Edge Detection and Shape Recognition in Neutron Transmission Images[*]

E. D. Sword[†] and S. M. McConchie

Oak Ridge National Laboratory, P.O. Box 2008, MS-6010, Oak Ridge, TN 37831-6010, USA

E-Mail Address: sworded@ornl.gov, mcconchiesm@ornl.gov

Abstract

Neutron transmission measurements are a valuable tool for nondestructively imaging special nuclear materials. Analysis of these images, however, tends to require significant user interaction to determine the sizes, shapes, and likely compositions of measured objects. Computer vision (CV) techniques can be a useful approach to automatically extracting important information from either neutron transmission images or fission-site-mapping images. An automatable approach has been developed that processes an input image and, through recursive application of CV techniques, produces a set of basic shapes that define surfaces observed in the image. These shapes can then be compared to a library of known shape configurations to determine if the measured object matches its expected configuration, as could be done behind an information barrier for arms control treaty verification inspections.

Introduction

Neutron transmission imaging is often used to produce radiographs of nuclear material as either an alternative or a supplement to more conventional photon (gamma, x-ray) imaging. The process generally makes use of an active neutron source (e.g., ^{252}Cf, D-T neutron generator) that produces time-tagged neutrons, which penetrate an inspection object and are detected on the other side [1][2]. The images produced by neutron transmission imaging are, at present, lower in resolution than most current photon techniques, but the greater penetration of neutrons through high-Z material produces images that offset this central weakness of x-ray transmission images. Additionally, for objects that undergo neutron-induced fission, similar imaging techniques may be used to generate a map of fission locations in the object [2].

At present, neutron transmission images are analyzed using visual inspection by an expert. Tests have been carried out to determine to what extent a relative novice user can be trained to identify important features in a neutron transmission image [3], although unpredictable systematic errors must be anticipated in general when relying on a visual inspection. Other efforts have been undertaken to implement a library-based approach, in which images representing components of a possible object are produced using computer simulations, combined, and compared with the measured object [4]. This methodology holds promise but requires either prior knowledge of the object's constituent parts or some information as to its internal construction to reduce the library of possible objects.

Imaging systems are able to provide additional geometric information about an object, which is not available from traditional nuclear measurement techniques such as passive gamma-ray spectroscopy or neutron multiplicity counting. This information comes at the price of additional invasiveness, resulting in a high probability for the release of sensitive information. Therefore, in some applications, it will be necessary to carry out the neutron transmission measurement behind an information barrier, to prevent undesired access to the potentially sensitive measured information [5]. An information barrier may be as simple as a shroud covering the object from view, or it may involve hardware and software features that prevent sensitive data from being stored or displayed. One possible approach to address the need of an information barrier for neutron transmission imaging is automatic decomposition of produced images into software-analyzable data using computer vision techniques. Additionally, if the surfaces in the object can be extracted, it may be possible to automatically model the geometry using Monte Carlo simulation techniques.

Background & Methodology

Computer vision (CV) techniques are used for an increasing variety of applications, including optical character recognition (OCR), object tracking in video, facial recognition, and medical image processing. The edge and surface identification process makes use of two steps. An image is first put through an edge detection algorithm, which is then followed by recursive Hough transforms, in order to identify likely lines and circles.

