DUT DURBAN UNIVERSITY OF TECHNOLOGY

DEPJ301 Project Report

Automatic Region-of-Interest Extraction in Low Depth-of-Field Images			
Vadia	**A. Y.**	**Mr.**	**21534684**
Automatic Region-of-Interest Extraction in Low Depth-of-Field Images			**Supervisor:** **Dr. S. Reddy**

Type of project: Design	Programme enrolled for: Electronic Engineering (Computer Systems)
Student declaration: I understand what plagiarism is, this project is my own work	A.Vadia ______ Student signature; 08.06.2017 ______ Date

Declaration by language editor (proofreader)	
I have been allowed adequate time to read this document carefully and to make corrections where necessary (date received indicated below). To the best of my knowledge, correct formatting, spelling and grammar are used throughout the document.	
Alka Singh ______ Example: A. M. Singh(language editor)	08.06.2017 ______ Date

Mark Allocation (to be completed by Supervisor/Examiner)	
Report (out of 100)	
Software (out of 100)	
Presentation (out of 100	
Total (70% Report + 25% Software + 5% Presentation)	
______ Dr. S. Reddy (Supervisor/Examiner)	______ Date
______ (Moderator)	______ Date

بسم الله الرحمن الرحيم

In the name of Allah [God], most merciful, ever merciful.

All praise is due to He alone. Peace and salutations upon his beloved messenger, our beloved master and guide Muhammed ﷺ. May He, Allah, burn the flame of his true love in our hearts & guide us on the path of remembrance and reflection leading to His proximity.

Contents

ABSTRACT

The automatic extraction of regions of interest in Low-Depth-of-Field (LDOF) images is a problem with no definite solution to date. In this paper, an attempt is made to reproduce a relatively efficient solution. A combination of ensemble clustering and graph cut modelling, together with traditional signal processing techniques are used to produce a novel algorithm, which filters regions of high information frequency from regions of low information frequencies, that is computationally more efficient than other such solutions. This design is principally based on the work of Dr Reza Rafiee [1] and is influenced heavily by the suggestions of Dr S Reddy [2].

1.**INTRODUCTION**

Image isolation and extraction is an important process in the field of digital image processing. Region-of-Interest(ROI) extraction is one such process used in image processing; it is a widely-used technique in computer vision and multimedia applications such as image compression, medical data analysis, understanding depth in 2D images, 2D to 3D conversion and various web applications. Despite the availability of photography techniques to capture foreground objects only, most settings for the capture of LDOF images render these techniques unsuitable. Consequently, the demand for a fast, efficient automatic solution is relevant.

The aim of this project is to realise a set of algorithms that perform efficient unsupervised extraction of Regions-of-Interests from LDOF images, this is performed by identifying the Object-of-Interest (OOI) using a combination of ensemble clustering, and wavelet analysis at a block level. Refining and removing unwanted details, by utilising minimal graph cuts and the Max-Flow Algorithm, from the in-focus region and after that delineation of the complete close bounded ROI.

This design is based primarily on the works of Dr Gholamreza Rafiee [1,3-6], specifically the technique of minimal graph cuts as proposed by his thesis. The completion of the algorithm implementation of this design in the allocated time is not foreseen, should this be the case then at the point at which the implementation ends, attention will be brought to this fact, and a theoretical design implementation will be detailed.

In section 2, the design specifications of the proposed solution are detailed. Section 3 presents a literature overview of the problem in the context of our solution. Section 4 provides the details of the realised solution. Section 5 will review the results achieved by the proposed solution. The following sections will review and conclude the proposed design.

2. DESIGN SPECIFICATIONS

The envisioned product is expected to differentiate between regions of interest and regions of non-interest automatically. Splitting will be achieved in broad terms by;

I. Breaking up the given image into blocks, so that it can be analysed.
II. Detecting the regions of potential interest by, in the case of this design, block wavelet analysis
III. Refinement of focus regions, using seeds obtained from step the preceding step.

The software will isolate interest regions by breaking down the given image into blocks. In doing so, analysis on the nature of each block can be performed, and the blocks can be classified. Classified blocks can then be seeded into an image segmentation algorithm to extrapolate the objects of interest in the given low depth of field image. The final output will consist of the region of interest in as it was presented to the system with defocused background regions on non-interest removed.

Aizawa, Kubota & Kodama [7] name the following applications of LDOF ROI extraction; "3D microscopic image analysis, image enhancement for digital cameras, arbitrarily focused image generation from two differently focused images, range segmentation for depth estimation, 2D-to-3D conversion for 3D TV, improvement of coding efficiency in multi-view coding, and fusion of multiple images which are focused to different degrees". Hence, the application of ROI extraction is typically employed as part of larger a system in the computer vision and digital signal processing discipline.

The Technical challenges expected in this project are: (I) To design an effective mechanism for separating regions of high information density from regions of low information density. (II) To develop an effective algorithm that implements the above design (III). To implement the code into a programming language (MATLAB) that the computer can process (using efficient and effective coding practices). (iv) To develop unit tests so that the algorithm can be effectively measured against comparable algorithms.

Limitations expected to be contended with in this project are the effective implementation of an algorithm that is of limited complexity while maintaining a conservative computational footprint within a short period.

3. LITERATURE STUDY

Preamble: The purpose of this project is to equip students with the skills and knowledge needed to work in the technical environment. Specifically, this project hones skills in the fundamentals of the digital image processing field. This project not only familiarises students with the tools and the different signal processing techniques that may be encountered during the prototyping of various computing systems typically encountered in the engineering and built environment but also teaches the correct processes and the best practices in applying these toolboxes correctly.

