RESTITUTION AUTOMATION FOR CLOSE-RANGE APPLICATIONS

Valanis A. [a], Georgopoulos A. [b]
[a] Student, of Photogrammetry, National Technical University of Athens, Greece – artvalanis@yahoo.gr
[b] Prof., Dept. of Photogrammetry, National Technical University of Athens, Greece – drag@central.ntua.gr

KEY WORDS: Automation, Restitution, Algorithms, Segmentation, Close-Range, Image, Application, Facades

ABSTRACT:

The most popular way of obtaining vector products from a single digital image is manual digitisation. Although this process gives reliable results of high accuracy, it is often characterized as laborious and therefore time-consuming. Systematic efforts have been made towards the automation of the restitution process, by employing digital image processing techniques ([2], [3]). However, to the best of our knowledge, the existing bibliography is rather poor and most of the methods developed are not close-range oriented and therefore give results that need a great deal of correction and human intervention. The basic aim of this paper is to exploit existing segmentation algorithms and methods with suitable addition and modification, in order to automate the restitution process. The method presented in this paper successfully attempts to automate a great part of the restitution process by combining adaptive thresholding techniques with segmentation and morphological processes and has successfully been applied to Byzantine monuments. All of the algorithms discussed were developed in the Matlab environment and applied using a fully functional interface. The user can work either with a semi-automated version or with the fully automated version of the process. The interface also provides the user with the capability to interact with the results and apply corrections where necessary. It also gives the opportunity to work with large (100MB), or even color, images. The proposed method has been put to test and the results were compared to a fully manual digitisation. The evaluation of this comparison is presented and discussed.

1. INTRODUCTION

The problem of automating the restitution process is highly complicated. Within the framework of this research, several image processing techniques were applied in order to solve the particular problem, but none of them gave a satisfying solution. This is mainly due to the fact that changes, for example, in the resolution of the images being used or in the rendering scale of the final products, can completely alter the course of action that should be taken. Another reason is the existence of a variety of structures. A great deal of the methods proposed so far can only be applied when a priori knowledge of the object itself is available, or when the object of interest is thoroughly examined. Another major drawback of those methods is that they suffer from a lack of information due to the fact that they rely on the use of ill-defined hard thresholds that may lead to wrong decisions [4]. The method proposed in this paper is free of such restrictions, as the user himself chooses the way he finds more fit to operate. There are no hard-set thresholds and the user defines any parameters needed for the operation of the basic algorithm. Everything is applied from within a fully functional interface, which gives the user the ability to work either in semi-automated or fully-automated mode, interact with the results and apply corrections where necessary. Another advantage is that the method can also be applied on colour images i.e. the user can choose the image to be processed among the components of the RGB or the HSI colour space. Additionally, there is a series of filters that can be used to pre-process the image, in order to obtain better results.

In the following the basic algorithm is firstly described. The basic algorithm is mainly a region-growing technique that has been adapted to solve such problems. Secondly the automation of the process is presented. If the method is applied in the semi-automated mode, the user has to give a sample for every single object that must be extracted. If the method is applied in the fully automated mode, the selection of the samples is done automatically. An extensive series of trial applications follows and the results are discussed thoroughly in the sections that follow.

2. THE BASIC ALGORITHM

The basic algorithm is, as already mentioned, a region-growing technique. The method described here is applied when the user works in the semi-automated mode. In the beginning, the user has to choose a confidence level for the algorithm to operate and give the maximum number of iterations, in order to define the area where the algorithm is permitted to operate. Afterwards, the user must give a sample i.e. two points that define a rectangle area that should belong in the interior of the object to be extracted. All the pixels that belong to the sample are given the value 1 in the binary image, where the results of the region growing process are temporarily stored. All other pixels are initially given the value 0 in this binary image.

