DETECTING OVERLAPPING SEMICONDUCTOR NANOPILLARS AND CHARACTERIZATION

Georges Chahine, Michael J Wishon

DETECTING OVERLAPPING SEMICONDUCTOR NANOPILLARS AND CHARACTERIZATION

Georges Chahine and Michael J. Wishon

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia

ABSTRACT

Scientists often individually count and sort items from images manually in a time-consuming and subjective process. Therefore, an automatic algorithm that can provide the same or better results in fractions of the time is desirable and has been done. However, detecting consistently uniform shapes is simple, but most algorithms that we are aware have difficulty with overlapping shapes. Here we demonstrate a relatively simple and fast algorithm to extract and characterize objects from images. Further, it is demonstrated how to detect and sort the blobs into overlapping and non-overlapping categories using a gradient method to create labeled data which is used to train a convolutional neural network. The algorithm shows great promise in the world of semiconductor object detection, growth characterization and can be generalized for other applications such as biomedical imaging.

Index Terms— Image Processing, Classification, Overlap Detection, CNN

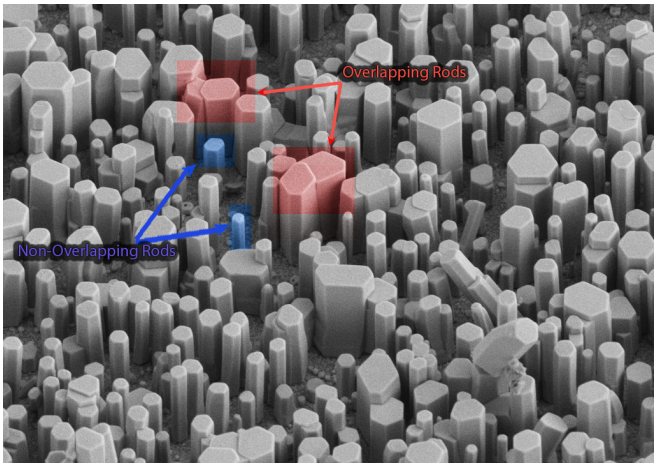

Fig. 1: Isometric view of the nanorods [3]. In contrast to literature dealing with structured overlap shapes such as [4], the above figure highlights the random and chaotic nature of the overlaps being characterized in this paper.

1. INTRODUCTION

Semiconductor nanopillars are of interest to many in the materials and electrical engineering community due to their promise at creating energy efficient lasers and LEDs [1]. Arrays of nanopillars are also explored as a way to create laser or LED arrays. For example in [2], arrays of such nanopillars grown on InGaN substrates were demonstrated. It is important for these arrays to not have overlapping pillars and to be as uniformly spaced as possible. Therefore, it is important to characterize the size, shape, and consistency of these arrays. Currently, this procedure is done manually by observing electron microscopy images and characterizing each nanopillar manually.

An example of an image from an electron microscope of nanopillars grown on InGaN can be seen in Fig. 1 [2]. In the image, it is clear that many of the pillars were grown together and in an non-uniform way creating overlaps. Therefore, detecting and characterizing such arrays in an automatic fashion is very desirable in order to dramatically reduce the amount of time it takes to analyze large or even small datasets because automatic schemes have the ability to analyze dozens of pictures in the matter of seconds where a human would take days. Further, such algorithms remove human subjectivity from the equation, which allows for more consistent results.

Previously, there have been algorithms which deal with detecting objects or blobs, but many fail when it comes to overlapping blobs [5, 6]. The most common uses of these blob detection algorithms is in biomedical research. Specifically, images from testing or imaging equipment are scanned for cells or various other organisms. For example in [7], blob detection was utilized in order to identify whether cancer was present or not. In other instances it was possible to detect nuclei [8]. In [9, 10] it was shown that a blob detection algorithms can be utilized to identify whether a cell is infected or not by scanning microscopy luminescence images. But generally, these algorithms fail when the cells or objects under search are overlapping. In [4], it was possible to detect overlapping cells, but the overlaps are still very uniform and consistent in appearance. Finally, most algorithms are exceedingly complex requiring many steps to smooth and pre-process the images and therefore have relatively slow execution times [4].

