Hough Transform-based Technique for Automated Carbon Nanocone Segmentation

Andries H. Smith[†] Jong Kwan Lee[†] Hao Hu[†] Eric S. Mandell[‡]
[†]Department of Computer Science [‡]Department of Physics & Astronomy
Bowling Green State University
Bowling Green, OH 43403, U.S.A.
{smithah, leej, haohu, meric}@bgsu.edu

ABSTRACT

A new technique to automatically segment the L-shaped carbon nanocone structures from Transmission Electronic Microscopy (TEM) images is described. The technique enables robust segmentation of the structures by exploiting a simplified Generalized Hough Transform (HT)-based processing. Exploitation of parallelism on commodity hardware is also explored for efficient processing. Effectiveness of the technique is evaluated through experiments on synthetic, simulated, and real images of carbon nanocone structures.

Keywords
Hough Transform, Carbon Nanocone Segmentation, Image Processing

1 INTRODUCTION

Automated feature segmentation has been utilized in many research and application domains (e.g., [Pie03, WCH04, YLL05, CNG09, Saf12, KlB12]) as it plays a key role in localizing objects of interest efficiently and effectively. Often, localization of objects leads to discovery of information which can help understanding of the underlying characteristics of the objects. For example, independent component analysis-based segmentation [WCH04] has been applied in medical imaging to characterize blood supply patterns which are critical for the profound analysis of cerebral hemodynamics. An automated method for localizing auroral ovals in satellite images was also introduced [CNG09]. Such a method can help scientists study the Earth's magnetic field. A fast on-line video motion segmentation method [KlB12] has also been recently presented in a multimedia application. In this paper, we introduce a new, robust automated technique for segmentation of L-shaped carbon nanocone structures from Transmission Electronic Microscopy (TEM) images.

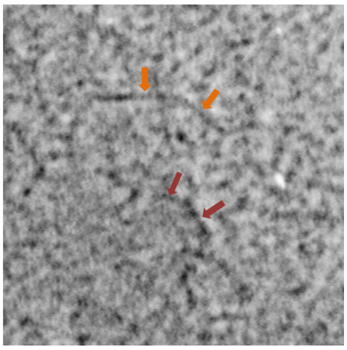


Figure 1: A sample of carbon nanocone TEM image

Carbon nanocone structures represent a class of novel materials that are of high interest to investigations of carbon's role in fossil fuel, hydrogen storage, and nanotechnologies. These structures appear as two linear structures joined at a common point in TEM images. Moreover, the linear structures form an angle of approximately 110° or 140° for the faceted nanocones and 113° or 150° for relaxed nanocones [Man08]. (Faceted and relaxed nanocones are types of carbon nano-structures that are studied by many scientists.) A robust localization of such carbon nanocone structures is a key precursor to effective use of the TEM images in many fields, such as biosensors or nanocomputing. A sample TEM image of the carbon nanocone structures is shown in Figure 1. In the figure, two sample carbon nanocone structures (i.e., dark linear features) are marked by two sets of colored arrows for readers. As shown in the figure, other non-nanocone features are also present, the carbon nanocone structures are oriented in various ways and have varying side lengths and varying contrasts along the structure, making automated segmentation of the structures very challenging.

The new automated segmentation of carbon nanocone structures is based on a variant of the Hough Transform (HT) [Hou62]. The new technique enables robust segmentation of the structures by utilizing a simplified binning process in the Generalized Hough Transform (GHT) [Bal81]. (GHT will be discussed in the next section.) We also exploited data parallelism on commodity hardware for efficient processing.

The paper is organized as follows. In Section 2, the related work is discussed. The new segmentation technique is introduced in Section 3. In Section 4, the experimental results and analysis are presented. Section 5 concludes this paper.

2 RELATED WORK

The related work, including the work our technique is based on, is discussed next.

2.1 Hough Transform and Its Variants

The Hough Transform (HT) [Hou62] is a well-known pattern detection method that enables recovery of global patterns in the image space via local pattern processing in the transformed parameter space. In a typical HT processing, the edge points (which present the object boundaries) in the image space are mapped to a *binned* parameter space in which dimensions of the space represent the pattern parameters. This parameter space is defined as the accumulator. After the binning process, the target pattern parameters are recovered by taking the parameters of the bins which contain the highest parameter vote counts. (While most of the HT-based methods considered the edge points in the image, direct application of HT processing on gray-scale images has also been reported [Sha96].)

The standard HT-based methods have been utilized successfully in many shape-based processing applications. However, the high memory and computational requirements of the standard HT make it difficult to apply for recovery of complex shapes. In addition, the standard HT aims to detect only analytic patterns; thus, it is difficult to apply it directly to arbitrary shapes (e.g., L-shaped carbon nanocone structures).

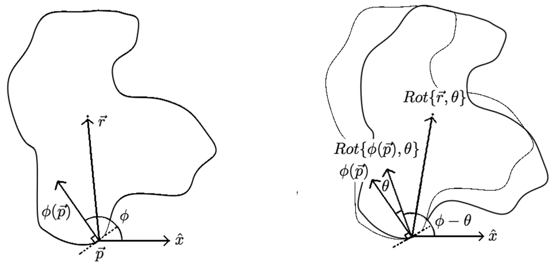
One way to reduce the HT's computational cost is to partition the high dimensional parameter space into lower-dimensional subspaces and then find the shape parameters in the series of lower dimensional subspaces (e.g., [HoC96]).

The Randomized Hough Transform (RHT) [XOK90] is a class of HT which can also reduce the standard HT's computational cost. While the standard HT considers all edge points, the RHT processes many sets of randomly selected edge points where each set maps to a single bin in the parameter space, allowing fewer computations during the binning process. An RHT-based method for ellipse detection was proposed by McLaughlin [McL98]. The method benefits from the RHT's random processing and the lower dimensional sub-parameter spaces. Cao et al. [CNG09] have extended McLaughlin's RHT to employ a linear least squares fitting for a more accurate and efficient binning process.

