ETRO
VUB-DEPARTMENT
OF ELECTRONICS
AND INFORMATICS

Vrije
Universiteit
Brussel

IMAGE PROCESSING

Projects 2016

Adrian Munteanu (titular)
Athanasia Symeonidou, Andrei Sechelea (TAs)

Guidelines

- The aim of your projects is to study, implement, test and demonstrate image processing algorithm(s)
- You should use Matlab or C/C++ for your implementation, unless other method is proposed.
- You have **to write your own Matlab (or C/C++) code**. Existing code may be used only for visualization purposes; this has to be appropriately acknowledged in the project and the report.
- **Concerning the code you have to write yourselves: copying existing code from external sources disqualifies the project, leading to a project score of 0 for the entire group.**
- You have to submit a written project report, 2000 words typical, 4000 words max
- Use IEEE conference paper as a template :
  http://www.ieee.org/conferences_events/conferences/publishing/templates.html
- Each group submits ONE report
- Sections (obligatory) : Introduction, Description of your algorithm or system, Experimental results, Conclusions, References, Appendix (listing briefly what each student did within the group)

Project submission

- Submit as pdf (along with the code, all in a zip file) to:
  asymeoni@etro.vub.ac.be, atsechelea@etro.vub.ac.be
- **Deadline: 05/06/2016**

Grading

- The project accounts for 40% of the final score for the image processing exam.
- The project grade is based on:
  - Technical quality (score per group)- 35%
  - Written report (score per group)- 25%
  - Oral defense (individual score)- 40%
- The oral presentation and Q&A (~10 minutes) will be done individually with a common Power point presentation on the day of the exam. Your score will be based on your understanding of the image processing techniques you used in your project.

Reference for all:
Gonzalez, Woods, "Digital Image Processing"

Project titles

01. Comparative study of denoising filters
02. Car plate detection and recognition
03. Image composition (collage) with Poisson blending
04. Measurement of diameter and identification of problems of cables
05. Fault detection by image processing
06. QR code decoder
07. HDR image processing
08. Single image super-resolution
09. Inpainting
10. Image compression
11. Multi-scale edge detection on medical images
12. Fingerprint verification
13. Graph Cut for Image Segmentation
14. Speckle Noise Reduction in Ultrasound Images
15. Solving a Jigsaw Puzzle
16. Image registration for panorama
17. Denoising for astrophotography

1. Comparative study of denoising filters

Description

Compare several denoising filters (Gaussian, Inverse, Wiener, Median) for various types of noisy images i.e. blur, Gaussian noise, salt & pepper, speckle, etc. Add also a state-of-the-art image denoising technique from the literature (you have to find the SoTa yourself per noise model, and justify the choice).

Technologies

1. Get an image
2. Add noise (different types)
3. Write your own code for the different denoising techniques
4. Denoise the noisy images
5. Compare using PSNR (Peak Signal to Noise Ratio) taking as reference the original noise-free image.
6. Select the best filter for each noise type.
7. Compare your results with the theory.

References

- A. Buades, B. Coll, J.M Morel, "A review of image denoising algorithms, with a new one"
- Check the State of the Art in denoising
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code (including all filters)

2. Car plate detection and recognition

Description

Perform detection and analysis of Belgian car plates

Use photos taken by you and try different type of distortions, e.g. viewing angle, extreme illumination and low resolution

