

Automatic Analysis of Radiographic Images for Non Destructive Test Applications

M.K. Felisberto, T.M. Centeno, L.V.R. Arruda, H.S. Lopes*

*Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI
Centro Federal de Educação Tecnológica do Paraná – CEFET-PR
Av. Sete de Setembro n° 3165, Curitiba, PR, Brazil
mkf, mezzadri, arruda, hslopes@cpgei.cefetpr.br

ABSTRACT

Radiographic inspection is a reliable non-destructive test, widely used for integrity evaluation of structures and equipments. Nowadays, high quality images with very accurate resolutions have been supported by modern digital radiographic systems. However, the image analysis for internal defect detection and geometric characterization is still a not totally automated task. The main reason is that image analysis is usually a very complex task, which involves heuristic decisions based on experiences, as object detection and recognition. For that reason, a new automatic radiographic image analysis system was developed in order to identify important components or component parts, which must be inspected separately, as weld joints, pipe walls, pipe wall thicknesses, valves and mechanical parts. The developed methodology involves the use of a genetic algorithm search to find desirable patterns on the image. Image indexing procedures are used for a final verification process. As a result, the system offers quick and correct answers and also flexibility to be applied in others applications.

1. Introduction

Radiographic inspection is a reliable non-destructive test, widely used for integrity evaluation of structures and equipments. [1] Nowadays, high quality images with very accurate resolutions have been supported by modern digital radiographic systems [2]. On this way, many efforts have been done in order to automate pipeline inspection tasks, through digital radiographic image analysis [3]. Some approaches consider the automation of specific defect detection on weld joints and pipeline [1,2]. However, the identification of the components that must to be inspected usually is a task performed by humans [4].

The present work is concerned with the design and construction of an object recognizing system, for radiographic image analysis applications, in order to automatically identify components for inspection. Such system can be used to automatically find weld joints, pipe walls, and many others components on pipeline radiographies, to support complete automation of radiographic inspection tasks on the industry.

By the proposed approach, image processing techniques are used to extract properties from an image model to construct an object model. A genetic algorithm was implemented to manage the search for the object model on an image input to the system. If the search result is correct, a final verification process validates the system response. As a result, the system offers quick and correct answers and also flexibility to be applied in others applications.

The next section will describe the radiographic image analysis problem and related works. Following, it will be described the applied methodology and the system implementation. After that, the test and results are presented. And, finally, the paper concludes with a brief discussion about the reached results and future works are considered.

2. The Problem Characterization and Related Works

On the present work, the identification of specific components on radiographic images is treated as an object verification problem, a particular image analysis case. The goal is to find where and how an object model appears on the input image. On agreement with Jain *et al.* [5], matching procedures as template matching, morphological approaches and analogical methods,

offers feasible solutions for the problem. However, it becomes hard to implement a quickly searching algorithm when it is necessary to consider that such object can appear in different rotational angles and scale factors, in high resolution images. For the traditional verification methods, such problem becomes an exhaustive search [5].

Fitzpatrick, Grefenstette, and Van Gucht [6, 7] faced a similar problem, when they implemented a system for medical x-ray images comparison in order to identify dye-coated region into the artery, after the dye injection. The problem, however, was to align the both image for comparison through images subtraction. The solution was to implement a genetic algorithm to find the best image transformation parameters, in order to align the pre-injection and post-injection images to compare themselves.

An example using genetic algorithms for image segmentation (term used for distinguishing important parts of an image [8]) and object recognition is the Bhanu and Peng [8] approach for adaptive image segmentation. Although it is not a radiographic image analysis example, such method contributed for the development of the present work. The Bhanu and Peng [9] method uses a genetic algorithm to find an appropriate set of parameters for an edge-detection algorithm. The set of parameters are evaluated based on an object recognition system performance.

An example of an artificial intelligence technique applied to radiographic weld joints image segmentation was the Lawson and Parker [10] approach. They implemented an artificial neural network for intelligent weld joint segmentation. However, such approach works better for weld joint images with less variation of the weld position.

Other image segmentation techniques have been used by different authors, for example: Lashkia [11] uses fuzzy logic for weld defect detection, Aoki and Suga [12] use the region growing for weld segmentation, Shafeek *et al.* [13] use a thresholding method and edge-detection for the same purpose.

