Quantitative analysis of skeletonisation algorithms for modelling of branches

Will Gittoes
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
Email: willgittoes@gmail.com

Tom Botterill
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand

Richard Green
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand

***Abstract*—Skeletonisation is an important low-level problem in computer vision with many applications in shape finding, motion tracking, character recognition and segmentation. This paper examines how skeletonisation can be used to find and model the path of plant branches in an image. A proposed method for quantitatively comparing the accuracy of skeletons is described, which compares a skeleton produced by a skeletonisation algorithm to a ground truth. This method is used to evaluate several skeletonisation algorithms within the context of branch modelling. The best single skeletonisation method is found to be morphological thinning, due to the highly connected nature of the skeleton.**

Keywords;* *skeletonisation, skeletonization, branch modelling, quantitative analysis

I. INTRODUCTION

Skeletonisation is the process of recovering a model from image of objects with a network structure, such as images of handwriting or characters [1-4], medical images of veins and organs [5, 6], or images of branching plants. Skeletonisation algorithms convert binary images of the objects (Fig. 1) to ‘skeleton images’; networks of lines describing the shape and topology of the object's structure (Fig. 2) [7, 8]. Good skeletons have the properties that they accurately represent the original image [8-10] and are easy to convert into more meaningful continuous models [8, 10]. Issues with contemporary skeletonisation algorithms include: not being centred within the shape described, being overly sensitive to small changes in the original image (both examples of poor localisation), extraneous branches (called “spurs”), not being thin and not being correctly joined up (errors in topology).

Figure 1. A black and white segmented image of a branching plant

Skeletonisation is a common operation in computer vision [8, 11] because it is often a requirement to find a representational model of an image on which program logic may operate; such as describing routes across a map [12-14] or representing and tracking body pose [15-17]. Each application has its own specific requirements of a skeletonisation algorithm and its own definition of robustness based on the relative importance of each of the issues described above. For instance handwritten character recognition emphasises unambiguous junctions where letter strokes cross and lack of extraneous skeleton artefacts (called “spurs”) [1, 2].

This paper presents an evaluation of the effectiveness of skeletonisation algorithms in creating models of branching plants. Branch skeletonisation has several similarities to handwriting recognition: the shapes are “ribbon-like” (a description used in character recognition papers to describe pen strokes [1, 4]), and the shapes must eventually be converted into a semantic model (for writing the character must be recognised, for branches a semantic, topological model must be built). These requirements must be reflected in the evaluation of the skeletonisation algorithms.

Because the applications of skeletonisation have these very specific requirements, quantitative analysis of skeletonisation measures is important. The metric that is most often evaluated is computation time (such as in [1, 12-14, 18, 19]), particularly when the application demands real-time performance. Some papers have measured skeleton suitability by counting the number of spurs. Ward and Hamarneh measure spur count using entropy by calculating

Figure 2. A binary skeleton image of the branch network shown in Fig. 1

the information content of skeletons [20], while Bai and Latecki evaluate spur count visually [7]. Other papers have measured connectivity of skeletons and number of end points [21]. Suen, Lam and Wang have proposed methods for measuring skeleton localisation accuracy by comparing skeletons with reference skeletons [9, 22]. Despite the creation of these evaluation methods and the importance of numeric comparison of skeletonisation algorithms, most other authors, for example [1, 5, 7, 10-12, 14], do not provide quantitative evaluations of skeleton accuracy and instead qualitatively compare computed skeletons by visual inspection.

This paper will examine the qualities of a skeleton that are important for creating a successful plant branch model, and present a method to quantitatively evaluate skeletonisation algorithms that reflects these requirements. Several skeletonisation algorithms will be briefly surveyed, and then evaluated on their suitability for branch modelling based on the presented method. While there have been several other surveys of skeletonisation [8, 9, 11], none of these surveys take a quantitative approach, and none are specific to branch modelling.

This paper is organised as follows: Section 2 proposes a set of metrics to evaluate skeletonisation algorithms, and describes the process for measuring those metrics. Section 3 surveys the classes of skeletonisation algorithm that will be evaluated. The results of the quantitative analysis are presented in Section 4, followed by conclusions on which skeletonisation algorithms are most suitable.

II. METHOD FOR QUANTITATIVE ANALYSIS

For a quantitative evaluation to be of use, the qualities measured must reflect desirable features of a skeleton for a given application. Several skeleton qualities have been discussed in previous papers, including thinness, localisation quality (accuracy), connectedness and number of spurs. The metrics chosen evaluate skeletons, specifically for the purposes of branch modelling, are thinness, connectedness and localisation quality.