This project also affords insight on how to correctly read and write technical documents, programming as it applies to the field of signal processing, design of a signal processing system, spatial filtering, colour image processing, Morphological Image Processing, Image segmentation and utilising software design tools and techniques and troubleshooting skills which can be employed in an array of applications in the electronic engineering environment.

Low Depth of Field Photography: “Depth of field is the effective distance between the nearest and farthest object in a scene that appears acceptably sharp in an image” [8]. The Low depth of field photography technique, also referred to as a low focus range or low effective focus range is commonly used in film and photography. The LDOF technique represents a photographer's intention by giving sharp focus to foreground OOI’s only. LDOF images are captured by the use of a large aperture, close viewpoints, or utilising a longer focal lens from a smaller distance.

The fundamental problem in the automatic extraction of LDOF images lies with the fact that the only context of identifying the OOI is that it is the part of a given image that is in sharp focus, in contrast to background regions which are blurred/out of focus. Accordingly, no ‘priori’ knowledge exists for the given image and the features that one may encounter in such an image. Features such as colour and texture are unsuitable on their own for the extraction of OOI’s as these contexts vary between images.

Aizawa, Kubota & Kodama [7] name the following applications of LDOF ROI extraction; “3D microscopic image analysis, image enhancement for digital cameras, arbitrarily focused image generation from two differently focused images, range segmentation for depth estimation, 2D-to-3D conversion for 3D TV, improvement of coding efficiency in multi-view coding, and fusion of multiple images which are focused to different degrees”.

Segmentation of LDOF images using techniques such as K-Means, mean shift and graph cut amongst others, on their own are ill-suited for the extraction of ROI's in LDOF images because the context of the ROI is unknown. Photographs captured in a natural setting, Regions of Interest infrequently share the same properties i.e. texture, intensity, colour, etc.

Presently, there are many methods of image extraction of LDOF images, these are explained in detail in the various literature. There are many more methods of emphasising the foreground of such images. Some methods of emphasising the foreground include edge detection, blind deconvolution, Bayes discrimination, non-reference blocking, wavelet analysis and LDOF segmentation [4]. Some techniques are significantly efficient; however, many algorithms work better with certain subsets of images and not with others, other algorithms may be effective in extracting information from an image but might be computationally expensive. Hence, the need to develop an efficient means of effectively emphasising the object of interest in an image, specifically images with low field depths.

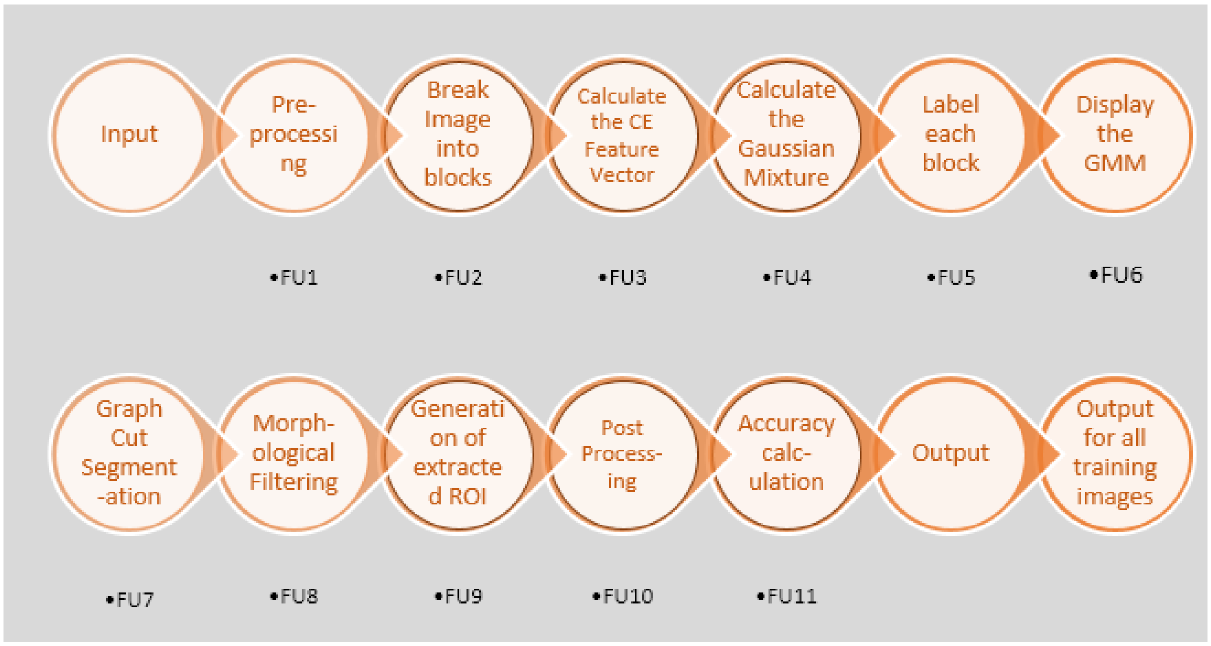
Two easy approaches were at first utilised for ROI extraction; these were direct edge analysis and region- or texture-based segmentation [2]. In the edge-based methods, their effectiveness is directly a function of the grey-level intensities. This reliance is shown to be limiting, more so for natural objects, where edges typically may be disconnected and vague. Even with the amalgamation of edge-linking techniques these methods often offer insufficient data required for effective ROI extrapolation.

