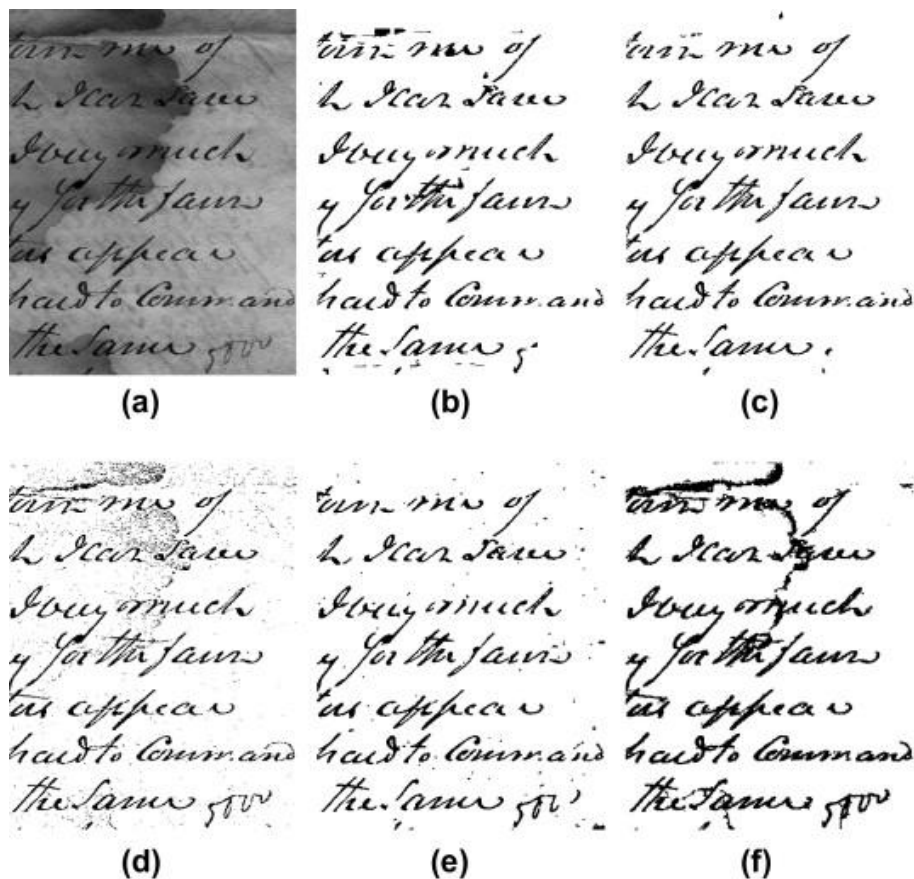


# Voting-Based Image Binarization

## Introduction

Image document analysis is a complex process that involves several steps of processing. However, due to their sensitivity to errors, most of these are not applied on the original image; instead, they use a simplified black and white version of the original image, which offers a clear separation between foreground and background. Unfortunately, achieving the optimal separation is difficult, as no proposed algorithm has managed to offer a solution that is adequate for any type of input.



(a) Representative image from DIBCO-2011 dataset [5]; corresponding binarization result produced by the method (b) Messaoud et al. (2011); (c) Su et al. (2010); (d) Ntirogiannis et al. (2009); (e) Howe (2011); (f) Gatos et al. (2008).

Over the past decades document creation and storage has slowly switched from physical to electronic support. This has greatly changed the way humans interact with their data: the search times have decreased as the documents can be easily accessed and investigated from virtually anywhere, the storage requirements no longer represent a problem and backing up information is handled automatically with little or no human intervention whatsoever.

Though it is great that newly created documents are no longer printed or typed on paper for storage, there are still millions more that were created before the advent of electronic storage.

These are either rare documents such as manuscripts or early printings, or archives that store large amounts of information that would be ideal for automatic processing.

Though content conversion has come a long way since its early days, processing old documents does pose a new set of challenges. These are far from the newly printed documents that are usually handled in everyday activities; instead, they show signs of deterioration caused by improper handling or storage, parasites attacks etc. Furthermore, the digitization process also introduces its errors, due to the poorly calibrated scanning devices, resulting in non-uniform brightness and noise.

As documents themselves have a complicated structure, the analysis components should not have to deal with the extra complexity previously presented. The documents are instead simplified, cataloguing, the pixels as either foreground or background through bitonal conversion or binarization [1]. A lot of work has been dedicated to image binarization achieving various degrees of success, depending on the input source.

## Project purpose

The purpose of the project is to develop an "Image Binarization System" (IBS). The IBS will consist from two parts:

- A "Binarization Algorithm Module" (BAM): this will be an executable which will receive an input continuous-tone image and will produce an output binary image.
- A "Voting Binarization Algorithm Module" (VBAM): using more BAMs a "smart-voting" technology will be used to blend the independent BAMs results into a binary image.

In every laboratory subgroup there will be 4 teams, each of them around 3 members for a total of around 12 members. Three of these teams will be responsible for 3 BAMs (one for each team), the fourth will be responsible for VBAM and project management activities regarding synchronization between teams.