In fact many effective segmentation techniques have been successfully applied on weld joint segmentation and defect detection on radiographic image. The problem however, is that, for the most, it is supposed that always a regular weld joint radiography will be used as the system input image. Otherwise, a window, enclosing the target, has to be defined manually [14].

Pipe wall thickness measurement for corrosion detection is another image analysis application where it is necessary to identify components on the radiography. On the Zscherpel *et. al.* [15] system, for example, the user draws the profile plot interactively with the mouse across the wall of the pipe. After that, the pipe wall thickness is measured by the system.

Currently, references about automatic weld defect evaluation systems [1, 3, 10-14] or corrosion detection and pipe wall thickness measurements [2, 15-17] usually present inspection systems and approaches that works on circumstances where the position and orientation of the inspecting component is trivial on the input radiography; otherwise, such information needs to be provided by the system user. Therefore, an automatic system for detection of the components to be inspected is necessary to allow the automatic inspection of complex radiographic images with multiple components, as shown on figure 1.

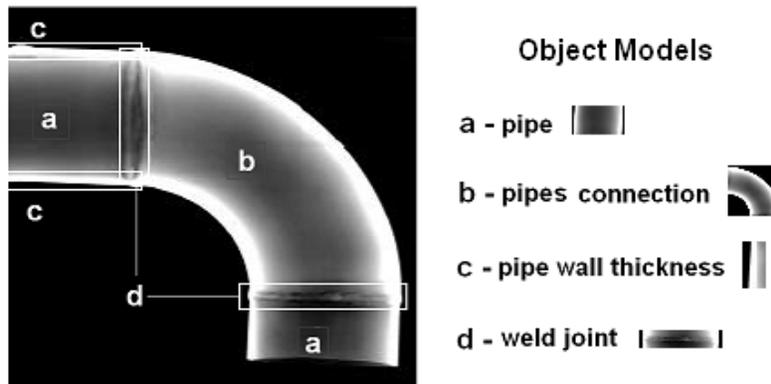


Figure 1: An example of digital radiographic image with many components to be inspected.

3. The Methodologies

Before implementing a genetic algorithm for object identification on radiographic images, it is necessary to model the problem in such way the genetic algorithm theory will be easy or feasible to be applied. For the actual problem it means to define how an object model will be represented, what will be the search objective, how a feasible solution will be represented, and how each solution will be evaluated. Such considerations and definitions are the main subjects of this section.

3.1 Object Model Representation

This sub-section intends to describe the two object model representation techniques used by this work on different circumstances. The procedures to generate the first object representation technique are described by the following steps:

1. The image is sliced by “n” horizontal strength line segments equally spaced by the distance “dy” (where $dy = \text{number of image lines} / (n+1)$, in pixels).
2. The image is sliced by “n” vertical strength line segments equally spaced by the distance “dx” (where $dx = \text{number of image columns} / (n+1)$, in pixels).
3. The strength lines crossing points are denominated points of reference, which are denoted by “P_{ij}”, where $i = 0, 1, \dots, n-1$, and $j = 0, 1, \dots, n-1$.
4. The point $P_{(n-1)/2 (n-1)/2}$ is denominated the central point of reference (“P₀”).
5. The function denoted by “f(P_{ij})” assigns to each point of reference (P_{ij}) the media value of the pixels on the “P_{ij}” neighborhood, as defined by the delimited region shown on the figure 2.
6. All the “f(P_{ij})” values are normalized through the [0 99] interval of integer numbers and achieved on a (n x n) matrix structure, called matrix of reference (“M_{ref}”).

The matrix of reference (“M_{ref}”), together with the distances “dx” and “dy”, complete the object model representation. The “n” value will be defined on the system implementation.

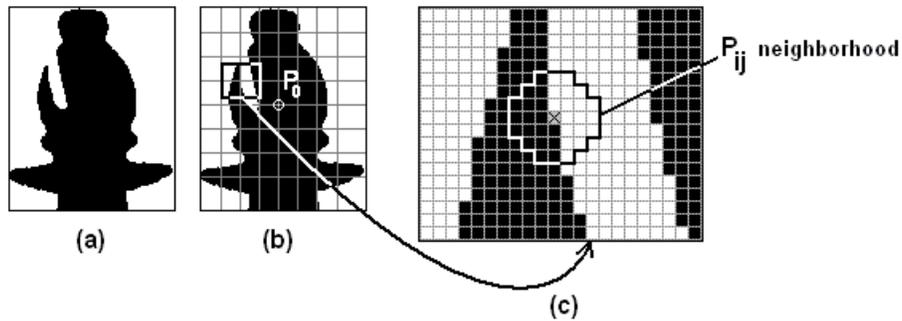


Figure 2: (a) An object model image; (b) The object model image sliced by horizontal and vertical strength lines, and the central point of reference identification; (c) The “P_{ij}” neighborhood definition.