"When digitally transposed, the ROI in an LDOF image may be mathematically described as regions of high-frequency and the background may be mathematically described as regions of low frequency" [2]. Multi-resolution wavelet analysis techniques try to attenuate the low-frequency components by exploiting the high-frequency parts in an image as a result of energy compaction through sub-band decomposition. A focused region in an image can exhibit low attenuation of its high-frequency components, while defocused regions can have a giant attenuation of the high-frequency components. Hence, the measure of the image blur caused by the defocus is directly related to the number of native spatial frequencies. Wavelet transforms differentiate saliencies that are used to analyse signal structures of various sizes and spatial frequencies. Since only local phase features are included, each wavelet constant is spatially localised, and since sub-band decomposition allows for non-overlapping of frequency bands, each wavelet constant is conjointly frequency localised.

4. DESIGN

4.1 PROPOSED APPROACH

The region of interest extraction system is a sequential system with three major stages: The pre-processing part, the image segmentation/background removal stage (processing) and post-processing/ clean-up segment.



Input:

The image to be processed.

FU1: Pre-Processing:

The image is converted to grayscale for use in the next stages. The size of the image and its ability to be processed is verified.

Input	IMG - The input image.
Output	IMG_GS – The grayscale image rotated such that the image is in landscape orientation. Rot - Boolean value indicating if the image has been orientated

FU2: Break Images into Blocks:

The image is converted into blocks and stored. The output is displayed.

Input	IMG_GS 2-Dimensional image

Output	CA - Cell array containing the given 2D image broken up into blocks.

FU3: Calculate the Contrast Energy Feature Vector:

The defining properties of the blocks are ascertained, in this case, the minimum wavelet variance coefficient of the high-frequency block and the dynamic contrast range of the block is calculated.

Input	CA - Cell array containing a 2D image broken up into n blocks
Output	X - 2*N Dimensional Array containing the contrast and energy of each block in cell array, which together makes up the CE Feature Vector (X)”

FU4: Calculate the Gaussian Mixture Model:

Gaussian mixture model values are calculated using an expectation maximisation algorithm.

Input	“X (N, D) – CE Feature vector (input data), n=number of observations, d=dimension of variable K - maximum number of Gaussian components allowed LTOL - percentage of the log likelihood difference between 2 iterations (is = none) MAXITER - maximum number of iterations allowed (= 10) PFLAG – Boolean flag for plotting the data INIT - structure of initial W, M, V (= none)”
Output	“W (1, K) - estimated weights of GM (Pi) M (D, K) - estimated mean vectors of GM (Mu) V (D, D, K) - estimated covariance matrices of GM (Sigma (standard deviation)) L - log likelihood of estimates”

FU5: Label Each Block:

Each block is assigned a weight based on its relation to the clusters defined by the GMM.

Input	W, M, V as detailed above
Output	XLABELS – Array of the weight of each block in the GMM.

FU6: Display the GMM:

The probability density function is shown.

Input	W, M, V, XLABELS
Output	Image displaying the Energy vs. Contrast graph from GMM along with the weights.

FU7: Graph Cut Segmentation:

The highest weighted blocks of both the background and foreground of the image are used to estimate the region of interest.

Input	XLABELS, X, CA, IMG_GS
Output	IMG_GC – A logical image describing the region of interest in the original image.

FU8: Morphological Filtering:

Morphological filtering is applied to the estimated region of interest.

Input	IMG_GC
Output	IMG_GC_MORPH – Filtered logical image.

FU9: Generation of the Extracted Region of Interest:

The ROI is defined.

Input	IMG_GC_MORPH
Output	IMG_ROI

FU10: Post-processing:

The ROI is extracted from the original image.

Input	IMG, IMG_ROI
Output	IMG_EXTRACTED – The extracted image with the correct orientation.

FU11: Accuracy Calculation:

The accuracy of the algorithm is quantified.

Input	GROUND_TRUTH, IMG_EXTRACTED
Output	Text output of the accuracy of the image.

Output:

The extracted region of interest in a LDOF image.

Output Iterations

The script runs through each image and performs the above functions.

4.2 DETAILED DESIGN AND TESTING

Note: The explicit definition of the nature of the variables in each functional unit is detailed in the preceding section. To avoid repetition, it has not been included here.

Stage 1: Pre-processing: The image is read and converted into a grayscale image, the grayscale image is briefly flashed on the screen to ensure that there are no errors. If the image is too small or is a logical image, the program is halted, detecting an object of interest in a very small image using the techniques presented in this design will not work. The image is then rotated such that the longer edge of a rectangular image is on its edge, i.e. The image is always processed in landscape operation; this is to simplify the processing in the next stage. The image is returned to its original orientation after the OOI has been identified.

Stage 2: The image is then broken up into square blocks, the size of each block is calculated such that the width of each block covers approximately 17% (1/6) of the width (longer edge) of the image. When the image is broken up into blocks, the last row, depending on whether the number of blocks in a row fits perfectly, may have a remainder, this is added to a list containing the places where the image is divided. The image is then broken up into a cell array; the image could have also been broken up into smaller sub-images using array slicing. However, a cell array is easier to work with, and the result is the same. The block size is selected at 1/6 of the width so that the image is broken up sufficiently to allow the local block energy and local block contrast to be calculated and compared statistically with other blocks. This size also allows the block to be broken up at a second level so that the sub-blocks can be measured against the main blocks to classify the blocks (as being either foreground/background). This is necessary because the expectation maximisation algorithm used in the next stage produces different results at every run, furthermore using block's and then sub-blocks significantly reduces the complexity of the algorithm.

