

VOTING-BASED IMAGE SEGMENTATION

Costin-Anton Boiangiu¹

Radu Ioanitorescu²

ABSTRACT

When it comes to image segmentation, there is no single technique that can provide the best possible result for any type of image. Therefore, based on different approaches, numerous algorithms have been developed so far and each has its upsides and downsides, depending on the input data. This paper proposes a voting method that tries to merge different results of some well-known image segmentation algorithms into a relevant output, aimed to be, as frequently as possible, better than any of the independent ones previously computed.

KEYWORDS: image segmentation, machine vision, voting, cluster identification, image processing

1. INTRODUCTION

In computer vision, segmentation refers to the process of partitioning an image into multiple sets of pixels based on similarities. This is useful when it comes to extracting specific data, for example, because it makes the image easier to analyze. Segmentation is used for locating particular elements or boundaries (lines, curves etc.) in the image.

The result of the image segmentation process is a set of collections of pixels – or clusters of pixels – that covers the entire image. Each of the pixels in one cluster is similar to the others in the same cluster regarding a certain characteristic or property, such as color or intensity, and based on the same criteria, adjacent clusters are significantly different.

This paper presents a method for segmenting an image using a voting technique that merges independent results of several known algorithms into a single final output, aimed to be, in as many cases as possible, better than any of the others in particular.

2. RELATED WORK

The number of image segmentation techniques has grown rapidly in the last decades and more and more fields benefit from its advantages. Having a wide range of applications, segmentation techniques tend to be specific and cannot be applied in a wide range of cases. This creates a burden on the user that has to select the right segmentation algorithm for a given task which is quite hard considering how complex the algorithms are and their sometimes chaotic behavior on some specific sample inputs. One such image segmentation application example would be for medical data sets [1], [2], which has to address problems like low image contrast, quality of input and high variance of input cases.

¹ Associate Professor PhD Eng., "Politehnica" University of Bucharest Romania, 060042 Bucharest, costin.boiangiu@cs.pub.ro

² Engineer, European Patent Office (EPO) Germany, Bayerstrasse 34, Munich, rioanitorescu@epo.org

A large number of papers [3-12] have been published with the purpose of evaluating the state of the art in the image segmentation domain and trying to provide explanations on what are the best choices for a given of image characteristics. However, this documentation includes only a limited number of segmentation algorithms, which are hard to evaluate and fine-tune for every image specific.

In [13] a generic evaluation framework for image segmentation is presented. Ana Fred proposes [14] a majority voting combination of clustering algorithms. In short, a set of partitions is obtained by running multiple clustering algorithms and then the outputs are merged into a single result. The main idea behind majority voting is that the judgment of a group is superior to that of the individuals.

Other papers like [15] try to use voting for removal of noise and other inherent artifacts that results from the segmentation process with efforts going as well into providing specific mathematical models, methods [16] and probabilistic analysis in order to evaluate the relative quality of segmentation [17].

The next section briefly presents the technique described in [14]. The results of the clustering algorithms are mapped into a matrix, called the co-association matrix, where each pair (i, j) represents the number of times the elements (pixels, in our case) i and j were found in the same cluster. A pseudo-code for the algorithm is:

1. *Input*: a set of A partitions obtained from running different A clustering algorithms on the initial data set of N samples
2. *Output*: a partition of the initial data set
3. *Initialization*: Set the *co-association matrix* to a null $N \times N$ matrix
4. for each clustering algorithm update the *co-association matrix*:
 - a. for $i = 1$ to A , let P be the i^{th} partition from the input set corresponding to the result of the i^{th} algorithm
 - b. for every sample pair (i, j) that belong to the same cluster in P set $coassoc(i, j) = coassoc(i, j) + 1/A$
5. obtain the output partition by thresholding on the matrix values:
 - a. for each sample pair (i, j) that has $coassoc(i, j) > threshold$, combine i and j in the same cluster and also combine their belonging clusters, if necessary
 - b. for the remaining samples form an one element cluster
6. return the result.

This served as a starting point for the implementation of the voting image segmentation algorithm proposed by this paper.

3. AUXILIARY ALGORITHMS

Different segmentation algorithms have been designed having qualities and shortcomings and articles like [3], [7], [9] have been dedicated to studying the different trade-offs that have to be made.