Another solution for object (image) representation comes from image indexing techniques, and it was proposed by Rudek [18]. Basically, the image of a model is divided in sub-regions and a value, based on the pixels gray level distribution, is calculated to each sub-region. The results are grouped on a vector (called behavior vector) and the sequence of the vector values components is used as an index to access the archived models (or images). This technique is very useful for fast image indexing/retrieving and its use on object representation can offers some important advantages. The behavior vector is color (gray level) sensible [19] and does not offers restrictions about the objects shapes. Besides, it has already been used successfully on object recognition applications [4, 20].

The 5 following steps describe the behavior vector construction to the image:

1. Divide the entire image in $l \times c$ blocks ('l' lines and 'c' columns)
2. For each block_{ij} (where i and j correspond the respective line and column of the block), do the steps 3 until 5, following the sequence: block₁₁, block₁₂, ..., block_{1c}, block₂₁, block₂₂, ... block_{2c}, ..., block_{lc}.
3. Count the number of pixels belonging to each range of gray levels. The table 1 shows the range value for each range of gray levels, considering 16 gray levels ranges (number of gray level ranges: Nr = 16).
4. Determine the predominant range of gray levels on the block and its respective range value (in case of equality, takes the maximum range value)
5. Assign to the current block the value of the predominant range of gray levels

The block values will correspond to the behavior vector elements on the same sequence the values were calculated.

Table 1: The values for 16 gray levels ranges [18]

range value	range of gray levels	range value	range of gray levels
00	-	08	119-135
01	000-016	09	136-152
02	017-033	A	153-169
03	034-050	B	170-186
04	051-067	C	187-203
05	068-084	D	204-220
06	085-101	E	221-237
07	102-118	F	238-255

Compared to the first proposed methodology for object representation, the behavior vector main disadvantage is the computational efforts. The previous method does not request the entire image sweeping, since just the points of reference neighborhoods are read. Such aspect is a strong argument to justify the use of the first method on genetic algorithm applications [6, 7, 21], since it will be necessary to repeat such process thousands of times during the genetic search. On the other hand, a more accurate object representation, as the behavior vector, is desirable for a final verification. So, the behavior vector will be used to represent the objects (images) during the image comparison procedures on the final verification, to validate the genetic search result.

3.2 Genetic Algorithms

Genetic algorithms are search and optimization techniques based on the mechanics of natural selection and natural genetics. An initial population of created individuals originates subsequent generations through mechanisms based on natural reproduction, crossover and mutation. The survival of the individual species is governed by principles, which are based on the Darwinian model of the natural selection. Such mechanisms preserve the genetic material of the best individuals through the next generations. As a result, it is expected that best individuals will predominate the last generations [21].

Mathematically, each individual is a string of zeros and ones that corresponds to a feasible solution for a mathematical problem. At the beginning, an initial set of strings (*population*) is generated with random values. Then, each string value is tested, and a function (*fitness function*) assign to each string a qualitative value (*fitness value*) that represent how good is such solution. The *fitness function* must to be on agreement with the main objective of the search, which is defined by the *objective function*. For the actual problem, such concepts will be defined on the following sub-sections [21].

3.2.1 The Individual Representation

To plot the points of reference of an object model on an input image, 6 parameters are necessary: the “dx” and “dy” distances, the scale factor (“s”), the rotation angle (“θ”), and the pair of coordinates (x0'; y0') of the central point of reference P0', which is the projection of P0 on the input image. Since “dx” and “dy” is defined on the object model representation, just 4 parameters should be generated to complete a possible solution, called individual “k”: (x0'_k ; y0'_k ; s_k ; θ_k). The use of (') means that the variable belongs to the input image domain instead of the model image.

