System for retrieval of defective electrical equipment infrared images using CBIR (Content Based Image Retrieval)

Hernan Dario Benitez R.
Department of Science and Engineering of Computing
Pontificia Universidad Javeriana Cali
Cali,Colombia
hbenitez@javerianacali.edu.co

Gloria Inés Alvarez V.
Department of Science and Engineering of Computing
Pontificia Universidad Javeriana Cali
Cali,Colombia
galvarez@javerianacali.edu.co

Beatriz Eugenia Marín O.
Department of Science and Engineering of Computing
Pontificia Universidad Javeriana Cali
Cali,Colombia
bmarin@javerianacali.edu.co

Juan Carlos Martinez A.
Department of Science and Engineering of Computing
Pontificia Universidad Javeriana Cali
Cali,Colombia
juancmartinez@javerianacali.edu.co

***Abstract*— In this document we present a design proposal of a system for retrieval of defective electrical equipment infrared images using CBIR (Content Based Image Retrieval). In the first part of this document we present the problem stament and the research hypothesis. Second, we show the steps to follow in order to have an infrared image representation based on texture (Gabor filters) and spatial relationships of pixel sets. Finally, we express the process of indexing and image retrieval.**

I. Problem statement

Thermographic inspections produce large amounts of information with infrared images and reports that contain data such as environmental conditions during the inspection, inspected component working conditions, data from the operator,etc. These images and reports must be stored to do comparisons with the results of previous and future inspections generating considerable amounts of data to be analyzed. However, companies dedicated to thermographic inspections tend to store these images without following an index protocol making the defective infrared images query a time consuming task. Additionally, the predictive maintenance through thermographic inspection is carried out with visual inspection on different thermal images taken on the same equipment at different periods of time. The performance of this process strongly depends on human intervention making it sensible to diagnosis errors. From this problem the following question comes out: How to retrieve infrared images of defective electrical devices based on visual content and reduce the semantic gap in this application?. This document proposes the use of spatial-based modeling for the design and implementation of a system based on CBIR (Content Based Image Retrieval) for retrieval of defective electrical equipment infrared images. This system will support thermographer decision making and ease the index and query of defective infrared images obtained from thermographic inspections of electrical devices.

II. Research hypothesis

1. Applying dissimilarity representation to CBIR could reduce the semantic gap between low level features (texture, color, shape) and high level features such as infrared image of defective or nondefective equipment.
2. Applying CBIR to infrared image retrieval of electrical devices could lead to correct diagnosis if the these are compared to manual classification

III. Image representation

In order to get an infrared image representation we follow some of the fundamental steps in digital image processing [1]:

1. Preprocessing
2. Segmentation
3. Features extraction
4. Representation
5. Image retrieval

III-A. Preprocessing

Infrared Thermography for nondestructive testing has two approaches : active and passive. For the former it is necessary to apply heat to the material and observe with an IR camera what is the evolution of temperature. For the latter ,the application of heat is not needed since the device or process inspected (i.e electrical device) emits heat by itself. In infrared images defects are revealed because they have higher temperature than the surrounding area, however sometimes the distortions due to thermal noise, optical distortion through the thermal imaging system, and of most concern, non-uniform heating through the uneven excitation of the surface makes difficult to distinguish the defective areas from the faultless ones [2], [3].

Therefore, it is needed to preprocess thermograms in order to eliminate noise and facilitates the application of defect detection algorithms.

There are several spatial images processing techniques [4] used to smooth images or suppress noise. Among these we have the gaussian window that fullfill the criteria of well behaved (nonoscillatory) both in frequency and spatial domains since it has identical forms in these domains. In 1D these forms are:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{\frac{-x^2}{2\sigma^2}\right\} \quad (1)$$

$$F(\omega) = exp\left\{-\frac{\sigma^2\omega^2}{2}\right\} \quad (2)$$

In 2D:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

where x and y are independent variables (i.e image coordinates) and σ is the standard deviation of the gaussian window. In this work we define the gaussian window as the filter to reduce the additive noise present in infrared images.

III-B. Segmentation

In several situations when Infrared Thermography for Nondestructive Testing (TNDT) is used, the important point is whether or not a defect is present in a given part of the sample [5]. The algorithm described next makes use of some heuristics and is based on the fact that TNDT images have a limited number of spatial features. The first step is to localize the defect and then specific thresholds are found in the image to estimate the border of the defects. Each threshold is found by means of a region growing approach, which starts at the central point of a defect and stops when either an image border is hit or the number of pixels agglomerated together around the seed (central point) increases abruptly.