This section briefly describes the individual image segmentation algorithms used in this paper to demonstrate the proposed concepts, with an emphasis on their upsides and downsides in terms of over/under segmentation i.e. computing too many/few segments as

output for a given input image than what is ideally expected. An in-depth analysis of each is not made, as this is not the purpose of the paper.

A series of representative algorithms from different categories in the image segmentation field are considered in order not to restrain the potential application domain. The paper makes use of the following segmentation techniques: histogram based segmentation, region growing based on neighboring pixels, segmentation using graphs, watershed transformation and mean shift segmentation.

Histogram based segmentation

This algorithm focuses exclusively on pixel intensity in an image. The general idea behind it is the use of local minima and maxima as the boundaries for a cluster of pixels. Thus, the histogram of the image is required.

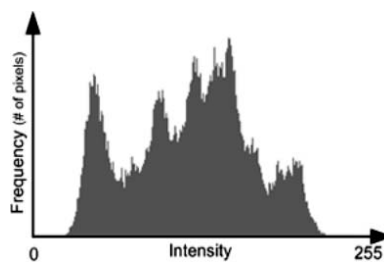


Figure 1. Histogram of a grayscale image

Fig. 1 displays a typical histogram of an 8-BPP (256 levels) grayscale image. The “peaks” represent the local maxima and the lower values represent the local minima. Between each adjacent minima/maxima or maxima/minima pair lies a cluster of pixels.

This technique is described in detail in [18], [19] and is known to give good results when the foreground of the image is clearly delimited from its background. Because the pixel distribution in the image is not taken into account, using this method on other types of images does not lead to very relevant results. A histogram based segmentation algorithm is generally considered to over-segment the input image.

Region growing based segmentation

This is a simple concept, based on the assumption that an object does not generally change color attributes in an abrupt manner. Therefore, neighboring pixels (that belong to the object in question) should have relatively close intensity values. Thus, starting with a random pixel and “growing” the region around it, pixels with similar characteristics can be found based on a homogeneity criterion – close intensity values – and clustered in the same collection. Optimized implementations of the region growing segmentation algorithms are the seeded region growing described in [20] and the adaptive region growing described in [21]. Both algorithms produce satisfactory results on most images, but generally have a tendency towards over-segmentation.

Graph based segmentation

The technique used is based on the “minimal spanning tree” concept and is described in detail in [22], [23]. In short, a graph is built with nodes as pixels and edges that have an associated weight which measures the dissimilarity of the connected pixels. The segmentation is obtained by partitioning the graph into connected components so that the edges between the nodes in the same component have higher weights (the higher the weight, the lower the dissimilarity).

The algorithm is relatively accurate, under-segmenting the input image in some cases.

Watershed based segmentation

The Vincent-Soille algorithm is the classic approach when it comes to watershed based segmentation. A detailed description of this technique can be found in [24].

A prerequisite for watershed transformation is blurring the input image first, so applying a Gaussian filter to it is a recommended pre-processing step. Even so, the algorithm generally tends to strongly over-segment the image.

Mean shift based segmentation

The theoretical background behind this approach can be found in [25] and an actual implementation is given in [26].

Most of the times the results obtained with this algorithm are accurate, with a slight tendency for under-segmentation.

4. THE VOTING ALGORITHM

The technique proposed is a variation of the more general algorithm based on majority voting clustering presented in the second section of this paper. Our algorithm mainly follows the same processing steps, with two major differences however: not all voting algorithms have the same “power of decision” and the actual implementation takes into consideration – and avoids – using too much of the system’s resources.

Coping with memory requirements

When images are used as input data, each pixel is a valid sample, so the total number of samples is given by the width of image times its height. Considering this and the fact that the co-association matrix is defined as an $N \times N$ matrix (where N is the number of samples), the result is that a very large structure will be needed to store the data. This could easily result in exceeding the addressing capabilities of a 32 bit operating system. To avoid such a scenario, a square-shaped sliding window across the image is used and only the pixels inside it are processed at a given time. The co-association matrix is thus dependent on the size of that window – adjustable by the user – instead of the image’s entire size.