To originate the initial population of individuals, a number “z” (population size) of individuals is generated with random values varying as follows:

$$0 \leq x0'_k \leq (\text{columns number of the input image} - 1) \quad (01)$$

$$0 \leq y0'_k \leq (\text{lines number of the input image} - 1) \quad (02)$$

$$0.5 \leq s_k \leq 2.0 \quad (03)$$

$$0 \leq \theta_k \leq 2\pi \text{ rad} \quad (04)$$

For the actual problem it will not be used images with more that 2047 lines or columns. Actually, 2^{11} bits will be enough to represent each P0' coordinate: (x0'; y0'). If the same number of bits were used for the other parameters “s” and “θ”, each individual will be composed of 4 strings of 2^{11} bit each one.

3.2.2 The Objective and the Fitness Functions

For any solution $(x0'_k ; y0'_k ; s_k ; \theta_k)$, the coordinates of any points of reference (P_{ij}') on the input image is given by following equations for translation and rotation [references]:

$$xi' = x0' + s [(xi - x0) \cos \theta + (yi - y0) \sin \theta] \quad (05)$$

$$yi' = y0' + s [(yi - y0) \cos \theta - (xi - x0) \sin \theta], \quad (06)$$

Here, xi and yi are the coordinates of the point P_{ij} in relation to $P0$ – on the object model image – and the xi' and yi' values are the coordinates of the point P_{ij}' – the projection of the point P_{ij} on the input image. For the central point of reference $P0'$, the equations (05) and (06) give $P0' = (x0' ; y0')$.

Supposing that the individual (23; 311; 6.28; 1) has been generated as a possible solution for the search of the object model (fig. 01), the points of reference would be located as shown on figure 3a, for that input image. To better visualize the result, the figure 3b shows the object model projection on the input image in agreement with the proposed solution. Also in the figure 3a, it is shown that, for some generated solutions, it is possible that some points of reference falls out of the image limits. Such points are called invalid points, and the $f(P_{ij}')$ values are denoted by “*”.

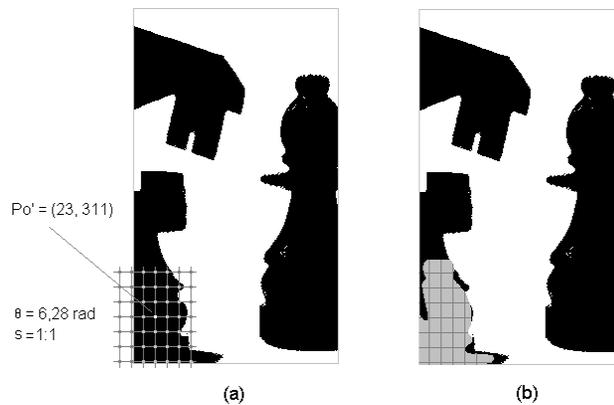


Figure 3: (a) The points of reference plotted on an input image, for the individual (23; 311; 6.28; 1); (b) The object model projection on the input image in agreement with the proposed solution.

Since the points of reference has been located, a new matrix of reference can be generated for the proposed solution, by following the steps 5 and 6, described on the sub-section 3.1. Such matrix is denoted by “ $M_{ref}' (x0'_k ; y0'_k ; s_k ; \theta_k)$ ” for the individual “k”.

To measure the similarities between $M_{ref}' (x0'_k ; y0'_k ; s_k ; \theta_k)$ and M_{ref} , the expression (07), based on the square error sum was used:

$$S_{SQE} (k) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} g(((M_{ref} - M_{ref}'(k)) \times (M_{ref} - M_{ref}'(k))^T)[i][j]), \quad (07)$$

Where:

For (i, j) , such as $f(P_{ij}') = “*”$: $g(M[i][j]) = 0$, otherwise: $g(M[i][j]) = M[i][j]$.

The less the $S_{SQE} (k)$ sum is, better is the proposed solution k . So, the *objective function* can be defined as the search for a solution $k = (x0'_k ; y0'_k ; s_k ; \theta_k)$, such as k minimize the $S_{SQE} (k)$ value:

$$\arg \min (k) = S_{SQE} (k) \quad (08)$$

Consequently, the *fitness function* will be defined in agreement with the *objective function*, by the following equation:

$$\text{fit} (x0'_k ; y0'_k ; s_k ; \theta_k) = (S_{SQE-MAX} - S_{SQE} (x0'_k ; y0'_k ; s_k ; \theta_k)) / S_{SQE-MAX} , \quad (09)$$

Where:

$$S_{SQE-MAX} = (99 (n - n^*))^2 \text{ is the maximum } S_{SQE} \text{ value,} \quad (10)$$

$n \times n$ are the M_{ref} dimensions, and

n^* is the number of invalid points of reference.