III-C. First part: Defect Localization

For each defect found in image , the first part of the algorithm produces localization of the corresponding hottest pixel. To limit the number of computations it is supposed that defects have at least one pixel at an intensity greater than the image Avg. With regard to this, many background pixels will be neglected since only pixels greater than Avg will be processed.

All the pixels $I(i,j)$ of the image are compare with average. If the pixel evaluated is larger than the average this pixel is included in a four - vector structure:

$vx(r)$ = j [i.e position of the pixel along the column]
$vy(r)$ = i , [i.e position of the pixel along the row]
$gl(r)$ = $I(i,j)$ [i.e gray levl of the pixel $I(I,j)$]
$lb(r)$ = label associated with the pixel $I(i,j)$

The four vector structure is sorted in a ascending order. This is done in order to adapt the algorithm to the image histogram distribution. The final step in this part consist of assigning the pixels to a given class, which is equivalent to determine what pixels belong to which defect.

The rule for assigning a pixel r_i to a given class is :
if $vx() - vx(r) > MND$ or if $vy() - vy(r) > MND$
The factor MDN means Minimum Neighbor Distance and is established by trial and error. This depends on the size of the defects necessary for detection and is independent of the image.

III-D. Second part:Defect edge estimation

At this stage of the algorithm, all the seeds corresponding to the defects are available. In this part, the gross defect shape will be estimated by growing a region around the seeds. Each seed is processed individually and one threshold per defect is established. If many defects are present in an image ,many thresholds (one per defect are obtained), the main purpose of this algorithm is to speed up the process of defect shape estimation.

In the image I,for the defect (i.e the seed) the threshold is first set:

$$Th = Th_{max} = I(i_d, j_d) \quad (4)$$

The number of neighbors $N_r(Th_{max})$ having the same gray-level value as $I(i_d, j_d)$ is computed using a recursive procedure and assuming and eight connectivity. The search is then redone with:

$$Th' = Th + 1 \quad (5)$$

Using a recursive procedure until an image border is hit. When the border is hit , the vector Nr has the number of pixels agglomerated around the seed located in i_d, j_d.

III-E. Results and discussion

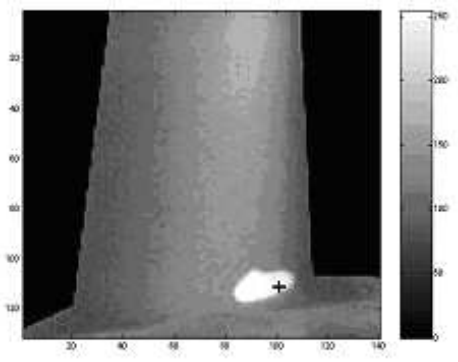

Fig. 1: Automatic defect detection algorithm applied to blade's thermogram

For figures 1 and 3 the unique defect was detected by the algorithm, for these cases the parameter MND (Minimum Neighbor Distance) is not important because there is only one

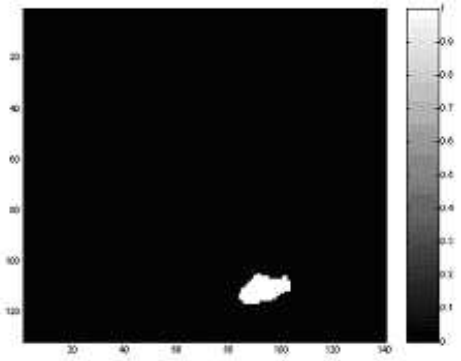

Fig. 2: Defect border detection of figure 20

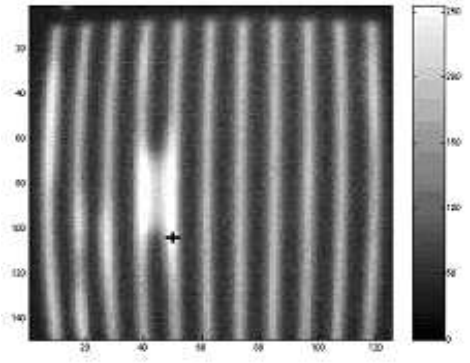

Fig. 3: Automatic defect detection algorithm applied to turbine cooling system's thermogram

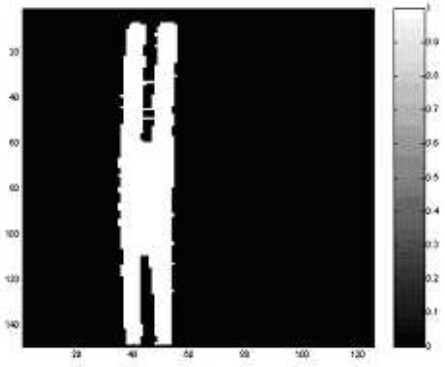

Fig. 4: Defect border detection of figure 22

class or defect in the image and therefore it is unnecessary to determine the distance between defects.

