A Methodology for the Characterization of the Performance of Thinning Algorithms

M. Y. Jaisimha, Robert M. Haralick

Dept. of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

Dov Dori*

Dept. of Computer Science
Technion, Israel Institute of Technology
Haifa 3200, Israel

Abstract

We measure the performance of thinning algorithms in the ideal world of noise-free Blum ribbons. Differences in the performance of thinning algorithms emerge even when they are applied on noise-free ribbon images. We define an error criterion function based on the Hausdorff distance that measures the deviation between the ideal and actual output of the thinning algorithms. We illustrate the process of performance evaluation by the application of ten state of the art thinning algorithms to the same (large) set of input images. We present results that show the mean value of the normalized error for each algorithm over each population of images. We also study how performance of an algorithm varies over different populations of images.

Keywords: Performance Evaluation, Thinning Algorithms, Objective evaluation.

1 Introduction

This paper presents a quantitative method for characterizing the performance of thinning algorithms. A large number of papers proposing thinning algorithms have been published in the literature. A comprehensive survey is given in [10]. A few papers comparing the performance of various thinning algorithms have also been published. Unfortunately, many of these papers provide no quantitative means of measuring the performance of the thinning algorithm. Quite often, the only measure of skeleton quality is a visual image of the output of the thinning algorithm. This is unfortunately true even when the authors wish to compare the performance of their thinning algorithm with that of other state-of-the-art thinning algorithms. Where an effort to provide a quantitative performance measure is made, the “gold standard” by which the algorithm is measured is quite often a subjective one provided by a “panel of experts.” [11].

Since thinning forms an important component in many OCR systems, it is important to accurately estimate the performance of thinning algorithms. When designing an OCR system, it is important to have objective gold standards with respect to which the skeleton quality is measured. This paper presents such an objective gold standard based on Blum ribbons.

We discuss briefly issues specific to the performance characterization of thinning algorithms in Section 2. We present the ideal world model for thinning based on Blum ribbons (which are generated when a disk is swept along a 2-D curve) in Section 2.1. The error criterion function we use that is based on the Hausdorff distance is described in Section 2.2. The performance characterization experiments and the results obtained are described in Section 4 and Section 5 respectively.

We present conclusions and further research directions in Section 6.

2 Performance Characterization of Thinning

Haralick [9] gives a detailed discussion of the philosophy of Performance Characterization and its application to the characterization of thinning algorithms. In this study we consider only noise-free images since we expect differences in thinning algorithm performance to appear even in this case since skeletons capture global topological information whereas many thinning algorithms are local in nature. This is referred to as Performance Characterization in the ideal. The steps in such a performance characterization of an algorithm are

- Choose the world of input imagery - Section 2.1.
- Devise an error criterion function that measures the deviation from the ideal of the output of the algorithm - Section 2.2.
• Define the population and a generation mechanism for the world of input imagery - Section 3.

• Carry out experiments studying the variation of the error criterion with respect to the image population and algorithm parameters for each algorithm according to a carefully designed experimental protocol - Section 4.

2.1 Ideal World Model for Thinning - Blum Ribbons

Considering the difficulty of assigning ground truth skeletons to real world images (the "expert" approach), we use synthetic imagery for performance evaluation of thinning algorithms. The ideal world of shapes to be thinned is the world of ribbons which are obtained when a shape referred to as the "generator" is swept along a two-dimensional curve referred as the "spine". An ideal ribbon is constructed from a simple (one that does not self-intersect) bounded curvature arc which is the spine, specifying the medial axis of the ribbon and a cross-section function giving the ribbon's width at each point of the spine. The cross-section function has bounded first derivatives to keep the width from changing too fast. The width itself is also bounded from both sides, to prevent too narrow or too wide ribbons. Additionally, there is a relation between the maximum allowed curvature (minimum local radius) of the arc and the maximum allowed width of the ribbon, in order to prevent a case in which the combination of sharp curvature and large width at that point cause the ribbon to overlap itself. An ideal digital image is constructed from a scene of ribbons by tessellating the scene of ribbons into pixels.