Since $S_{SQE-MAX} \geq S_{SQE}(k)$ for any feasible solution k , the fitness function values will be defined on the interval $[0; 1]$:

$$0 \leq \text{fit}(k) \leq 1 \quad (11)$$

A restriction to the maximum number of invalid points was incorporated to the fitness function definition to limit the maximum number of n^* occurrences. So, the fitness function must to be redefined, as follow:

$$\begin{aligned} \text{If } n^* > (n \times n)/2 : \text{fit}_{WR}(k) &= \text{fit}(k), \\ \text{Otherwise: } \text{fit}_{WR}(k) &= 0. \end{aligned} \quad (12)$$

The “WR” index on “ fit_{WR} ” means “with restriction”.

3.2.3 The Genetic Search Working

The genetic search starts with the generation of a set of individuals (feasible solutions) that will originate the initial population of “ z ” individuals: $(x0'_1; y0'_1; s_1; \theta_1)_{G1}, (x0'_2; y0'_2; s_2; \theta_2)_{G1}, \dots, (x0'_z; y0'_z; s_z; \theta_z)_{G1}$. Where “ z ” is the population size and the index “ $G1$ ” means 1st generation.

Each individual of the initial population is evaluated through the fitness function. The probability of each individual to be selected, from the population, increases with a high fitness value. An appropriated method for selection [21, 22] can be used to select candidates for *crossover* and *mutation* [21]. Such genetic operators will generate new individuals to form a new population. Some population generating strategies include *elitism*, which means to *clone* the fitness individuals of the last population and insert then on the next one. Basically the same procedures are used to generate the next populations until some stop condition be satisfied. Usually a maximum number of generations or a minimum fitness value reached is used as stop condition [21].

A great variety of selection methods and population generating strategies can be found on genetic algorithms books and publications. To better understanding the mechanism behind such genetic operators, it is indicated the Goldberg text book [21], and [22] for extended selection methods.

3.3 The System Implementation

The figure 4 presents a block diagram that illustrates how the information flows through the system components, which are described as follow:

- If testing new genetic algorithm parameters is desirable, the default parameters can be changed manually in the *parameters up-dating* block.
- The *image processing I* block applies operators to the object model image in order to construct the object model representation (M_{ref}, dx, dy), as explained on the section 3.1 (object model representation).
- The *image processing II* block applies operators to the input image in order to enhance image visual aspects. It uses an image expansion algorithm and a histogram equalization to improve the input image contrast.
- The *genetic search* block uses the genetic algorithm resources to find a region of the input image, which contends the supposed object model. Such region is saved as new image, labeled “object image i ”. Where, the “ i ” index indicates the number of object copies found until the moment.
- The “object image i ” is verified on the *final verification* block in comparison with the object model image. The behavior vector is constructed for each image and compared, one to the other. The sum of the absolute errors is used to evaluate if the found solution is acceptable or not. In case it is not accepted, it means that no more copies of the same object can be found on the input image, and the search is finished. Otherwise, the “object image i ” is saved, “ i ” is updated to “ $i+1$ ”, and the correspondent region is extracted from the original input image, which is feed-backed to the genetic search block. And a new search is started.

This closed loop finished when no more valid copies of the same object model can be found on the input image.