After the matrix is filled with data, the second step of the algorithm takes place, where pixels are being merged into clusters. This is done by assigning a label to those pixels – all

the pixels in a cluster will be given the same label, which is different from labels associated with pixels in adjacent clusters. Once this is complete, the window slides forward and the process is repeated.

Another aspect is the increment used to slide the window. If it were to slide by its entire size, then different clusters would be obtained in each region of the image the window overlaps, although a certain cluster could easily span across multiple such regions. A workaround to this is not to slide the window by its entire size, but rather by its size minus one row of pixels – both horizontally and vertically, where applicable. This way an overlap with previously considered regions is obtained and the erroneous splitting of a pixel set is avoided, because the first row and/or column of pixels are already labeled and therefore newly considered pixels can be added to previously found clusters.

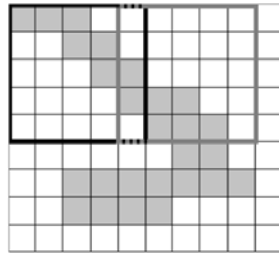


Figure 2. The sliding window technique

Fig. 2 displays the process described above. The sliding window is showed in two consecutive iterations – the black thick frame and the grey thick frame. Notice the overlapping area between the two windows, which allows the grey pixels inside the second one to be merged into the cluster found by the first one.

Weighted voting

Because each of the segmentation algorithms used had its “strengths and weaknesses”, a method of selecting the better results was needed. The main choice criterion was the one used for analysis in the third section of our paper: the over/under segmentation tendency of an algorithm. Over-segmentation implies that different clusters contain pixels with the same characteristics and thus can be combined in a larger one. Contrary, under-segmentation represents the case when clusters span over regions with different characteristics that should be split.

The assumption used was that if an algorithm is known to strongly over-segment an image and determines that two pixels are in the same cluster, then this is highly probable to be true. Analogous, if an algorithm is known to strongly under-segment an image and determines that two pixels are not in the same cluster then this is highly probable to be true as well. Considering all aforementioned assumptions, a performance evaluation parameter was introduced for each considered segmentation algorithm - called the over-segmentation factor. This parameter ranges from 0 to 1, where 0 represents the factor for the most extreme under-segmentation algorithm which would consider all pixels to be in one cluster, 1 the factor for the most extreme over-segmentation algorithm which will consider each pixel to

have its own cluster, and 0.5 being the optimal segmentation algorithm which depends on the image characteristics.

Based on the dynamic analysis of the output results of the independent segmentation algorithms used, an over-segmentation factor is associated with each as follows: a low one (user defined) for the algorithm that produces the fewest clusters, a high one (also user defined) for the algorithm that produces the most clusters and a variable one to the other algorithms (computed using linear interpolation based on the number of clusters they produce). This is a preprocessing step. Next comes the filling of the co-association matrix: when an algorithm finds two pixels that are/aren't in the same cluster, a value proportional to its over-segmentation factor is added/subtracted to/from the corresponding matrix element. Thus the "weight" of each segmentation algorithm is determined in a manner that encourages its decision if the probability of it being true is high.

The second step of the original approach remains the same – mainly clusters are determined using a threshold value on the matrix. This is also user-defined.

Because the lowest and the highest values of the over-segmentation factor, as well as the threshold value used in the cluster creation process are user-defined, the accuracy of the final result is dependent on how well these parameters are chosen.

Complexity analysis

For the first step of the algorithm, the complexity is $O(N^2)$. The second step introduces the cluster merging, which is done in $O(\log N)$ using a tree forest with path reduction. Therefore the total complexity of the algorithm is $O(N^2 \log N)$.

It should be noted that the complexity depends on the image pixel size, so N represents the image width times its height.

5. EXPERIMENTAL RESULTS

The algorithm performed well in general, giving good results on several test images considered. The system used for testing had an Intel® Core™2 Duo 2.4 GHz Processor and 2 GB RAM.

The first test considered was the classic "Lena", a perfect image for its natural color balance and great range of frequencies encountered across different areas:



Figure 3. Lena input image

The results obtained when running the algorithm set on the image shown in Fig. 3 are presented in Fig. 4 (the outputs for each independent algorithm, as well as the final output of the voting algorithm).