ϕ	R
0	$\vec{r}_{0,0}, \vec{r}_{0,1}, \vec{r}_{0,2}, \cdots$
$\Delta\phi$	$\vec{r}_{1,0}, \vec{r}_{1,1}, \vec{r}_{1,2}, \cdots$
$2\Delta\phi$	$\vec{r}_{2,0}, \vec{r}_{2,1}, \vec{r}_{2,2}, \cdots$
$\vdots$	$\vdots$

Table 1: R-table in GHT

The Generalized Hough Transform (GHT) [Bal81] is another class of HT that allows detection of both analytic and non-analytic shapes in images. Instead of binning the votes via an analytically-defined mapping scheme, the GHT utilizes a model of the object that consists of the boundary gradient direction and the distance from a reference point of the object to each point on the object boundary in a lookup table, called the *R-table*. (A reference point of the object can be any image point which can be used as an "anchor" for the object.) Specifically, the R-table defines a multi-valued mapping among the table indices corresponding to the gradient directions $\phi(\vec{p}_i)$, at the boundary points $\vec{p}_i$, and the vectors $\vec{r}$ from the boundary point to the reference point $\vec{c}$ of the object. The R-table is constructed by iterating over the object boundary points, calculating the set of $\vec{r}_i$ for each index $\phi(\vec{p}_i)$, where i indexes a set of boundary points with the same gradient direction and $\vec{r}_i$ is the vector from $\vec{p}_i$ to $\vec{c}$. A sample format of the R-table is shown in Table 1. As shown in the table, there are multiple values $\vec{r}_{i,j}$ for each index. For each edge pixel $\vec{p}_i$ with R-table index $\phi(\vec{p}_i)$, the potential reference points $\vec{c}_i$ of the object is determined as $\vec{c}_i = \vec{p}_i + \vec{r}_i$. Then the binning process is performed for the corresponding values of $\vec{c}_i$. The GHT allows rotation-



(a) (b)

Figure 2: Illustration of the Generalized Hough Transform (GHT): (a) $\vec{p}_i$, $\phi(\vec{p}_i)$, and $\vec{r}$, (b) rotation invariant for θ by re-indexing R-table

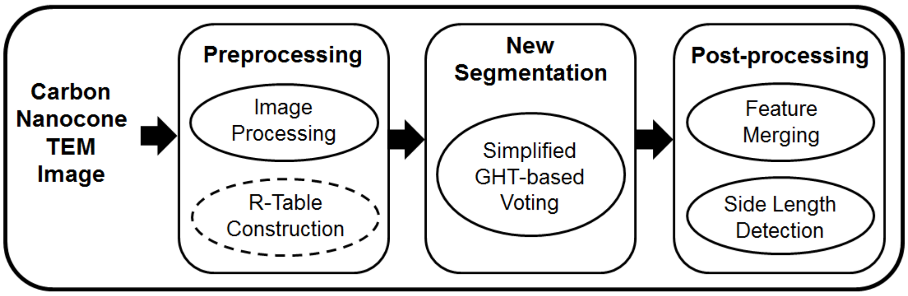


Figure 3: Overview of new GHT-based technique

invariant and scale-invariant object detection by modifying the R-table according to the rotation and scaling parameters; for scale invariant and rotation invariant detection, the R-table values are multiplied by the scaling vector or re-indexed, respectively. Figure 2 illustrates the key components of the GHT. In Figure 2 (a), the boundary point, $\vec{p}_i$, the gradient direction, $\phi(\vec{p}_i)$, and the vector $\vec{r}$ are shown. Figure 2 (b) illustrates that the vector $\vec{r}$ needs to be rotated by θ and ϕ needs to be replaced by $\phi - \theta$. This can be done by simple re-indexing of the R-table contents.

2.2 GPU-based Hough Transform

Computation using programmable graphics processing units (GPUs) has been utilized in traditional graphics as well as in an increasing number of more general-purpose applications, including simulation, data mining, analytics, databases, etc. GPU computation has also been employed in HT-based work. Here, we briefly discuss some of the key GPU-based HT work.

One early GPU-based HT was introduced by Strzodka et al. [SIM03]. In their method, programmable GPU shaders were utilized to speed up the binning process using the GPU textures as the lookup table for the shape parameter mapping. Ujaldon et al. [URG07] have presented a circle HT which uses a specific graphics pipeline (i.e., the rasterizer) for voting the circle candidates from a set of seeds computed by GPU vertex shaders. Other GPU-based HT methods include detection of more complex shapes (e.g., [ZhL05, LWN08, Deg10]). One efficient GPU HT for ellipse detection has been employed in real-time face detection [ZhL05]. Lee et al. [LWN08] presented a GPU-based HT for efficient ellipse detection in satellite imaging. Their HT uses a small set of CPU-generated random seeds in efficient GPU-based RHT processing for localization of the aurora oval.

2.3 Carbon Nanocone Segmentation

Although many methods to automatically-segment objects of interest in other types of imagery have been reported in the literature (e.g., [PRN04, LWN08]), the recently presented work by Ngatuni et al. [NLW12] is the only automated method for carbon nanocone segmentation. Their HT-based shape detection method introduced a new parameter space for the carbon nanocone structures. Their parameter space is defined by the pre-computed distances and the orientations of a set of points on the structure. However, while the method produced accurate detection results, the binning process was applied for all combinations of the edge point pairs; thus, the method was slow. In addition, since the method weighs the pairs of the edge points that have the same distance from the point joining the two sides of the structure, the method fails to detect the structures that have different side lengths. (In Section 4, we briefly compare the method with our new technique.) The new automated segmentation technique introduced in this paper aims to overcome these weaknesses. The new technique is based on a new type of Generalized Hough Transform (GHT) which simplifies the original GHT's binning process.