A. Thinness

One requirement of a skeleton is that it should be thin [8, 10]. Some applications demand the more explicit thinness requirement that the skeleton should be exactly 1 pixel thick. One example of this strict requirement is given in [10], which uses pixel adjacency information to build a topological graph. Skeletons thicker than 1 pixel will create graphs with erroneous loops. This is also an important requirement within the context of branch modelling. Because the branch model will need to be represented semantically, a method such as the one described in [10] will need to be used to generate a topological graph. This therefore extends the strict 1 pixel thin requirement to skeletons used to model branches.

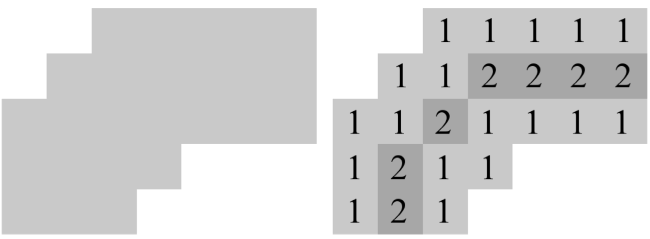

Figure 3. Left: A section of a skeleton, right: the skeleton pixels overlaid with the distance to the edge and the local maxima highlighted.

A simple method has been developed to calculate mean skeleton thickness. Firstly, for each pixel on the skeleton, the distance to the nearest non-skeleton pixel d_p is calculated. These distances are compiled into a distance map (Fig. 3). Next, the local distance maxima are selected. The mean thickness of the skeleton is calculated like so:

$$\overline{d_p} = \frac{1}{N}\sum_{i=0}^{N} 2d_{pi} - 1, \qquad (1)$$

where N is the number of local maxima distance values.

The mean thickness for the enlarged example line shown in Fig. 3 is 3.

B. Connectedness

Skeleton connectedness is another metric that is important when a semantic model needs to be built from a skeleton [10]. To be connected, a skeleton should maintain a topology consistent with original shape [8, 9]. Producing connected skeletons is an extremely important requirement when modelling branches, because a topologically accurate model ensures that consistent high-level decisions can be made using the model. To correctly estimate the branch structure, it is important that no topological information is lost during skeletonisation. There are methods of describing branch topology such as L-systems [23], however these descriptions apply only to semantic graphs. A method of topological comparison needs to be chosen that can work with discrete models composed only of pixels.

The Betti numbers are a series of quantities that can be calculated for topological graphs and spaces that encode information about how those spaces are connected [5, 24].

Skeleton connectedness can be measured by calculating the 0^{th} and 1^{st} Betti numbers of the skeleton, such as in [5, 24]. The 0^{th} number is simply the number of connected components, and the 1^{st} number measures the number of holes in the skeleton [5], and can be computed by:

$$B_1 = m - n + k \qquad (2)$$

Where m and n are the number of edges and nodes respectively in a topological graph and k is the number of connected components. Fig. 4 shows the procedure for calculating the number of connected components.

In [10] Reinders, Jacobson and Post describe how to turn a 1 pixel thin skeleton into a topological graph. The number of edges and nodes can then be found, allowing calculation of the 1^{st} Betti number.

C. Localisation quality

Skeletons should be accurate, or well localised; this is a measure of how well centred a skeleton is within the region

For each unvisited pixel on the skeleton:
1) Initialise a list containing that pixel's location
2) While the list is not empty:
    a. Remove a pixel from the list
    b. Mark the pixel as visited
    c. Add all the surrounding skeleton pixels to the list
3) Increment the number of connected components

Figure 4. Procedure for calculating the number of connected components in a 1 pixel thin skeleton image.

Figure 5. An example of a ground truth skeleton.

it describes [8]. Each skeletonisation algorithm inherently contains and implements its own definition of what centred means. Therefore, comparing the output of a skeletonisation algorithm using a mathematical or algorithmic definition for "centre" in analogous to calculating the similarity between two skeletonisation algorithms. Instead, skeletons should be compared against a hand-made ground truth skeleton. This allows the ground truth author to create a basis for comparison that encodes the features of a skeleton that are most useful for a specific application. Fig. 5 is an example of a ground truth skeleton, representing the branches shown in Fig. 1.

Suen, Lam and Wang compare generated skeletons against ground truth skeletons by calculating the mean distance between a skeleton pixel on one image and the closest skeleton pixel on another image [9, 22]. Their method produces a similarity metric in terms of pixel distance. To calculate skeleton accuracy for branch modelling a similar method has been developed, but similarity is instead expressed as a percentage rather than a distance.