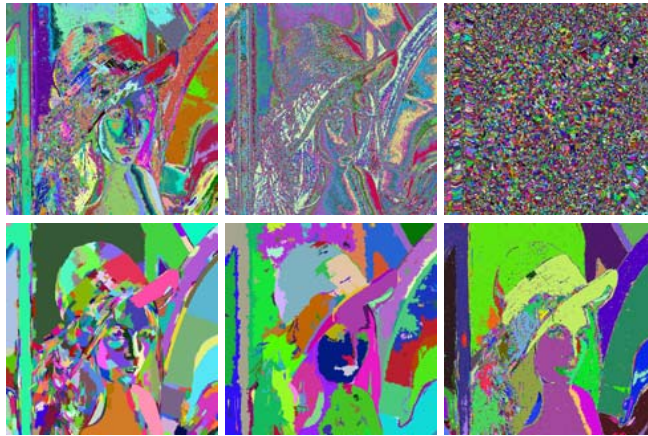


Figure 4. Lena segmentation processing results. From left to right and top to bottom: Region segmentation, Histogram segmentation, Watershed segmentation, Mean shift segmentation, Graph segmentation and (the final result) the Voting segmentation

As can be seen in Fig. 4, the region growing, histogram and watershed transformation based algorithms tend to over-segment the input image – especially watershed, which therefore will have the highest over-segmentation factor associated. On the other hand, it can be observed that the mean shift and graph segmentation methods tend to under-segment the image. In this case, the graph based segmentation algorithm will have the lowest over-segmentation factor associated.

The voting based image algorithm obtains a balance between the over and under-segmentation characteristics of the individual algorithms.

The current application is built with extensibility in mind, taking a framework approach where adding a new algorithm is as simple as implementing an interface. The framework is efficient because it runs all configured input segmentation algorithms in parallel.

The approach presented in this paper is an important part of a bigger project: a complete, modular, fully automatic content conversion system developed for educational purposes. Once finished, the system will allow large batches of documents to be processed fully automatically and as a result more complex algorithms like [27] will be employed to provide input segmentations. The voting process tuning will also benefit from the availability of more diverse test data, allowing every input algorithm to adjust over/under segmentation by using its running history.

Here are some other test images and the associated results for each of them (only the final voting based result is shown):