Stage 3: To measure each block against the other, and thus classify it, the texture details of the block is determined. For each block, the local(block) minimum moment of wavelet energy coefficient and the local dynamic contrast range is considered.

The local dynamic contrast range is calculated by subtracting the local minimum intensity from the local maximum intensity and dividing that by the sum of the minimum intensity and maximum intensity:

$$x_c = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \text{ [14]}$$

In order to extract the energy of a block, the block’s frequency is decomposed (classified). The discrete wavelet transform is a multi-scale frequency decomposition technique. The Haar discrete wavelet transform is applied at a single level to achieve this. The Haar transform is employed, because it is computationally inexpensive [9], it is used at a single level, again, to conserve computational power, a highly accurate transformation is unnecessary because only an approximate relative energy is needed to classify a block’s energy. The Haar transform decomposes a block into for frequency bands, The HH, HL, LH and LL frequencies. Only the high-frequency bands (HFB) defined as the HH, HL and LH bands are evaluated in determining a block’s frequency, this is intuitive. The square root of the second order moment of wavelet coefficients in each high-frequency block, i.e. the variance is calculated. The minimum energy is the lowest variance in frequency value calculated. It is not necessary to normalise this value. However, this is done for convenience when evaluating the Energy Contrast graph.

$$x_c = \min\left(\frac{1}{2}\sqrt{W_k^2}\right)$$

where W_k represents the summed matrix of wavelength coefficients of the HFB. The wavelet energy coefficient x_c and dynamic contrast x_e for each block in an image together form the contrast-energy feature vector x of that image.

Stage 4: The expectation maximisation algorithm is a model-based clustering approach that utilises models to attempt to find a fit between the data and K models. The EM algorithm can be considered as a soft version of the K-means clustering algorithm [13], however, unlike the K-means system of classification a point in space need not belong to a cluster, this is essential in labelling uncertain regions and is the reason why the K-means algorithm was not applied in this design [11]. In this approach, the EM algorithm aims to estimate the maximum likelihood parameters of a bivariate mixture [1]. Because of the nature of the data in the expectation maximisation pool, it is very easy to ascertain the cluster corresponding to sharp regions as being the cluster with the largest mean Euclidian distance from the centre of

Contrast-Energy feature space [1]. The cluster corresponding to the background region will also have a relatively much lower dynamic contrast range.

Stage 5: All blocks of a given low depth if field image can be classed into one of:

1. In focus/Sharp
2. Out of focus/Blurred
3. Uncertain

Regions defined as out of focus typically have uniform intensities through the entire block. Conversely, Regions labelled as being “sharp” typically exhibit distinctive contrast and energy features. However, there exist regions in LDOF images for which adequate distinctive features exists, these regions can be evaluated by examining their neighbouring regions to classify them. To classify the blocks into one of the three classes an expectation maximisation algorithm is applied to the contrast energy feature vector and based on the results of the expectation maximisation algorithm a block may be classified. It should be noted here that the expectation maximisation algorithm used in this design was not developed from first principles. However, the source code provided by Patrick Tsui [10], did not classify data, as is required by this application. The original code has thus been modified to suit the needs of this design.

Stage 6: To simplify processing this stage was combined with the previous step in the implementation.

Stage 7: Graph Cut Segmentation. Block based interest regions identified in the previous stage are used as a seed for a minimal graph cut. Only strong seeds (pixels) from both the foreground and background regions are used i.e. seeds that are found to be close to the centroids of the k clusters in the contrast energy Gaussian Mixture Model. The seeds impart prior knowledge on the image, which is used to reduce the search space for a feasible region of interest. The logical area representing the region of interest is developed from this. “Yuri Boykov's and Vladimir Kolmogorov's work on graph cuts and MRF optimisation has been extensively cited in academia; their maximum flow implementation is widely used in computer vision and image processing research” [11]. Hence, their work has been used to compute the max flow algorithm

Stage 8: Morphological filtering is applied to the region of interest to remove any noise that may exist due to inconsistencies in the previous stage. This stage is not needed if the previous stage works correctly. Morphological filtering is achieved by relatively simple operations such as STREL dilation and erosion. This operation is simply to remove small amounts of noise from the previous stage.

Stage 9: The filtered morphological binary image is returned to the orientation of the original image, and the two images are compared. The region of interest is extracted by the multiplication of the constituent vectors of the original image.

Stage 10: The region of interest is displayed and saved.

Stage 11: The accuracy of the region of interest is calculated using common and accepted algorithms in the field of computer vision.