This similarity score is calculated as follows:

$$S = 50\left(\frac{M_{c,t}}{N_c} + \frac{M_{t,c}}{N_t}\right) \quad (3)$$

$M_{c,t}$ is the number of skeleton pixels in the calculated skeleton that are within a certain threshold distance of a corresponding skeleton pixel in the truth skeleton; likewise $M_{t,c}$ is the number of skeleton pixels in the truth skeleton that are within the threshold distance of a pixel in the calculated skeleton. N_c and N_t are the total number of skeleton pixels in the calculated and truth skeletons respectively.

III. SKELETONIZATION ALGORITHMS

There are two main categories of skeletonisation algorithm; discrete and continuous [8, 25].

Discrete skeletonisation algorithms are the most common class. They reduce a region into a minimal skeleton defined by a set of pixels (or voxels if done in 3D [8, 10]). Continuous skeletonisation algorithms create a continuous representation of the original image, for example with a function or graph [8, 26]. Continuous skeletonisation algorithms usually require initialisation, e.g. from manual initialisation or from a discrete skeleton [26, 27], so this paper will only evaluate discrete skeletonisation methods.

Discrete skeletonisation algorithms are subject to two main types of errors; firstly extra/combined branches, and secondly spurs [1]. Extra branches are common at crossing points or sharp angles [11] and form when two ridges combine into one, and spurs are caused by edge noise [7]. Many papers present a spur elimination algorithm along with a skeletonisation method, or present a skeletonisation method designed to reduce these errors.

Discrete skeletonisation algorithms can produce skeletons either iteratively, or non-iteratively [9]. Iterative methods require multiple passes over the image.

A. Ridge Finding

A discrete skeleton can be considered the set of all pixels that are locally in the centre of a shape [28]. To compute the skeleton, each pixel must be tested to see if it is a ridge pixel, and either accepted or rejected. Ridge finding is therefore a non-iterative skeletonisation algorithm.

If we consider ridge points as local maxima, then the assumption is made that pixel intensity is correlated with being centred in the shape. This assumption may not be true for every image; consider a binary image of a shape, where the cross section would look rectangular. This assumption can be enforced by applying a Gaussian blur to the image, essentially examining the image in a lower scale space [29].

If we consider a 2D cross section of a branch or line shape that is perpendicular to the direction of the line, in terms of position and intensity, then the peak intensity on the cross section is a point on the ridge. This maximum can be found by examining the directional derivative of the intensity function [28].

Ridge detection in this way is sensitive to two parameters: the scale of the image (equivalent to the level of Gaussian blur applied) and the scale at which the derivative is taken. If the discrete derivative is found using finite differences, then the second parameter is the distance offset value. The values of these parameters that are evaluated in this paper were chosen by comparing every combination of the parameters using the same localisation quality and connectedness metrics described in Section 2.

B. Medial Axis Transformation

A medial point is a point in the exact centre of a shape. Medial axis transformations (MAT) find medial points by finding the set of points that are local maxima in terms of distance from the edge of the shape [20]. This can be done in two ways, by fitting circles and selecting the centre points, or by creating a distance map and finding local maxima [20].

As described above, when considering skeletons as ridges in an image, the assumption is made that intensity is correlated with a central position in the shape. This correlation can be made explicit by creating a distance map and using this to find the medial axis. Fig. 3 gives an example of a distance map.

Once a distance map has been calculated, the set of pixels with an intensity that is maximal compared to their neighbours constitute the skeleton.

Medial axis transformations are a non-iterative skeletonisation method, since the algorithm always requires two passes: firstly computation of the distance map, and secondly, selection of local maxima.

Many different implementations and extensions have been proposed for the medial axis transform algorithm such as using discrete contour partitioning to prune spurs [7], calculating global significance values for each branch [20] or joining branches using an Euclidian distance-based skeleton strength map [30]. This paper is focused on evaluating the different classes of skeletonisation algorithm rather than comparing specific implementations. Therefore, only a simple MAT algorithm that finds local maxima of the distance map will be evaluated here.

C. *Morphological Thinning*

Morphological thinning takes a region, and gradually reduces the boundaries of that region until they are only one pixel apart [11]. The results are similar to the medial axis transformation, because pixels are effectively classified by distance from the edge of the shape. However, instead of trying to explicitly locate individual medial points, non-medial pixels are pruned. This means that connectivity is implicitly guaranteed because no pixel that is the only connecting pixel between two sections is removed [10].