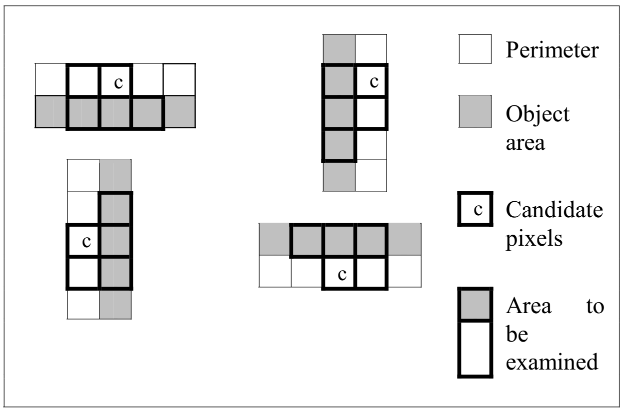

Figure 1. Regions of pixels to be examined according to the relative position of the candidate pixel and the examined area

From that point on, the algorithm examines the surrounding pixels in order to find which ones should be included in the area of the object. In order to decide whether a pixel should be included in the pixels that belong to the object or not, certain pixels that belong to the surrounding area of the candidate are used. Depending on whether the candidate belongs on the upper, right, lower or left side of the exterior perimeter of the area that has already been examined, the pixels that are examined by the algorithm are shown in Figure (1).

Using the pixels of the sample, the arithmetic mean (m_s) and standard deviation (s_s) of the gray values are calculated. In order to find which pixels belong to the area of the object, there are two criteria. The first one is the criterion of connectivity. The pixel, which is being examined, is initially given the value 1 in the binary image. If at least two of the other pixels belonging to the currently examined region also have the value 1, then the connectivity criterion is met, in which case the homogeneity criterion is also checked. Otherwise, if the connectivity criterion is not met, the next pixel is examined. In order for the homogeneity criterion to be checked, the mean gray value (m_r) of the pixels of the currently examined region, which in the binary image have the value 1, is calculated. If Equation (1) is satisfied, then the homogeneity criterion is met and the candidate pixel is included to the pixels of the area of the object. Otherwise, the next pixel is examined.

$$m_s - z \cdot s_s \leq m_r \leq m_s + z \cdot s_s \qquad (1)$$

where m_s = the arithmetic mean of the gray values of the pixels of the sample
s_s = the standard deviation of the gray values of the pixels of the sample
m_r = the arithmetic mean of the gray values of the pixels of the currently examined region, that in the binary image have the value 1
z = a value which depends on the chosen level of confidence

The algorithm stops when, during a pass, no pixels are added to the area of the object, or when the maximum number of iterations is reached.

The typical region-growing algorithms tend to introduce some problems. These methods make use of relatively large neighbourhoods in order to obtain sufficient information to decide whether or not a pixel should be aggregated into a region. Consequently, the region approach tends to sacrifice resolution and detail in the image to gain a sample large enough for the calculation of useful statistics for local properties. This can result in segmentation errors at the boundaries of the regions, and in a failure to distinguish regions that would be small in comparison with the block size used. Furthermore reasonable initial seed points and stopping criteria are often difficult to choose in the absence of a priori information [4]. All of these problems are dealt with by the proposed method. The sample is given by the user and is therefore in most cases representative of the area of the object. Additionally, due to the way the sample is chosen and the homogeneity criterion that has been formed, there is no loss of detail and the boundaries are detected as accurately as possible. Furthermore, the fact that the user gives the parameters required for the operation of the algorithm, makes a priori information unnecessary. It must be noted that the parameters that the user chooses, i.e. the confidence level and the number of iterations, are usually fit for quite a few objects belonging to an area of an object.

In Figure 2 an example is given in order to show the results of the algorithm. The first picture illustrates the sample given by the user in order to detect the object of interest. In the second picture, the results of the region-growing process are illustrated.

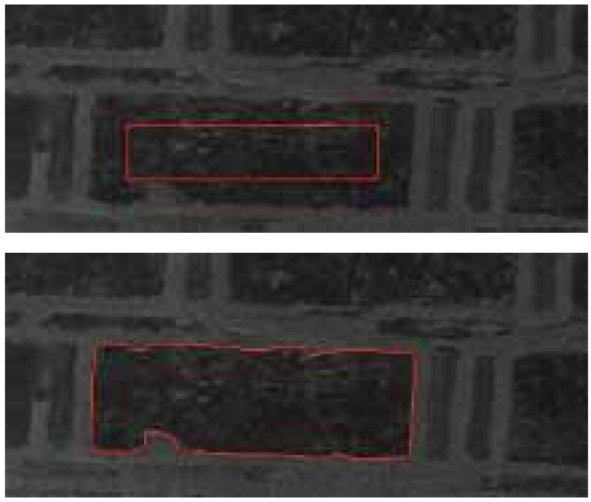

Figure 2. The upper part illustrates the sample that the user gave in order to detect the borders of the particular object. The lower part illustrates the boundaries that were detected by the proposed method.