Analyzing figures 2,4 and 6 it is observed that the borders are well defined and they correspond with the defect's shape.

On the other hand, for the thermogram in figure 5 there are three defects that were found by the algorithm. Initially , a MND of 50 was chosen but with this value the algorithm wrongly detected three defects in faultless areas. When the value 3 was chosen the defects were correctly detected as well as the shape defects.

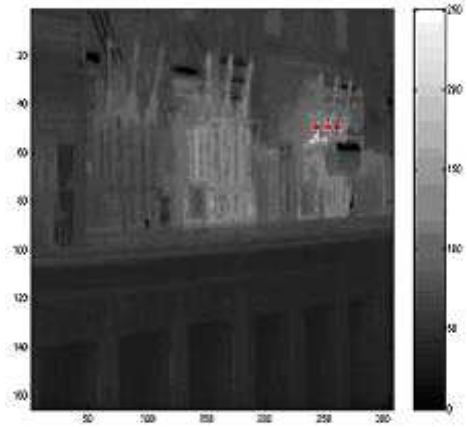

Fig. 5: Detection of defects in passive thermography application

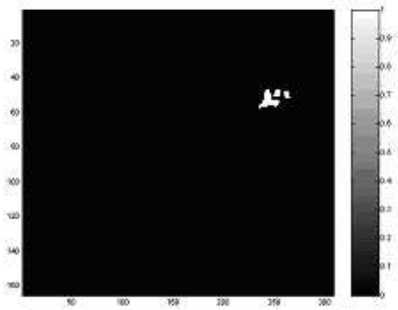

Fig. 6: Defect border detection of figure 24

III-F. Features extraction

This stage follows the segmentation stage whose output is raw pixel data that in this case constitutes all the points in the defects detected. In the feature extraction stage the data is converting in a form suitable for computer processing. The data is represented as complete region since we are interesting in a internal property of the defect: its texture. We use texture instead of color given the variability of color palettes used during the inspection that generates images with different pseudo-colors. For the description of defects detected in the segmentation stage we use Gabor filter banks [6]. Mathematically, a 2D Gabor function, g, is the product of a 2D Gaussian and a complex exponential function. The general expression is given by:

$$g_{\theta,\gamma,\sigma}(x,y) = exp\left\{\frac{x^2+y^2}{2\sigma^2}\right\} exp\left\{\frac{j\pi}{\gamma\sigma}(xsin\theta - ycos\theta)\right\} \quad (6)$$

The Gabor response obtained from Eq. 6 can emphasize basically three types of characteristics in the image: edge oriented characteristics, texture-oriented characteristics and a combination of both. In order to emphasize different types of image characteristics, we must vary the parameters σ, θ and γ of the Gabor function.

The variation of θ changes the sensitivity to edge and

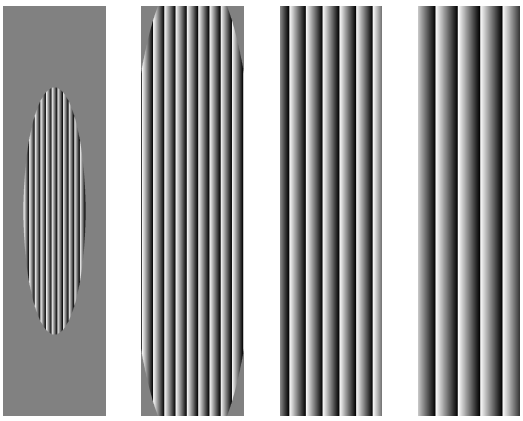

Fig. 7: $\sigma = \{4,\ 8,\ 12,\ 16\}$

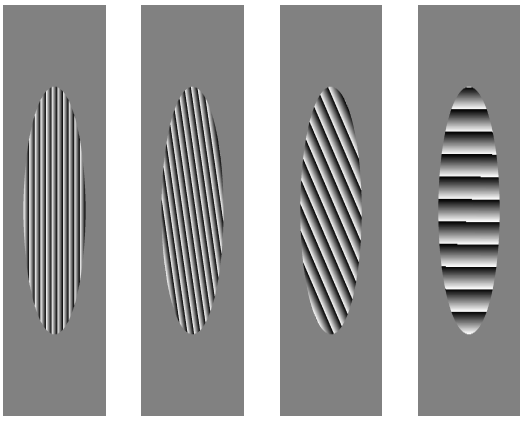

Fig. 8: $\theta = \left\{0,\ \frac{\pi}{6},\ \frac{\pi}{3},\ \frac{\pi}{2}\right\}$

