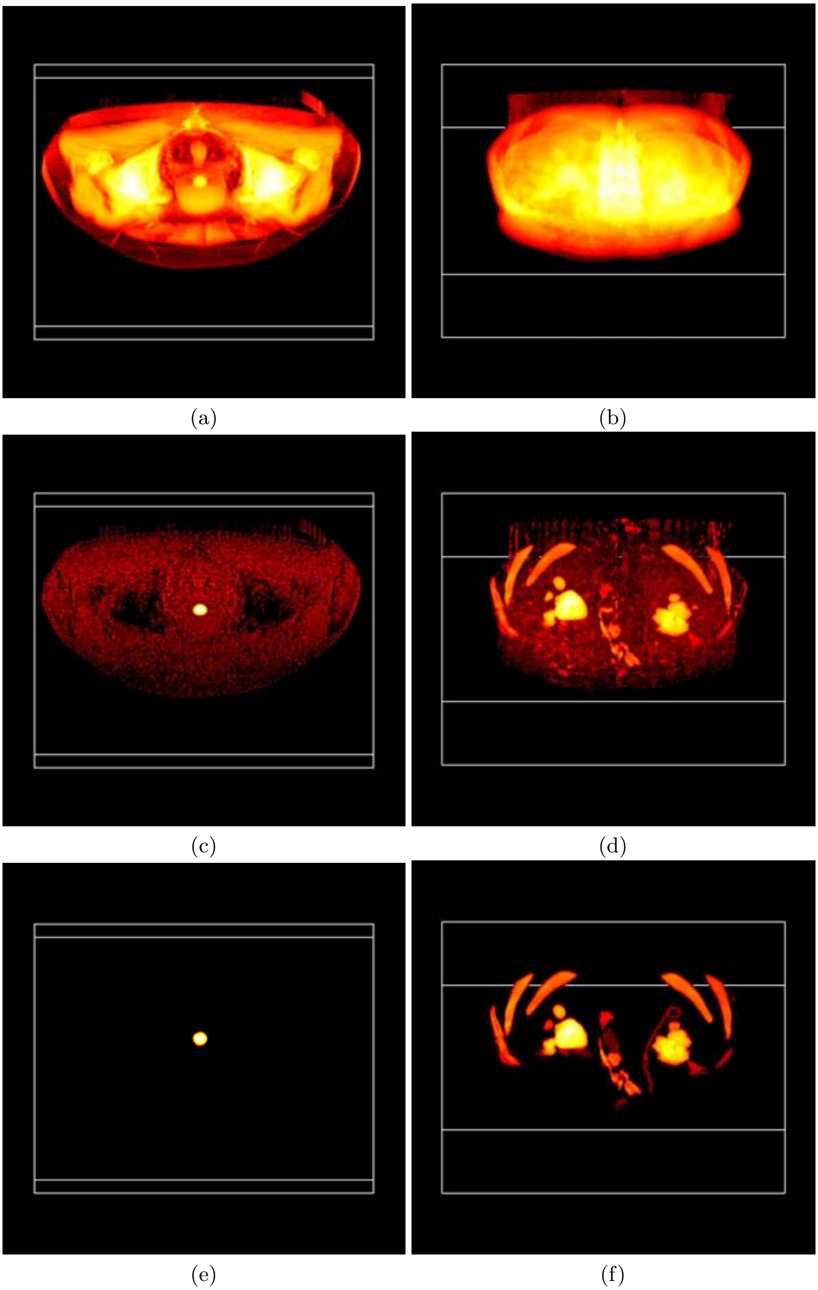
The computational complexity of this algorithm is $O(N\Delta G)$ with N being the number of cells or voxels, and ΔG is the mean grey level range within each cell. Thus the algorithm should slow down as grey level resolution is increased. Memory costs of both algorithms are the same, requiring storage of two doubles per node: one for volume and one for surface area.