3. AUTOMATION OF THE PROCESS

The semi-automated method gives very satisfying results, accelerates the restitution process and, if appropriate attention is given, the corrections that must be applied in the end are relatively few. However, automating a greater part of the process is always a challenge. Therefore, a method was developed in order to automate the sampling stage. This is achieved by combining an adaptive thresholding technique with morphological processing. This process yields an improved binary image, which is later used for the definition of the samples. In the beginning, the user selects an area of the image, within which objects will be detected. The user also gives a sample of the objects, which should be detected. For this sample, the arithmetic mean (m_s) and standard deviation (s_s) of the gray values is calculated. Given a selected level of confidence, it is assumed that the gray values of the pixels of the objects lie within the interval $[m_s - z \cdot s_s, m_s + z \cdot s_s]$ and that all other pixels belong to the background. In this way, the distributions of the object gray values and the background gray values are approximated. Using this information, an adaptive thresholding method is applied [1]. The thresholding process yields a binary image, where pixels that belong to the objects are given the value 1. This binary image is processed in order to find clusters that correspond to actual objects. At this stage, the binary image is firstly smoothed with an average filter (3x3) and the pixels that still have the value 1 are retained. All other pixels are given the value 0. The holes in the image are filled, and the smoothing process is repeated with an average filter (5x5). Again, the pixels whose value is 1 are retained. In this way, most of the small clusters are removed.