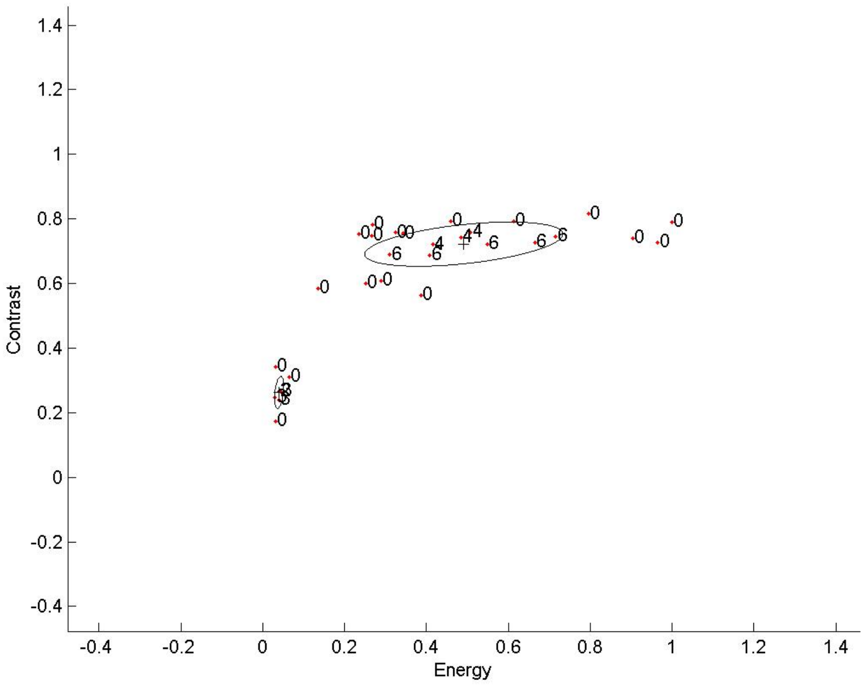
5. RESULTS

5.1 SUMMARY OF THE RESULTS ACHIEVED

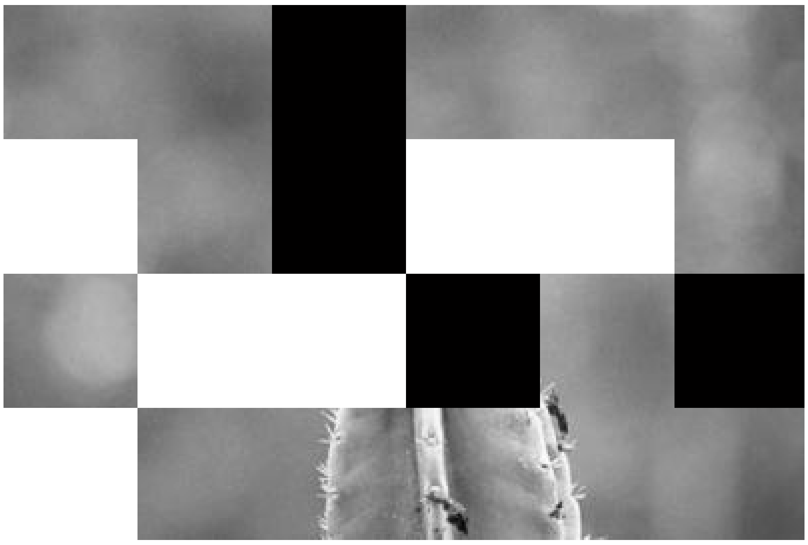
[“Note to supervisor, this sections are incomplete and will be submitted conclusively with the revised presentation as scheduled for the 9/6/2017]

The performance of this design cannot cannot be quantified as the implementation was not completed successfully in the allocated time.

5.2 RESULTS AND OBSERVATIONS

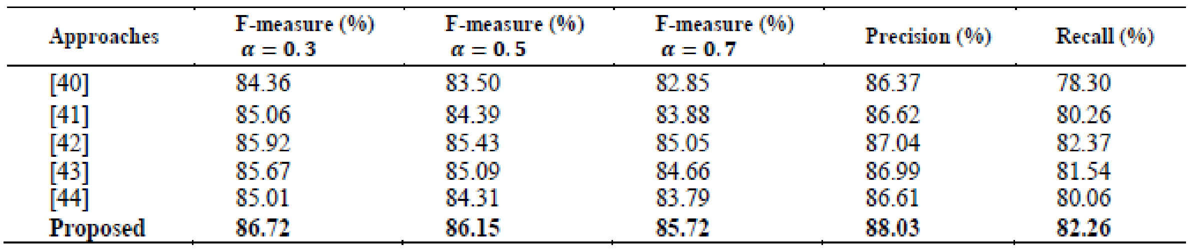


A typical Gaussian Mixture model of energy and contrast vectors after the expectation maximization of the feature space is calculated.



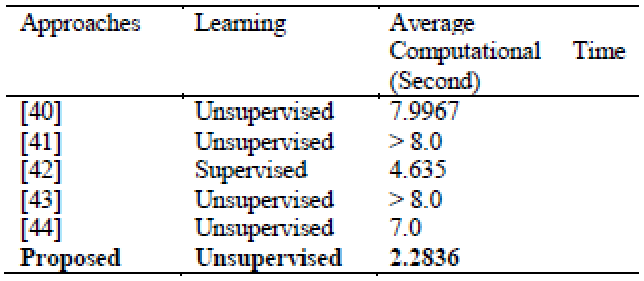
A labelled image. Black represents the foreground, white background and grey, regions of uncertainty

Dr G. Rafiee's implementation of a similar design boasted the following results amongst four images from the Corel Draw LDOF image dataset, against leading algorithms:



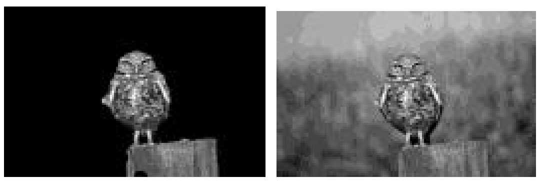
Approaches	F-measure (%) $\alpha = 0.3$	F-measure (%) $\alpha = 0.5$	F-measure (%) $\alpha = 0.7$	Precision (%)	Recall (%)
[40]	84.36	83.50	82.85	86.37	78.30
[41]	85.06	84.39	83.88	86.62	80.26
[42]	85.92	85.43	85.05	87.04	82.37
[43]	85.67	85.09	84.66	86.99	81.54
[44]	85.01	84.31	83.79	86.61	80.06
Proposed	**86.72**	**86.15**	**85.72**	**88.03**	**82.26**

Comparison of average computational time results for the 117 test images.