5 Results and Discussion

5.1 Computational Cost

After computing the Max-Tree representation of a data set, the attribute of each node is computed from its auxiliary data. In the filtering stage and based on each node's attribute value, a decision is made and its level is modified according to the rule chosen. To measure the algorithm's computational performance, we ran some timing experiments on a Core 2 Duo E8400 at 3.0 GHz. Attribute computation times in seconds are shown in the Table. 1. We ran tests on the `mrt16_angio2` and `mrt8_angio2` time-of-flight magnetic resonance angiograms (MRA) from `http://www.volvis.org` and the `fullhead` CT data set included with VTK at 8 bit and 12 bit grey-level resolution for several attributes. As expected, the iso-surface based attribute is slower than the other attributes.

As expected the iso-surface algorithm performs more slowly on 12-bit resolution data sets than the equivalent 8-bit data sets. The effect is most pronounced on the `fullHead` data set, in which the grey level range is 4095 for the 12-bit data. The 12-bit `mrt16_angio2` has only a 576 grey level range, which explains why it only doubles in computing time with respect to the 8-bit version.



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

Fig. 2. Sphericity filtering of CT scans along the urinary tract in X-ray rendering mode: (a) the unfiltered view of a bladder calculus and (b) of kidney calculi; the results of the sphericity filter for each set with $\lambda = 0.94$ (c), and $\lambda = 0.39$ (d) respectively; the results of the volume filter following the sphericity filter with $\lambda = 78$ (e) and $\lambda = 450$ (f) respectively

Table 1. Computing time (in seconds) of various attributes

	`fullHead`		`mrt*angio2`		`vessels`
Attribute	12 bits	8 bits	12 bits	8 bits	8 bits
Sphericity(Iso.)	123.13	8.64	26.77	14.10	2.94
Sphericity(Adj.)	1.44	0.98	1.44	1.30	0.85
Non Compactness	0.70	0.57	0.66	0.62	0.83
Sparseness	0.98	0.83	0.92	0.85	1.18
Volume	0.15	0.08	0.14	0.12	0.12