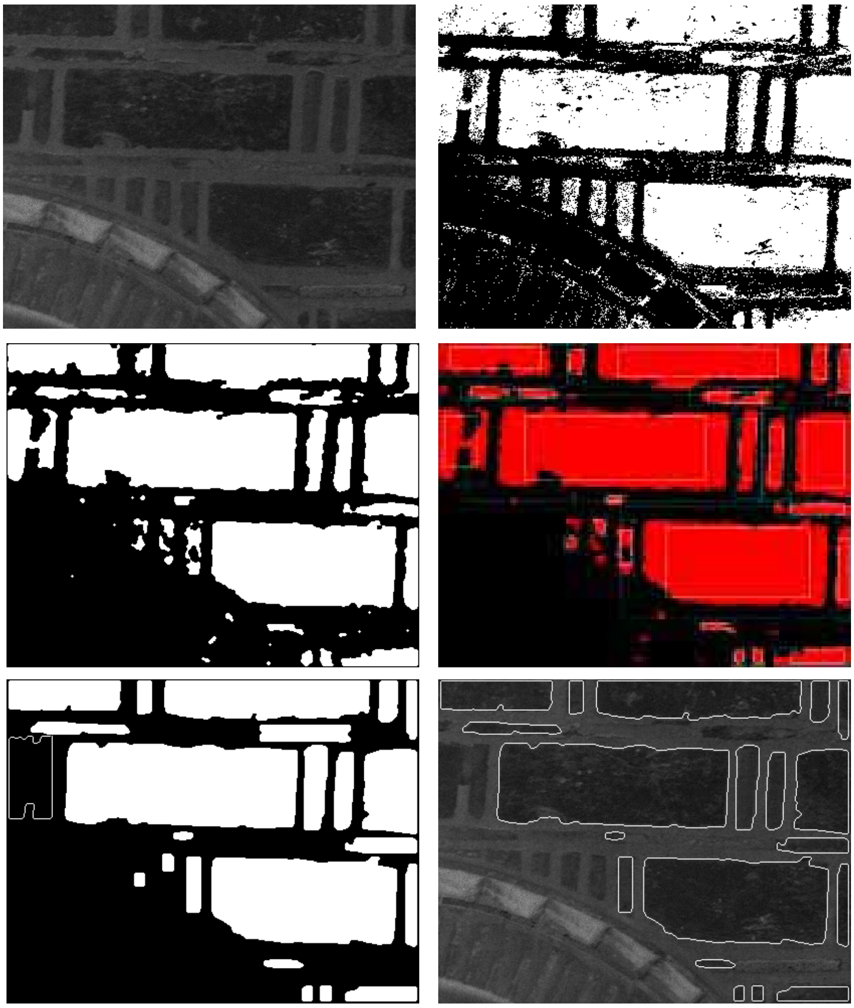

Figure 3. Upper section: Left, the original gray image and right, the binary image that the thresholding process yields. Middle section: Left, the morphologically processed image and right, the image of the samples. Lower section: Left, the objects that were detected by the automated process and right, the borders of the segments that were selected to be saved, overlaid on the original image for comparison.

Afterwards, by successively applying a set of morphological processes, the segments are refined and their compactness is improved. The processes applied at this stage are: cleaning, skeletonization, thinning, spurring (x10), cleaning, thickening (x2) and majority (x10). In Figure 3, the original gray image, the image that is yielded by the thresholding process and the improved version of the same image are presented.

After the binary image that the thresholding process yields is improved, the samples are defined. At this stage, using a labeling process, each one of the initial segments is identified and its dimensions (width and height) are calculated. Additionally, the centroid of each segment is determined. Using this information, a rectangle is defined as a sample for each segment. Furthermore, another larger rectangle is defined, in order to determine the area where the region-growing algorithm will be permitted to operate for each one of the objects. All this information is stored in a matrix and used by the region-growing algorithm. The region-growing algorithm is applied as described in the previous section. The only difference is that instead of using the maximum number of iterations as a stopping criterion, the information, which was collected during the automated sampling stage, is used. The region-growing algorithm is applied for each one of the samples, and every new segment is stored in a binary image.

In the end, this binary image is presented to the user. In order for the user to decide which of the segments have successfully been detected and store them, another image is created. This is the original image, where the borders of the detected segments are overlaid. In this way, the user can select which segments he wishes to save, and does so by simply clicking somewhere in their interior in the binary image.

Everything described in this and the previous section, is applied from within a fully functional interface, which was also created within the Matlab environment. The final product of this process is a binary image where all of the detected segments are stored. Vector products can later be obtained by applying a raster to vector conversion, using another software such as Corel OCR-trace. Finally, the vector file can be imported in a CAD software, such as AutoCAD, where further corrections may be applied.

It must be noted that the results of the fully automated method are still not considered to be satisfactory. The final products rendered in this case need a great deal of correction and therefore an extensive application of the fully automated method is not presented, as in the case of the semi-automated method. For the time being, efforts are being made so as to improve the performance of the automated process and a comparison to the semi-automated one should definitely be presented in the future. Nevertheless, it was considered worthwhile to make an initial theoretical presentation in order to demonstrate the basic idea of the automated algorithm.

4. AN EXTENSIVE TEST APPLICATION

In order for the described theoretical research considerations to be put to practical use, it was decided to perform an extensive test application and assess the results.

The object of the application is a digital photomosaic of the eastern façade of the Daphni Monastery (Figure 4). The Daphni Monastery is considered to be one of the finest and most important specimens of Byzantine architecture and art. The stonework of the façades is highly complex, mostly composed by stones, which are surrounded by bricks.

Using the developed method, the stones of the façade were restituted. For the restitution process, the semi-automated method was applied, as the fully automated method is still in an experimental phase. Using the semi-automated method, 687 stones of the façade were detected in 5 hours (approximately 130 items detected per hour). The task of the restitution of this particular object is considered to be difficult, even when applying manual digitization. This is due to the complexity of the object, the similarity between the appearance of the stones and the joints, and the existence of shadows.

The original image was a colour image of 104MB. For most of the stones, the detection process was carried out using the saturation component of the image. However, for some objects that were difficult to identify from the saturation component, the intensity component of the HSI colour space was used. In other cases, where the objects of interest could not be identified in either of those two images (i.e. the saturation and the intensity component of the original image), the components of the RGB system were used.