In this paper, we sort and characterize overlapping and non-overlapping semiconductor nanopillars grown on InGaN substrates [2]. The images being analyzed were taken from a

top-view perspective of the semiconductor nanopillar arrays. For the remainder of this paper, we will refer to the individual pillars in the top-view as blobs, as shown in Fig. 4. In our algorithm, the electron microscopy images are scanned and the individual nanopillars are extracted using simple thresholding and Canny edge detection [11]. The individual or groups of nanopillars are then analyzed for circularity, surface area, and classified into overlapping and non-overlapping using a gradient method. Ideally the observed nanopillars should be hexagonal or having 6 sides. Therefore, we examine the number of unique sides in order to determine if a blob is overlapping (with another blob) or not using the gradient method, detailed in Section 4. But, due to several factors such as the chaotic nature of the overlaps, this method breaks down because it is susceptible to noise. Therefore, we used the data obtained from the gradient method as a labeling scheme for a convolutional neural network (CNN). By training a CNN using Caffe Caffenet [12], we aim at improving the recall rates even with nonuniform, overlapping blobs. Finally, the algorithm is relatively fast, and we suspect that it can be further generalized to work with any non-uniform object if the appropriate gradient method threshold is chosen.

2. OVERLAP DETECTION SYSTEM

In this section, we detail the setup and training process performed with the aim of detecting overlapping blobs using Caffe CaffeNet [12] model.

2.1. The Gradient Method

Given the complexity of the problem at hand, having an approximate guess of the overlapping blobs provides a starting point for a CNN solution. As shown in Fig. 1, the nanorods have a distinctive hexagonal shape that can be used to check for overlaps. To this end, irregularities in the expected shape could potentially indicate an overlapping blob. Given that our blobs can be modeled as irregular polygons, we propose a gradient method $i.e.$, a derivative-based scheme, that counts the number of edges that form the polygon. As shown in Algorithm 1, a threshold on D limits the number of times the curvature of a blob can change. For instance, the threshold for hexagonal rods is set to 7 $i.e.$, any blob that exhibits more than 7 changes in its curvature is labeled as overlapping.

2.2. Dataset Generation

In Section 2.1, we presented a gradient-based method to generate initial guesses of overlapping blobs. However, the presented method is sensitive to noise with mediocre recall as perceived in Fig. 4. A closer inspection of the detected blobs shows that the method is nevertheless precise, meaning that detected blobs are most likely to be truly overlapping. The gradient method would therefore act as a precursor to the CNN, providing labels for the training process.

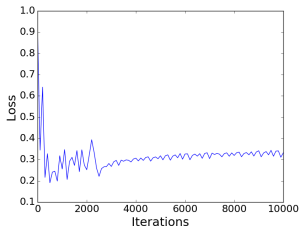

Fig. 2: Loss Function of trained convolution neural network, showing the optimum value after 3000 iterations.

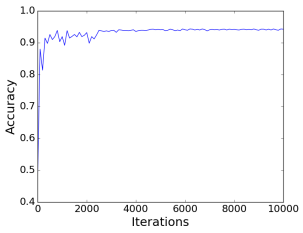

Fig. 3: Accuracy of the trained network against the validation dataset, showing no significant improvement after 3000 iterations.

The generated training dataset consisted of 1996 extracted and labeled blobs, half of which are overlapping. Given that the number of non-overlapping blobs far exceeds that of detected overlapping blobs, and since the training dataset should be balanced and representative, the number of overlapping blobs had to be artificially increased. This was accomplished by performing rotations of that overlapping blob. The required number of rotations k per overlapping blob can be calculated using $k = [(N_i - O_i)/O_i] + 1$, where O_i is the number of overlapping blobs, and N_i is the number of non-overlapping blob. In an effort to reduce discrepancies between blobs of the same nature (overlapping and non-overlapping), and since CNNs learn from recurring patterns, a final step consisted of resizing extracted blobs to 64 X 64 images, while preserving the aspect ratio. Further, the latter normalization reduces the possibility of distracting the classifier, especially because we are only interested in detecting overlapping blobs, regardless of dimensions. As previously discussed, the training labels were generated using the gradient method shown in Section 2.1 and Algorithm 1.

2.3. Training the CNN

The input layer for the CNN consisted of the labeled data, and the dimension of the output layer was set to two. As shown in Fig. 2 and Fig. 3, there is no major improvement in accuracy after 3000 iterations, with the loss function reaching an optimum value at 3000 iterations, before starting to slightly increase as the number of iterations increases.

Training was performed using a computer equipped with a Nvidia K20 GPU (around 2500 cores and 5 GB of RAM). Training took an average of 20 minutes.