Approaches	Learning	Average Computational Time (Second)
[40]	Unsupervised	7.9967
[41]	Unsupervised	> 8.0
[42]	Supervised	4.635
[43]	Unsupervised	> 8.0
[44]	Unsupervised	7.0
Proposed	**Unsupervised**	**2.2836**

The above results were also performed at an average computational time of half of the computational time of peers.



The results of Dr Rafiee’s segmentation of an image

6. DISCUSSION

A novel perspective was applied to the approach in this design. Unfortunately, this meant that the support for the problems encountered during this design was unsupported by peers, this was a primary cause of failure. The design could be improved further by fixing errors. The results of Dr Rafiee prove that the potential for the accuracy of this design is quite high, however this remains unproven. Classifying ROI regions using high frequency components should be combined with other techniques.

7. CONCLUSION

Much was learnt during this design. Perfection cannot always be attained.

8. REFERENCES

[1]G. Rafiee, "Automatic Region-of-Interest Extraction in Low Depth-of-Field Images", Ph.D, University Newcastle, 2017.

[2]S. Reddy, "Automatic 2D-TO-3D Conversion of Single Low Depth-of-Field Images", Ph.D, University of Cape Town, 2016.

[3]G. Rafiee, S. Dlay and W. Woo, "Region-of-interest extraction in low depth of field images using ensemble clustering and difference of Gaussian approaches", Pattern Recognition, vol. 46, no. 10, pp. 2685-2699, 2013.

[4]G. Rafiee, S. S. Dlay, and W. L. Woo, "Unsupervised Segmentation of Focused Regions in Images with Low Depth of Field," in Multimedia and Expo (ICME), 2013 IEEE International Conference on, San Jose, USA, in Press.

[5]G. Rafiee, S. S. Dlay, and W. L. Woo, "Automatic Segmentation of Interest Regions in Low Depth of Field Images Using Ensemble Clustering and Graph Cut Optimization Approaches,", Multimedia (ISM), 2012 IEEE International Symposium on, California, USA, pp. 161-164.

[6]G. Rafiee, S. S. Dlay, and W. L. Woo, "A Review of Content-Based Image Retrieval," in Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on, Newcastle, UK, pp. 775-779.

[7]K. Aizawa, A. Kubota and K. Kodama, "Implicit 3D Approach to Image Generation: Object-Based Visual Effects by Linear Processing of Multiple Differently Focused Images", Multi-Image Analysis, Theoretical Foundations of Computer Vision, vol. 10, pp. 226-237, 2001.

[8]D. Stump, Digital Cinematography, 1st ed. Oxon: Focal Press, 2014, p. 188.

[9]R. Stanković and B. Falkowski, "The Haar wavelet transform: its status and achievements", *Computers & Electrical Engineering*, vol. 29, no. 1, pp. 25-44, 2003.

[10]P. Tsui, *EM_GM Expectation Maximization algorithm for Gaussian mixture*. University of Waterloo: Mathworks, 2016.

[11]A. D’Souza, *Using EM To Estimate A Probablity Density With A Mixture Of Gaussians.*, 1st ed. cise.ufl.edu/.

[12]Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, 2004.

[13]Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, 2004.

[14]J. Wu, "Photolithography Optics", University of Tennessee, Knoxville.