As far as the parameters that should be defined are concerned, if there was enough contrast between the object and the background, the confidence level chosen was 95% or greater. If there was little difference between the object and the background, the confidence level chosen was 90% or smaller. The maximum number of iterations was chosen depending on the objects to be identified; some typical values for this parameter are 10, 20 or 30, depending on the size of the object to be identified and the contrast between the objects and the background.

After the objects of interest were detected, a binary image such as the one that is presented in Figure 4 was created. This image was imported in the Corel OCR-trace environment and a raster to vector conversion was applied. The method chosen for the conversion was the trace-by-outline method. The parameters chosen for the conversion were Node Reduction = 0 and Noise Reduction =0, in order to retain all the detail of the original binary image. The results of the vectorization were exported in a .dxf file.

The dxf file that was created was imported in the AutoCAD environment, where overall corrections were applied. Initially, the vectorized borders of the detected objects were converted to closed polylines. Afterwards, the drawing cleanup processes were applied. Considering that the rendering scale of the final products would be 1:25, the desired tolerance is 6.25mm in object coordinates. Defining this value for the tolerance, the following cleanup processes were successively applied: a) Break crossing objects, b) Extend undershoots, c) Snap clustered nodes, d) Erase short objects and Dissolve pseudonodes, and e) Delete duplicates. Finally, the corrected polylines were simplified by selecting only the Simplify linear objects option and using the same tolerance i.e. 6.25 mm. The results were stored in a dwg file and were later used for the evaluation of the process.

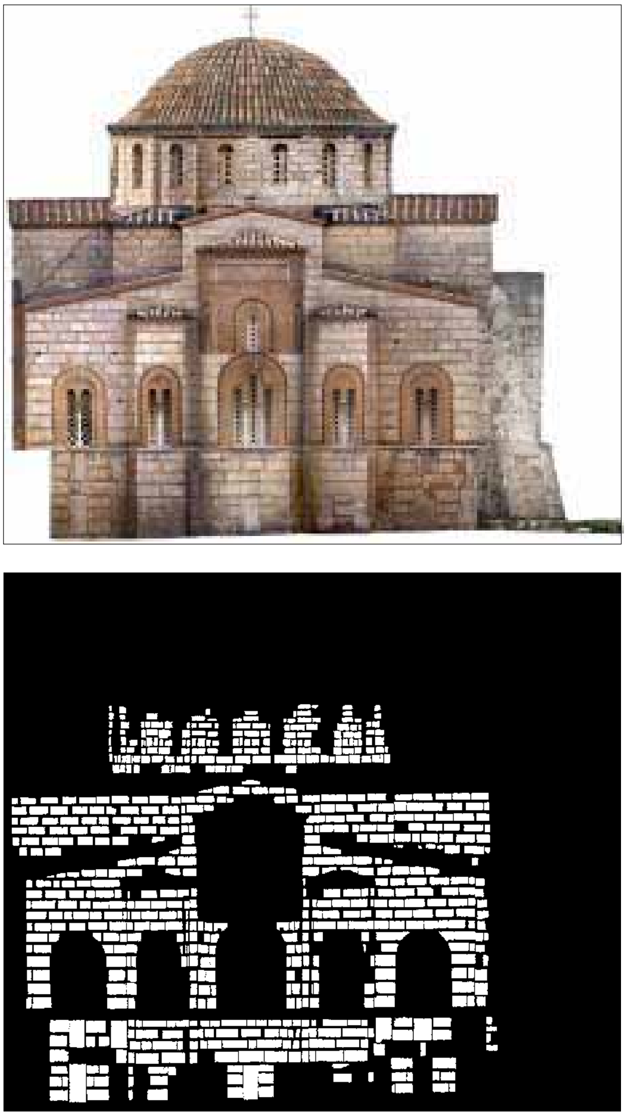

Figure 4. Up: Colour photomosaic that was used for the application. Down: Binary image where all of the detected objects appear.

