

AUTOMATIC SURFACE DEFECT DETECTION AND SIZING FOR SUPERCONDUCTING RADIO FREQUENCY CAVITY USING HAAR CASCADES*

D. Iriks, Santa Rosa Junior College, Santa Rosa, CA, U.S.A.
G. Ereemeev[†], Jefferson Lab, Newport News, VA, U.S.A.

Abstract

Serious albeit tiny surface defects can remain on the surface of superconducting radio frequency (SRF) cavities after polishing and cleaning. These defects reduce the efficiency of cavities and often limit the maximum attainable fields. We applied a Haar cascade artificial vision technique for automated identification, counting, and sizing of defects induced on niobium surface by Nb-H precipitates formed at cryogenic temperatures. The defects were counted and sized by a computer program and also counted and measured manually to estimate detection rate and accuracy of sizing. The overall detection rate was 53%, and the overall false positive rate was 29%. The technique that was used to automatically size the features was found to oversize the features, but oversize them consistently, resulting in a size histogram that represents the defect size distribution on the sample. After scaling the histogram data, the average defect area was found to be $90 \mu\text{m}^2$ with the standard deviation of $70 \mu\text{m}^2$.

INTRODUCTION

Superconducting RF structures made of sheet metal niobium are widely used around the world to accelerate particle beams [1]. Being cooled to liquid helium temperatures, they provide an efficient way to deliver electric energy to particle beams. While very efficient, the cavities are still prone to limitations such as multipacting, field emission, Q-slopes, and defects. For the best cavities to date, the two main gradient limitations are defects and field emission. Larger defects left on the cavity surface after cavity preparation lead to higher dissipation and eventually quench, while smaller defects contribute to dissipation and reduce the quality factor, which is the measure of cavity efficiency.

A number of tools have been developed for internal inspection of cavity surface for small surface imperfections. The tool most widely used presently is the so-called KEK system [2] and its modifications. The tool is capable of capturing images of the cavity surface at the equator and iris regions. The modifications to this tool include imaging of 9-cell cavity equator regions with automatic cavity

positioning and focusing [3, 4]. The pictures captured by these imaging systems still need to be reviewed in order to identify, count, and record defects on the inner surface. To achieve full automation of cavity inspection process, defects need to be identified automatically. Hence, there is an interest in exploring different techniques for automated defect identification.

METHOD

Sample pictures were captured with a HIROX KH-7700 digital microscope. The MXG-2500REZ lens was used at high-range (x1000), as was an angular light filter. The samples were made out of high residual resistivity ratio (RRR 300) niobium. The defects on the samples surface were formed by hydrogen precipitates formed in the bulk at about 100 K. More details about these defects can be found in [5]. The photographing procedure was to take a picture, move the stage so that there was a small overlap between frames, adjust the lens focus, take another picture, move the stage a little again, and so on. About 2000 negative pictures and 1000 positive pictures were taken. An attempt was made to seek negative pictures generally with backgrounds similar to those in the positives, however some with less similar backgrounds were also intentionally taken. Of the pictures taken, a certain number of positives and negatives were reserved for testing only.

For defect recognition, a C++ program was written in Visual Studio C++ 2008 that uses the OpenCV 2.1. library and a Haar cascade [6, 7]. Instructions and materials for training the cascade can be found online, e.g. [8]. In total, 731 positive pictures were taken for training, and the total count of training defect samples obtained from these images was 1148. The total number of the negative training pictures was 1917.

For testing, four folders of images that were not used in cascade training were assembled. Every test image was a jpg image with dimensions of 1600 by 1200 pixels. One was a folder of 90 negatives. The other three were folders of positives which were separated according to how difficult it was thought it would be for the program to detect the defects. There was an easy folder with 8 images and medium and hard folders with 21 images each. The assignment of positives into these folders was based on personal judgment of the quality of surface cleanliness and de-

* Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177.

[†] grigory@jlab.org

fect distinction. Generally, if nearly all defects had distinct edges and had no other significant surface features in their immediate proximity (e.g., dirt, scratches), the image was placed in the Easy folder. If between about a quarter to two-thirds of the defects were indistinct or had other significant features in their immediate proximity, the image went to the Medium folder. Otherwise, the image went to the Hard folder.