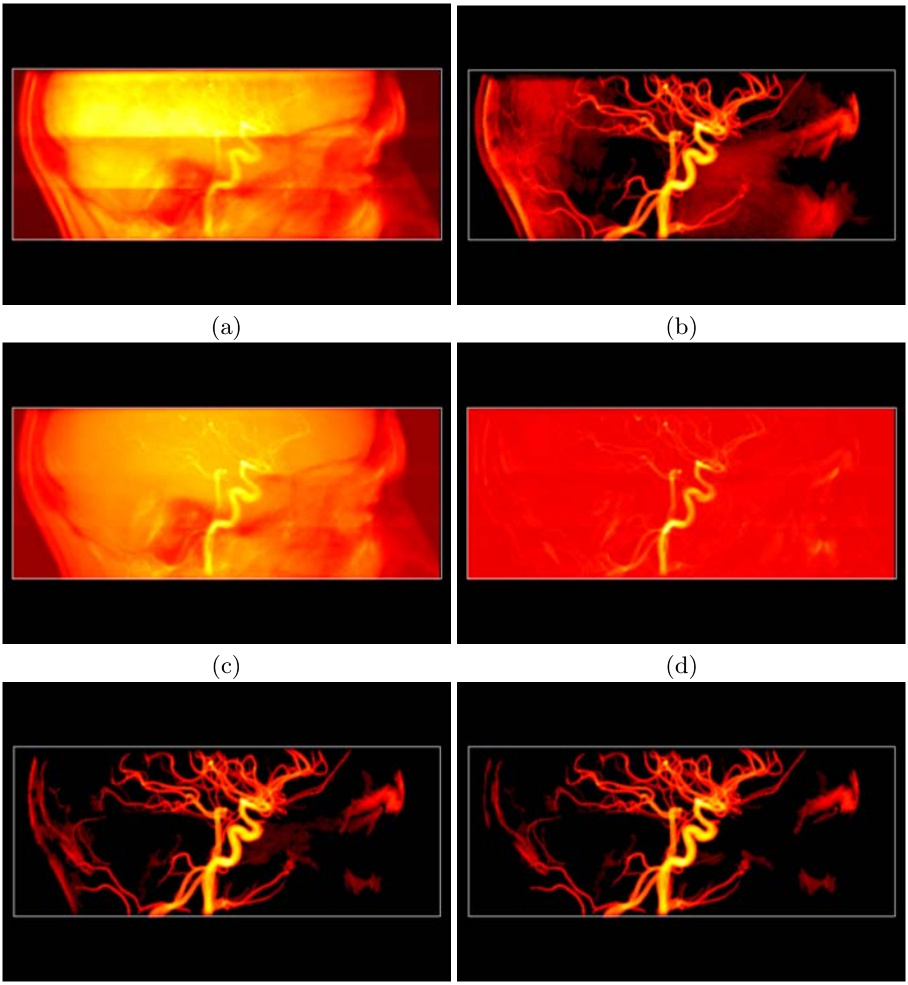
5.2 Performance Evaluation

CT Scans of the Urinary Tract. To evaluate the performance of the sphericity filter in isolating compact structures, we used two 3D CT data sets of patients suffering from urolithiasis. Both are courtesy of the Department of Radiology and Medical Imaging, University General Hospital of Alexandroupolis, Greece. The first one shown at the top of left column of Fig. 2 is of a patient diagnosed with a bladder calculus. The stone is rather compact allowing for high values of λ. Fig. 2 (c) shows the sphericity filter output for $\lambda = 0.94$, for which all other anatomical structures are suppressed. Followed by a volume opening with $\lambda = 78$, we can completely isolate the stone from the remaining noise - Fig. 2 (e).

The second data set shows a severe case of nephrolithiasis, in which a stent is inserted to one of the kidneys to bypass the obstruction of urine flow. In each kidney there exists one major calculus accompanied by other smaller ones. Due to their arbitrary shape, extracting them all (Fig. 2 (d)) requires a low value for the sphericity threshold – $\lambda = 0.39$. To enhance the scene further, a volume opening is applied with $\lambda = 450$, shown in Fig. 2 (f).

Time-of-Flight Angiography. Time-of-Flight angiograms are very difficult to filter. On the `mrt16_angio2` data set (from `http://www.volvis.org/`, the performance of the two methods is hard to distinguish as shown in Fig. 3(c) and (d). Both methods perform equally in terms of retaining or discarding specific features. However, both struggle in suppressing the background without removing the vessels. By contrast, the moment-based non-compactness attribute from [6,7] removes background much more effectively (see Fig. 3(b)). However, it too retains unwanted features. In Fig. 3(e) and (f) we see the effect of applying the sphericity filters to the volume in Fig. 3(b). Both show a distinct improvement on background suppression. Evidently, the blood vessels have a larger sphericity than most unwanted features retained by the non-compactness filter.

Accuracy and Rotational Invariance. To measure the accuracy of the methods we measure the sphericity for several synthetic objects of known sphericity. The first volume is an inverted Euclidean distance map from the Volume Library (`http://www9.informatik.uni-erlangen.de/External/vollib/`). This yields a series of Euclidean spheres with increasing radii. A separate volume of cubes and blocks of various sizes was made. Finally, a volume consisting of a rod of 4×4×32



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Sphericity filtering of time-of-flight MRA: (a) Xray rendering of original volume; (b) filtered with non-compactness attribute at $\lambda = 3.7$; (c) original filtered with iso-surface sphericity attribute at $\lambda = 0.1320$; (d) original filtered with adjacency sphericity attribute at $\lambda = 0.1124$; (e) volume (b) filtered with iso-surface sphericity attribute at $\lambda = 0.1320$; (f) volume (b) filtered with adjacency sphericity attribute at $\lambda = 0.1124$

rotated from 0° to 45° in 5° steps. For both algorithms we computed the sphericity attributes. For an axis aligned cube of 32^3 the sphericity of the iso-surface method yielded 0.821 versus 0.806 for the adjacency method. The latter is the theoretical value. The difference is due to the rounding of the corners by the iso-surface method.