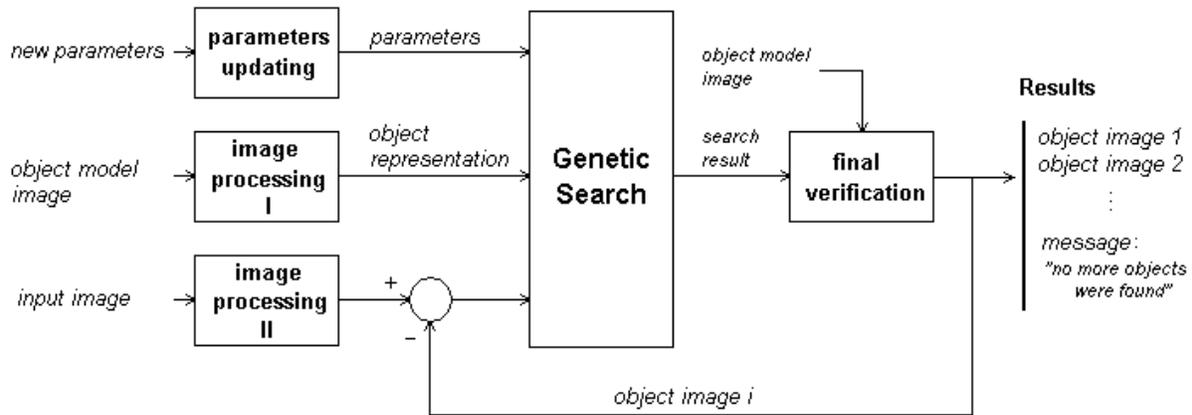


Figure 4: Blocks diagram showing the system components and the information flow.

The system was implemented on C++ object oriented programming language on windows platform. For the image processing routines it was used the Dilabien 6.11 package [23], and for the genetic algorithm implementation it was used the GALib genetic algorithm package [24].

4. Tests and Results

For the first tests series, it was used binary images of chess peaces. An image having an object alone was used as image model. Others images, where the same object appears on different orientation and position, as well as multiobject scenes and partial occlusion occurrences, were tested. Different parameters, as population size (z), maximum number of generations (g), crossover probability (pc), mutation probability (pm), as well as different types of genetic selection methods were tested.

The best performance was reach using the following parameters, which were kept for the others tests series:

- Genetic Algorithm Parameters: $z=100$, $g=500$, $pc = 0.9$, $pm = 0.2$, Tournament Selection ($k=2$), elitism.
- Object Model Representation (based on the matrix of reference) Parameters: $n = 15$.
- Object Model Representation (behavior vector) Parameters: $c = 20$, $l = 20$.

The number of supposed solution to be analyzed by the algorithm on each search is $z \times g = 100 \times 500 = 50\,000 = 5 \times 10^4$. The number of possible solution, considering an image of 512×512 pixels, 720 possible values for the rotational angle, and 100 scale factors, the search space size would be 1.89×10^{10} . So, instead to sweep the entire search space in an exhaustive searching, the algorithm finds an acceptable solution spending much less computational efforts.

For the second test series it was used 20 weld joints radiographic images and 100% of the weld joints were found by the system. The figure 5b shows some results for the searching of object the model shown on the figure 5a. Note that, the genetic algorithm selected the region 5 as a weld joint candidate; but the final verification process rejected it.

Other radiographic images, with different components, as pipe, pipe connections and valves were tested. And the system was able to find almost all of the desirable objects. The image 6 shows the reached results of a valve searching, on two different radiographies.

5. Conclusions

A system for automatic components identification on radiographic images was design and constructed. The system performance on the tests shown that the proposed method is appropriated to find specific objects, and check their correct position, orientation and scale on the image. As a result, the identified objects can be inspected separately through other methods.

By now, it is desirable to attach an image indexing/retrieving algorithm to the system in order to manage a database of object model images. The main objective, however, is to implement inspection routines to inspect the objects found by the system.

