Proceedings of 2016 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Harbin, China

Vision-based Keyhole Detection and Parameter Extraction in Door Opening Task

Jiajun Li, Xu Xu, Jianguo Tao*, Liang Ding, Haibo Gao, Zongquan Deng

State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, China
Corresponding Author*: jgtao@hit.edu.cn

***Abstract*—It is important for a robot to be able to open a door in indoor environment. There are occasions when the robot has to open a locked door with a key. Problem arises when the robot tries to insert the key into the keyhole; it has to know the location and direction of the keyhole. In this paper, a procedure is developed to deal with this problem. First we detect the keyhole with the HOG features and a SVM classifier. The detection process has an average precision of 89%. Then the detected target image is processed by the opening operator in order to filter out the disturbance caused by the error in assembling. The resulting image is a binary image which contains only the pixels corresponding to the keyhole. Finally, we calculate the location of the keyhole with the image moment. The direction of the keyhole is computed with two different methods. We compared and analyzed these two method and the result shows that the calculated dominant direct has a constant error about one degree.**

Keywords—keyhole; detection; image opening; dominant direction; spectrum method

I. INTRODUCTION

Nowadays there is an increasing demand for robots working as home servers or nuclear disaster rescuers. In these tasks, the robots have to work in complex indoor environment which imposes additional requirements to the robots compared to the traditional structured environment in factories.

In indoor environment, it is inevitable for the robots to navigate through a door to explore the space behind it. Now the problem is what if the door is closed, moreover, what if the door is locked, and can only be unlocked by a key. Many people think it is a trivial task. It is trivial for human because we do open doors every day, but it non-trivial and important for robots working in indoor environment to be able to unlock and open doors.

Researches have been carried out to tackle the door opening problem during the past decade. In [1], a framework was proposed to open doors and drawers in unknown environment by Tomas and Jurgen. The frame work consists of three components: door handle detection and localization element, kinematic model learning and operating element and a semantic map used to integrate the learned model for further reuse. The authors tested the framework on PR2 robot and the success rate is 51.9% among 104 trails.

Sachin Chitta and Benjamin Cohen proposed a graph-based representation to overcome the planning problem in high dimension space [2]. This representation is small enough for efficient planning yet rich enough to contain feasible motions needed for door opening tasks. The authors utilized graph-based search method to handle the wide variance of conditions under which doors need to be opened.

Fig. 1. Keyhole used in this paper

Saleh Ahmad and Hongwei Zhang [3] designed a modular and reconfigurable robot (MRR) mounted on a wheeled mobile platform. A switching mode control strategy was proposed to prevent the occurrence of large internal forces which arise because of the positioning errors or imprecise modeling of the robot or its environments by utilizing the multiple working modes of the MRR modules.

The researches that addressed the door opening problem can be divided into three classes according to the specific sub-problem they focused on: door handle detection and localization problem, motion planning of the manipulator and the mobile base, control strategy design to handle the forces and uncertainties. Most of the researches which handle the last two problems avoid the circumstances that the door is locked. They assume that the door can be opened by the manipulator under feasible motion planning.

In this paper, we will focus on unlocking the door. Similar to the door opening problem, the unlocking problem can also be roughly divided into three consecutive sub-tasks: keyhole detection and localization, motion planning and control strategy design. The last two sub-tasks are beyond our scope, and we focus on dealing with the first detection and localization task.

The detection and localization task consists of two goals: (a) keyhole detection and localization which means the robot must localize the keyhole in the color image collected by the camera. (b) The calculation of the direction of the keyhole. The direction of the keyhole is one of the most important parameter needed in the following motion planning step.

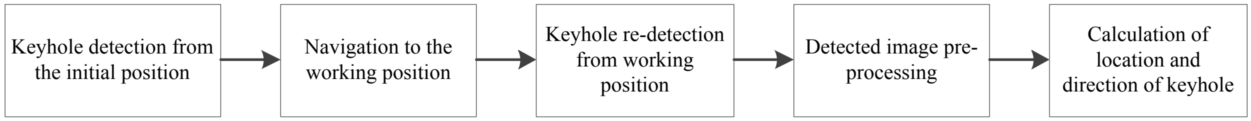


Fig. 2. Overview of the proposed method

As shown in Fig. 1, the robot is required to detect and localize the keyhole in the field of vision. This is achieved by utilizing detection techniques in computer vision. Specifically, we use Histogram of Oriented Gradient (HOG) as features to represent the keyhole, Support Vector Machine as classifiers to distinguish keyhole region from non-keyhole region, Sliding Window detector to localize keyhole in color image. These will be further discussed in Section IV.