In this analysis, the first technique applied to a neutron radiograph is edge detection. Edge detection examines the image to find pixels that are a local maximum for the image gradient. The algorithm used for this application was described by Canny in 1986 [6]. An example of this technique is shown in Figure 1, for Inspection Object #9 (IO9) from a series of measurements carried out at the Idaho National Laboratory Zaro Power Physics Reactor facility [1]. Canny edge detection involves first applying a Gaussian blur filter to the image, smoothing areas of noise in the data [Figure 1(b)]. After smoothing the image, a Sobel operator is used to convert the image "intensity" (in this case, the neutron attenuation) into a gradient (c), which has large values when the intensity is changing rapidly. In order to produce sharp edges in the final image, all points that are not local maxima are removed (d). Finally, the Canny method uses edge tracing (e and f), in which two thresholds are applied to the gradient image: one high and one low. Any points that pass the high threshold are considered a part of the edge. To extend the edge into a more continuous line, the algorithm begins at points that pass the high threshold and applies a lower threshold to adjacent pixels. Points that are connected to "high threshold" points and that pass the lower threshold are included as parts of the edge. The algorithm continues applying the lower threshold until the adjacent points fail to pass. This produces a simple line drawing that is capable of emphasizing important features in an image. A final comparison overlaying the edge-detection result (black lines) on the original image is shown in Figure 2.
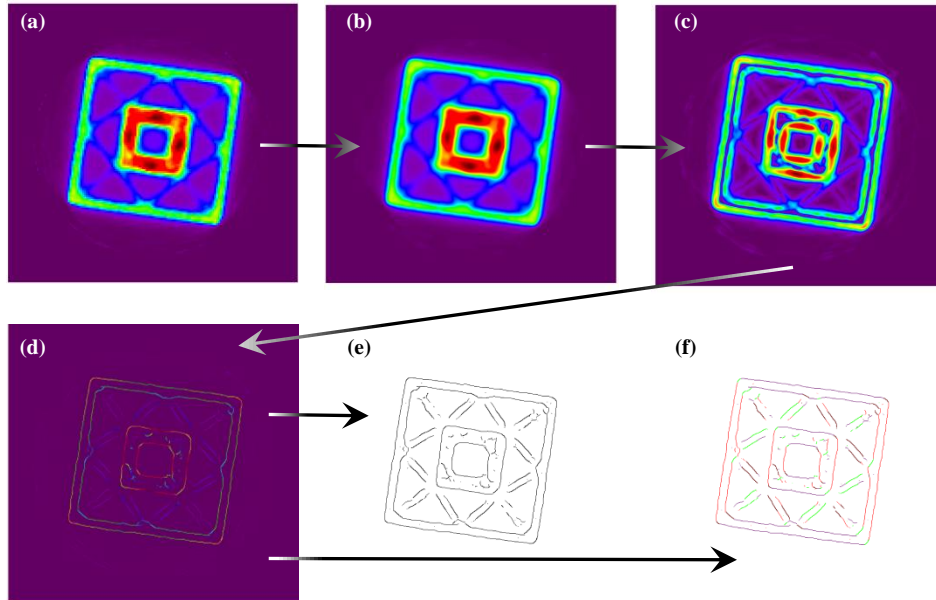
Figure 1: Steps involved in Canny edge detection using IO9 [1]. Beginning with the input image (a), a Gaussian blur is applied (b), followed by a gradient (c). Points in the gradient image that are not local maxima are suppressed (d). Finally, the thresholds are applied and the edges are traced, giving the final edge-detected image (e). Additionally, lines surviving in the edge-detected image can be colored by the direction of the gradient (f) if the additional information is desirable.
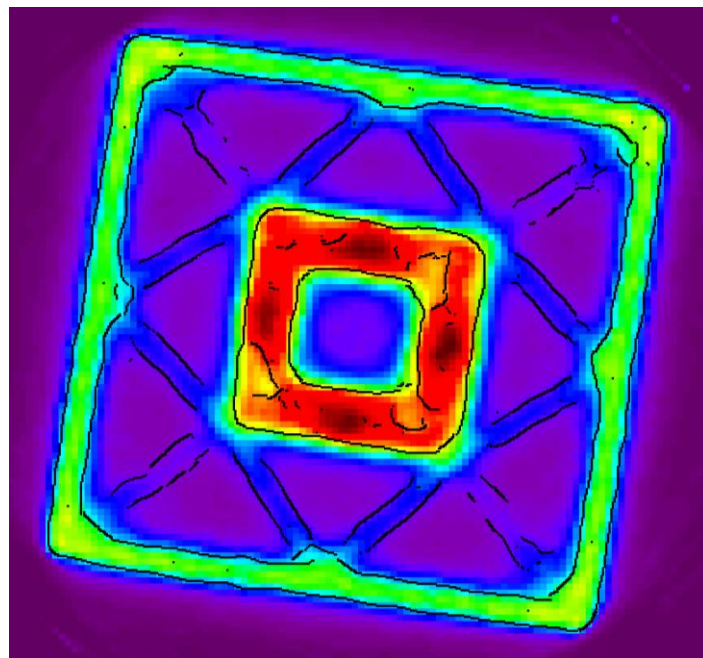

Figure 2: Overlay of the original image of IO9 [1] with the result of edge detection.