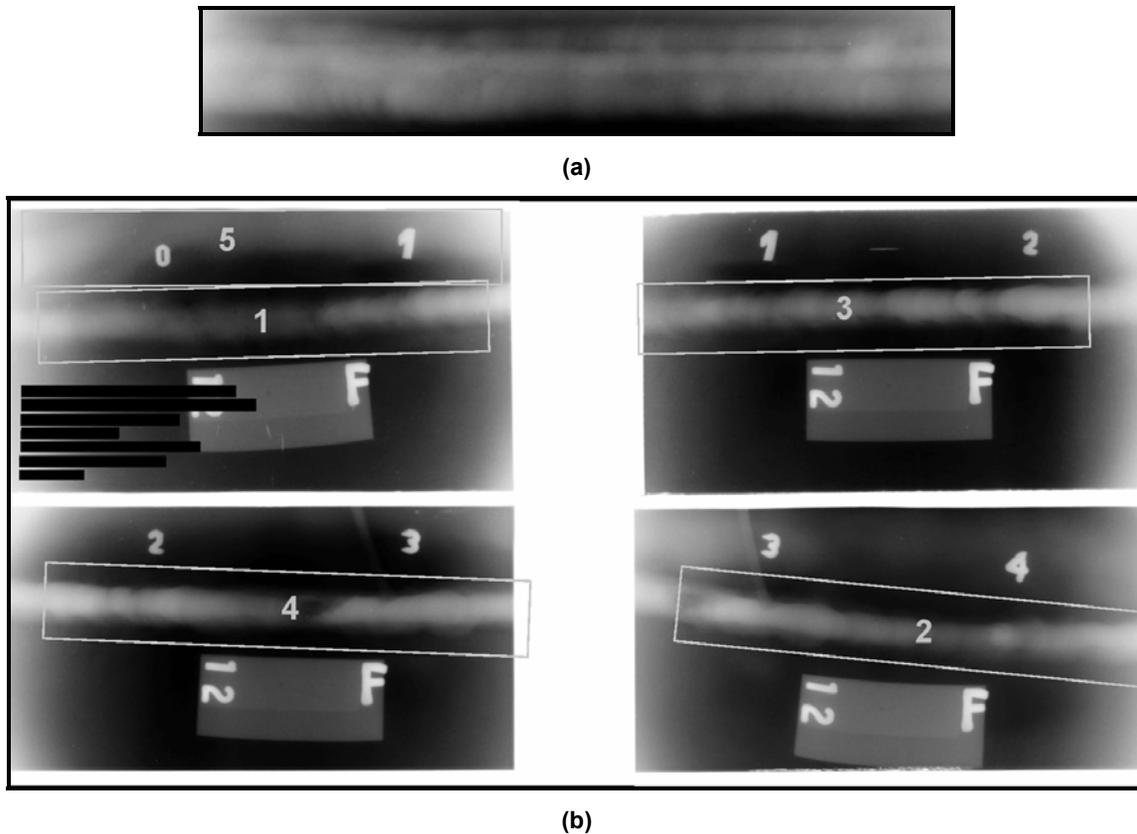


Figure 5: (a) The weld joint image model; (b) Five regions were identified as weld joint on the input image.

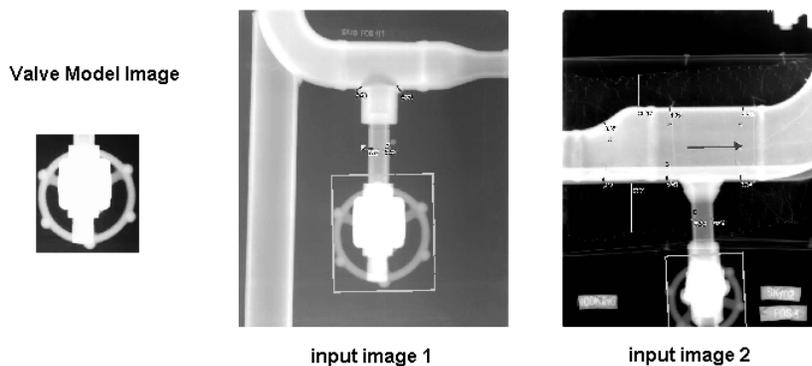


Figure 6: identification of a pipeline equipment in two radiographic image, based on an object model

6. Acknowledges

This work has been partially supported by Agência Nacional do Petróleo (ANP) and Financiadora de Estudos e Projetos (FINEP) - ANP/MCT (PRH10-CEFET-PR).

The weld joints radiographic images for the tests were provided by ARCTest – Technical Services of Industrial Inspection and Supporting, from Paulínea, SP, Brazil.

The software for this work used the GALib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology, and the Dilabien 6.11 image processing package, written by Jefferson Osowsky at the Centro Federal de Educação Tecnológica do Paraná.