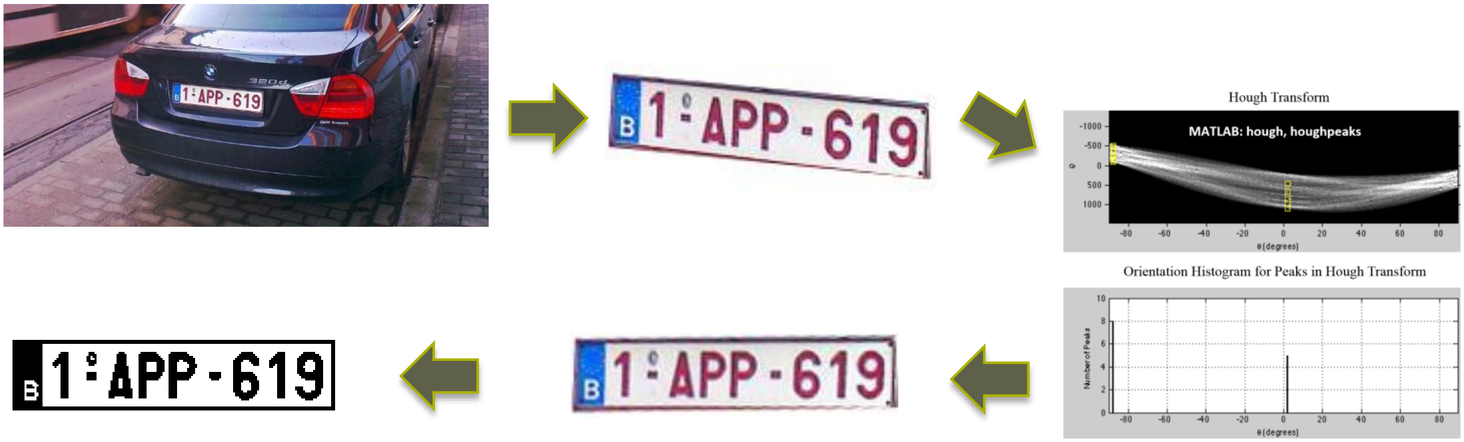
Technologies

1. Use Otsu method to binarize
2. Use Hough Transform to rotate image
3. Use perspective transformation to align
4. Binarize
5. Define and extract your own features for binary image
6. Make a letter/number database for number plates
7. Perform license plate identification for a number of input images which are NOT part of the training phase
8. Export the result in **text** format (e.g. 1 APP 619)

References

- Ondrej Martinsky, "Algorithmic and mathematical principles of automatic number plate recognition systems" (2007)
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



3. Image composition (collage) with Poisson blending

Description

Manually compose pieces from *several* images

Automatically blend these pieces and produce an output image

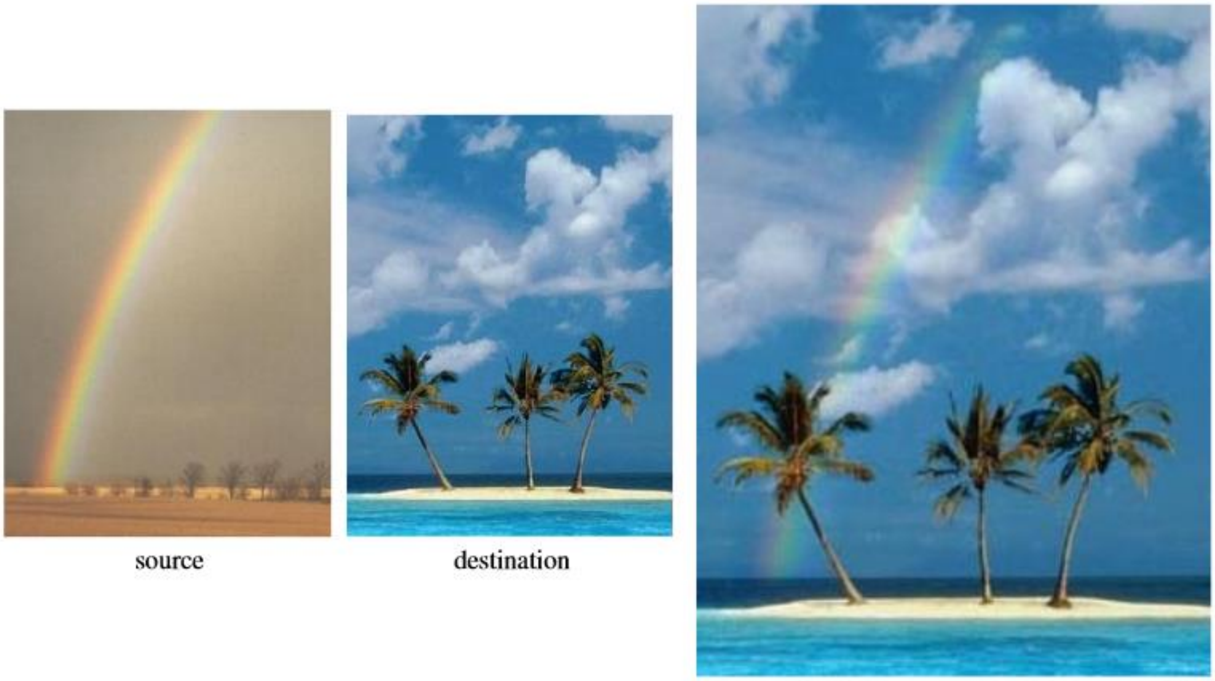
Technologies

- Define regions of interest in each input image that need to appear in the final result
- Poisson blending, Edge detection/segmentation starting from several colour images
- You need to implement a GUI (you may use existing SW).