The results for the Euclidean distance map are in Fig. 4. This shows that the iso-surface method approaches the theoretical value of 1.0 as radius r increases.

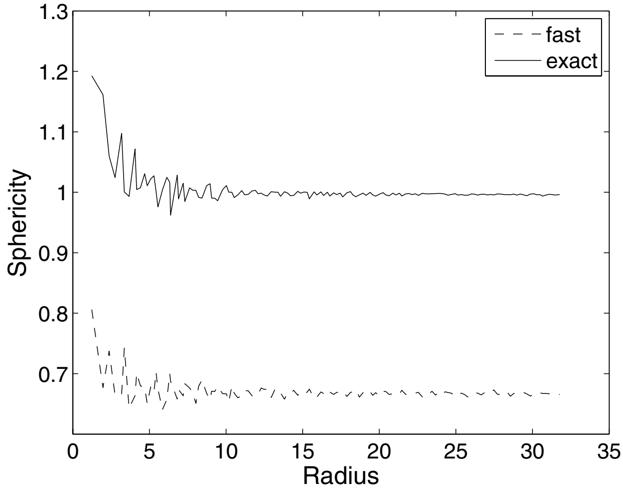


Fig. 4. Attribute measure as a function of sphere diameter for Euclidean spheres, for both the iso-surface (solid) and adjacency method (dashed)

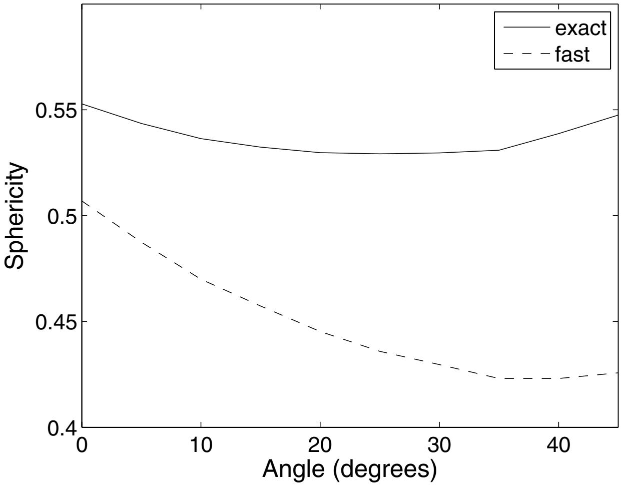


Fig. 5. Sphericity of a 4 × 4 × 32 rod rotated from 0° to 45° in steps of 5°, for both the iso-surface (solid) and adjacency method (dashed)

The values larger than 1 at small r are caused by the fact that the volume is computed voxel-wise, and the surface from the iso-surface. This means that the iso-surface can enclose a smaller volume than that measured by the voxel-based volume computation. By contrast the adjacency method consistently approaches 2/3 asymptotically, which is what is expected from theory.

Fig. 5 shows the results of the rotation invariance test. Clearly, as we rotate the rod the iso-surface method shows more rotation invariance than the adjacency method. Between 45° and 90° the results mirror those in the graph. In a second test we compared the sphericity measure of a $32 \times 32 \times 32$ cube (axis aligned) with the same cube rotated by 45° over the z-axis. For the axis aligned cube we have 0.806 for the adjacency method and 0.821 for the iso-surface method as noted above. When we rotate over 45 degrees, the values become 0.637 for the adjacency method (theoretically approaching 0.632 or 22% off), versus 0.818 for ours (or 0.4% lower than unrotated) again proving our improved rotation invariance.