This connectivity guarantee is an important feature for branch model fitting, but thinning algorithms can be slow [11, 19] and are not well-suited to parallelization. Never-the-less, thinning has been described as easier to parallelize than medial axis transformations [19].

The thinning algorithm that will be evaluated in this paper is described in [11], and uses the concept of a neighbourhood matrix to encode adjacency information for each pixel. Pixels are iteratively pruned according to their neighbourhood value.

D. *Ridge Finding Using Steerable Filters*

Steerable filters are a class of image filter that allow extrapolation of filter responses at arbitrary rotations by taking a combination of the output from a few basis filters [31-34]. The response from as few as three basis filters can be interpolated to describe every possible rotation [31, 34].

Steerable filters can be used for both edge detection (using antisymmetric filters created by odd Gaussian derivatives) and ridge detection (using symmetric filters created by even derivatives) [32]; the latter can therefore be used for skeletonisation. The filter does not output a binary skeleton and therefore a secondary non-maximum suppression step must be executed to select ridge pixels; this method is outlined in [34]. An advantage of using ridge-detecting steerable filters for branch model fitting is that the orientation of the branches would be given by the angle of the filter response, and that the scale of the filter kernel detects only branches with the corresponding thickness. This can be used to filter out small shoots or thick trunks.

IV. RESULTS

Each of the skeletonisation algorithms described in Section 3 were measured using the metrics for localisation quality, connectedness and thinness described in Section 2.

A. *Localisation quality*

Fig. 7 shows the localisation quality of the skeletonisation algorithms. The most accurate skeletons are produced by the medial axis transform and by thinning, with ridge finding producing the lowest quality skeletons. Fig. 6 shows an enlarged view of a section of the branch image, along with the skeleton calculated by each method.

The ridge finding algorithm has found undesirable extra

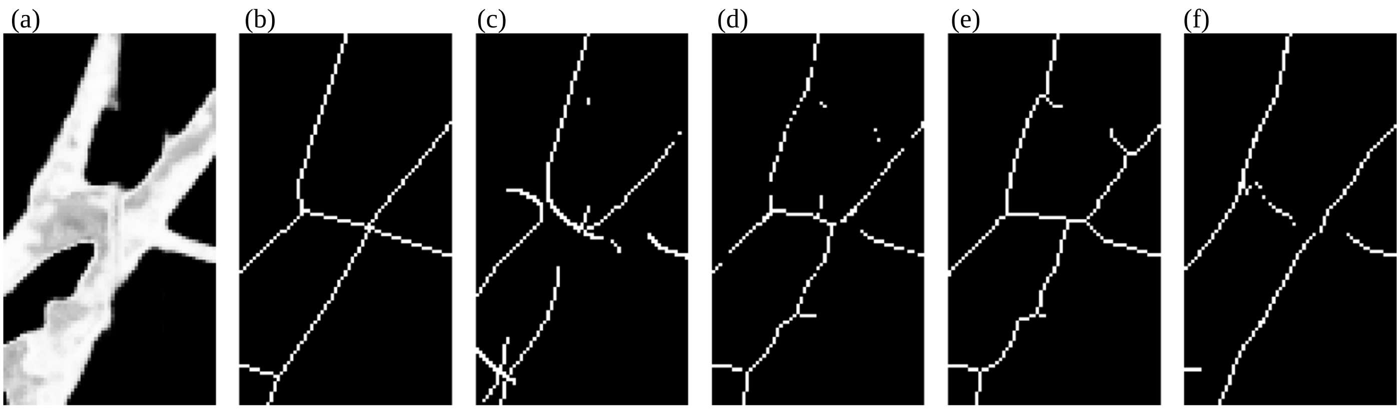

Figure 6. An enlarged view showing the results of the various skeletonisation methods on (a) the original image of branches compared to (b) the perfect skeleton. The algorithms are produced using: (c) ridge finding, (d) medial axis transformation, (e) thinning and (f) steerable filters.