Any team will have to cover (basically) three roles: research, development and test. The teams members are encouraged switching their roles between them whenever this seems appropriate for the benefit of the final product.

## BAM Overview

A BAM will be an executable which will receive from command line two file names (*input\_image* and an *output\_image*). The BAM will return an error-code: zero for no error (results are valid and will be used for voting purposes) and nonzero in case of an error occurrence (the error code should specify the error type; in this case the result will not be considered).

The output of the BAM is a 1bpp image, *output\_image* and an 8bpp image, *output\_image-confidence*. The first image is the actual binarization and the second a gray-scale image containing the confidence for the binarization for every pixel. 0 means that the respective pixel was randomly assigned a color (black or white); 255 means that the algorithm is absolutely certain that the respective color of the pixel is correctly assigned.

Construction of a BAM can be something as easy as a thresholding operation using a global threshold [8], fixed or adaptive local thresholding, etc... Researchers may find a lot of useful binarization ideas on internet; also some of Costin-Anton Boiangiu articles/ideas may be downloaded from his ResearchGate profile [2].

## VBAM Overview

The VBAM will receive at input any number of BAMs and will perform an “educated voting” algorithm. This can be something like taking for every pixel the majority decision, taking into account a weighted-decision mechanism using the confidence images; taking into account specific algorithm behavior (in having problems with excessive noise, biasing towards black/white, biasing toward unnecessary split/merge of image segments), etc...

## Evaluation criterias

In the evaluation of the binarization results there will be used both ground-truth image files, evaluation tools like the ones in DIBCO [4][5][6], the quality of the OCR output using the Tesseract OCR engine [7], the value of the ideas behind the algorithm and the execution time (complexity of the algorithm).

In order to fully evaluate the potential of the algorithm it is strongly recommended that the programming language is C/C++. Some simple skeletons to isolate you from the programming I/O will be provided for both BAM and VBAM. Also, usage of any optimization/parallelization technique (including multi-core or GPGPU) is optional but encouraged.

Any approach used both in BAMs and VBAM must be parameter free. If any of the algorithms involved require one or more parameters to be set for every image processing, these shall be computed/estimated by the program itself.

## A Blended Contest/Collaboration Approach...

- All subgroups will compete one against another using the VBAM results.
- All BAMs from all subgroups will compete one against another. The top N will be selected.
- All VBAMs from all subgroups will compete against one another. The top one will be selected.
- The selected BAMs and VBAM will be fine-tuned to work together by the ones of you who want an extra bonus, in the last weeks of the semester.

## Project expected outcome

Hopefully, the package containing the fine-tuned BAMs and VBAM and the used approaches/methods/algorithms technical descriptions will be presented in the international “Document Image Binarization Contest” (DIBCO) during the very prestigious international conference ICDAR 2015 [3] on behalf of the “Politehnica” University of Bucharest.

## Conclusion

While the end result of a binarization is definitely important, the means of obtaining it is also relevant, and this aspect is often downplayed in many papers. Algorithms that show good results for a wide range of input images are often parameterized, requiring human intervention for fine tuning. Setting these parameters requires some experience with the algorithm and its output, and obtaining the best out of an algorithm becomes more of an art than science. This could hardly be considered a “sin”, but such an approach renders an algorithm unusable for automatic processing, at least in its original implementation. To compensate for this aspect, some authors propose dynamic mechanisms for setting the values, which can, to some extent, yield results that are almost just as good as the human fine tuned ones.

Another aspect that is rarely tackled is resource usage. Hardly any algorithm requires more memory than what is usually available for modern commodity computers. On the other side, execution time can spiral out of control. Whereas for Otsu’s algorithm [8] the image pixels are inspected once for histogram creation and once again for the actual image binarization, applying local Otsu for 10 by 10 square windows results in 100 more inspections than the original approach. The processing can become even slower if the computations performed on the window are more complicated than basic pixel counting. The time complexity is usually linear, directly proportional to the number of pixels in the image. However it can sometimes depend on the square of the image size, when variable size windows are used, and even when it does not, the proportionality constant can be really high. Still, for large scale document analysis projects, the time constraint is not a deal breaker; the processing involved is usually easy to parallelize so more hardware and efficient implementations should normally solve, at least partially, this problem.

## References

1. [http://en.wikipedia.org/wiki/Binary\\_image](http://en.wikipedia.org/wiki/Binary_image)
2. [https://www.researchgate.net/profile/Costin-Anton\\_Boiangiu](https://www.researchgate.net/profile/Costin-Anton_Boiangiu)
3. <http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=27021&copyownerid=20950>
4. <http://users.iit.demokritos.gr/~bgat/DIBCO2009/>
5. <http://utopia.duth.gr/~ipratika/DIBCO2011/>
6. <http://utopia.duth.gr/~ipratika/DIBCO2013/>
7. <https://code.google.com/p/tesseract-ocr/>
8. [http://en.wikipedia.org/wiki/Otsu's\\_method](http://en.wikipedia.org/wiki/Otsu's_method)