3. RESULTS AND DISCUSSION

To validate our model, we ran our code against different images acquired using electron microscopy, such as the ones dis-

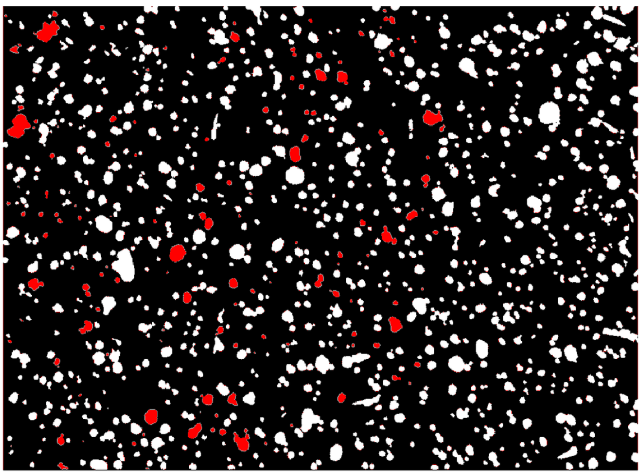

Fig. 4: Overlap Detection using the Gradient Method.

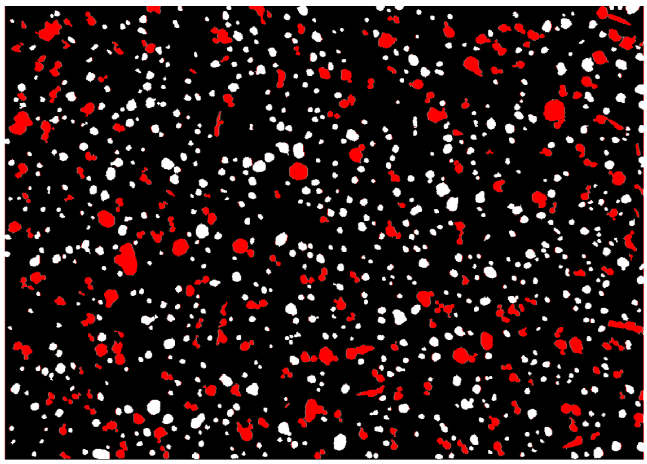

Fig. 5: Overlap Detection using the Trained Model.

played in Fig. 4 and Fig. 5. Table 1 shows the precision and recall rates. These numbers were calculated by slightly modifying the code to pause on each overlapping blob in order to determine the precision rates, while doing the opposite to determine the recall rates. Subjective human evaluation was then used to measure the number of true positive and true negative samples, respectively.

Algorithm 1 Dataset Generation and CNN Training

Input: Training Image I
Output: Trained Model S

for *every blob* M_i *in* I **do**
    Compute Edges
    Extract Contour Points $C_{i,j}$
    Filter M_i ▷ filter very small blobs
    Resize M_i
end for

O=[] ▷ empty list of overlapping blobs
N=[] ▷ empty list of non-overlapping blobs
for *every blob* M_i *in* I **do**
    for *every point* $C_{i,j}$ **do**
        $D = Derive\{C_{i+1,j+1}, C_{i,j}\}$
        if $D > Threshold$ **then**
            $append(O, M_i)$
        else
            $append(N, M_i)$
        end if
    end for
end for

v=size(O)
w=size(N)
$k = [(w - v)/v] + 1$ ▷ required number of rotations
$\alpha = 360/k$
$R_k = [0, \alpha, 2\alpha ... (k-1)\alpha]$
for *every blob* O_i *in list* O **do**
    Apply Rotations $R_k\{O_i\}$
end for
Train CNN with the list of positive examples O and the list of negative examples N

As shown in Fig. 6 and Fig. 7, sampling continued up to the point where no significant change is perceived in both recall and precision. The authors would like to note a noticeable pattern of erroneous detection pertaining to the use of smaller blobs. For instance, as shown in Table 2, the average area per blob is 73.78 square pixels, with a standard deviation of 105.72, with blobs falling on the smaller side of the curve more likely to be falsely classified. To address this problem, all blobs smaller than 7 pixels in either dimension were filtered. Table 2, also shows the corrected number of blobs and other statistics related to Fig. 5.

For an image containing 1308 blobs, it took 20 seconds to detect and segment all the blobs, using a regular computer (4 Cores @ 3.1GHZ and 16 GB of RAM). This includes blob extraction and label generation for comparison purposes. Classification was done in real-time, as shown in the attached video. A simple visual comparison of Fig. 4 and Fig. 5 shows the superiority of the trained model over the gradient method used in the training process.

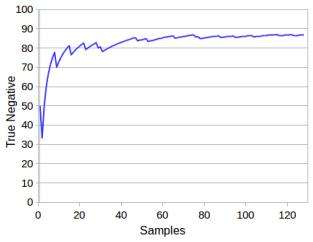

Fig. 6: Sampling and human evaluation of the trained model, showing minor changes to the recall rate after 100 samples.