In Blum's medial axis representation, called "Blum ribbon," [6] the generator is a disk with its center on the spine. The Blum ribbon is illustrated in Figure 4. Rosenfeld [12] discusses various ribbon descriptions from the standpoint of both generation and recovery. He notes that Blum ribbons are both constructible and uniquely recoverable. These two features make this type of ribbon a natural choice for our purpose. In addition, Blum ribbons agree quite well with both the "prairie fire" analogy for a skeleton and the concept of a medial axis.

To avoid problems of instability when the slope of the spine approaches 90°, and to enable self intersection of the spine, we adopt the following parametric description for the spine S:

\[ S(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} \]

The scalar function that describes the radius of the generating disc sweeping along S is called the contour function and denoted by C(s).

The region contained by the Blum ribbon is defined

\[ R(s, \theta) = \left\{ \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} \middle| \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = S(s) + \lambda (\cos \theta t(s) + \sin \theta n(s)) \right\} \]

where \( 0 \leq \theta \leq 2\pi; 0 \leq s \leq 1; 0 \leq \lambda \leq C(s), \) and

\[ t(s) = \frac{\hat{n}(s)}{||\hat{n}(s)||} \]

is a unit velocity vector tangent to the spine S, and \( n(s) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} t(s) \) is a unit velocity vector normal to the spine. The boundary of the ribbon is given by its envelope, which is tangent to the curves \( R(s, \theta) \) for every s.

To avoid self-intersection of the ribbon we have to impose constraints that relate value of the radius function to the radius of curvature of the spine in order to generate ribbons suitable for the characterization of thinning algorithm performance. Specifically, the constraint takes the form

\[ \rho_{max} < \frac{(x^2 + y^2)^{1/2}}{y - y_1} \]

where \( \rho_{max} \) is the maximum value of the radius function and the terms \( x, y, \) and \( \hat{x}, \hat{y} \) are the first and second order derivatives of the x and y components of the spine. These components are expressed in parameteric form parameterized by arc length. Then, \( \rho_{max} \) is the minimum value of the radius of curvature of the spine. Once we assume a form for the functions \( x(s), y(s) \) and \( C(s) \) (the contour function, which, in our case, is the radius function), the problem of generating a ribbon becomes one of choosing the parameters of the form such that the constraints mentioned above are satisfied. For the purposes of this study we assume that the spine polynomial components are of the form

\[ x(s) = \sum_{i=0}^{n} a_i s^i \quad \text{and} \quad y(s) = \sum_{i=0}^{n} b_i s^i \]

Thus, the highest power of s in either the x or y component is 3. Furthermore, we assume that the radius function takes the form

\[ C(s) = \sum_{i=0}^{n} c_i s^i \]

Generating a ribbon thus reduces to the problem of choosing the coefficients \( a_i, b_i \) and \( c_i \) to satisfy the constraints on the curvature of the spine relative to the value of the radius function. Two images are generated. One contains the discretized version of the spine and the other contains a discretized version of the ribbon. The ribbon is generated by stepping along the spine and rendering digital disks of a radius specified by the radius function.

2.2 Error Criterion Function

In this discussion we assume that the purpose of thinning to be the identification of the pixels through which the spine passes. With this purpose in mind, the goal of any thinning algorithm is to erode all but the spine pixel. The measure of the algorithm's quality then inversely depends on the deviation of the resulting skeleton from the source spine. We use as the error criterion, a metric based on the Hausdorff distance [8] which provides a measure of the farthest distance between sets. It has an advantage over Euclidean error measures that compute pointwise distances in that the ideal and actual output can have
3 Computational Procedure for Generating Ribbon Images

Given the choice of the form of the spine and radius functions in Section 2.1 the computational procedure for generating the images is as follows. The user supplies as input to the procedure the maximum degree of the spine polynomials for the $x$ and $y$ components, the maximum degree of the radius polynomial, the number of rows and columns in the image and the aspect value (this is the ratio of the length of the spine to the maximum value that the radius function can take). The procedure for generating random ribbons has to pick coefficients for the radius and spine polynomials in such a way as to satisfy the given constraints. The procedure for picking coefficients is as follows:

1. Pick $a_i \sim U(-1, 1)$ and $b_i \sim U(-1, 1)$ where $U(-1, 1)$ is uniformly distributed on the interval $[-1, 1]$.

2. Compute the minimum radius of curvature for the coefficients $a_i$ and $b_i$ for all values $t_i$ in the interval $(0, 1)$ where $t_j = j \times \delta, j = 1, \ldots, (1/\delta) - 1$ and $\delta$ is the discretization increment in the value of the parameter $t$. Given the forms we have assumed for $x$ and $y$ we have $\hat{z}(t_j) = a_1 + (2 \times a_2 \times t_j) + (3 \times a_3 \times t_j^2)$ and $\hat{y}(t_j) = 2 \times a_2 + (6 \times a_3 \times t_j)$.

The values of $\hat{x}$ and $\hat{y}$ are given in a similar fashion. The minimum value of the radius of curvature $r_{\text{min}}$ is given by

$$ r_{\text{min}} = \min_j \left( \frac{(\hat{z}(t_j))^2 + (\hat{y}(t_j))^2}{\hat{z}'(t_j)\hat{y}'(t_j)} \right)^{3/2} $$

3. Pick $c_i \sim U(0, 1, 1)$ for $i > 0$ Scale $c_i$ to satisfy the Blum ribbon generation condition given by $|C'(s)| < 1$

4 Experiments for Performance Characterization under Noise-free conditions

4.1 Population of Input Images

We have to define populations of images based on the parameters used as input to the image generation process. We vary only the degree of $x$ and $y$ components of the spine polynomial and the degree of the radius function polynomial. We hold the image size fixed at $128 \times 128$ and the aspect value fixed at $5.0$. Letting $d_x$ be the degree of the $x$ component of the spine polynomial, $d_y$ be the degree of the $y$ component of the spine polynomial and $d_r$ be the degree of the radius function we define six image populations by taking $d_x = 1$, $d_y = 0, 1$ or 2 and $d_r = 0, 1$ or 2. In Figure 1 through Figure 3 we show example images of some of these images with the spines overlaid on the ribbons.

4.2 Measurements

We make measurements of the error criterion function value for different image population and algorithm combination. If we use a large enough number of images as input to the thinning algorithms we can then compute statistics of the error criterion value which reflect how the algorithm behaves on average over the input population. For each algorithm and for each population of input images we generate 300 images which are provided as input and measure the resulting error criterion function value. For square images, this maximum value the error criterion function can take is equal to half the length in pixels of each side of the square. In order to make the error measure independent of the image size we normalize the error criterion values with respect to this maximum value so it takes values in the interval $(0, 1)$.

4.3 Thinning Algorithms

5 Results and Discussion

The following table presents the tabulated values of the average error and the standard deviation for each algorithm for each of the image populations when the same 300 images (for each image population) are input to the algorithms. For each algorithm the mean and the standard deviation of the relative error are given (one below the other in that order).