3 NEW SEGMENTATION TECHNIQUE FOR CARBON NANOCONE STRUCTURES

In this section, the new technique for the automatic segmentation of the carbon nanocone structures from TEM intensity images is detailed. The key component of the technique is a simplification of GHT that enables extraction of credible L-shaped carbon nanocone structures. Exploitation of parallelism on commodity hardware is also discussed.

3.1 Overview

Figure 3 presents an overview of the technique. The technique includes preprocessing steps that "clean" the image, determine the potential carbon nanocone pixels, and compute a lookup table, the simplified GHT-based bin voting step that exploits the characteristics of L-shaped object, and post-processing steps that merge segmented structures that are very close to each other and determine the structure side lengths. (In the figure, the dotted oval for the lookup table construction step indicates that it is computed only one-time.)

3.2 Preprocessing

In the new carbon nanocone structure segmentation, a set of image processing steps are applied (1) to highlight carbon nanocone structure, (2) to remove the non-nanocone structures, and (3) to generate the edge point image for HT's binning process.

First, median filtering is applied to the intensity-inversed image to remove high spike noise and smooth out the structures. We empirically determined that 7×7 median filtering could eliminate much of the high spike noise and fill the gaps within the structures. Second, an intensity cumulating scheme is applied to the median filtered image to highlight the carbon nanocone structures. This intensity cumulating is done by replacing each pixel intensity with the maximum of the pixel intensity sums in narrowed rectangular-shaped (e.g., 15×5) image regions centered at the pixel. Since the carbon nanocone structures are linear features, this intensity cumulating step highlights (i.e., increases the contrast of) the structures and fills in the gaps within the same structures.

Third, a modified version of the Strous linear feature pixel labeling algorithm [Str00] is applied to the image "cleaned" in the first two steps. The Strous algorithm allows determination of local *bright* pixels by considering 3×3 local image region. We have modified the algorithm to consider 5×5 local image region with a less restricted constraint to label a pixel as a potential point on the carbon nanocone structures. This modification allows more points on the structures to be labeled as the potential nanocone pixels. However, this modification results in "thick" potential carbon nanocone structures. To select the points along the centerline of the structures, the morph-thinning [Dou92] is applied to the result of the modified Strous algorithm. (Morph-thinning is one of the widely-used thinning operators which employs a morphological erosion-based hit-or-miss operator to the original image with a pair of structuring elements, and then subtracts the result from the original image.)

Lastly, global and adaptive thresholdings are applied to the "cleaned" image to remove non-nanocone features. Specifically, the binary result of thresholdings is used as a mask on the morph-thinned image. For the global thresholding, its threshold is the median intensity T of the filtered image; all pixels whose intensities are less than T are considered to be non-nanocone pixels. For the adaptive thresholding [GoW02], the filtered image is sub-divided with overlapping and thresholding is applied in each sub-region. We have found empirically that sub-dividing the image into 64×64 pixel tiles with 50% overlap (e.g., the top-half of a sub-region overlaps the bottom-half of the regions that is above sub-region) produces the best results for the TEM images.

3.3 New HT for carbon nanocone segmentation

The carbon nanocone segmentation technique presented here exploits a simplified lookup table-based GHT binning scheme to recover the L-shape structure parameters.

In the technique, the coordinates of the reference point and the orientation of the L-shape are used as the shape parameters for the carbon nanocone structure. Figure 4 shows a L-shaped structure and its parameters. The reference point O is defined as the point joining the two structure edges and the orientation γ is defined as the angle between the x-axis and the vector halving the two edges. (In the figure, the side edges of the structure are denoted as the counter-clockwise and clockwise edges.)

The new segmentation technique utilizes a simplified lookup table that is designed to recover the L-shape parameters through a GHT binning process. We will call this lookup table the *L-table* in this paper. Unlike the standard GHT's lookup table, the L-table does not include many entries for different gradient directions since the gradient directions are the same for all the points on the same edge of the structure; thus, the L-table includes entries for only two gradient directions, one for each edge of the linear structure. In addition, the L-table is indexed by the distance between the reference point to the points on the on the edges. This indexing strategy yields a very simple scalar-valued lookup table mapping from the distance to the orientation. In addition, unlike the standard GHT, the new HT binning pro-

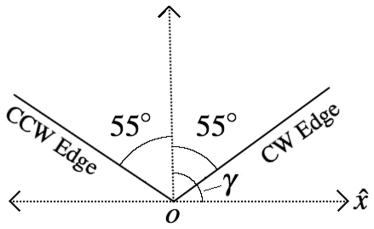


Figure 4: Illustration of a L-shape structure (representing a 110° faceted nanocone): O and γ represent the reference point and the structure orientation, respectively.

CCW edge		CW edge	
Distance	Orientation	Distance	Orientation
1	55°	1	−55°
2	55°	2	−55°
3	55°	3	−55°
⋮	⋮	⋮	⋮

Table 2: A diagrammatic representation of the L-table for 110° faceted nanocones

cess does not include a scaling of values in the L-table. Instead, a simple post-processing step to determine the edge lengths of the structure is included to reduce the computational cost. (This post-processing step is discussed in the next subsection.)

In our HT binning process, each bin of the accumulator contains two vote values; one vote counts for each edge. This two-value bin enables segmentation of L-shapes with imbalanced side lengths—even if the vote counts from one short edge is low, the vote counts from the other longer edge is high; thus, the structure parameters can be recovered.

3.4 Post-processing

It is possible for the new HT binning process to produce structures that are very close to each other. In addition, the binning process determines only the coordinates of the reference point and the orientation of the structure (but not the edge lengths). To merge structures that are close to each other into one structure and to determine the edge lengths, we post-process the output of the binning process.