texture orientations. The variation of σ will change the *scale* at which we are viewing the world, and the variation of the sensitivity to high/low frequencies. We would like to find the most adequate combinations of σ, θ and γ and to represent particular parts of objects for recognition/detection tasks. According to a statistical study developed in [6] the most sensitive parameter is σ since represents the shape factor of the Gaussian surface: this determines the greater or less selectivity of the filter in the spatial domain. In figures we illustrate the variation of parameters σ, θ and γ in the shape of the Gabor function. The features to represent defects will be chosen by a backward feature selection from a larger set of Gabor filters with different smoothing, frequency and direction parameters [7], [8].

IV. Spatial Relations Representation

An image is composed by entities spatially related. Once we have decided how to represent entities it is necessary to

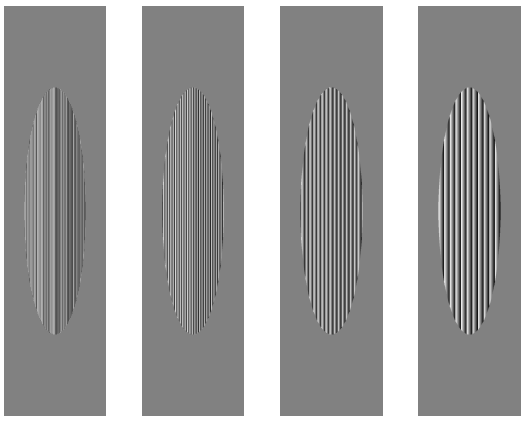

Fig. 9: $\gamma = \left\{\frac{1}{2},\ \frac{3}{2},\ \frac{5}{2},\ \frac{7}{2},\right\}$

choose a representation for spatial relations. Some criteria for this selection are: representation should be tolerant to translations and rotations, it should include a distance metric to be able of decide if two spacial relations are alike or not, and finally representation should be as efficient as possible in computational terms.

The representation proposed for each spatial relation is based on weighted walkthroughs [9] which is tolerant to translations and rotations, it includes a distance metric, and it is efficient in space and affordable in time. Assuming two entities A and B, each one represented by a cloud of dots, their spatial relation is expressed by a matrix M of size 3 $\times$ 3, which counts in each position the number of dots of both images with a given spacial relationship between them. In Figure 10 image A is represented by a rounded rectangle and image B by a triangle.

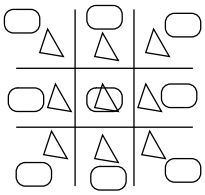

Fig. 10: Spatial relationships reflected in the 3×3 matrix

The meaning of the values $M[i,j]$ into the matrix is as follows:

$$i = \begin{cases} -1 & \text{if } x_B < x_A \\ 0 & \text{if } x_B = x_A \\ 1 & \text{if } x_B > x_A \end{cases}$$

$$j = \begin{cases} -1 & \text{if } y_B < y_A \\ 0 & \text{if } y_B = y_A \\ 1 & \text{if } y_B > y_A \end{cases}$$

The next example is taken from [9]:

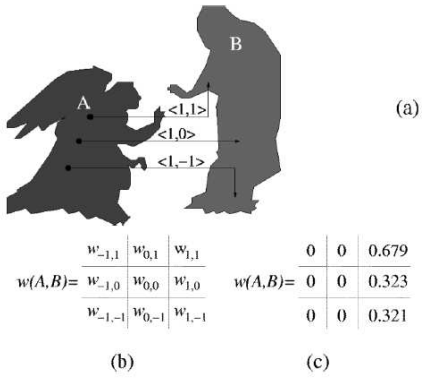

Fig. 11: Example of use of the 3×3 matrix

Provided that each couple of entities into the same image have been related by one of theses matrices, next step is to calculate the distance between two spatial relationships. The distance suggested by Berreti et. al. is:

$$\begin{aligned} D_s(w,w') &= \lambda_H d_H(w,w') + \lambda_V d_V(w,w') + \lambda_D d_D(w,w') + \\ &\quad \lambda_{H_0} d_{H_0}(w,w') + \lambda_{V_0} d_{V_0}(w,w') + \lambda_{00} d_{00}(w,w') \end{aligned}$$