<table>
<thead>
<tr>
<th>$d_y = 1$</th>
<th>$d_y = 2$</th>
<th>$d_y = 1$</th>
<th>$d_y = 1$</th>
<th>$d_y = 2$</th>
<th>$d_y = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_x = 0$</td>
<td>$d_x = 0$</td>
<td>$d_x = 0$</td>
<td>$d_x = 0$</td>
<td>$d_x = 0$</td>
<td>$d_x = 0$</td>
</tr>
<tr>
<td>Alg. 1</td>
<td>.0427</td>
<td>.2577</td>
<td>.1606</td>
<td>.0564</td>
<td>.0580</td>
</tr>
<tr>
<td></td>
<td>.0571</td>
<td>.4000</td>
<td>.0757</td>
<td>.0779</td>
<td>.0572</td>
</tr>
<tr>
<td>Alg. 2</td>
<td>.0744</td>
<td>.1691</td>
<td>.0312</td>
<td>.1005</td>
<td>.3263</td>
</tr>
<tr>
<td></td>
<td>.0473</td>
<td>.1651</td>
<td>.0979</td>
<td>.1141</td>
<td>.4342</td>
</tr>
<tr>
<td>Alg. 3</td>
<td>.0472</td>
<td>.0825</td>
<td>.1618</td>
<td>.0809</td>
<td>.1477</td>
</tr>
<tr>
<td></td>
<td>.1585</td>
<td>.0966</td>
<td>.1402</td>
<td>.1017</td>
<td>.1147</td>
</tr>
<tr>
<td>Alg. 4</td>
<td>.2429</td>
<td>.0417</td>
<td>.0322</td>
<td>.0812</td>
<td>.0372</td>
</tr>
<tr>
<td></td>
<td>.2058</td>
<td>.0266</td>
<td>.0227</td>
<td>.0580</td>
<td>.0252</td>
</tr>
<tr>
<td>Alg. 5</td>
<td>.0580</td>
<td>.0397</td>
<td>.0931</td>
<td>.0561</td>
<td>.0491</td>
</tr>
<tr>
<td></td>
<td>.0201</td>
<td>.0260</td>
<td>.0551</td>
<td>.0526</td>
<td>.0289</td>
</tr>
<tr>
<td>Alg. 6</td>
<td>.0553</td>
<td>.0369</td>
<td>.0275</td>
<td>.0326</td>
<td>.0365</td>
</tr>
<tr>
<td></td>
<td>.0322</td>
<td>.0254</td>
<td>.0121</td>
<td>.0197</td>
<td>.0246</td>
</tr>
<tr>
<td>Alg. 7</td>
<td>.0264</td>
<td>.0441</td>
<td>.0347</td>
<td>.0355</td>
<td>.0375</td>
</tr>
<tr>
<td></td>
<td>.0105</td>
<td>.0335</td>
<td>.0201</td>
<td>.0303</td>
<td>.0246</td>
</tr>
<tr>
<td>Alg. 8</td>
<td>.0309</td>
<td>.0401</td>
<td>.0398</td>
<td>.0327</td>
<td>.0446</td>
</tr>
<tr>
<td></td>
<td>.0183</td>
<td>.0298</td>
<td>.0433</td>
<td>.0334</td>
<td>.0317</td>
</tr>
<tr>
<td>Alg. 9</td>
<td>.0414</td>
<td>.0447</td>
<td>.0256</td>
<td>.0357</td>
<td>.0377</td>
</tr>
<tr>
<td></td>
<td>.0644</td>
<td>.0281</td>
<td>.0108</td>
<td>.0158</td>
<td>.0241</td>
</tr>
<tr>
<td>Alg. 10</td>
<td>.0231</td>
<td>.0510</td>
<td>.0452</td>
<td>.1122</td>
<td>.0506</td>
</tr>
<tr>
<td></td>
<td>.0088</td>
<td>.0280</td>
<td>.0176</td>
<td>.0825</td>
<td>.0221</td>
</tr>
</tbody>
</table>

From this table we can see that for a given population of images, differences in algorithm performance emerge. For example in ribbons with $d_y = 1$ and $d_x = 0$, Algorithm 4 does substantially worse than all the other algorithms since the mean normalized error is the highest (0.242917). For ribbons with $d_y = 2$ and $d_x = 2$ on the other hand we find that Algorithm 1 has the poorest performance. We can also use the table to verify hypotheses about how various algorithms function for different types of images. For example Algorithm 6's performance does not deviate much with respect to the type of ribbon. Thus if one has prior information on the nature of shapes present in the input image (for example if one were provided with a drawing where all the lines were of constant width or linearly varying width) then we could use this table to decide on which thinning algorithm would be most appropriate - one possible answer on examining the table is Algorithm 7.

6 Conclusions and Future Work

In this paper we presented a quantitative performance evaluation methodology for thinning algorithms. We describe the mechanism whereby the world of ideal input images is generated. We also describe an error criterion function that describes how the output of the thinning algorithm deviates from the ideal. We present results for a trial run of 300 images each of six different image populations for ten distinct thinning algorithms. In work that is currently in progress we are in the process of designing ANOVA data analysis procedures that examine the output and enable the testing of statistical hypotheses about the functioning of algorithms.

References