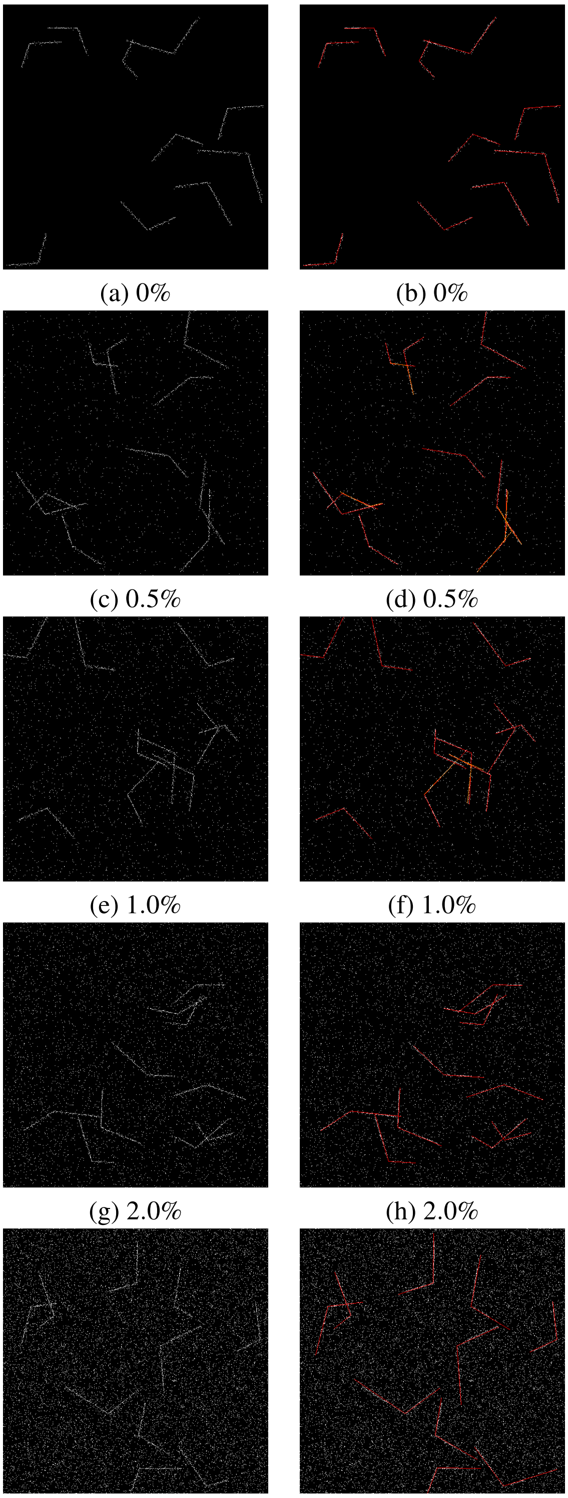
To merge the structures that are very close to each other, we find the peaks in the parameter space (i.e., 3D accumulator) via a $N \times N \times N$ local search. Specifically, the merging is done by averaging the structure parameters for the local maxima above a threshold value within the local search space. We have used a threshold value that is dependent of the total number of the edge points to allow varying threshold value for different noise levels. (We empirically determined that about 3% of the total number of edge points is a good threshold value and 16×16×16-accumulator-bin is a reasonable local search space for the carbon nanocone segmentation.)

The lengths of the structure edges are determined via a simple edge extension scheme that extends the edge from the reference point along the edge directions. Each side edge is extended if there exists "enough" evidence of potential nanocone pixels within a rectangular (e.g., 7×3) region along the edge direction.

3.5 Efficient Processing

To efficiently process the new technique's binning process, we have also explored the data parallelism on commodity hardware. While other types of parallelization are possible, we focused on multithreading in a shared memory environment. (We have also reported our preliminary results on GPU-based processing in Section 5.) Both OpenMP and Pthreads have been used in our parallel implementation. In the multithreading, a set of edge points are assigned to each thread and processed independently. The updates to the accumulator in the shared memory is done via atomic addition.

To increase the performance of the parallel implementation, we manually set the processor affinity of each OpenMP thread using the Pthreads library. Each OpenMP thread's processor affinity was set to a particular CPU core corresponding to its thread



(a) 0%

(b) 0%

(c) 0.5%

(d) 0.5%

(e) 1.0%

(f) 1.0%

(g) 2.0%

(h) 2.0%

(i) 5.0%

(j) 5.0%

Figure 5: Results on synthetic images with perturbing noise of $\sigma = 1.5$ and five different background noise levels (0%, 0.5%, 1.0%, 2.0%, and 5.0%)

```
for $r$ from 1 to 360° do
    for $d$ from 1 to $Max_d$ do
        for $b_x$ from 1 to $Width$ do
            for $b_y$ from 1 to $Height$ do
                for $p$ from 1 to $NumPatterns$ do
                    $o_x, o_y = L(b_x, b_y, r, p, d)$
                    increment $A[p][o_x][o_y][r]$
```

Algorithm 1: An example of a naive implementation of the binning process

```
for $p$ from 1 to $NumPatterns$ do
    for $b_x$ from 1 to $Width$ do
        for $b_y$ from 1 to $Height$ do
            for $d$ from 1 to $Max_d$ do
                for $r$ from 1 to 360° do
                    $o_x, o_y = L(b_x, b_y, r, p, d)$
                    increment $A[p][o_x][o_y][r]$
```

Algorithm 2: A better implementation of the binning process with increased spatial locality of reference

ID using a pthread_setaffinity_np() call. We also increased locality of reference by re-ordering the algorithm. Setting processor affinity allowed for significant performance improvements for the parts of the algorithm with increased locality of reference. This can be attributed to the increased effectiveness of each of the CPU caches under these conditions.

One simple example that illustrates the consideration of locality of reference is shown in Algorithms 1 and 2. Algorithm 1 typifies an implementation of HT binning process that does not exploit the locality of reference well. Due to the row-major-order access pattern of the multidimensional accumulator, the memory access for each iteration of the nested loop results in very high number of cache misses. Algorithm 2 optimizes the naive algorithm by considering more spatial locality of reference.