The number of defects was counted for each of the positive test images. Defect, in this case, was defined according to the following criteria: a surface feature caused by the

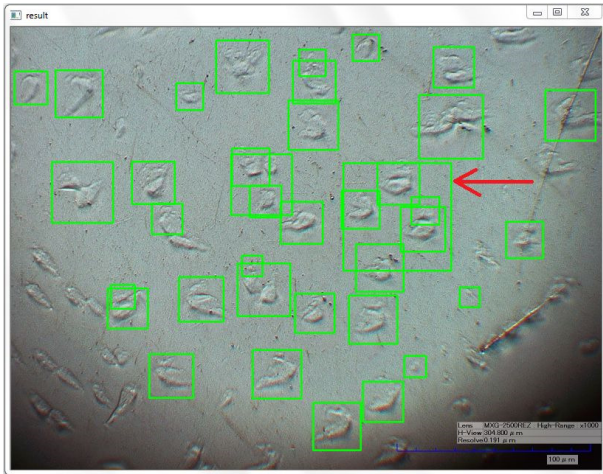


Figure 1: This is the output image for one of the positive test pictures from the Easy folder. One of the false positives is indicated by the arrow. It counted as a false positive for two reasons: more than 50% of contained defect area was contained in other, true positives, and it contained defects separated by more than 10 μm .

formation of Nb-H precipitates in the metal (and when in doubt, to not count it), surrounded completely by negative space (and when in doubt, to count it), and having a major axis of ten micrometers or greater (and when in doubt, to count it). The total number of defects from all three positive folders was found to be 1709.

To calculate the size of detected defects we used a built-in OpenCV function, which identifies rectangular regions containing defects. The pixel area of each region is calculated and scaled using the known physical size of the image to determine the physical size of the identified defect. The program also counts and outputs how many defects are detected within certain size ranges, and from this data, a histogram demonstrating defect size distribution is created.

For the positive test images, other built-in OpenCV functions were applied so that a copy of the input image with rectangles drawn around the identified defect regions was outputted. As the program outputted the total number of detections, the false positives could be counted in these output images and the number of true detections obtained as well. In order to make this counting process more objective and consistent, false positive was defined to mean not containing 50% or more of a defect, containing two defects

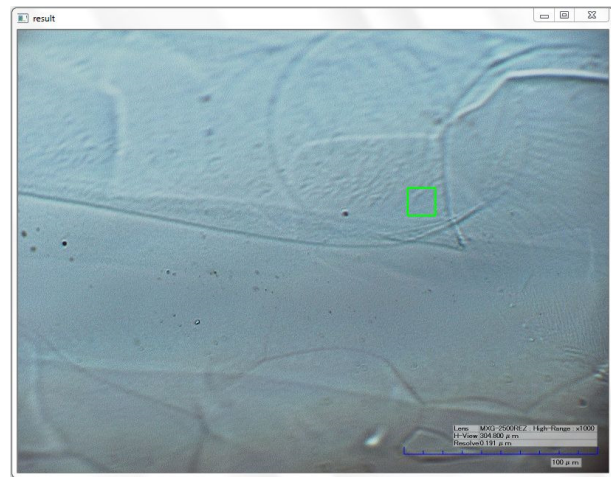


Figure 2: This is the output image for one of the negative test pictures. As it was a negative image and thus contained no defects, any detection would be a false positive. The detection here is a false positive specifically because it does not contain at least 50% of a defect.

separated by more than ten micrometers, or having more than 50% of contained defect area contained in another detection that is true. When any of these requirements was doubted, the detection was not counted as a false positive.

For each positive image, one true positive identification was chosen to test sizing performance. The chosen true positive identification was the one whose rectangles center was nearest to the center of the output image. The rectangles and defects areas were then measured on the computer screen using a ruler. When a positive identification contained two or more defects, they were treated as one larger one. The defects area was approximated by measuring its major and minor axes and multiplying them together.

DISCUSSION

In Figures 1 and 2, sample output windows are shown. In Figure 1, the result of automatic identification on an image with defects is shown. Green boxes are drawn by the program around the area identified as defective. The red arrow was drawn manually in the picture and marks a false positive identification on this image. This identification is a false positive because more than 50% of contained defect area was contained in another, true identification and because this identification contained defects separated by more than 10 μm . In Figure 2, the result of automatic identification on an image without any defect is shown. The picture shows a false positive identification by the program. Figure 3 is a histogram constructed using the defect size data outputted for all of the positive test folders. It demonstrates the distribution of the calculated areas for detected defects amongst 20 area ranges. The results of testing are summarized in Table 1. It contains, for each positive test folder, the detection rate, false positive rate, and the average and relative average deviation of the quotients of calcu-