After we obtain the detected region of keyhole, we process the image to filter out the unwanted area which will give us no information about the parameter of the direction of the keyhole. The details are presented in Section V. What follows is the calculation of the direction of the keyhole in Section VI. The conclusion is shown in Section VII

II. Related Work

In this section, we will briefly introduce some of the related works and method. As there is no direct related works on keyhole detection and localization, we present the related topic on door handle detection and localization.

Ellen and Ashutosh [4] tried to locate a door handle with color images and then make use of the 3D range information to extract the “3D key locations” of the handle. They utilized the sliding window method to locate the handles and tuned the empirical constants which indicate the “3D key locations” to get the geometry feature of the handles.

Thomas and Jurgen [1] proposed a method with two functional elements: one for detecting specular handles and the other for non-specular handles. These two components are complimentary because the RGBD sensors use infrared light to get the depth information which may cause problems with specular surfaces.

Wim and Melonee carried out the door and handle detection with both RGBD sensors and laser scanners [5]. During the handle detecting process, an object classifier which is implemented by OpenCV’s HaarClassifier Cascade is utilized, and then the 3D range information from the stereo camera is used to filter out the false positives.

III. Overview of the Method

In this section, we briefly introduce the structure of our method shown in Fig. 2. As we can see, the method consists of several consecutive actions.

- The robot detects the keyhole in color image. Combining the depth information, we can compute a motion primitive for the robot to navigate to the front of the door.
- The robot navigates to the door.
- The robot re-detects the localization of the keyhole to obtain a refined image of the keyhole for the follow process.
- Filter the refined keyhole image to extract the keyhole pattern.
- Compute the location and direction of the keyhole.

Following the steps above, we will eventually get the location and direction of the keyhole. Note that before we continue the paper, we have to make the following few assumptions about the work in this paper.

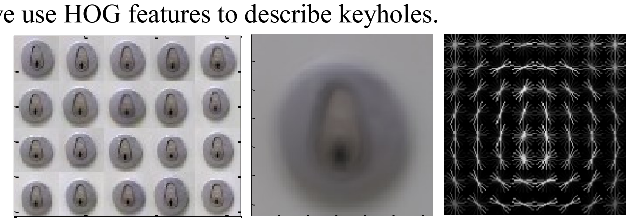
- The robot can navigate to the front of the door automatically or with the assists of human operator. So in this paper, we will not discuss the navigation process. Similar assumptions are made for the manipulator.
- In this paper, we consider only one kind of keyhole, though there are hundreds of different kinds of door latches and keyholes. This assumption is appropriate in many situations. In our lab, there is only one kind of keyhole. In nuclear power station in China, the situation is the same.
- The robot has access to the depth data collected by the Kinect sensor. So when the location and direction of the keyhole is calculated, the corresponding 3D information is also obtained. We will not further discuss how we obtain and process the depth information.

IV. Keyhole Detection

In this section, we will try to detect and localize the keyhole by a keyhole detector which is based on HOG, SVM and Sliding Window technique. This is the first time these techniques being applied to keyhole detection.

A. HOG Features and SVM Classifier

HOG (Histogram of Oriented Gradient) was first proposed by Dalal and Bill Triggs in the CVPR 2005[6]. It’s a kind of computer vision and image processing techniques used for object detection feature descriptor. It captures local gradient directions of the color image to form features. The HOG features are widely used in different kinds of tasks because of its strong ability to describe image features. The HOG feature is invariant of changes in light intensity so that it can be applied to detect keyholes in dim environment. In this paper, we use HOG features to describe keyholes.



(a) (b) (c)

Fig. 3. (a) Some Training images (positive samples) with keyhole. (b) The average features of the training images. (c) HOG model extracted from positive samples.