References

- P. Prez, M. Gangnet, and A. Blake, Poisson image editing, ACM Transactions on Graphics (2003)
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



source

destination

4. Measurement of diameter and identification of problems of cables

Description

Metrology by image processing (video data and calibration data is also provided for this problem – contact the teaching assistants)

Technologies

Image format conversion (read video frames from two different video sources)

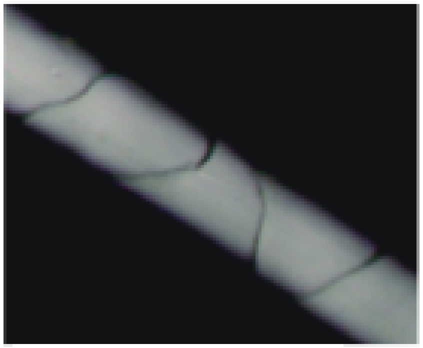
Image denoising, image segmentation, edge detection (using Canny), object extraction from the background, pixel measurement (use calibration data for that)

Account for the fact that the distance between the cable and the cameras is varying (i.e. use both cameras to perform the measurements)

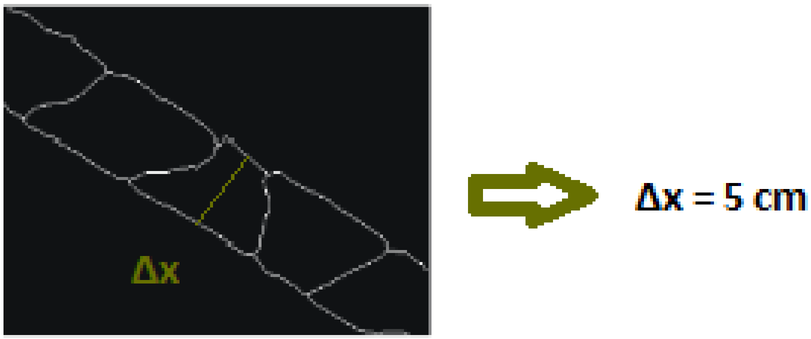
References

- Li, Yi, Shang, "Measurement of Diameter and Scale of Cashmere Fibers by Computer Images Analysis"
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



Original image



result

5. Fault detection by image processing

Description

Fault detection by image processing. Scan through the frames of 2 videos in real time and show an alert when a fault in the cable is identified showing also the frame number.