It must be noted that some of the objects were impossible to detect, due to poor contrast. Additionally, as far as the dome and the decorated with bricks areas are concerned, only experimental applications of the method took place and the results were not included in the final product. The objects in these areas were either very small or presented a great lack of homogeneity, and therefore their detection was poor. However,

Figure 5. Samples used for the evaluation of the results of the proposed method. Left, part of the drum that supports the dome. Right, part of the stonework of the middle-leveled area of the façade

presenting all the details of the applications that were made exceeds by far the scope of this paper. The following section is concerned with the evaluation of the application. The accuracy of the final products is estimated and the results are discussed.

5. EVALUATION AND DISCUSSION

In order to evaluate the results of the proposed method, a manually created line drawing of the object was used. The corrected vectorized borders of the detected objects were imported in a new layer in the project of the manually created line drawing.

For the estimation of the accuracy, two samples were used. The first sample illustrates part of the area of the drum that supports the dome. The other sample illustrates part of the stonework of the middle-leveled area of the façade. In Figure 5 these samples are illustrated. It is noted that the borders of the objects that were detected by the proposed method are overlaid on those images.

For each of the samples, four images were created. The first image contained the borders of the objects that were detected by the proposed method. In order to assess the accuracy of the detection, the manually created line drawing was used. Assuming that the manually digitized borders of the objects were correct, the authors created buffer zones from both sides of those borders. In the first image, the width of the buffer zones was such, that all of the points lying within $\pm\sigma$ from the correct borders are covered i.e. $\pm\sigma = \pm 0.25$mm on the printed document or $\pm\sigma = \pm 6.25$mm in object coordinates for a restitution scale of 1:25. In the second image, the width of the buffer zones was $\pm 2\sigma$ from the correct borders, and in the third image the width of the buffer zones was $\pm 3\sigma$ from the correct borders. For each sample, by comparing the image of the detected borders with the images of the buffer zones, the accuracy of the results was assessed. Initially, all of the images were imported into the Matlab environment and the pixels that corresponded to the detected borders were counted. Afterwards, using logical operators, the pixels that belonged to the detected borders and also lied within the buffer zone, were detected and counted.

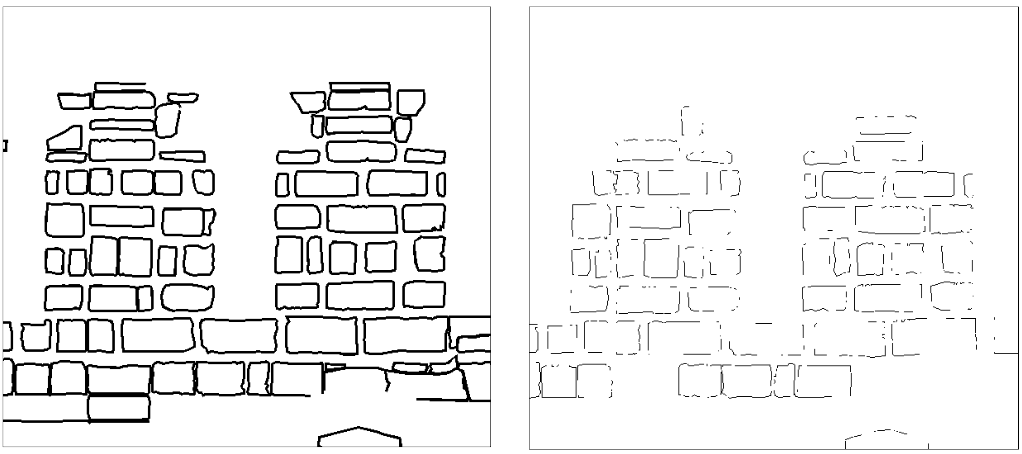

Figure 6. Images used for the estimation of the accuracy of the process for the case of the sample of the drum. Left, image which illustrates the buffer zones of $\pm\sigma$ tolerance. Right, the linear segments of the detected borders that lie within $\pm\sigma$ from the correct borders

Distance from the correct borders	**Sample 1 (11559 pixels)**		**Sample 2 (8485 pixels)**		**Total (20044 pixels)**	
(mm)	pixels	percentage	pixels	percentage	pixels	percentage
d ≤ ±σ	8020	69%	5371	63%	13391	67%
±σ < d ≤ ±2σ	2251	20%	1909	23%	4160	21%
±2σ < d ≤ ±3σ	740	6%	747	9%	1487	7%
d> ±3σ	548	5%	458	5%	1006	5%

Table 1. Estimation of the accuracy of the proposed process. The first sample is the middle-level stonework image whereas the second sample is the drum image

This process was repeated for all of the buffer zone images and for both of the samples. In Figure 6 an example of the use of the buffer zone images is set. The image on the left is the buffer image that corresponds to ± σ tolerance for the case of the sample of the drum. The image on the right presents the linear segments of the detected borders that lie within ± σ from the correct borders.