In order to produce additional information regarding shapes in the image, the Hough transform is used to detect particular shapes [7]. For this application, the majority of shapes are either circles or lines, which makes shape detection much simpler. This process transforms an image into a parameter space that is defined by the particular shape being searched for (e.g., lines, circles) with the dimensionality set by the number of independent parameters required to define the shape (e.g., 2 for lines, 3 for circles). The algorithm then populates the Hough transform space by a voting scheme. Each point in the image votes for a set of points in the Hough space, and the points with the greatest number of votes (local maxima) define the shapes that were detected. The Hough transform can be carried out on a raw image gradient, or it can follow a preprocessing step such as edge detection. Preprocessing the image results in a much simpler transformed space.

An example of the transform used in the search for lines is shown in Figure 3. The algorithm transforms a single pixel's x and y coordinates into a sinusoid in the Hough space, so that

$$r(\theta) = (x - x_0)\cos(\theta) + (y - y_0)\sin(\theta), \qquad (1)$$

where x_0 and y_0 denote the center of the image. Likewise, each point in the Hough space produces a vector in the image, where the radius is taken from the center of the image, the angle is taken from the horizontal (increasing clockwise), and the corresponding line runs perpendicular to the defined vector. The set of 'detected' lines can be determined by applying a score cutoff to the transformed space, with only a few successful lines surviving. Circles proceed analogously; however, they generally require a three-dimensional Hough transform corresponding to the center point of the circle and its radius [9][10].
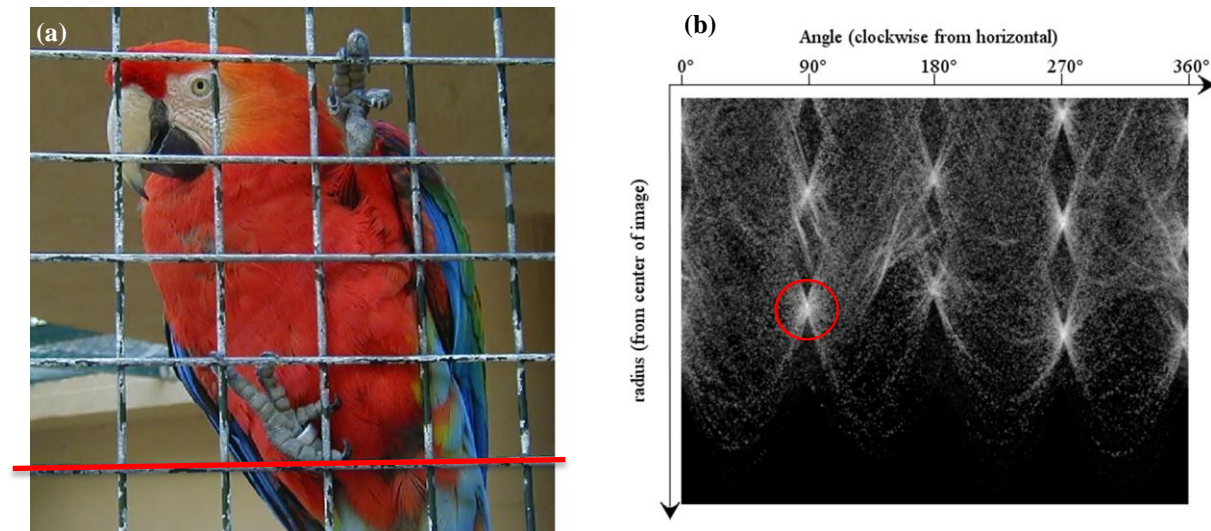

Figure 3: Example of Hough transform for line-shapes. (a) The source image, with a line in the image marked by a red line. (b) The transformed space, where the brightness of points corresponds to the score of the line defined by the (r,θ) value. The transform point corresponding to the red line in (a) is circled in red. These images were produced using the C++ Template Image Processing Toolkit (CImg) [8].