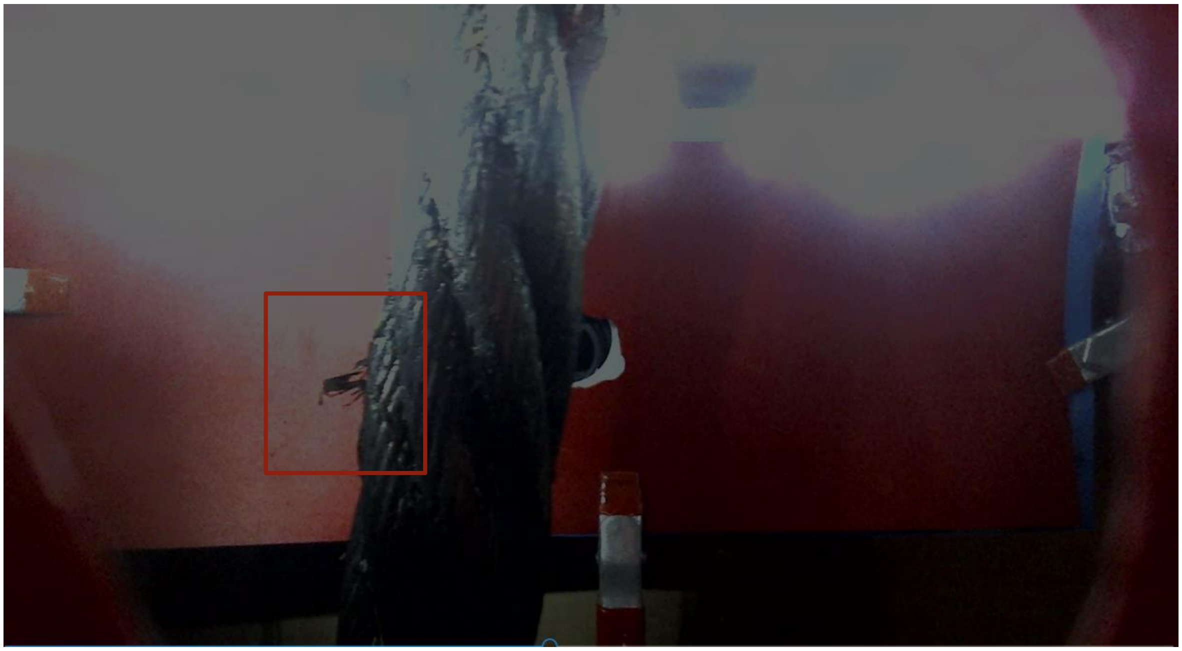
Technologies

- Image format conversion (read video frames from two different video sources)
- Perform contrast correction – ensure that the illumination is uniform across the cable
- Identify the median axis of the constituting cables in the cable rope Identify anomalies in object's outer structure (see example), and trigger and alarm

References

- Zhang Yan, Zhong Weiliang, "Online Fault Detection Method for Steel Cord Conveyor Belt Using the Regularity Band"
- Markus Oberweger, Andreas Wendel, and Horst Bischof "Visual Recognition and Fault Detection for Power Line Insulators"
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



6. QR code decoder

Description

Read QR code and decode message

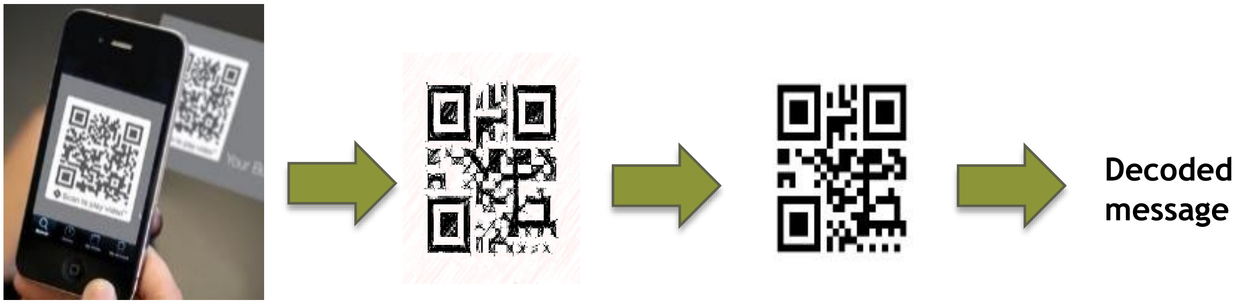
Technologies

Edge and pattern detection, perspective distortion, morphological operators, counting lines and distances.

References

- Weibing Chen, Gaobo Yang, Ganglin Zhang, "A Simple and Efficient Image Pre-processing for QR Decoder"
- Jeng-An Lin, Chiou-Shann Fuh, "2D Barcode Image Decoding"
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



Decoded message

7. HDR image processing

Description

Rendering High Dynamic Range Images on a GUI from a single “raw” image

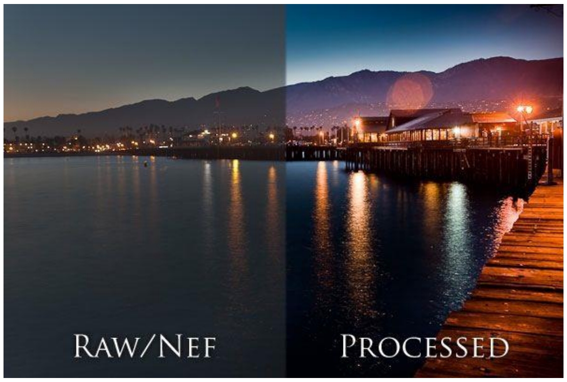
Technologies

- Perform tone mapping: e.g. compute the logarithm of the HDR image (or a window of values in the HDR image) and normalize it to the interval [0 1], thereby compressing the dynamic range of HDR data to an interval that a LDR device can display.
- Convert the normalized image to the CIE L*a*b* color space, in which luminance can be manipulated directly.
- Implement and apply a contrast-limited *adaptive* histogram equalization for colour images (produce uniform, exponential, Gaussian distributions for the output).
- Adjust image contrast according to a user-specified parameter.
- Increase or decrease color saturation and hue as requested by the user.
- Adjust color temperature (from warm to cold colormaps); create your own “vivid” colourmap
- You need to implement a Matlab GUI (you may use existing SW). You need to be able to change these parameters in real time through the GUI, and allow windowing (show the various tone mappings)

References

- Debevec, and Malik, "Recovering High Dynamic Range Radiance Maps from Photographs"
- Reinhard, Ward, Pattanaik, and Debevec, “High Dynamic Range Imaging”
- http://digital-photography-school.com/correcting-and-creating-hdr-images-in-lightroom/
- http://nl.mathworks.com/company/newsletters/articles/rendering-high-dynamic-range-images-on-the-web.html
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



8. Single image super-resolution

Description

Recover a high resolution image from one or more low resolution input images and do better than bi-cubic interpolation.