To compute HOG features, we firstly divide the image into several small connected regions, and each connected component is called a cell unit. Then calculate and accumulate the gradient direction histogram or edge direction histogram in each cell. Finally, the combination of these histograms constitutes the HOG features of images.

A SVM [7] (support vector machine) constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Categories can be determined by adjusting the sample size threshold.

B. Keyhole Detection and Localization

The keyhole detection and localization process consists of several consecutive sub-process including feature extraction, classifier training, hard mining and detection. In this section, we will show the details of our method.

1) Train and Test Set

We collect 200 samples for train set and test set respectively and in both the train and test set, half of the samples are positives and the other half are negatives. We assume that in each positive sample there is only one keyhole. The region of the keyhole has been recorded as a rectangle. This rectangle can be of any size, but should roughly be square with some background included in the region, as shown in Fig. 3(a).

The negative samples which do not contain the target object are chosen randomly from the images in the train set. There are totally 200 negative samples.

2) HOG Feature Extraction

HOG feature extraction process includes the calculation of the gradient value, and the construction of the gradient histogram.

- Use discrete differential template to calculate the gradient of each cell in the horizontal and vertical direction at the same time. And get the average features of positive samples, as shows in Fig. 3(b). Note that the cell unit size is set to 8 x 8 pixels, when extracting HOG features.
- Then we construct gradient direction histogram for the cell unit. Each pixel point in the cell votes for the weighted histogram channel. And the weights are calculated according to the pixel gradient amplitude. The extracted HOG feature show in the Fig. 3(c).

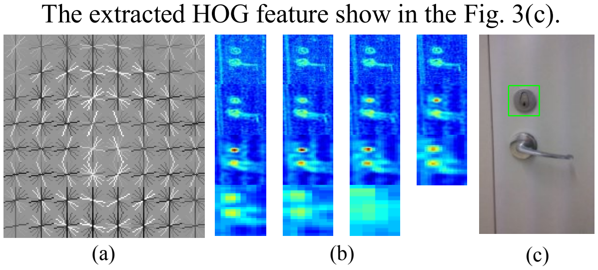


Fig. 4. (a) SVM HOG model. (b) The test image score cloud picture. (c) SVM detector output.

3) SVM Classifier Training

We use the HOG features extracted in the previous step to train the SVM classifier as shown in Fig. 4. The weights of the initial SVM classifier are tuned according to the samples in the train set. The learned model so far is too weak to work well. So we continue to retrain the model in the next step with the hard negative mining technique.

4) Hard Mining

The weak classifier trained in the previous step will produce more false positive predictions than we expect. We will further utilize these false positive samples to retrain our SVM classifier to get a more robust model. By iteratively carrying out this process, the average precision of the classifier on this specific data set will increase dramatically.

5) Sliding Window Detection

After we get a robust classifier from the previous step, we use it to detect keyholes in images in test set. The sliding window detector is utilized here. As objects exist in images at sizes different from one of the learned template, we use multi-scaled sliding window detector in order to find target object at different scales [8].

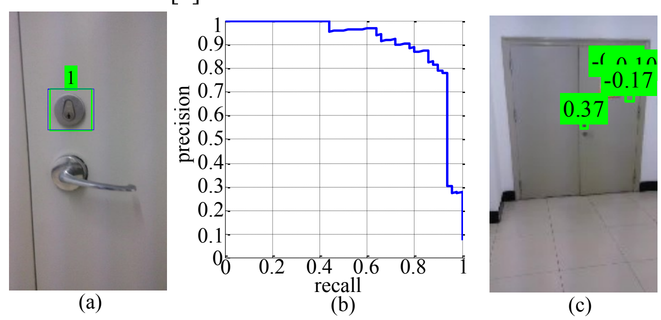


Fig. 5. (a) Sample detection with SVM. (b) The precision of these images. (c) Evaluate the learned model.Model Evaluation

We carry out the experiments on the collected data set, and the result is shown in Fig. 5. Fig.5 (a) shows that the detector can successfully detect target object about 0.5 meters away in front of the camera. Fig. 5(b) shows that the detector works well when the target object is about 2.5 meters away from the camera. This result indicates that the multi-scale sliding window detector is suitable for the task. Fig. 5(c) shows the precision and recall curve of the detector. The average precision of the proposed detector in this specific test set is 89%.