Table 1 shows the results of the estimation process. For each one of the samples the results are expressed both in pixels and percentages; additionally, an overall estimation is obtained by calculating the corresponding weighted averages for both of those samples. As indicated by the results, which are presented in Table 1, the accuracy of the detection was within the ±σ for the greatest part of the borders i.e. 67%. About 21% of the detected borders lie within ±σ < d ≤ ±2σ i.e. approximately 88% of the detected borders lie within ±2σ from the manually digitized borders.

In this experiment, the authors used the proposed method to detect the borders of the stones that are illustrated in this particular façade of the Daphni Monastery. During this process it was possible to collect 687 objects out of a total of 735 (i.e. 93%) in 5 hours. Taking into consideration the results of the statistical estimation, 67% of the semi-automatically collected objects i.e. approximately 60% of the whole, satisfy the accuracy specifications for this particular application. This means that there is another 40%of the objects remaining that must be collected with another method e.g. manually. In order to collect all of the objects of interest manually, the estimated time of work is about 30 hours. Consequently, collecting the rest 40% of the objects should take about 12 hours. Taking all of the above into account, using the semi-automated method and then working manually to complete the project takes about 17 instead of 30 hours. This means that the project is completed 1.7 times faster than it would, if everything were collected manually. However, it must be noted that the demand in accuracy for this particular project is very high, because of the large rendering scale (1:25). For a smaller scale, such as 1:50, 82% of the results of the method are satisfactory, and the estimated time of manual work required in order to complete the project is reduced to approximately 5.5 hours (2.9 times faster). Finally, for even smaller scales e.g. 1:100, manual work in order to complete the project is almost unnecessary.

6. CONCLUSIONS

In this paper, a complete approach in order to automate a part of the restitution process was presented. Several image processing techniques, such as image enhancement methods (both in the spatial and the frequency domain), edge detection, image classification etc. have been investigated and discarded, mainly due to the fact that their successful application is possible only when a priori knowledge of the properties of the object of interest are well known. This very reason renders such methods highly inappropriate for such complex objects as the Daphni Monastery. The semi-automated method presented in this contribution, proved to be flexible and therefore was successfully applied for the case of the Daphni Monastery. The problems that arose are taken under consideration so as to improve the semi-automated method and to develop the fully automated method. Furthermore, the method is currently being tested for other kinds of monuments, such as neoclassic and ancient ones. The basic algorithm in its current form examines fairly simple properties of an object so as to detect its borders. Nevertheless, the extension of the algorithm so as to investigate more properties of higher complexity, such as texture, might also be promising, in order for the method to be applicable in the case of other kinds of monuments too. However, the proposed method is considered to be a good basis for further research on the field of automating the restitution process.

REFERENCES

[1]Gonzalez R., Woods R. *Digital Image Processing*, Second edition, Prentice Hall, USA, 2002

[2] Kalenderidis K., Alexiou N. *Edge extraction from digital images*, Diploma Thesis, Laboratory of photogrammetry, NTUA, Athens, 1997, (In Greek)

[3]Karantzalos K. *Edge detection in digital images*, Diploma Thesis, Laboratory of photogrammetry, NTUA, Athens, 2000, (In Greek)

[4]Muñoz X., Freixenet J., Cufí X., and Martí J. *Strategies for image segmentation combining region and boundary information*, Pattern Recognition Letters, Vol.24, Issues 1-3, Pages 375-392, 2003

[5]Valanis A., *Automation of the restitution process for close range applications using photogrammetric image products*, Diploma Thesis, Laboratory of photogrammetry, NTUA, Athens, 2003, (In Greek)