Technologies

- Bicubic interpolation, Self-similarity
- interpolation of the input low-resolution image into the desired scale
- generation of a set of candidate images based on patch-wise regression: kernel ridge regression is utilized;
- combining candidates to produce an image: patch-wise regression of output results in a set of candidates for each pixel location;
- post-processing based on the discontinuity prior of images: as a regularization method, kernel ridge regression tends to smooth major edges; References

References

- K.I. Kim, Y. Kwon, "Example-based Learning for Single-Image Super-resolution"
- D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in Computer Vision, 2009 IEEE 12th International Conference on, 29 2009-oct. 2 2009, pp. 349 –356.
- S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," Signal Processing Magazine, IEEE, vol. 20, no. 3, pp. 21 – 36, may 2003.
- S. Baker and T. Kanade, "Hallucinating faces," in Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, 2000, pp. 83 –88.
- http://www.wisdom.weizmann.ac.il/~vision/SingleImageSR.html

NOTE: Please create your own MATLAB code.

9. Inpainting

Description

Remove selected objects and inpaint content, restore image

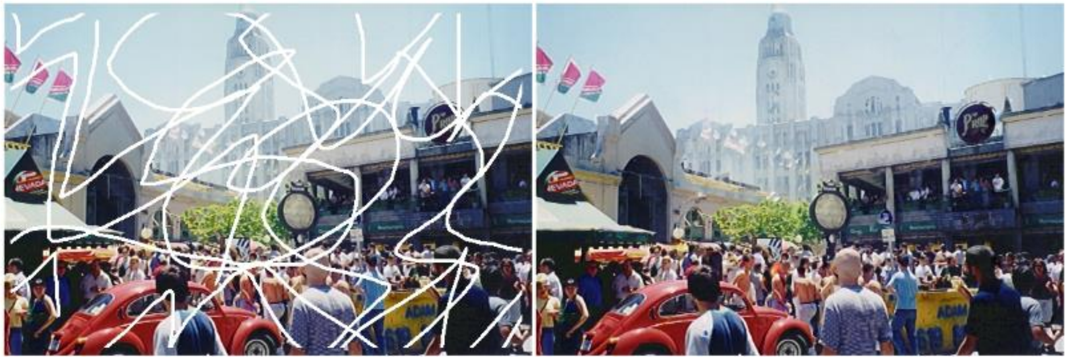
Technologies

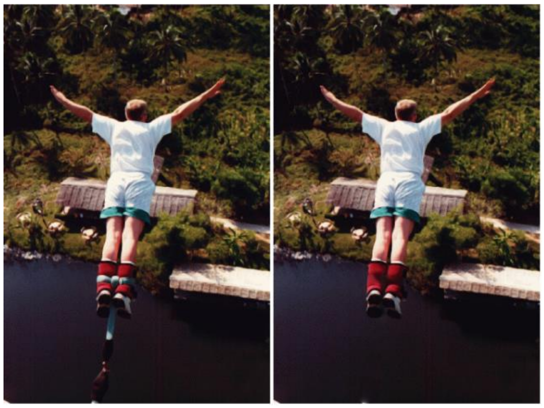
Inpainting based on masked regions

References

- Marcelo Bertalmio and Guillermo Sapiro, ”Image Inpainting”
- Criminisi, P. Perez and K. Toyama, “Region Filling and Object Removal by Exemplar-Based Image Inpainting”
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.