V. Image Pre-processing

In previous section, we implement the detection of the keyhole. The result of the detection is a small rectangular region that contains the target keyhole in it. In this section, we will further process the resulting image to get the exact pixels that corresponds to the keyhole. The location and shape of the pixels are needed in next section where we compute the location and direction of the keyhole in working space.

We randomly select a detected keyhole as shown in Fig. 6. The pixels inside the solid rectangle are the target pixels we want. They share a same feature that all these pixels are black. The pixels in the dashed rectangle are the disturbance which means that these pixels have the same value as the pixels in the solid rectangle and we want to filter them out.

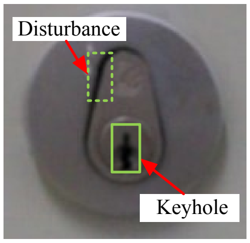


Fig. 6. The disturbance and keyhole pixels

There are many ways to weaken the influence of the disturbance. In this section, we utilize a method that contains the following three steps: opening process, gray scale conversion and binarization. Then we can get rid of the influence of the disturbance, and the resulting image can be used to compute the location and direction of the keyhole.

A. Opening Process

The opening process is a combination of Erosion and Dilation:

- Erosion[9]: with A and B as sets in z, the erosion of the image A by the structuring element B is defined by:

$$A \Theta B = \{z \mid (B)_z \subseteq A\} \tag{1}$$

  The Erosion process is like a local minimum operator. As the kernel B is scanned over the image, we replace the pixel under the anchor of the kernel by the minimum pixel value in the region overlapped by B.

- Dilation: with A and B as sets in z, the dilation of the image A by the structuring element B is defined by:

$$A \oplus B = \left\{z \middle| (B)_z \cap A \neq \varnothing\right\} \tag{2}$$

  The Dilation is the converse operation. The action of the dilation operator is equivalent to computing a local maximum over the area of the kernel B.

- Opening: Opening [9] is the dilation of the erosion of a set A by a structuring element B, as shown in Fig. 7. The opening operator is defined by:

$$A \bullet B = (A \Theta B) \oplus B \tag{3}$$

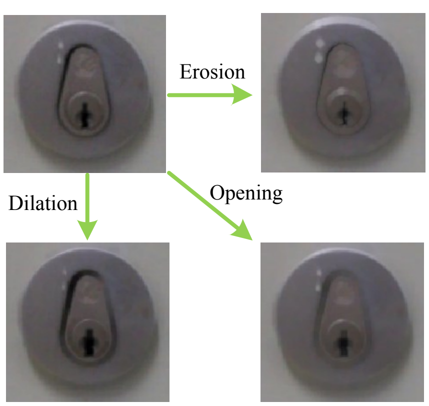


Fig. 7. The effect of three opertor to the original image

B. Image Grayscale Conversion and Binarization

In the previous step, the image is processed by the opening operator. The resulting image has black pixels only in the region of the keyhole. To further extract the location of the keyhole, we need to convert the image to a gray-scale image, as shown in Fig. 8(b), in order to reduce the dimension of the target image to provide appropriate data form for the next binarization step.

In image binarization step, by properly choosing a threshold, we convert the gray-scale image into a binary image. In the resulting binary image, the only non-zero pixels are those corresponding to the keyhole, as shown in Fig. 8(c).

(a) (b) (c)

Fig. 8. (a) The original target image. (b) Image gray processing. (c) Image binarization processing

VI. CALCULATION OF THE LOCATION AND DIRECTION

In the previous section, we process the detected image which contains the keyhole with a series of operators and actions in order to get the binary image that contains the pixels corresponding to the keyhole only. In this section, we will calculate the location and direction of the keyhole from the binary image.

A. Location Calculation

Image moment [10] is a representation of weighted average of the image pixels' intensities. It usually has some attractive property or interpretation. As the binary image obtained from the previous section only contains the pixels corresponding to the keyhole, in this section we will use image moment to calculation the center of the gravity of the binary image to get the location of the keyhole.