9. APPENDIX

Printout of Code:

```
function [img,rot,imgC] = FU1( imgName )
%FU1 Pre-Processing
%   Gets the ball rolling for the next functional unit..
   %
%   Input       -   imgName An image with a region of interest that needs
%   to be extracted can be RGB or Grayscale, It would be pointless to
%   process a logical Image
%   Output      -   IMG   An image grayscale image rotated such that the
longer edge is the width
%                   ROT   Parameter specifying if the image has been
rotated
%   Processing  -   Check if the image is the right size Draw inputted iage
img = imread(imgName);
imgC = img;
img = rgb2gray(img);

figure(1),imshow(imgC,[]), title('Input Image'); drawnow;
% Enlarge figure to full screen.
set(gcf, 'units','normalized','outerposition',[0 0 1 1]);
drawnow;
%Make sure image is greater than size 24*24

[rows, columns] = size(img);
if(rows<32 || columns <32||islogical(img))
    error('The input image is too small or is logical, please try again
with a more detailed image');
    %disp('<a href="matlab:dbquit;">Yes</a> / <a
href="matlab:dbcont;">No</a>');
    %hlt;
end

if (rows>columns)
    rot=true;
    img=imrotate(img , -90);
else
    rot=0;
end
figure(2),imshow(img,[]), title('Grey Scale'); drawnow;
end
```

```matlab
function [ ca ] = FU2(img)
%FU2 Breaks up the image into blocks
%   Input       -   Image
%   Processing  -   Image is broken up
%   Output      -   Cell array with broken up image

[rows, columns] = size(img);
blockSize = ceil(columns/6);% Columns in block.
% Figure out the size of each block in rows.
% Most will be blockSize but there may be a remainder amount of less than
that.
wholeBlockRows = floor(rows / blockSize);
if(rem(rows, blockSize)>1)
    blockVectorR = [blockSize * ones(1, wholeBlockRows), rem(rows,
blockSize)];
else
    blockVectorR = blockSize * ones(1, wholeBlockRows);
end
% Figure out the size of each block in columns.
wholeBlockCols = floor(columns / blockSize);
if(rem(columns, blockSize)>1)
    blockVectorC = [blockSize * ones(1, wholeBlockCols), rem(columns,
blockSize)];
else
    blockVectorC = blockSize * ones(1, wholeBlockCols);
end

    ca = mat2cell(img, blockVectorR, blockVectorC);
end
```

```
function [ X ] = FU3( ca )
%FU3 Works ot the contrast x-e feature vector
%   Input       -   Cell Array with broken up image blocks
%   Processing  -   Work out the Contrast & Energy for each block and
%   stores  it in an array. Displays the cell aray ca this is to reduce
%   redundancy and should be in the previous FU to be technically correct.
%   Output      -   X(2xn) Contrast Energy Feature Vector
%                   n is the number of blocks in the cell array ca
plotIndex = 1;
numPlotsR = size(ca, 1);
numPlotsC = size(ca, 2);
X = zeros((numPlotsR*numPlotsC),2);
figure(3),imshow([],[]), title('XE'); drawnow;
for r = 1 : numPlotsR
    for c = 1 : numPlotsC
        %fprintf('plotindex = %d,   c=%d, r=%d\n', plotIndex, c, r);
        % Specify the location for display of the image. For testing
        subplot(numPlotsR, numPlotsC, plotIndex);
        currentBlock = ca{r,c};
        imshow(currentBlock);
        [rowsB, columnsB, ~] = size(currentBlock);
%         Make the caption the block number.
        caption = sprintf('Block #%d of %d\n%d rows by %d columns', ...
        plotIndex, numPlotsR*numPlotsC, rowsB, columnsB);
        title(caption);drawnow;

        blockMaxInt=double(max(currentBlock(:)));
        blockMinInt=double(min(currentBlock(:)));
        blockDynamicRange=double(double((blockMaxInt-
blockMinInt))/double((blockMaxInt+blockMinInt)));
        %[a,d] = haart(currentBlock,1);
        [~,LH,HL,HH] = dwt2(currentBlock,'haar');

        LHA= ((1/4)*(sum(sum(LH.^2))))^(1/2);
        HLA= ((1/4)*(sum(sum(HL.^2))))^(1/2);
        HHA= ((1/4)*(sum(sum(HH.^2))))^(1/2);

        blockEnergy = min((LHA),(HLA));
        blockEnergy= min(blockEnergy,(HHA));
        %figure; imshow(currentBlock);
        X(plotIndex,2) = blockDynamicRange;
        X(plotIndex,1) = blockEnergy;

        plotIndex = plotIndex + 1;

    end
end
X(:,1)= X(:,1)/max(X(:,1));
X(isnan(X)) = 0 ;
end
```

```
function [W,M,V,L] = EM_GM_FU4(X,k,ltol,maxiter,pflag,Init)
% [W,M,V,L] = EM_GM(X,k,ltol,maxiter,pflag,Init)
% Code not included as it is mostly reused code
% EM algorithm for k multidimensional Gaussian mixture estimation
%
% Inputs:
%   X(n,d) - input data, n=number of observations, d=dimension of variable
%   k - maximum number of Gaussian components allowed
%   ltol - percentage of the log likelihood difference between 2 iterations
([] for none)
%   maxiter - maximum number of iteration allowed ([] for none)
%   pflag - 1 for plotting GM for 1D or 2D cases only, 0 otherwise ([] for
none)
%   Init - structure of initial W, M, V: Init.W, Init.M, Init.V ([] for
none)
%
% Ouputs:
%   W(1,k) - estimated weights of GM
%   M(d,k) - estimated mean vectors of GM
%   V(d,d,k) - estimated covariance matrices of GM
%   L - log likelihood of estimates
%
% Written by
%   Patrick P. C. Tsui,
%   PAMI research group
%   Department of Electrical and Computer Engineering
%   University of Waterloo,
%   March, 2006

% Modified By
%   Vadia A. Y.
%   Department of Electrical and Computer Engineering
%   Durban University of Technology,