10. Image compression

Description

Image compression codec, comparison to JPEG/JPEG200

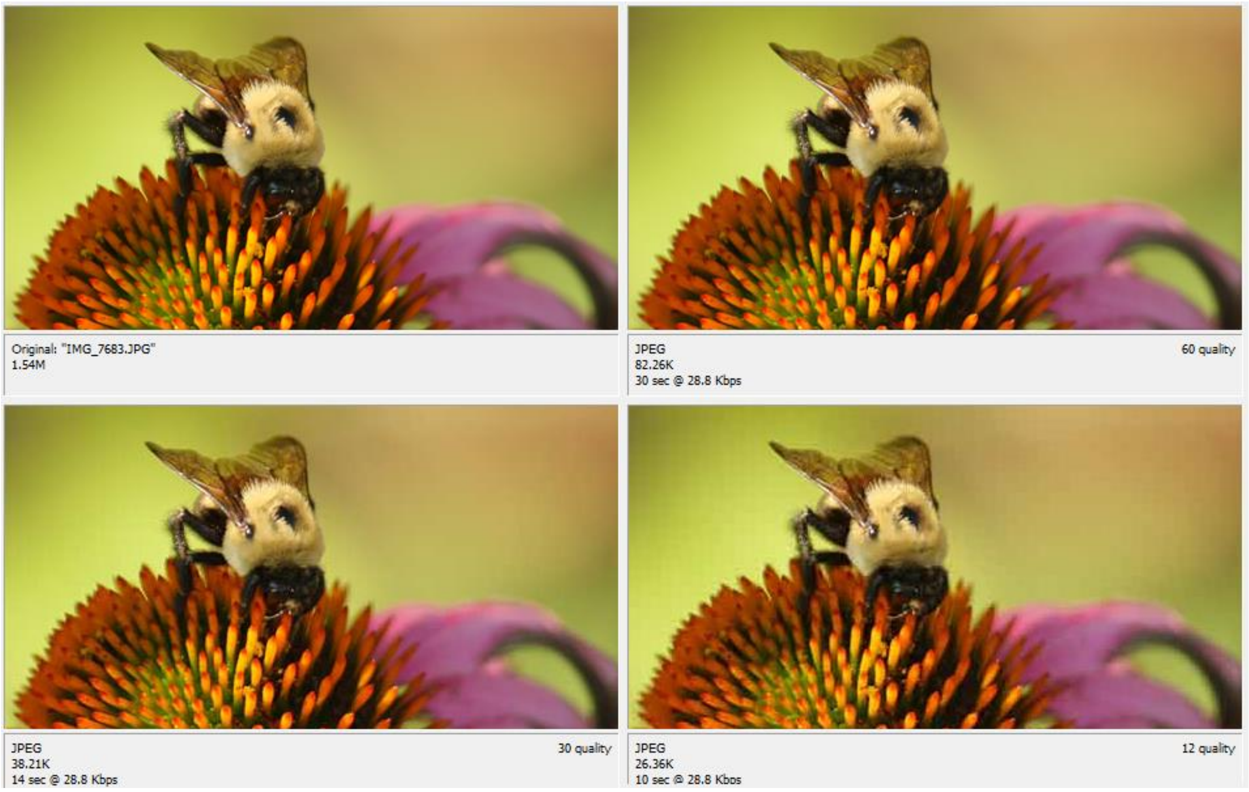
Technologies

- Design and implement you own quality / resolution scalable wavelet codec
- Test and evaluate the degradation in image quality while the compression ratio increases
- Compare to JPEG and JPEG-2000

References

- Wei-Yi Wei, “An Introduction to Image Compression“
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



Original: "IMG_7683.JPG"
1.54M

JPEG
82.26K
30 sec @ 28.8 Kbps
60 quality

JPEG
38.21K
14 sec @ 28.8 Kbps
30 quality

JPEG
26.36K
10 sec @ 28.8 Kbps
12 quality

11. Multi-scale edge detection on medical images

Description

Edge detection of medical images

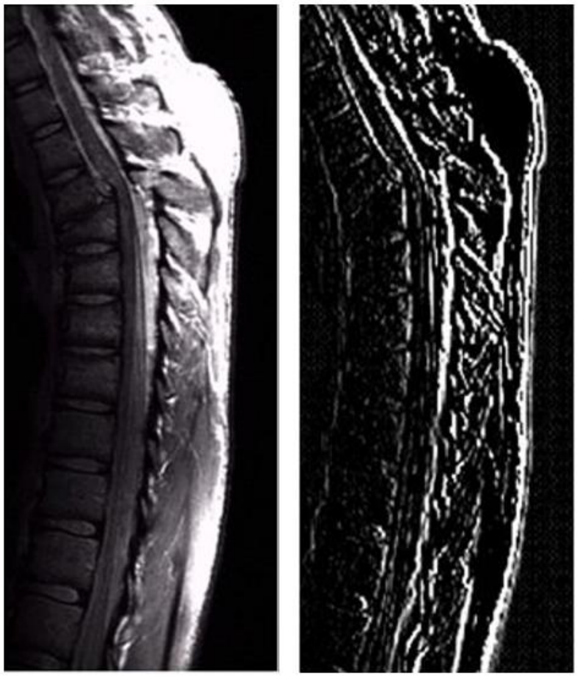
Technologies

- Design and implement your own multiscale wavelet-based edge detection method
- Compare it against single-scale edge detection methods (Roberts, Prewitt, Sobel, single-scale Canny...) as well as against edge detection with morphological operators