Where all λ's coefficients are nonnegative numbers with their sum equal to one. And terms $d_H, d_V, d_D, d_{H_0}, d_{V_0}$ y , d_{00} are calculated as follows:

$$\begin{aligned} d_H(w,w') &= |(w_{1,1} + w_{1,-1}) - (w'_{1,1} + w'_{1,-1})| \\ d_V(w,w') &= |(w_{-1,1} + w_{1,1}) - (w'_{-1,1} + w'_{1,1})| \\ d_D(w,w') &= |(w_{-1,-1} + w_{1,1}) - (w'_{-1,-1} + w'_{1,1})| \\ d_{V_0}(w,w') &= |(w_{-1,0} + w_{1,0}) - (w'_{-1,0} + w'_{1,0})| \\ d_{H_0}(w,w') &= |(w_{0,-1} + w_{0,1}) - (w'_{0,-1} + w'_{0,1})| \\ d_{00}(w,w') &= |(w_{0,0} - w'_{0,0})| \end{aligned}$$

Image representation consists on a mixture of entities and spatial relationship representations, and distance between two images can be established as follows:

$$\mu^{\Gamma}(Q,D) = \lambda \mu_a^{\Gamma}(Q,D) + (1-\lambda)\mu_a^{\Gamma}(Q,D)$$

Where

$$\begin{aligned} \mu_a^{\Gamma}(Q,D) &= \sum_{k=1}^{N_q} D_a(q_k, \Gamma(q_k)) \\ \mu_s^{\Gamma}(Q,D) &= \sum_{k=1}^{N_q} \sum_{h=1}^{k-1} D_s([q_k,q_h],[\Gamma(q_k),\Gamma(q_h)]) \end{aligned}$$

Where λ is a nonnegative number in the range $[0,1]$ and distances D_a, D_s have been previously described.

V. Image Retrieval

In an image retrieval system each image into the database must be represented by a graph as have been explained in previous sections that is, a pair (E,R) where E is a vector of entity representations and R is a vector of spatial relation representations. When a query image is presented to the system, it calculates the representation for such new image, a pair (E',R'), and proceeds to compare this image against database ones obtaining their distances.

In this system will be used A^* algorithm to do this matching [10]. Algorithm 1 shows the main tasks to be developed to retrieve images based on content. There is an important issue related with this comparison, it is the problem of stablishing which entity of image Q matches with which entity of image D. In graph theory, this problem corresponds to a subset isomorphism problem and it is NP-hard, so backtracking is an alternative to get the solution but paying a high computational cost. In this first year of the project, system will use A^* algorithm to do this matching.

Algorithm 1 $CBIR(Q, BD, k)$

```
// Q is the query image
// BD is the image data base
// k is the threshold to accept to images as similar
(E_Q, R_Q) = getRepresentation(Q)
for image = (E_BD, R_BD) in BD do
    Γ = getInterpretation(E_Q, E_BD)
    d_e = getEntitiesDistance(Γ, E_Q, E_BD)
    d_s = getRelationsDistance(Γ, R_Q, R_BD)
    d = getImageDistance(d_e, d_s)
    data = data ∪ {(id(Q), d)}
end for
sortedData = sortDescendent(data)
answer = getFirst(sortedData)
while getDistante(answer) < k do
    print getId(answer)
    answer = getNextData(sortedData)
end while
```

References

[1] Rafael Gonzalez and Richard Woods. *Digital image processing*. Prentice Hall, 2001.

[2] C. Ibarra-Castanedo, D. Gonzalez, M. Klein, M. Pilla, S. Vallerand, and X. Maldague. Infrared image processing and data analysis. *Infrared Physics and Technology*, 46(1-2):75–83, 2004.

[3] N. Rajic. *Noise in infrared thermography, In: NONDESTRUCTIVE TESTING HANDBOOK, vol 3, Infrared and Thermal Testing X. Maldague and Patrick O. Moore (eds.).* American Society for Nondestructive Testing, 2001.

[4] J. Russ. *The image processing handbook*. CRC Press, 2002.

[5] X. Maldague. *Theory and Practice of Infrared Technology for Nondestructive Testing*. Wiley Interscience, New York, 2001.

[6] F. Bianconia and A. Fernández. Evaluation of the effects of gabor filter parameters on texture classification. *Pattern recognition*, 40:3325–3335, 2007.

[7] Lai.C, D. TAX, R Duin, E. Pekalska, and P. Paclick. A study on combining image representation for image classification and retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(5):867–890, 2004.

[8] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier, third edition, 2006.

[9] S. Berretti and A. Del Bimbo E. Vicario. Weighted walkthroughs between extended entities for retrieval by spatial arrangement. *IEEE TRANSACTIONS ON MULTIMEDIA*, 5(1):52–70, March 2003.

[10] S. Berretti, A. Del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 23(10):1089–1105, 2001.