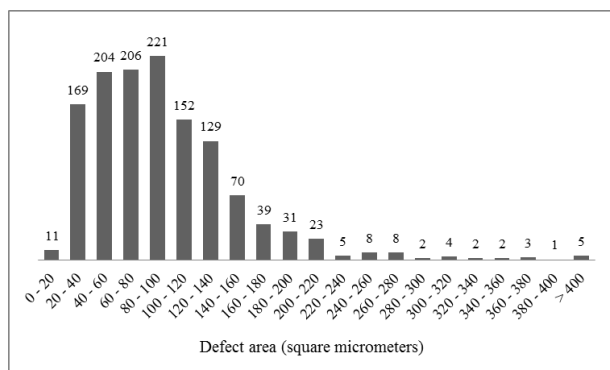


Figure 3: The program counts and outputs how many defects are detected within certain area ranges. This histogram is constructed using the data from all of the positive test folders. The size ranges are scaled by one-fifth since the calculated area was found to be five times larger than the actual value on average. From this data, it can also be determined that the average defect area (using the scaled values) was $90 \mu\text{m}^2$ with a standard deviation of $70 \mu\text{m}^2$.

lated defect area to measured defect area. It also contains, for each folder, the average run time per image. Overall values over all the folders are included also, calculated as the average of all the values from each folder. The number of negative test images that had at least one false positive detection and the total number of false positive detections for the negative test folder are also included. The overall detection rate and false positive rate were, respectively, 53% and 29%. The detection rate was as high as 71% for the Easy folder and as low as 40% for the Hard one, and the false positive rate was as low as 19% for the Easy folder and as high as 37% for the Hard one. The overall time per image was 2.1 seconds. Overall, the rectangle area was 5 times larger than the defect area with a percent relative average deviation of 30%. 11 out of the 90 test negatives had at least one false positive, and there were 18 such false positives in all.

CONCLUSION

Haar-cascade feature identification technique was applied to a set of images of niobium surface with well-defined defects. The overall detection rate was 53%, and the overall false positive rate was 29%. The detection rate was 71% and the false positive rate was 19% for images where defects had distinct edges and had no other significant surface features in their immediate proximity. The technique was also used to automatically size the features, and was found to size the features consistently, resulting in a size histogram that represents the defect size distribution on the sample. The average defect size after scaling was found to be $90 \mu\text{m}^2$ with the standard deviation of $70 \mu\text{m}^2$.

Table 1: Results of testing: (-) column denotes negatives, #1 – "Easy" folder images, #2 – "Medium", and #3 – "Hard".

	(-)	#1	#2	#3	Overall ^a
Detection rate(%)		71	59	40	53
False positive rate (%)		19	29	37	29
Time per image (s)	1.81	3.0	2.8	2.5	2.1
Average ra/da ^b % rad (%) ^c		6	4	5	5
Number of negative test images with false positive detection(s) ^d					11
Total number of false positive detections in negative test images					18

^aAverage of all data values from each of the folders

^b(Rectangle area) / (defect area)

^cPercent relative average deviation of ra/da quotients

^d90 negative test images total

ACKNOWLEDGMENTS

We would like to thank James Lyons for initial testing of OpenCV library for niobium defect recognition, and Samantha Oswald for initial testing of Haar algorithm. We would also like to acknowledge J. Spradlin for his assistance in polishing the niobium samples and the College of William & Marys Surface Characterization Lab and the staff there for the use and assistance with their equipment for sample photography. D.I.s work on this project was performed as part of the Research Experiences for Undergraduates Program conducted at Jefferson Lab in partnership between Jefferson Lab and Old Dominion University. He would like to express his deep gratitude to these two institutions and the program administrators for this wonderful experience, as well as to the National Science Foundation for funding this program.

REFERENCES

- [1] RF Superconductivity: Science, Technology and Applications, Wiley-VCH (2009).
- [2] Y. Iwashita, Y. Tajima, and H. Hayano, Phys. Rev. ST Accel. Beams 11, 093501 (2008).
- [3] E. Toropov, D. A. Sergatskov, Fermi National Accelerator Laboratory report, TD-11-004 (2011).
- [4] A. Navitski, E. Elsen, B. Foster, R. Laasch, D. Reschke, J. Schaffran, W. Singer, X. Singer, Y. Tamashevich, Proc. of SRF 2013, MO053 Paris, France (2013).
- [5] F. Barkov, A. Romanenko, and A. Grassellino, Phys. Rev. ST Accel. Beams 15 122001 (2012).
- [6] P. Viola and M. Jones, Int. J. Comput. Vision, 57(2), 137-154 (2002).
- [7] Intel, Intel Open Source Computer Vision Library.
- [8] <http://nayakamitarup.blogspot.com/2011/07/how-to-make-your-own-haar-trained-xml.html> (posted July 29, 2011).