References

- S. Mallat, S. Zhong, "Characterization of signals from multiscale edges," PAMI, vol. 14, no. 7, July 1992.
- S. Mandal, V.P. Sudarshan, Y. Nagaraj, X.L. D. Ben, D Razansky, "Multiscale edge detection and parametric shape modeling for boundary delineation in optoacoustic images"
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



12. Fingerprint verification

Description

Fingerprint verification

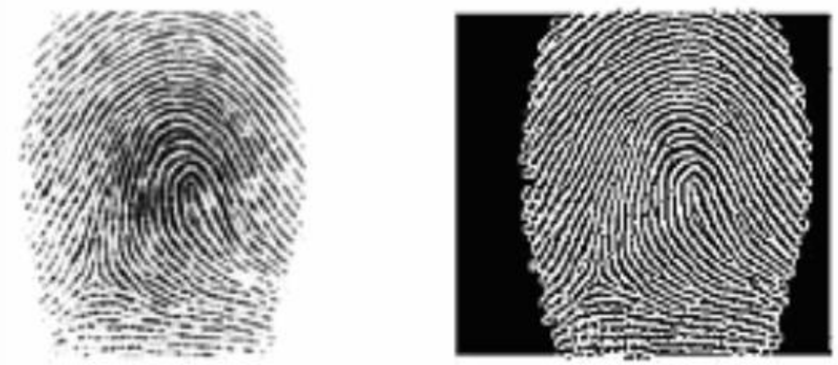
Technologies

- Rotate, Binarize image, Feature extraction with mathematical morphological operator
- Do feature extraction for the fingeprint database (Ref. 4) and find the correct match for your initial fingerprint.

References

- Humbe, Gornale, Manza, Kale, "Mathematical Morphology Approach for Genuine Fingerprint Feature Extraction"
- Kumar, Verma, "Fingerprint image enhancement and minutia matching"
- http://www.math.tau.ac.il/~turkel/imagepapers/fingerprint.pdf
- Fingerprint images: http://www.advancedsourcecode.com/fingerprintdatabase.asp
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



13. Graph Cut for Image Segmentation

Description

- Segmentation of colour images

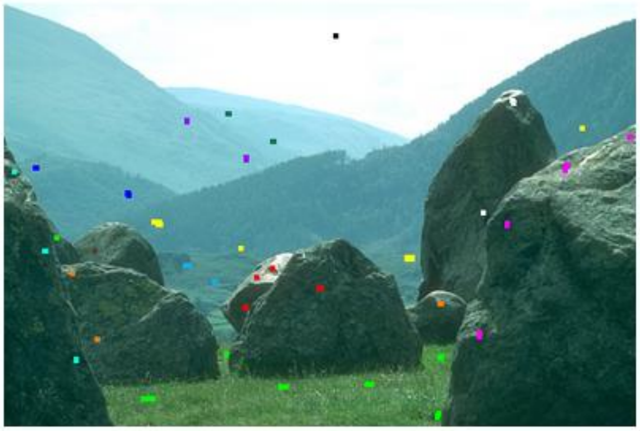
Technologies

- Image segmentation using graph cuts
- Apply image segmentation using the watershed transform
- Compare the segmentation results

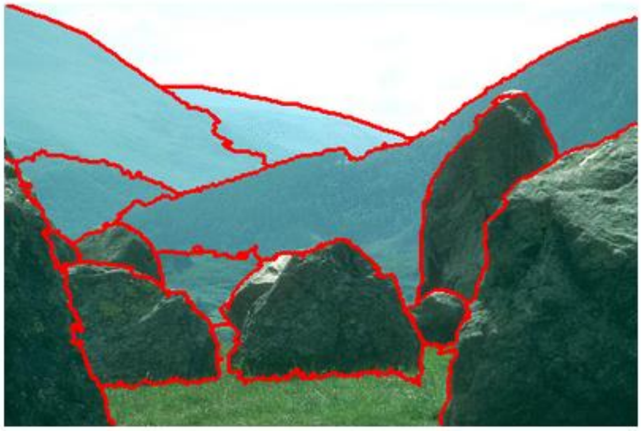
References

- C. Couprie, L. Grady, L. Najman and H. Talbot, "Power Watersheds: A New Image Segmentation Framework Extending Graph Cuts, Random Walker and Optimal Spanning Forest"
- T.Cour, F. Benezit, J. Shi. ”Spectral Segmentation with Multiscale Graph Decomposition”
- James Malcolm, Yogesh Rathi, Allen Tannenbaum, “A Graph Cut Approach to Image Segmentation in Tensor Space”
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