Challenges

The types of objects commonly imaged using neutron transmission imaging consist of relatively simple geometries; however, there remain a number of challenges to automating image analysis on real objects. These challenges include noise in image reconstructions and irregularities caused by limited spatial resolution

Noise in an image can cause confusion in the edge and shape detection process. This problem is particularly significant in tomographic images produced using the filtered back-projection (FBP) technique [11]. When using FBP for image reconstruction, a radial striping effect occurs due to the limited number of projections acquired in the sinogram. This effect lessens as the number of projections increases, but it can be avoided altogether using another tomographic reconstruction technique. The method of maximum-likelihood expectation maximization (MLEM) is an iterative reconstruction technique that refines an initial guess to match the input sinogram [12]. Because this method approximates the answer, it does not produce the noise pattern found in images constructed using FBP.
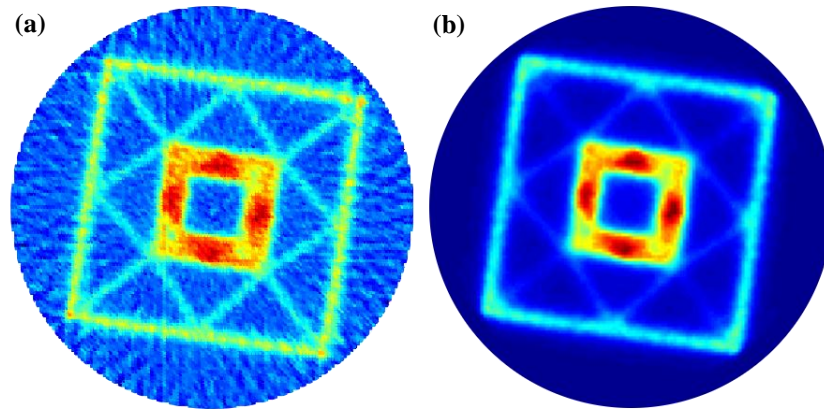

Figure 3: Comparison of images of IO 9 [1] produced using (a) filtered back-projection (FBP) and (b) maximum likelihood expectation maximization (MLEM).

Because of the fixed size of pixels and imperfections in reconstructing the edges of some objects, a single edge may produce a high Hough transform score for a large number of very similar shapes. An example of such a feature can be seen in the corners of the object shown in Figure 2. When attempting to detect all shapes in an image, this causes a problem, as a family of shapes at a high score value will obscure other relevant shapes with a smaller score value, as shown in Figure 4 (a). In order to correct this, as shapes are detected, they are removed from the edge-detected image to prevent them from interfering with the detection of less heavily weighted shapes. The result produces a set of shapes at the same threshold, but with fewer detected shapes, as shown in Figure 4 (b). This may, however, cause additional problems in images where the edge lines cross frequently, as it will eventually erode a line that intersects several others. An example of this can be seen in Figure 4 (c), where a number of the short lines around the perimeter seen in (b) have been erased by the crossing lines.
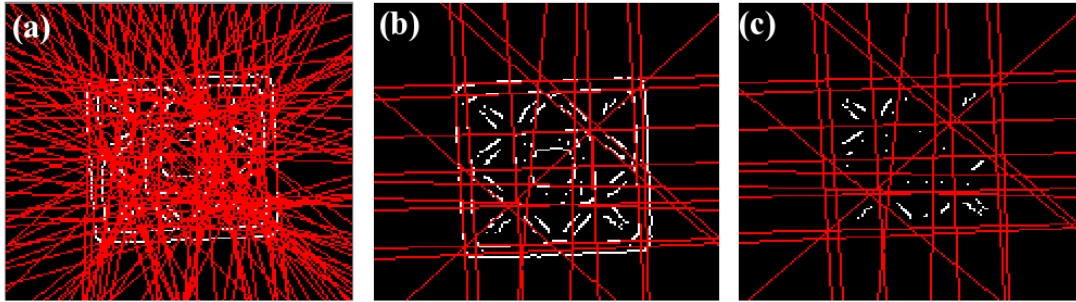
Figure 4: Comparison of shape-finding examples on IO 9 [1] using a Hough transform with low threshold (a) with no edge removal (b) with edge removal, superimposed on the original edge-detected image, and (c) with edge removal, showing the remaining edges only.