4 EXPERIMENTAL RESULTS

The effectiveness of the new carbon nanocone structure segmentation technique has been benchmarked using over 1,000 synthetic images and 20 simulated and real carbon nanocone TEM images. (All of the images tested were of size 512 × 512.) The benchmarking included measuring the effectiveness of the technique by considering four different errors–errors in reference point position, structure orientation and side lengths–, the false positive rate, and the match rate, and measuring the efficiency of our parallelized implementation.

We have used the Oakley cluster system of the Ohio Supercomputing Center in our experiments. The system supports 12 cores (Intel Xeon x5650 2.66 GHz CPUs) and 48 GB memory per node (up to thousands of cores in total) and 128 Nvidia Tesla M2070 GPUs in total. Here, we note that while the cluster system provides the state-of-art computing power and our parallelized implementation is scalable to use all the system resources, our experiments were limited to amount of resources that are available on a typical PC configuration (e.g., a hyper-threaded, 6-core CPU and one programmable GPU). This limited resource usage allows to show the technique's performance for typical PC users.

For the synthetic images, we modeled the carbon nanocone structures using two straight lines joined at an angle of 110° or 140°. Modeled structures also had different random side lengths. In addition, two different types of image noise were considered; random background noise and random noise that perturb the positions of the edge points. We have considered 0%, 0.5%, 1.0%, 2.0%, and 5.0% background noise. Gaussian random noise with zero mean and σ's of 0.0, 1.5, and 3.0 was used to perturb the edge point positions.

Figure 5 shows the five samples of the synthetic images on the left column and the segmentation results as red (and orange) overlays for the same images on the right column. These synthetic images contain five different levels of the background noise with the perturbing noise of $\sigma = 1.5$. As shown in the images, the new technique well-segmented all the structures (even for the image with the high background noise). Here, we note that the technique also produced some false positives. While some of the false positives were caused by image noise, many of them were caused by randomly oriented, nearby structures in which the edges from different structures made up a L-shape similar to a carbon nanocone. Such false positives are shown as or-

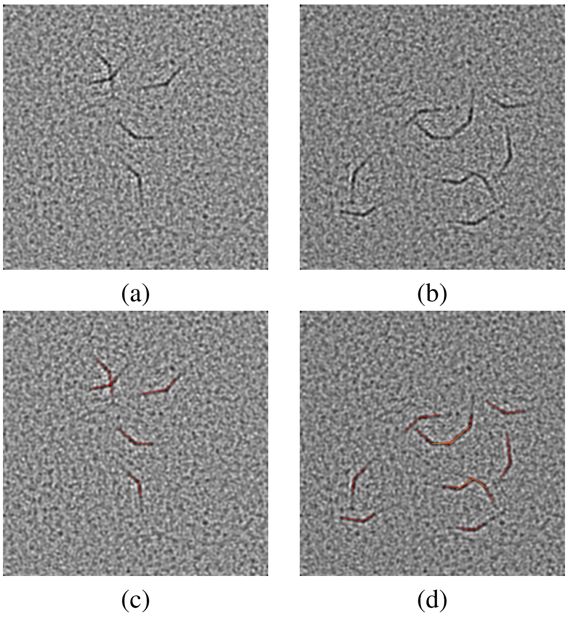

(a) (b) (c) (d)

Figure 6: Results on simulated nanocone TEM images

Noise Level	0.0	0.005	0.01	0.02	0.05
(# of Edge Pt.)	(704)	(2811)	(3308)	(5885)	(18845)
Ref. Pt. Position	0.51	0.41	0.51	0.47	0.67
Orientation	0.31	0.33	0.39	0.58	0.99
Side Length	1.43	1.69	2.11	3.81	16.95
Match Rate	0.99	0.99	1.00	1.00	1.00

(a) With perturbing noise of $\sigma = 0.0$

Noise Level	0.0	0.005	0.01	0.02	0.05
(# of Edge Pt.)	(610)	(2683)	(3209)	(5787)	(18714)
Ref. Pt. Position	0.71	0.72	0.70	0.73	0.93
Orientation	0.52	0.55	0.63	0.83	1.19
Side Length	1.63	2.38	2.91	4.78	16.83
Match Rate	1.00	0.99	1.00	1.00	0.95

(b) With perturbing noise of $\sigma = 1.5$

Noise Level	0.0	0.005	0.01	0.02	0.05
(# of Edge Pt.)	(616)	(2686)	(3220)	(5793)	(18723)
Ref. Pt. Position	1.42	1.35	1.43	1.57	1.52
Orientation	0.79	0.80	0.91	1.11	1.43
Side Length	2.98	3.37	3.45	6.02	14.95
Match Rate	0.92	0.914	0.88	0.78	0.16

(c) With perturbing noise of $\sigma = 3.0$

Table 3: Average errors on synthetic image tests: reference point position error (in pixels), orientation error (in degrees), side length error (in pixels) and average match rate

Rate	Max.	Mean	Min.	Std. Dev.
False Pos. Rate	0.22	0.14	0.00	0.05

Table 4: False positive rate on synthetic images

ange overlaps in Figure 5. For instance, all four false positives in Figure 5 (b) were from the edges of two different structures that made up a L-shape.

Tables 3 and 4 summarize the segmentation results for all the synthetic image tests. The average errors in the reference point position, orientation, and side lengths of structures, and the match rate are shown in Table 3 and four metrics (i.e., maximum, mean, minimum, and standard deviation) of the false positive errors are shown in Table 4. Table 3 also includes the number of edge points considered for different noise levels. As shown in Table 3, the new technique produced very low errors with high match rates for most of the cases. For example, synthetic image tests with a perturbing noise of σ=1.5 and background noise of 0.01% had the average errors of 0.70 in pixels, 0.63 in degrees, 2.91 in pixels for the reference point position, orientation, and side length, respectively, and had a 100% match rate. For most of the synthetic image tests, the new segmentation technique had a match rate of higher than 90%. While the average match rate for the image with the highest noise level (i.e., 5% background noise and perturbing noise of σ=3.0) was only 16%, a typical "cleaned" TEM images of carbon nanocone does not contain similar noise levels. As shown in Table 4, the technique produced about 14% false positive errors on average. However, many of the false positives were caused by the randomly oriented, nearby structures that were explained previously.