For the binary image, we can compute the image moment which represents the center of the gravity of the image by utilizing the pixel values corresponding to the keyhole. The image moment can be expressed using (4):

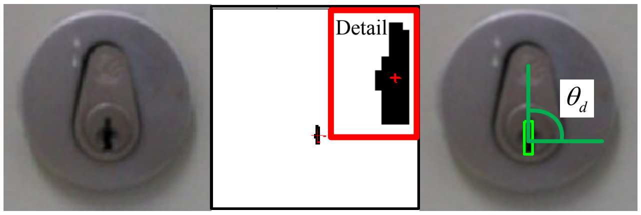
$$M_{pq} = \sum_{x=0}^{w} \sum_{y=0}^{h} x^p y^q f(x, y) \tag{4}$$

Where x, y are the coordinates of the pixel; ω, h represent the width and height of the image. p, q indicate order coefficient. $f(x, y)$is an image, and its value is continuous.

Let x_c,y_c represent the coordinates of the center of the gravity of the image. x_c,y_c can be calculated using (5):

$$\begin{aligned} x_c &= M_{10} / M_{00} \\ y_c &= M_{01} / M_{00} \end{aligned} \tag{5}$$

where M_{00} represents the area of the binary image. The result is shown in Fig. 9(b).



(a) original image (b) location of the image (c) direction of the image

Fig. 9. Location and direction of the keyhole

B. Direction Calculation

In this section, we will calculate the direction of the keyhole using two different ways.

1) PCA-based method

PCA (Principal Component Analysis) is a commonly used method of data analysis. The PCA-based method can transform the original data into a set of linearly independent representation of each dimension. It can be used to extract the main component of the original data. The method consists of the following actions.

- Extract the pixels representing the keyhole. Resize the coordinates of these pixels to a matrix of 2 by n (n represents the number of all the black pixels).
- Compute the zero-mean by subtracting each row of the matrix X by the mean of the corresponding line.
- Calculate covariance matrix
- Calculate eigenvalues of the covariance matrix and the corresponding eigenvectors.

The direction of the keyhole is the same as the direction of the eigenvector which corresponds to the maximum eigenvalue.

2) Spectrum Analisys based method

We utilize the notion of the dominant direction of the texture to calculate the direction of the keyhole [11]. The process is as follows:

- The binary image from the previous step can be viewed as a texture image. We first resize the image to a square for convenient. The width of the window is set to *M*, and the number of the samples is set to *n*.
- Use an edge detector to extract the edges of the texture image. The resulting image is denoted as $f_{M\times M}(i,j)$.
- We apply the Fourier transform to the image $f_{M\times M}(i,j)$, and get $F_M(u,v)$. Then we use this item to get the power spectrum denoted as $P_M(u,v)$:

$$P_M(u,v) = \left|F_M(u,v)\right|^2 \tag{6}$$

where $u,v = 1,2,\cdots,M$.

- The power spectrum is represented in Cartesian coordinate system, and we transform it to the Polar coordinate system to get $p(r_i,\theta_j)$, where

$$\begin{aligned} r_{i+1} - r_i &= M/(2n); \\ \theta_{j+1} - \theta_j &= \pi / n \end{aligned} \tag{7}$$

- Then we sum out the variable r to get the function of P with respect to θ as shown in (8):

$$P(\theta) = \sum_{i=1}^{n} P(r_i,\theta_j)\ (\theta = \theta_1,\theta_2,\cdots,\theta_n) \tag{8}$$

- The θ_d that corresponds to the direction of the keyhole can be compute using (9):

$$\theta_d = \arg\max_{\theta}(P_\theta) \tag{9}$$

3) Comparsion and Error Analysis of the two methods

We compare the two methods by using a series of texture images with known dominant direction. As shown in Fig. 10, the dominant direction of the texture is rotated from 0° to 180° with 1° increment at a time. In each time step, the dominant direction is calculated by the two methods. The predicted error of the two methods during this process is shown in Fig. 11.

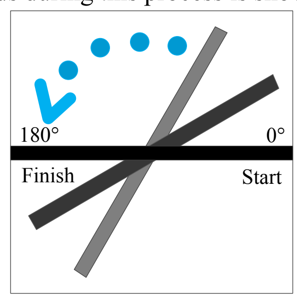


Fig. 10. Samples of the angle of rotation

With the increase of the angle of the dominant direction, the predicted error of the Spectrum based method stabilizes at 1°. The error curve shows no change except some pulses at certain angle of rotation. The predicted error of the PCA based method oscillates during this process. The mean error is -0.500 degrees and the variance are 0.117. The maximum and minimum errors are 0.019 and -1.161 degrees respectively.