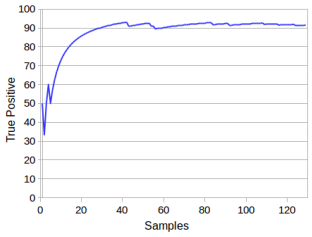

Fig. 7: Sampling and human evaluation of the trained model, showing minor changes to the precision after 60 samples.

Table 1: Precision and Recall, Reflecting the Data Generated in Fig. 6 and Fig. 7.

$Metric$	$O.D.S.$
Precision	91.70%
Recall	85.61%

Table 2: Blob Count Statistics for Fig. 5.

Total Number of Blobs	1308
Number of Overlapping Blobs	202
Average Area per Blob	$73.78\ px^2$
Corrected Average Area per Blob	$< 63.87\ px^2$
Standard Deviation of the Area	105.72

4. CONCLUSION

The method presented here has a 91.7% precision and 85.6% recall when classifying overlapping, non-uniform semiconductor nanopillars with fast processing speed on a standard computer. These results allow for a significantly simpler and faster way to analyze large datasets. Further, the methods presented here could be generalized, if the appropriate gradient method threshold is chosen, to classify other overlapping objects such as cells which can be difficult to detect due to overlaps.

5. ACKNOWLEDGMENTS

The authors would like to thank Dr. Cédric Pradalier for his guidance and comments that greatly improved this paper.

6. REFERENCES

[1] Daniele Bajoni, Pascale Senellart, Esther Wertz, Isabelle Sagnes, Audrey Miard, Aristide Lemaître, and Jacqueline Bloch, "Polariton laser using single micropillar gaas- gaalas semiconductor cavities," *Physical review letters*, vol. 100, no. 4, pp. 047401, 2008.

[2] S Sundaram, Xin Li, Youssef El Gmili, PL Bonanno, R Puybaret, C Pradalier, K Pantzas, G Patriarche, PL Voss, JP Salvestrini, et al., "Single-crystal nanopyramidal bgan by nanoselective area growth on aln/si (111) and gan templates," *Nanotechnology*, vol. 27, no. 11, pp. 115602, 2016.

[3] Suresh Sundaram, Xin Li, Yacine Halfaya, Taha Ayari, Gilles Patriarche, Christopher Bishop, Saiful Alam, Simon Gautier, Paul L. Voss, Jean Paul Salvestrini, and Abdallah Ougazzaden, "Large-area van der waals epitaxial growth of vertical iii-nitride nanodevice structures on layered boron nitride," *Advanced Materials Interfaces*, vol. 0, no. 0, pp. 1900207.

[4] João C Neves, Helena Castro, Ana Tomás, Miguel Coimbra, and Hugo Proença, "Detection and separation of overlapping cells based on contour concavity for leishmania images," *Cytometry Part A*, vol. 85, no. 6, pp. 491–500, 2014.

[5] Tony Lindeberg, "Feature detection with automatic scale selection," *International journal of computer vision*, vol. 30, no. 2, pp. 79–116, 1998.

[6] Stefan Hinz, "Fast and subpixel precise blob detection and attribution," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. IEEE, 2005, vol. 3, pp. III–457.

[7] Woo Kyung Moon, Yi-Wei Shen, Min Sun Bae, Chiun-Sheng Huang, Jeon-Hor Chen, and Ruey-Feng Chang, "Computer-aided tumor detection based on multi-scale blob detection algorithm in automated breast ultrasound images," *IEEE transactions on medical imaging*, vol. 32, no. 7, pp. 1191–1200, 2013.

[8] Anthony Santella, Zhuo Du, Sonja Nowotschin, Anna-Katerina Hadjantonakis, and Zhirong Bao, "A hybrid blob-slice model for accurate and efficient detection of fluorescence labeled nuclei in 3d," *BMC bioinformatics*, vol. 11, no. 1, pp. 580, 2010.

[9] Pedro Leal, Luís Ferro, Marco Marques, Susana Romão, Tânia Cruz, Ana M Tomá, Helena Castro, and Pedro Quelhas, "Automatic assessment of leishmania infection indexes on in vitro macrophage cell cultures," in *International Conference Image Analysis and Recognition*. Springer, 2012, pp. 432–439.

[10] PA Nogueira, "Determining leishmania infection levels by automatic analysis of microscopy images," *arXiv preprint arXiv:1311.2621*, 2013.

[11] John Canny, "A computational approach to edge detec-

tion," *IEEE Transactions on pattern analysis and machine intelligence*, , no. 6, pp. 679–698, 1986.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.