For the simulated image tests, the carbon nanocones were created using the TEM image simulation presented by Kirkland [Kir88]. (This image simulation has been used by other scientists in the field.) Figure 6 shows the segmentation results on two samples of simulated TEM carbon nanocones images. In the figure, the sub-images (a) and (b) are the simulated images with different numbers of nanocones and the sub-images (c) and (d) show the segmentation results as red (for matched structures) and orange (for false positives) overlays. As shown in the figures, the new technique segmented all of the structures for these images. Some false positives were also segmented for the simulated images. Many false positives were also caused by nearby structures in which the edges from different structures made up a L-shape similar to a carbon nanocone. For instance, Figure 6 (d) contains two of such false positives.

To analyze the effectiveness of the new segmentation technique for the real carbon nanocone TEM images, the manual segmentation results were used as the gold standard since the actual positions of the real carbon nanocone structures were unknown. For the manual segmentation, the reference point positions, the structure orientations, and the side lengths were manually-selected by a field expert. (Here, we note that the field expert indicated that manual segmentations of the structures are ambiguous and can often lead to mis-segmentations.)

Figure 7 shows the segmentation results on three samples of real images. In the figure, the sub-images (a)–(c) show the input images, the sub-images (d)–(f) show the manual segmentation results as green overlays, and the sub-images (g)–(i) show the automated segmentation results as red (for matched structures) and orange (for false positives) overlays. As shown in the figures, while the new technique segmented the carbon nanocone structures reasonably well, it also produced relatively high number of false positives (i.e., the orange overlays in the figure). Some of the false positives were from side edges from two different nearby structures making a L-shape. Others were from human bias on the subjective manual segmentation of the field expert; the field expert has agreed that some of the carbon nanocone structures were missed in the manual segmentation results after comparing the results with the automatic segmentation results. This differ-

Errors & Rates	Simulated	Real
Ref. Pt. Position	1.94	2.41
Orientation	1.14	1.95
Side Length	3.33	26.04
Match Rate	0.94	0.86
False Pos. Rate	0.05	0.59

Table 5: Average errors on simulated and real image tests: reference point position error (in pixels), orientation error (in degrees), side length error (in pixels), match rate and false positive rate

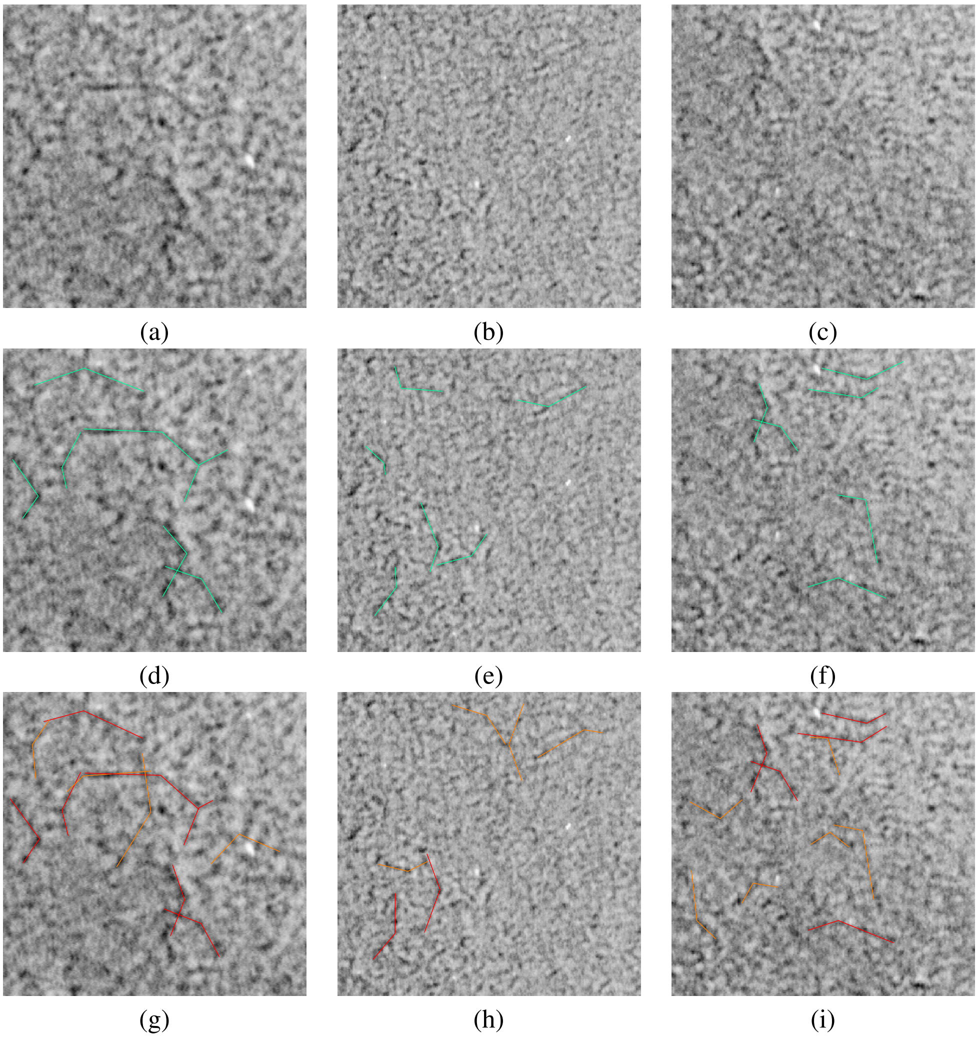


Figure 7: Results on real carbon nanocone TEM images

ence suggests that the results from the new automated segmentation technique could be the "better" segmentation results than the manually-segmented results in some cases.