Results

A software package was developed using the OpenCV framework [12] to carry out edge detection and shape finding. The shape identification routine makes use of an edge-detected image as input and, as shapes are found, they are removed from the image. This ensures that the algorithm does not continue to consider data that have already been accounted for when searching for "weaker" shapes. An example of the starting point and the final result of this process is shown in Figure 4 and Figure 5 for a test geometry consisting of a steel brick and an annular depleted uranium casting inside a box, which produces an image with a combination of lines and circles.
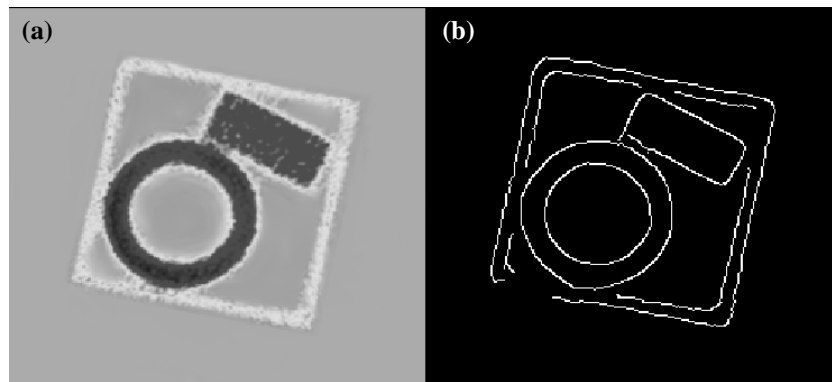

Figure 4: Starting image for shape detection. (a) Grayscale image to be analyzed. (b) Image following edge-detection.
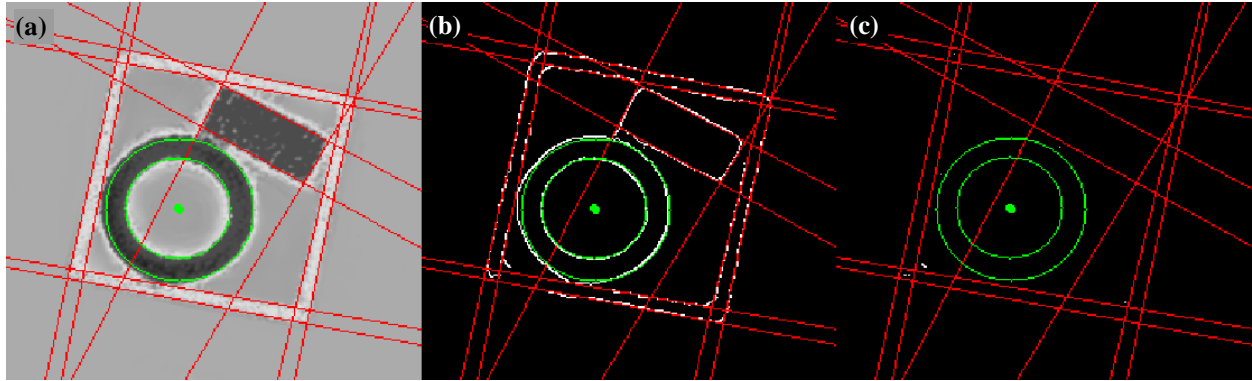
Figure 5: Result of shape detection. Lines are shown in red, circles are shown in green. (a) Overlay of detected shapes on starting image. (b) Overlay of detected shapes on edge-detected image. (c) Overlay of detected shapes on the edge-image following removal of edges that have already been accounted for.