7. References

- [1] WANG, G., LIAO, T. W. Automatic Identification of Different Types of Welding Defects in Radiographic Images NDT&E International, vol. 35 , pp.519-528, May 2002.
- [2] GIBSON, J., WYSNEWSKI, D. Inspection of Pipe for Erosion/Corrosion Using Digital Radiography AGFA Technical Papers, 2002. <http://ndt.agfa.com/bu/ndt/index.nsf/EN/rvproductliterature.htm>
- [3] DA SILVA, R. R., SIQUEIRA, M. H. S., CALOBA, L. P., REBELLO, J. M. A. Radiographic Pattern Recognition of Welding Defects Using Linear Classifiers INSIGHT, vol. 43 No. 10, pp. 635 – 637, October 2001.
- [4] FELISBERTO, M.K., CENTENO, T.M., ARRUDA, L.V.R. An Object Recognizing System for Industrial Applications. 17th International Congress of Mechanical Engineering, São Paulo, 2003.
- [5] JAIN, R., KASTURI, R., SCHUNCK, B. G. Machine Vision New York: McGraw-Hill and MIT Press, 1995.
- [6] FITZPATRICK, J.M., GREFENSTETTE, J.J., VAN-GUCHT, D. Image registration by genetic search, Proceedings of Southeastcon 84, Louisville, KY, 460-464, Apr 1984
- [7] GREFENSTETTE, J.J., FITZPATRICK, J.M. "Genetic search with approximate function evaluations," Proc. Intl. Conf. on Genetic Algorithms and their Applications, Pittsburgh, PA, 112-120, Jul. 1985.
- [8] ROSENFELD, A. From Image Analysis to Computer Vision: An Annotated Bibliography, 1955-1979 Computer Vision and Image Understanding. Academic Press, n. 84, p.298-324, Seattle, 2001.
- [9] BHANU, B., PENG, J. Adaptive Integrated Image Segmentation and Object Recognition IEEE Transactions on Systems, Man and Cybernetics, 30(4): 427-441, November 2000.
- [10] LAWSON, S. E., PARKER, G. A. Intelligent Segmentation of Industrial Radiographic Images Using Neural Networks Machine Vision Applications, Architectures and Systems Integration III, Proc of SPIE, vol. 2347, pp. 245-255, 1994.
- [11] LASHKIA, V. Defect Detection in X-ray Images Using Fuzzy Reasoning Image and Vision Computing. , New York, Elsevier, n. 19, p.261-269, 2001
- [12] AOKI, K., SUGA, Y. Intelligent Image Processing for Abstraction and Discrimination of Defect Image in Radiographic Film The International Society of Offshore and Polar Engineers (ISOP-98), Vol.4, pp.527-531, Honolulu, May 1997.
- [13] SHAFEEK, H. I., GADELMAWLA, E.S., ABDEL-SHAFY, A. A. & ELEWA, I. M. Assessment of welding defects for gas pipeline radiographs. NDT&E International, Oct 2003.
- [14] SHAFEEK, H. I., GADELMAWLA, E.S., ABDEL-SHAFY, A. A. & ELEWA, I. M. Automatic inspection of gas pipeline welding defects using an expert system. NDT&E International, Oct 2003.
- [15] ZSCHERPEL, U., ONEL, Y., EWERT, U. New Concepts for Corrosion Inspection of Pipelines by Digital Industrial Radiology (DIR). Proceedings of the 15th World Conference on Nondestructive Testing, Roma (Italy) Oct 2000
- [16] ONEL, Y., EWERT, U., WILLEMS, P. Radiographic Wall Thickness Measurement of Pipes by a New Tomographic Algorithm. Proceedings of the 15th World Conference on Nondestructive Testing, Roma (Italy) Oct 2000
- [17] REDOUANE D., YACINE K., AMAL A., FARID A., AMAR B. Evaluation of Corroded Pipelines Wall Thickness Using Image Processing in Industrial Radiography. Proceedings of the 15th World Conference on Nondestructive Testing, Roma (Italy) Oct 2000
- [18] RUDEK, M. Uma Proposta para Indexação e Recuperação Automática de Imagens e Reconhecimento de Cheques Bancários Baseadas no Vetor de Comportamento, Dissertação de Mestrado, CPGEI CEFET-PR. Curitiba: 1998.
- [19] RUDEK, M., COELHO, L. S., CANGIOLIERI JR., O., Visão Computacional Aplicada a Sistemas Produtivos: Fundamentos e Estudo de Caso, XXI Encontro Nacional de Engenharia de Produção - 2001, Salvador: 2001.
- [20] CENTENO, T. M., BAGATELLI, R. C. Indexing Techniques applied in Machine Vision for Object Recognition, Irish Machine Vision & Image Processing Conference. Belfast: ago. 2000.
- [21] GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning Addison-Wesley, 1989.
- [22] BÄCK, T., HOFFMEISTER, F. Extended selection mechanisms in genetic algorithms. In: Proceedings of IV International Conference on Genetic Algorithms - ICGA, San Diego, USA, p. 92-99, 1991.
- [23] OSOWSKY, J. DILabien Image Processing Package vs. 6.11, 2004
- [24] WALL, M. GALib A C++ Library of Genetic Algorithm Components vs. 2.4.5, 2003, <http://lancet.mit.edu/ga/>