%   June 2017
%
```

```
function XL = Plot_GM_FU5(X,k,~,M,V)
[~,d] = size(X);
S = zeros(d,k);
R1 = zeros(d,k);
R2 = zeros(d,k);

for i=1:k,  % Determine plot range as 4 x standard deviations
    S(:,i) = sqrt(diag(V(:,:,i)));
    R1(:,i) = M(:,i)-4*S(:,i);
    R2(:,i) = M(:,i)+4*S(:,i);
end
Rmin = min(min(R1));
Rmax = max(max(R2));
clf, hold on
if d==2
    plot(X(:,1),X(:,2),'r.');
    XL = zeros(1,max(size(X)));
    for i=1:k,
        XL = XL + Plot_Std_Ellipse_FU6(M(:,i),V(:,:,i),X,XL);
    end
    xlabel('Energy');
    ylabel('Contrast');
    axis([Rmin Rmax Rmin Rmax])
end
XL= transpose(XL);
    dx = 0.005; dy = 0.005; % displacement so the text does not overlay the
data points
text((X(:,1))+dx, (X(:,2))+dy,num2str(XL));
%saveas(gcf,'Barchart','jpg')
%title('');
end
```

```matlab
function XL = Plot_Std_Ellipse_FU6(M,V,X,XL)
[Ev,D] = eig(V);
d = length(M);
if V(:,:)==zeros(d,d),
    V(:,:) = ones(d,d)*eps;
end
iV = inv(V);
% Find the larger projection
P = [1,0;0,0];  % X-axis projection operator
P1 = P * 2*sqrt(D(1,1)) * Ev(:,1);
P2 = P * 2*sqrt(D(2,2)) * Ev(:,2);
if abs(P1(1)) >= abs(P2(1)),
    Plen = P1(1);
else
    Plen = P2(1);
end
count = 1;
step = 0.001*Plen;
Contour1 = zeros(2001,2);
Contour2 = zeros(2001,2);
for x = -Plen:step:Plen,
    a = iV(2,2);
    b = x * (iV(1,2)+iV(2,1));
    c = (x^2) * iV(1,1) - 1;
    Root1 = (-b + sqrt(b^2 - 4*a*c))/(2*a);
    Root2 = (-b - sqrt(b^2 - 4*a*c))/(2*a);
    if isreal(Root1),
        Contour1(count,:) = [x,Root1] + M';
        Contour2(count,:) = [x,Root2] + M';
        count = count + 1;
    end
end

Contour1 = Contour1(1:count-1,:);
Contour2 = [Contour1(1,:);
Contour2(1:count-1,:);
Contour1(count-1,:)];

plot(M(1),M(2),'k+');
plot(Contour1(:,1),Contour1(:,2),'k-');
plot(Contour2(:,1),Contour2(:,2),'k-');

for i=1:numel(XL)
%      if((X(i,1)> M(2,1)))
%          %       XL(i) = 5;
%      end
%      if((min(size(M)==1)))
%          if((X(i,1)> M(1,2)))
%            %  XL(i) = 20;
%         end
%      end
    if((inpolygon(X(i,1),X(i,2),Contour1(:,1),Contour1(:,2))))
        XL(i) = 2;
    elseif (inpolygon (X(i,1),X(i,2),Contour2(:,1),Contour2(:,2)))
        XL(i) = 3;
    end
end
```

```matlab
function flow = FU7(im,Sed)

m = double(rgb2gray(im));
[height,width] = size(m);

disp('building graph');
N = height*width;

% construct graph
E = edges4connected(height,width);
V = abs(m(E(:,1))-m(E(:,2)))+eps;
A = sparse(E(:,1),E(:,2),V,N,N,4*N);

% terminal weights
% connect source to leftmost column.
% connect rightmost column to target.
% Sed = sparse([1:height;N-
height+1:N]',[ones(height,1);ones(height,1)*2],ones(2*height,1)*9e9);

disp('calculating maximum flow');

[flow,labels] = maxflow(A,Sed);
labels = reshape(labels,[height width]);

imagesc(labels); title('labels');
end

function [ img ] = FU8(in)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
%in = imread('trainmask_1.bmp');
%in = rgb2gray(in);
%figure(15),imshow(in,[]); drawnow;
%in = im2bw(in, 0.5);
in = im2bw(in, 0.5);
figure(15),imshow(in,[]); drawnow;
SE=strel('diamond',5);
img = imdilate(in,SE);
img_blank_2_fill = imfill(img, 'holes');
img=imerode(img_blank_2_fill,SE);
figure(20),imshow(img,[]); drawnow;
end

function [ maskedRgbImage] = FU9(imgL,img_colour,rot)
%UNTITLED6 Summary of this function goes here
%   Detailed explanation goes here

%imgL = imread('trainmask_1.bmp');
%rot = false;
imgL = im2bw(imgL, 0.5);% Just in case
if (rot==true)
    imgL=imrotate(imgL, 90);
end
%img_colour = imread('train_1.jpg'); %loads colour image into img_colour

maskedRgbImage = bsxfun(@times, img_colour, cast(imgL,class(img_colour)));
figure(18),imshow(maskedRgbImage,[]), title('Output [Colour]'); drawnow;

end
```

```
function
[Fvalue,precision,recall,accuracy,JaccardIndex,TP,FP,TN,FN,FPrate,TPrate,MC
C] = compareBinaryImages( reference, toTest )
%COMPAREBINARYIMAGES Compute various similarity metrics between two binary
images
%   reference = grouth truth binary image
%   toTest = binary image to be compared to the reference image
% https://github.com/nicjac/PHANTAST-MATLAB
% https://github.com/nicjac/PHANTAST-
MATLAB/blob/master/GUI/OptimizationTool/compareBinaryImages.m
% Nicolas Jaccard, University College London
if(ndims(reference)~=2 && ndims(toTest)~=2)
    error('Inputs must be two 2-dimensional matrices');
end;

 TP = nnz(reference==1 & toTest==1); % True positive
 FP = nnz(reference==0 & toTest==1); % False positive
 TN = nnz(reference==0 & toTest==0); % True negative
 FN = nnz(reference==1 & toTest==0); % False negative

P = TP + FN; % Total positive for the true class (= reference)
N = FP + TN; % TOtal negative for the true class (= reference)

FPrate = FP/N; % False positive rate
TPrate = TP/P; % True positive rate

precision = TP/(TP+FP);
recall = TP/P;
accuracy = (TP+TN)/(P+N);

MCC = (TP*TN-FP*FN)/sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN));
2/((1/precision)+(1/recall));

% Alternative form for Fvalue
2*(precision*recall/(precision+recall));

% Avoid getting a division by 0 if only negatives and perfect detection
if(TN==numel(reference))
    'gaga'
    Fvalue = 1;
    warning('FValue was set to 1 as all pixels were true negatives');
else
    Fvalue=2*TP/(FP+TP+P);
end

2*TP/((FP+TP)+(TP+FN));

JaccardIndex = TP / (FP+TP+FN);

end
```