After shape detection is completed, equations describing the detected surfaces can be written to file in a format convenient for later use (e.g., visualization, modeling, comparison to expected values). This process is currently carried out by manually changing threshold values, but it may be possible to automate this process, removing user bias and leading to a capability for automatic analysis of geometric features. At present, a feature is implemented that allows the user to store the threshold values and steps taken to detect the surfaces in a single image and allows the user to "replay" these steps with other images. This allows semi automatic analysis of a family of similar images using a simple script-like framework.

Future Work
This work provides a proof-of-concept that computer vision techniques can be successful and useful in decomposing of a radiation map into (1) a simple binary map of edges and (2) a list of surfaces detected in the image. This provides a basic functionality that allows for several areas of refinement and expansion.

The algorithms used for these tests were the basic edge-detection and shape-finding capabilities built into the OpenCV framework [13]. There are a number of alternative techniques used for identifying objects of interest in images, including making use of a number of different edge- or shape-detection algorithms [14], detecting additional features, such as corners [15], machine-learning approaches [15], or the generalized Hough transform [17], among others.

The ability of this software to be used as an input to other algorithms, such as modeling or library comparison, is an attractive application. It is, therefore, important to examine the ability to take the list of detected shapes and automatically produce, not only surfaces, but also volumes. This proves to be conceptually difficult, as there are a number of ways to combine an arbitrary set of surfaces into logical volumes. It may be possible, however, to incorporate values from the original image to produce a sort of surface-constrained "fuzzy select" (or "magic-wand") tool, similar to the tools used by popular image-editing software.

References

[1]. J. Mullens, et. al., "Neutron Radiography and Fission Mapping Measurements of Nuclear Materials with Varying Composition and Shielding", Institute of Nuclear Materials Management Conference, July 2010.

[2]. P. Hausladen, et. al., "Induced Fission Imaging of Nuclear Material", Institute of Nuclear Materials Management Conference, July 2010.

[3]. A. Swift, et. al., "Attributes from NMIS Time Coincidence, Fast Neutron Imaging, Fission Mapping, and Gamma-Ray Spectrometry Data", Institute of Nuclear Materials Management Conference, July 2012.

[4]. B.R. Grogan, et. al., "Identification of Shielding Matrerial Configurations using NMIS Imaging", Institute of Nuclear Materials Management Conference, July 2011.

[5]. D. MacArthur, et. al., "Use of Information Barriers to Protect Classified Information", Institute of Nuclear Materials Management Conference, July 1998.

[6]. J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8.6, 1986.

[7]. R. O. Duda and P. E. Hart, "Use of the Hough Transform to Detect Lines and Curves in Pictures", *Comm. ACM*, Vol 15, p. 11-15, 1972.

[8]. "The C++ Template Image Processing Toolkit", http://cimg.sourceforge.net, Accessed May, 2012.

[9]. M. Rizon, et. al., "Object Detection using Circular Hough Transform", *American Journal of Applied Sciences* Vol. 2.12, 2005.

[10]. M. Smereka and I. Duleba, "Circular Object Detection using a Modified Hough Transform", *Int. J. Appl. Math. Comput. Sci.*, Vol. 18, No. 1, 2008.

[11]. A.M. Cormack, "Sampling the Radon Transform with Beams of Finite Width", *Phys. Med. Biol.*, Vol. 23, No. 6, 1141-1148, 1978.

[12]. K. Lange and R. Carson, "EM Reconstruction Algorithms for Emission and Transmission Tomography", *Journal of Computer Assisted Tomography*, Vol. 8(2) 1984, pg. 306-316.

[13]. Open CV, http://opencv.willowgarage.com/wiki/, Accessed May, 2012.

[14]. C. Rothwell, "Hierarchical Object Description Using Invariants", *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, p. 397-414, 1994.

[15]. C. Harris and M. Stephens, "A Combined Corner and Edge Detector", *Proceedings of the 4th Alvey Vision Conference* p. 147-151, 1988.

[16]. A.P. Ashbrook, N.A. Thacker and P.I. Rockett, "Multiple Shape Recognition Using Pairwise Geometric Histogram Based Algorithms", *Proceedings of the Fifth International Conference on Image Processing and its Applications*, 1995.

[17]. D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", *Pattern Recognition*, Vol.13, No.2, p.111-122, 1981.