6 Conclusions

Rotation invariance is an important property for object detection, and it is a challenge to achieve it in the case of surface-area-based attributes. In this paper we presented a rotation invariant, iso-surface method whose performance was compared to the adjacency method from [11]. The performance of both methods is shown for images rotated at different angles. While the iso-surface method performs well in terms of rotation invariance, the adjacency method [11] is faster. The accuracy of the iso-surface method was better than that of the adjacency method for large structures, though on small spheres the accuracy was reduced. This should be improved in the future. Both methods struggled when applied to very noisy data sets. This is because sphericity is less robust to noise than the moment-based non-compactness measure from [6]. The reason for this is that surface area changes dramatically as small holes appear in a component due to noise, whereas both volume and moment of inertia are affected in more or less the same way by small holes. However, the combination of sphericity filtering and non-compactness filtering in time-of-flight MRAs is clearly superior to only using non-compactness.

Though we focused on sphericity measures in this paper, surface area itself, and many other surface-area-based attributes can be derived in the same way. All code, data sets and an installer program for Microsoft Windows demonstrating these possibilities can be downloaded from http://www.cs.rug.nl/~michael/MTdemo. This source code can also be compiled under Linux and Apple's OS-X. Finally, the iso-surface method presented here only uses a single core of the CPU. Parallelization of this algorithm should be straightforward, and will be done in the future.

References

1. Heijmans, H.J.A.M.: Connected morphological operators for binary images. Comp. Vis. Image Understand. 73, 99–120 (1999)
2. Salembier, P., Serra, J.: Flat zones filtering, connected operators, and filters by reconstruction. IEEE Trans. Image Proc. 4, 1153–1160 (1995)
3. Breen, E.J., Jones, R.: Attribute openings, thinnings and granulometries. Comp. Vis. Image Understand. 64(3), 377–389 (1996)

4. Salembier, P., Oliveras, A., Garrido, L.: Anti-extensive connected operators for image and sequence processing. IEEE Trans. Image Proc. 7, 555–570 (1998)
5. Meijster, A., Wilkinson, M.H.F.: A comparison of algorithms for connected set openings and closings. IEEE Trans. Pattern Anal. Mach. Intell. 24(4), 484–494 (2002)
6. Wilkinson, M.H.F., Westenberg, M.A.: Shape preserving filament enhancement filtering. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 770–777. Springer, Heidelberg (2001)
7. Westenberg, M.A., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Volumetric attribute filtering and interactive visualization using the max-tree representation. IEEE Trans. Image Proc. 16, 2943–2952 (2007)
8. Urbach, E.R., Roerdink, J.B.T.M., Wilkinson, M.H.F.: Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. IEEE Trans. Pattern Anal. Mach. Intell. 29, 272–285 (2007)
9. Naegel, B., Passat, N., Boch, N., Kocher, M.: Segmentation using vector-attribute filters: Methodology and application to dermatological imaging. In: Proc. Int. Symp. Math. Morphology (ISMM 2007), pp. 239–250 (2007)
10. Purnama, K.E., Wilkinson, M.H.F., Veldhuizen, A.G., van Ooijen, P.M.A., Lubbers, J., Sardjono, T.A., Verkerke, G.J.: Branches filtering approach for max-tree. In: Proc. 2nd Int. Conf. Comput. Vision Theory Applic (VISAPP 2007), Barcelona, March 8-11, pp. 328–332 (2007)
11. Ouzounis, G.K., Giannakopoulos, S., Simopoulos, C.E., Wilkinson, M.H.F.: Robust extraction of urinary stones from CT data using attribute filters. In: Proc. Int. Conf. Image Proc. (in press, 2009)
12. Udupa, J.K.: Multidimensional digital boundaries. CVGIP: Graphical Models and Image Processing 56(4), 311–323 (1994)
13. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Proc. 15, 3531–3539 (2006)
14. Sladoje, N., Nyström, I., Saha, P.K.: Measurements of digitized objects with fuzzy borders in 2d and 3d. Image Vis. Comput. 23, 123–132 (2005)
15. Kaufman, A., Mueller, K.: Overview of volume rendering. In: The Visualization Handbook. Academic Press, London (2005)
16. Cignoni, P., Marino, P., Montani, C., Puppo, E., Scopigno, R.: Speeding up isosurface extraction using interval trees. IEEE Trans. Visualization Comp. Graph. 3, 150–170 (1997)