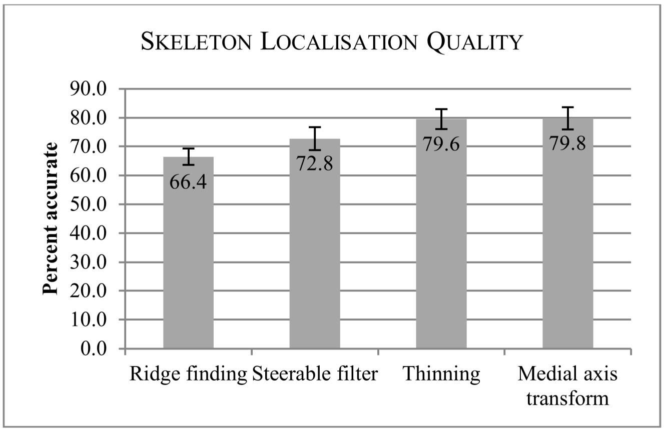

Figure 7. Accuracy of branch image skeletons produced by the surveyed skeletonisation algorithms, for error < 0.05

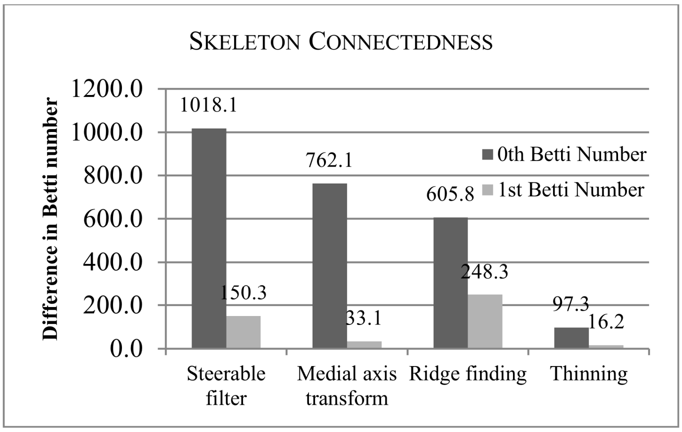

Figure 8. Chart of the average difference in Betti number between the computed skeleton and the truth skeleton. Lower is better.

ridges that negatively affect the quality of the skeleton and the skeleton localisation is also warped where the branches cross each other. The skeleton produced by the steerable filter is localised well, but there are some large gaps in the skeleton that account for its relatively low quality score.

B. Connectedness

Fig. 8 compares the connectedness of the skeletons produced by each algorithm. The value shown is the average difference in Betti number between the calculated skeleton and the truth skeleton, so a lower value is more desirable, with a value of 0 indicating complete topological accordance.

Thinning is by far the best skeletonisation algorithm for producing connected skeletons. The reason for this is that thinning guarantees connectivity.

The ridge finding algorithm produces skeletons that have breaks at the branch crossing points. The reason for this is that the direction of the ridge is not defined at the crossing points, as crossing points represent peaks, not ridges.

The medial axis transformation appears to produce skeletons that are disconnected along edges. This can be caused by two pixels that are adjacent in the direction perpendicular to the axis having an identical distance map value.

The steerable filter algorithm only produces well-connected skeletons of branches that closely match the size of the filter. If the branch is thicker or thinner than the diameter of the filter, then the filter response will be very low, and will therefore produce a broken response when discretised to produce a skeleton.

C. Thinness

The average thinness of the skeletons produced by each algorithm is shown in Table 1. All the algorithms produce satisfactorily thin skeletons. The medial axis transform and thinning methods both utilise each pixel’s distance from the shape edge, and therefore always produce skeletons exactly 1 pixel thin.

V. CONCLUSIONS

A proposed group of metrics have been described that measure the quality of skeletonisation algorithms for producing models of plant branches. The algorithms were evaluated based on the localisation quality, connectedness and thinness of the skeletons they produce. Four discrete skeletonisation algorithms were evaluated using these measures: ridge finding, medial axis transformation, morphological thinning and steerable filters. The only appropriate discrete skeletonisation algorithm for applications requiring connectedness is morphological thinning. Plant branch modelling has a strong requirement for connected skeletons, and therefore thinning is most suitable algorithm; both because of its connectedness and its high localisation accuracy. Skeletons produced by morphological thinning localised well, are the most connected and were found to be satisfactorily thin. The results show that the proposed method is a useful metric for quantitatively comparing the accuracy of thinness, connectedness and localisation of skeletons produced by skeletonisation algorithms.

TABLE I. SKELETON THINNESS

Ridge finding	**Medial axis transform**	**Thinning**	**Steerable filters**
1.01	1.00	1.00	1.01

VI. FUTURE WORK

In future research we will evaluate combinations of skeletonisation algorithms. For example, a branch image may be adaptively filtered using steerable filters before being skeletonised by morphological thinning, or a distance map may be calculated and then a ridge finding operation could be run to find the skeleton rather than finding local maxima.