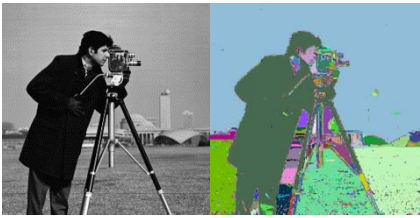


Figure 5. Cameraman segmentation processing results



Figure 6. Livingroom segmentation processing results

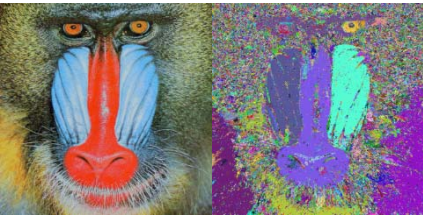


Figure 7. Mandril segmentation processing results



Figure 8. Peppers segmentation processing results

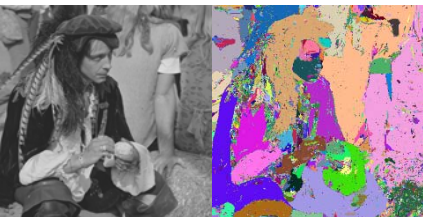


Figure 9. Pirate segmentation processing results

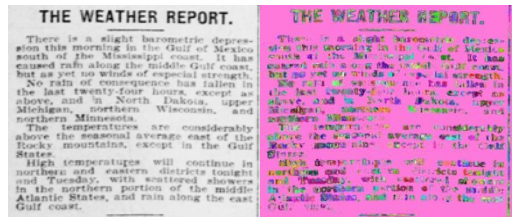


Figure 10. Text segmentation processing results

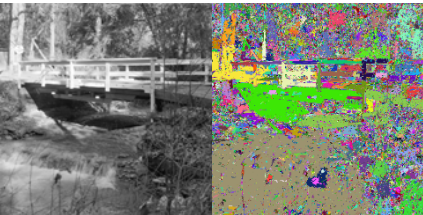


Figure 11. Walkbridge segmentation processing results



Figure 12. Blonde woman segmentation processing results

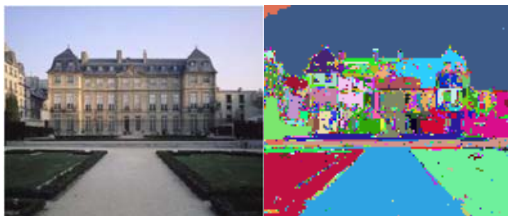


Figure 13. Museum segmentation processing results

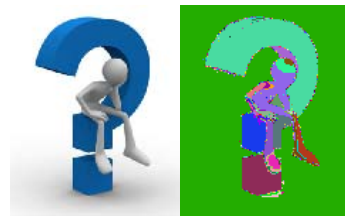


Figure 14. Question mark segmentation processing results

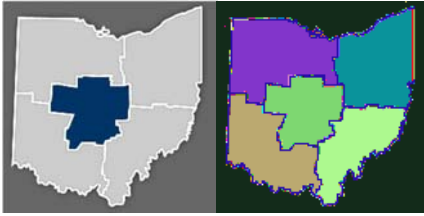


Figure 15. Map segmentation processing results

In the following table are presented the times obtained (for the voting algorithm only) on the considered testing system:

Input image	Image size		Run time (seconds)
	Size (pixels)	Color depth (bits)	
Lena	512 x 512	24	22.03
Cameraman	512 x 512	8	19.14
Livingroom	512 x 512	8	13.26
Mandrill	512 x 512	24	13.82
Peppers	512 x 512	8	14.44
Pirate	512 x 512	24	14.48
Text	975 x 821	8	50.40
Walkbridge	512 x 512	8	12.62
Blonde woman	512 x 512	8	15.20
Question mark	300 x 375	24	11.21
Museum	584 x 504	24	23.48
Map	512x512	24	16.16

Table 1 Output result analysis

6. CONCLUSIONS AND FUTURE WORK

The aim of this work is to propose a weighted voting algorithm that segments an image based on the results of other well-known segmentation algorithms. The algorithm copes with large memory requirements and also takes into account the tendency of the individual considered segmentation algorithms to over/under segment and has encouraging higher accuracy results.

Using the proposed algorithm, the results obtained were promising for most images tested. It can be therefore concluded that the idea of using several segmentation algorithms and merging their output to obtain a relevant result is viable.

Another algorithm optimization approach could use a feedback loop by considering the results of some segmentation algorithms to adjust the input of the other segmentation algorithms from the framework. By knowing the general characteristics of a segmentation algorithm and its results on an image, some meaningful information about the processed image can thus be derived and looped back either into the voting process or the individual segmentation parameters.

As future work there are two improvements to be taken into consideration. The first is adding an extra post-processing step for eliminating all the small clusters which may appear as being noise in the final result. The second is dynamically adapting the threshold parameter used for the merging/splitting of clusters, thus ensuring a higher chance of getting improved output results for a very varied range of input images.

7. ACKNOWLEDGMENT

The authors would like to thank Constantin Manoila and Lucian-Ilie Calin, for their great ideas, support and assistance with this paper.