The same types of errors were measured to benchmark the new technique's effectiveness on the simulated and real images: the errors in the reference point position, orientation, side lengths, and the match rate and false positive rates. Table 5 summarizes the benchmarking. As shown in the table, the new segmentation technique produced very low errors with high match rate for the simulated images; it produced average errors of about 1.94 pixels, 1.14 degrees, 3.33 pixels for the reference point position, orientation, and side lengths, respectively, with 94% match rate and 5% false positive rate on average. The technique also produced very low average errors in the reference point position and the orientation (i.e., about 2.41 pixels and about 1.95 degrees, respectively) with a match rate of 86% for the real image testings. However, it produced somewhat a high average side length error (i.e., about 26 pixels for both edges) and 59% false positive errors. These high side length and false positive errors were caused by the image noise on the carbon nanocone TEM images, the low-contrasted and blurry nanocone structures, and by the human bias on the gold standard (i.e., manually-segmented structures) used for the comparison.

Next, the efficiency of the new segmentation is benchmarked by measuring the execution times on our multithreading-based parallelization of the technique using the synthetic image tests. Figure 8 summaries the efficiency benchmarking results. In the figure, the speedups of the multithreading using 1, 2, 4, 6, 8, 10, and 12 threads are shown for the images with different noise levels. For each noise level, the total number of edge points is also noted in parentheses in the figure. (We note again that only up to 12 threads were used in our experiments for the reason discussed previously. However, our parallelization is scalable to use more threads to achieve real time processing.) As shown

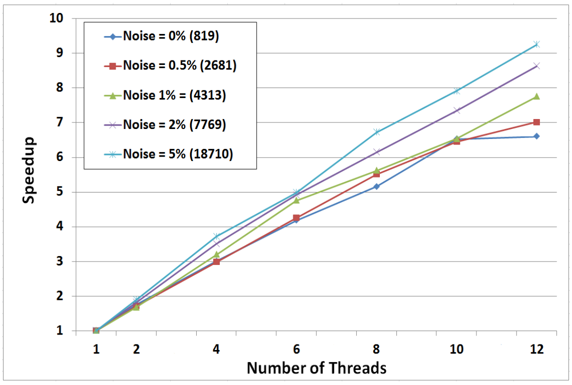


Figure 8: Speedups of multithreading-based parallel processing on synthetic images with different background noise levels with perturbing noise of $\sigma = 1.5$, (the total # of edge points are shown in parenthesis)

Noise Level (# of Edge Pt.)	0.0 (819)	0.005 (2681)	0.01 (4313)	0.02 (7769)	0.05 (18710)
1 thread	34.48	47.56	62.23	93.67	187.60
2 threads	19.60	27.62	37.10	51.39	98.81
4 threads	11.46	15.94	19.48	26.70	50.43
6 threads	8.26	11.16	13.08	19.07	37.65
8 threads	6.69	8.63	11.08	15.24	27.94
10 threads	5.29	7.37	9.53	12.77	23.73
12 threads	5.23	6.78	8.03	10.87	20.31

Table 6: Execution times (in seconds) of multithreading-based parallel processing on synthetic images with different background noise levels with perturbing noise of $\sigma = 1.5$, (the total # of edge points are shown in parenthesis)

in the figures, the multithreaded version performed and scaled reasonably well. For example, when using 12 threads, we have achieved 6.59 (i.e., from 34.48 seconds to 5.23 seconds) and 9.24 (i.e., from 187.60 seconds to 20.31 seconds) speedups for noise levels of 0.0% and 0.05%, respectively over the single threaded version. The execution times (shown in Table 6) varied for different noise levels by about 18% to 23% since the total number of edge points considered in the images varied about from 819 to 18,710 edge points on average. Here, we also note that the performance improvements discussed in Section 3.5 contributed about 10% performance increase.

Lastly, a brief comparison of our automated segmentation and the only prior carbon nanocone segmentation method by Ngatuni et al. [NLW12] is reported. The comparison experiments included measuring the same types of the errors and the execution times using synthetic images with different types of L-shaped linear structures and using one real image (shown in Figure 7 (a)). While our technique performed well on the synthetic image (i.e., errors, match rate, and false positive rate similar to the ones reported in Table 3), their method performed poorly for the synthetic images with L-shapes that have different side lengths since their method weighs the pairs of edge points that have the same distance from the reference point in their binning process. Their method also segmented only one structure for the real image. In addition, their method ran about 3–10 times slower than the serial version of the new segmentation technique.

5 CONCLUSION

We have presented and evaluated a new technique of segmenting L-shaped structures in high-resolution carbon nanocone TEM images. The new technique involves use of a simple lookup table that enables efficient GHT binning process. The new automated segmentation technique produces reasonably good results for a variety of carbon nanocone images and provides a consistent solution. In addition, the multithreading-based parallelization of the technique with the exploitation of locality of reference performs the computationally expensive processing of the segmentation efficiently.

For the future work, other preprocessing steps to distinguish the carbon nanocone structures from other features in the image can be explored. Improvement of the load balancing in the multithreading may also be possible. We plan to achieve real-time processing of the technique by considering a fine-grained parallel approach using programmable GPU processing (e.g., using CUDA) and by considering MPI-based multiprocessing. Our preliminary results on CUDA-based GPU processing shows about 4–6 times performance speedups. However, the limit on the GPU's shared memory and the inefficient atomic operation on the accumulator update make the GPU-based algorithm challenging. One challenge of the MPI-based multiprocessing is the overhead to gather the large accumulator. For example, the MPI_Allreduce operation for gathering the 3D accumulator took about 2.2 seconds in a 12-node configuration. One simple solution to reduce this overhead could be to increase the size of each bin in the accumulator (with a cost of relatively small accuracy).

6 ACKNOWLEDGEMENTS

This work was supported in part by an allocation of computing time from the Ohio Supercomputer Center. We also appreciate Ronald Ngatuni for providing assist on running experiments using their method [NLW12].