REFERENCES

[1] J. J. Zou and H. Yan, "Skeletonization of ribbon-like shapes based on regularity and singularity analyses," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 31, pp. 401-407, 2001.

[2] S. A. Mahmoud, *et al.*, "Skeletonization of arabic characters using clustering based skeletonization algorithm (CBSA)," *Pattern Recognition,* vol. 24, pp. 453-464, 1991.

[3] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 31, pp. 216-233, 2001.

[4] M. Z. W. Liao, *et al.*, "Ribbon-like skeletonization based on contour reconstrction on intersection regions," presented at the Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, San Antonio, TX, 2009.

[5] M. N. O. K. G. Szekely, *et al.*, "Characterization and recognition of 3d organ shape in medical image analysis using skeletonization," 1996, p. 0139.

[6] C. Fouard, *et al.*, "Skeletonization by blocks for large 3D datasets: application to brain microcirculation," 2004, pp. 89-92 Vol. 1.

[7] X. Bai, *et al.*, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Transactions on pattern analysis and machine intelligence,* pp. 449-462, 2007.

[8] J. K. Lakshmi and M. Punithavalli, "A Survey on skeletons in digital image processing," 2009, pp. 260-269.

[9] L. Lam, *et al.*, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence,* pp. 869-885, 1992.

[10] F. Reinders, *et al.*, "Skeleton graph generation for feature shape description," 2000, pp. 73–82.

[11] K. Saeed, *et al.*, "K3M: A universal algorithm for image skeletonization and a review of thinning techniques," *International Journal of Applied Mathematics and Computer Science,* vol. 20, pp. 317-335, 2010.

[12] B. Y. Ko, *et al.*, "Real-time building of a thinning-based topological map with metric features," 2004, pp. 1524-1529.

[13] C. H. Choi, *et al.*, "Topological map building based on thinning and its application to localization," 2002, pp. 552-557 vol. 1.

[14] T. B. Kwon and J. B. Song, "Real-time building of a thinning-based topological map," *Intelligent Service Robotics,* vol. 1, pp. 211-220, 2008.

[15] C. Menier, *et al.*, "3d skeleton-based body pose recovery," 2006.

[16] L. Herda, *et al.*, "Using skeleton-based tracking to increase the reliability of optical motion capture," *Human movement science,* vol. 20, pp. 313-341, 2001.

[17] J. Gall, *et al.*, "Motion capture using joint skeleton tracking and surface estimation," 2009, pp. 1746-1753.

[18] H. Fujiyoshi and A. J. Lipton, "Real-time human motion analysis by image skeletonization," 1998, pp. 15-21.

[19] K. H. Kim, *et al.*, "Real-time skeletonization using FPGA," pp. 1182-1186.

[20] A. D. Ward and G. Hamarneh, "The groupwise medial axis transform for fuzzy skeletonization and pruning," *IEEE Transactions on pattern analysis and machine intelligence,* pp. 1084-1096, 2009.

[21] L. Lam and C. Y. Suen, "An evaluation of parallel thinning algorithms for character recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 17, pp. 914-919, 1995.

[22] L. Lam and C. Y. Suen, "Automatic evaluation of skeleton shapes," 1992, pp. 342-345.

[23] P. Prusinkiewicz, *et al.*, "Development models of herbaceous plants for computer imagery purposes," 1988, pp. 141-150.

[24] P. K. Saha, *et al.*, "Three-dimensional digital topological characterization of cancellous bone architecture," *International Journal of Imaging Systems and Technology,* vol. 11, pp. 81-90, 2000.

[25] D. Attali, *et al.*, "Pruning discrete and semicontinuous skeletons," 1995, pp. 488-493.

[26] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image understanding,* vol. 53, pp. 211-218, 1991.

[27] J. De Vylder and W. Philips, "A computational efficient external energy for active contour segmentation using edge propagation," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 661-664.

[28] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," 1996, p. 465.

[29] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales,"" *Journal ofapplied statistics,* vol. 21, pp. 225-270, 1994.

[30] L. J. Latecki, *et al.*, "Skeletonization using SSM of the distance transform," 2007, pp. V-349-V-352.

[31] W. T. Freeman, *et al.*, "The design and use of steerable filters," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 13, pp. 891-906, 1991.

[32] C. Burlacu and G. Burlacu, "Color image segmentation using steerable filters," pp. 347-350 Vol. 1.

[33] K. R. Castleman, *et al.*, "Simplified design of steerable pyramid filters," 1998, pp. 329-332 vol. 5.

[34] M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criteria," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 1007-1019, 2004.