8. REFERENCES

- [1] P. Herghelescu, M. Gavrilescu, V. Manta, "Visualization of Segmented Structures in 3D Multimodal Medical Data Sets", *Advances in Electrical and Computer Engineering*, vol. 11, no. 3, 2011, pp. 99 - 104.
- [2] L. D. Pham, X. Chenyang and J. L. Prince, "Current Methods in Medical Image Segmentation". *Annual Review of Biomedical Engineering*, vol 2, pp. 315–337, 2000.
- [3] S. K. Pal, N. R. Pal, "A Review on Image Segmentation Techniques", *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, 1993.
- [4] Fu K.S., Mui J.K., "A survey on image segmentation", *Elsevier Pattern Recognition journal*, Volume 13, Issue 1, pp. 3–16, 1981.
- [5] R.M. Haralick, L.G. Shapiro, "Image Segmentation Techniques", *CVGIP: Graphical Models and Image Processing* vol. 29, pp. 100-132, 1985
- [6] A. Hoover, "An experimental comparison of range image segmentation algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 18, Issue 7, pp. 673-689.
- [7] A. A. Aly, S. B. Deris, N. Zaki, Research review for digital image segmentation techniques, *International Journal of Computer Science & Information Technology (IJCSIT)* vol. 3, no. 5, oct 2011
- [8] Y.J. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335C1346, 1996.
- [9] H. Zhang, J. E. Fritts, S. A. Goldman, "Image segmentation evaluation: a survey of unsupervised methods", *Computer Vision and Image Understanding* vol. 110, Elsevier Science Inc., New York, pp. 260-280, 2008.
- [10] J. Freixenet, X. Munoz, D. Raba, J. Martí and X. Cufí, "Yet Another Survey on Image Segmentation: Region and Boundary Information Integration", *Lecture Notes in Computer Science*, vol. 2352/2002, pp. 21-25, 2002.
- [11] K. McGuinness, N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms", *Pattern Recognition* vol. 43, Elsevier Science Inc., New York, pp. 434-444, 2010.
- [12] S. Chabrier, H. Laurent, B. Emile, C. Rosenburger, and P. Marche, "A comparative study of supervised evaluation criteria for image segmentation", *EUSIPCO*, pp. 1143-1146, 2004.
- [13] J. S. Cardoso and L. Corte-Real, "Toward a Generic Evaluation of Image Segmentation", *IEEE Transactions on Image Processing*, No. 11, pp. 1773-1782.
- [14] A. L. N. Fred, Finding consistent clusters in data partitions, *Lecture Notes in Computer Science* vol. 2096, Springer-Verlag, London, pages 309-318, 2001.
- [15] P. Jonghyun, T. K. Nguyen and L. Gueesang, "Noise Removal and Restoration Using Voting-Based Analysis and Image Segmentation Based on Statistical Models", *Energy minimization*

methods in computer vision and pattern recognition, Lecture Notes in Computer Science, vol 4679/2007, 242-252, 2007.

- [16] P. Correia and F. Pereira, "Objective evaluation of relative segmentation quality", Image processing, vol 1, pp. 308-311, 2000.
- [17] K. Vincken, A. Koster and M. Viergever: Probabilistic multiscale image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:2, pp. 109–120, 1997.
- [18] L. G. Shapiro, G. C. Stockman (2001): "Computer Vision", New Jersey, Prentice-Hall, ISBN 0-13-030796-3, pp 279-325, 2001.
- [19] R. Ohlander, P. Keith, R. D. Raj, "Picture Segmentation Using a Recursive Region Splitting Method", Computer Graphics and Image Processing, vol. 8, no. 3, pp. 313–333, 1978.
- [20] R. Adam, L. Bischo: "Seeded Region Growing", IEEE Transactions On Pattern Analysis Machine Intelligence, vol. 16, no. 6, pp. 641-647, 1994.
- [21] Y. L. Chang, X. Li: "Adaptive Image Region-Growing, IEEE Transactions on Image Processing", vol. 3, no. 6, pp. 868-872, 1994.
- [22] P. F. Felzenswalb, D. P. Huttenlocher, "Efficient graph-based image segmentation", International Journal of Computer Vision vol. 59, Kluwer Academic Publishers, Hingham, pp. 167-181, 2004.
- [23] J. Shi, J. Malik, "Normalized Cuts and Image Segmentation". IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 888-905, 2000.
- [24] J. Roerdink, A. Meijster, "The watershed transform: definitions, algorithms and parallelization strategies", Fundamenta Informaticae vol. 41, IOS Press, Amsterdam, pp. 187-288, 2000.
- [25] D. Comaniciu, P. Meer, "Mean shift: a robust approach towards feature space analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, IEEE Computer Society, Washington DC, pp. 603-619, 2002.
- [26] C. M. Christoudias, B. Georgescu, P. Meer: "Synergism in low level vision". 16th International Conference on Pattern Recognition., Quebec City, Canada, August 2002, vol. IV, 150-155, 2002.
- [27] E. Ganea, D. D. Burdescu, M. Brezovan, "New Method to Detect Salient Objects in Image Segmentation using Hypergraph Structure", Advances in Electrical and Computer Engineering, vol. 11, no 4, pp. 111-116, 2011