7 REFERENCES

[Bal81] Ballard, D., Generalized Hough Transform to Detect Arbitary Patterns, IEEE Trans. of Pattern Analysis and Machine Intelligence, Vol. 13 (2), 1981, pp. 111–122.

[CNG09] Cao, C., Newman, T.S., and Germany, G.A., New Shape-based Auroral Oval Segmentation Driven by LLS-RHT, Pattern Recognition, Vol. 42 (5), 2009, pp. 607–618.

[Deg10] Degirmenci, M., Complex Geometric Primitive Extraction on Graphics Processing Unit, Journal of WSCG, Vol. 18, 2010, pp. 129–134.

[Dou92] Dougherty, E.R., Introduction to Morphological Image Processing, SPIE-International Society for Optical Engine, 1992.

[GoW02] Gonzalez, R.C. and Woods, R.E., Digital Image Processing, second ed., Prentice Hall, 2002.

[HoC96] Ho, C., and Chen, L., A High-speed Algorithm for Elliptical Object Detection, IEEE Transaction on Image Processing, Vol. 5 (3), 1996, pp. 547–550.

[Hou62] Hough, P.V.C., Method and Means for Recognizing Complex Patterns, U.S. Patent, 3,069,654, 1962.

[Kir88] Kirkland, E.J., Advanced Computing in Electron Microscopy, Plenum Press, 1988.

[KlB12] Klicnar, L., and Beran, V., Robust Motion Segmentation for On-line Application, in Proc., 20th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'12), Plzen, Czech Republic, 2012, pp. 205–212.

[KXO90] Kultanen, P., Xu, L., and Oja, E., Randomized Hough Transform (RHT), in Proc., 10th Int'l Conf. on Pattern Recognition, Atlantic City, 1990, pp. 631–635.

[LWN08] Lee, J.K., Wood, B.A., and Newman, T.S., Very Fast Ellipse Detection using GPU-based RHT, in Proc., 19th Int'l Conf. on Pattern Recognition, Tampa, 2008, pp. 1–4.

[Man08] Mandell, E.S., Electron Beam Characterization of Carbon Nanostructures, Ph. D. Dissertation at the University of Missouri-Rolla, 2008.

[McL98] McLaughlin, R.A., Randomized Hough Tranform: Improved Ellipse Detection with Comparison, Pattern Recognition Letters, Vol. 19 (3–4), 1998, pp. 299–305.

[NLW12] Ngatuni, R., Lee, J.K., West, L., and Mandell, E.S., New Hough Transform-based Algorithm for Detecting L-shaped Linear Structures, in Proc., 2012 Int'l Conf. on Image Processing, Computer Vision, and Pattern Recognition (IPCV'12), Las Vegas, 2012, pp. 641-646.

[Pie03] Pietrowcew, A., Face Detection in Colour Images using Fuzzy Hough Transform, Opto-Electronics Review, Vol. 11 (3), 2003, pp. 247–251.

[PRN04] Pintaviroj, C., Romputtal, A., Ngamlamiad, A., Withayachumnankul, W., and Hamanoto, K., Ultrasonic Refractive Index Tomography, Journal of WSCG, Vol. 12, 2004, pp. 333–339.

[Saf12] Safdar, K., Detecting and Removing Islands in Graphics-Rendering-Based Computations of Lower Envelopes of Plane Slabs, in Proc., 20th Int'l Conf. on Central Europe on Computer Graphics, Visualization, and Computer Vision (WSCG'12), Plzen, Czech Republic, 2012, (abstract).

[Sha96] Shapiro, V.A., On the Hough Transform of Multi-level Pictures, Pattern Recognition, Vol. 29 (4), 1996, pp. 589–602.

[SIM03] Strzodka, R., Ihrke, I., and Magnor, M., A Graphics Hardware Implementation of the Generalized Hough for Fast Object Recognition, Scale, and 3D Pose Detection, in Proc., 12th Int'l Conf. on Image Analysis and Processing '03, Mantova, Italy, 2003, pp. 188–193.

[Str00] Strous, L.H., Loop Detection, http://www.lmsal.com/ãschwand/stereo/2000_easton/cdaw.html, 2000.

[URG07] Ujaldon, M., Ruiz, A., and Guil, N., On the Computation of the Circle Hough Transform by a GPU Rasterizer, Pattern Recognition Letters, Vol. 29 (3), 2007, pp. 309–318.

[WCH04] Wu, Y.-T., Chen, H.-Y., Hung, C.-I., Kao, Y.-H., Guo, W.-Y., Yeh, T.-C., and Hsieh, J.-C., Segmentation of Hemodynamics from Dynamic-Susceptibility-Contrast Magnetic Resonance Brain Image Using Sequential Independent Component Analysis, in Proc., 12th Int'l Conf. on Central Europe on Computer Graphics, Visualization, and Computer Vision (WSCG'04), Plzen, Czech Republic, 2004, pp. 267–274.

[XOK90] Xu, L., Oja, E., and Kultanen, P., A New Curve Detection Method: Randomized Hough Transform (RHT), Pattern Recognition Letters, Vol. 11 (5), 1990, pp. 331–338.

[YLL05] Yu, X., Lai, H.C., Liu, S.X.F., and Leong, H.W., A Gridding Hough Transform for Detecting the Straight Lines in Sports Video, in Proc., IEEE Int'l Conf. on Multimedia and Expo, Amsterdam, The Netherlands, 2005, pp. 1–4.

[ZhL05] Zhang, S., and Liu, Z., A Robust, Real-Time Ellipse Detector, Pattern Recognition, Vol. 38 (2), 2005, pp. 273–287.

[ZhS84] Zhang, T.Y., and Suen, C.Y., A Fast Parallel Algorithm for Thinning Digital Patterns, Communications of the ACM, Vol. 27 (3), 1984, pp. 236–239.