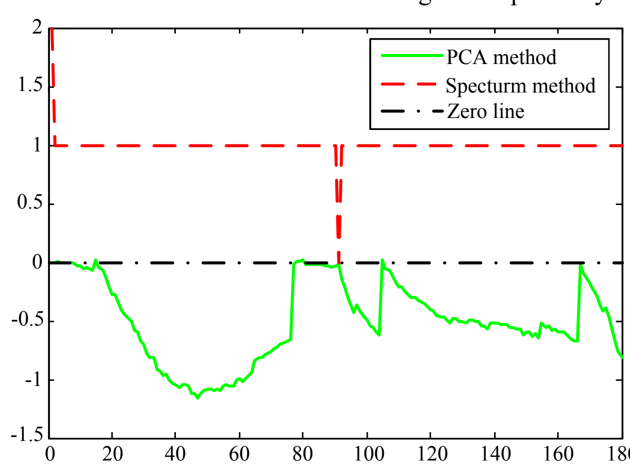


Fig. 11. Error curve of the two methods

VII. Conclusion

Door opening problem has been studied by many researchers for the past decade. Most of the research papers made the assumption that the door is unlocked and can be opened directly. We tackle the problem of opening a locked door. Specifically, we focus on the problem of the keyhole detection and parameter extraction.

A keyhole detector which consists of a SVM classifier and multi-scale Sliding Window detector is trained using HOG features to represents the keyhole. The detected image is then processed with opening operator to filter out the disturbance caused by the error in assembling. The location is calculated using image moment. The direction of the keyhole is calculated using two different methods: a PCA based method and a Spectrum based method. The two methods are compared and analyzed and we found that the Spectrum based method has a constant error about 1°.

ACKNOWLEDGMENT

This work was supported by the National Key Basic Research Development Plan Project (973) (2013CB035502), Harbin Talent Program for Distinguished Young Scholars (No. 2014RFYXJ001), Research Project of State Key Laboratory of Mechanical System and Vibration (MSV201610), Fundamental Research Funds for the Central Universities (Grant No. HIT.BRETIII.201411), and “111” Project (B07018).

REFERENCES

[1] T. Rühr , J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, “A generalized framework for opening doors and drawers in kitchen environments,” in *Robotics and Automation (ICRA)*, May 2012, pp. 3852-3858.

[2] S. Chitta, , B. Cohen, and M. Likhachev, “Planning for autonomous door opening with a mobile manipulator,” in *Robotics and Automation (ICRA)*, May 2010, pp. 1799-1806.

[3] S. Ahmad, H. Zhang, and G. Liu, “Multiple working mode control of door-opening with a mobile modular and reconfigurable robot. Mechatronics,” *IEEE/ASME Transactions on*, 2013, pp.833-844.

[4] E. Klingbeil, A. Saxena, and A.Y. Ng, “Learning to open new doors,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* ,October 2010, pp. 2751-2757.

[5] Meeussen, Wim, et al. “Autonomous door opening and plugging in with a personal robot,” in *Robotics and Automation (ICRA)*, 2010, pp. 729-736.

[6] N. Dalal, and T. Bill, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 886-893.

[7] C. Cortes, V. Vapnik, “Support-vector networks,” *Machine learning*, Sep 1995,20(3),pp.273-97.

[8] T.M. Koller, G. Gerig, G. Szekely, D. Dettwiler, “Multiscale detection of curvilinear structures in 2-D and 3-D image data,” in *Computer Vision, 1995. Proceedings., Fifth International Conference on* , Jun 1995 ,pp. 864-869.

[9] R.C. Gonzalez, and R.E. Woods, *Digital image processing*,3rd ed,2002.

[10] Hu MK. “Visual pattern recognition by moment invariants,” *information Theory, IRE Transactions on*. Feb 1962,8(2).179-87.

[11] Vaidyanathan G, Lynch PM. “Texture direction analysis using edge counts,” in *Southeastcon'89. Proceedings. Energy and Information Technologies in the Southeast*, Apr 1989, pp. 733-738.