Input seeds



Power Watershed

14. Speckle Noise Reduction in Ultrasound Images

Description

Remove speckle noise from ultrasound images

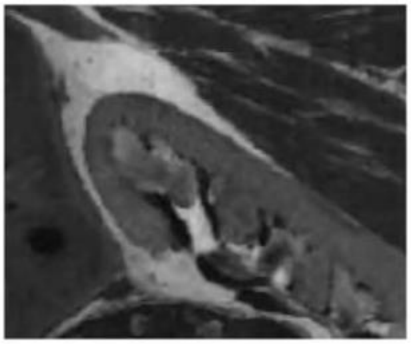
Technologies

Contrast adjustment, Laplacian of the Gaussian, structure tensor

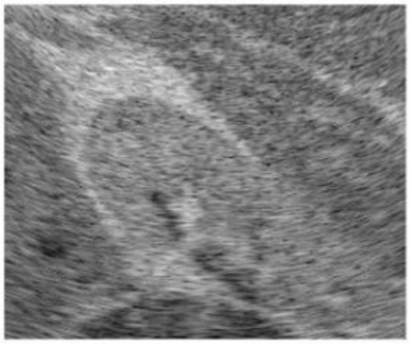
References

- Juan L. Mateoa, , Antonio Fernández-Caballeroa, "Finding out general tendencies in speckle noise reduction in ultrasound images"
- Zapata, Ruiz, "On Speckle Noise Reduction In Medical Ultrasound Images"
- Database: http://www.medison.ru/uzi/eng/all/
- http://www.mathworks.com/help/images/functionlist.html

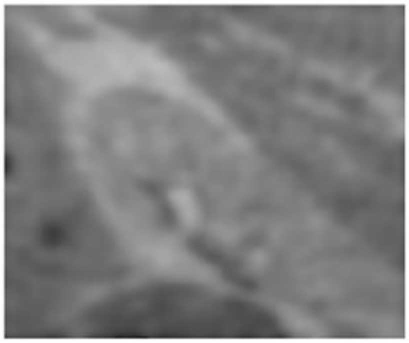
NOTE: Please create your own MATLAB code.



Original image



Speckled image



De-speckled image

15. Solving a Jigsaw Puzzle

Description

Solve a jigsaw puzzle by image processing

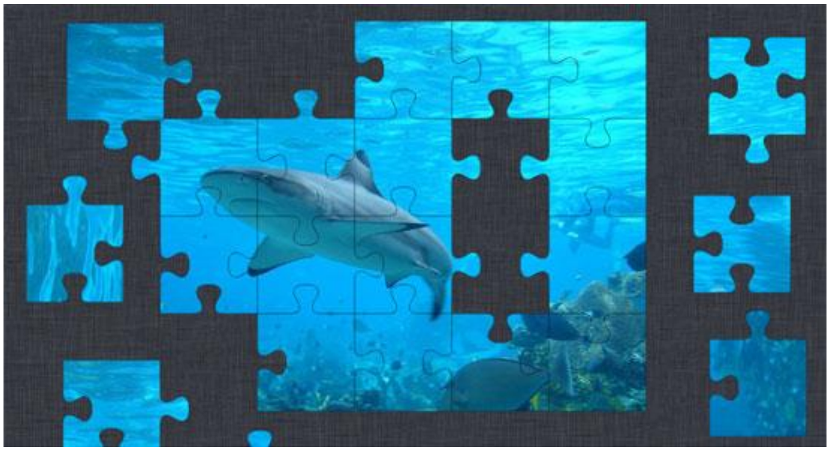
Technologies

- Region labeling, template matching, Eigenimages, Edge Detection
- Take an image, separate it to puzzle pieces based on a puzzle template, Make an image with all the pieces in random positions, do segmentation and feature extraction for the pieces and match their edges, solve the puzzle by identifying the correct position of the sub-images.

References

- Feng-Hui Yao, Gui-Feng Shao, "A shape and image merging technique to solve jigsaw puzzles"
- Daniel J. Ho, Peter J. Olver, "Automatic Solution of Jigsaw Puzzles"
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own MATLAB code.



16. Image registration for panorama

Description

Perform image registration of photos.

Technologies

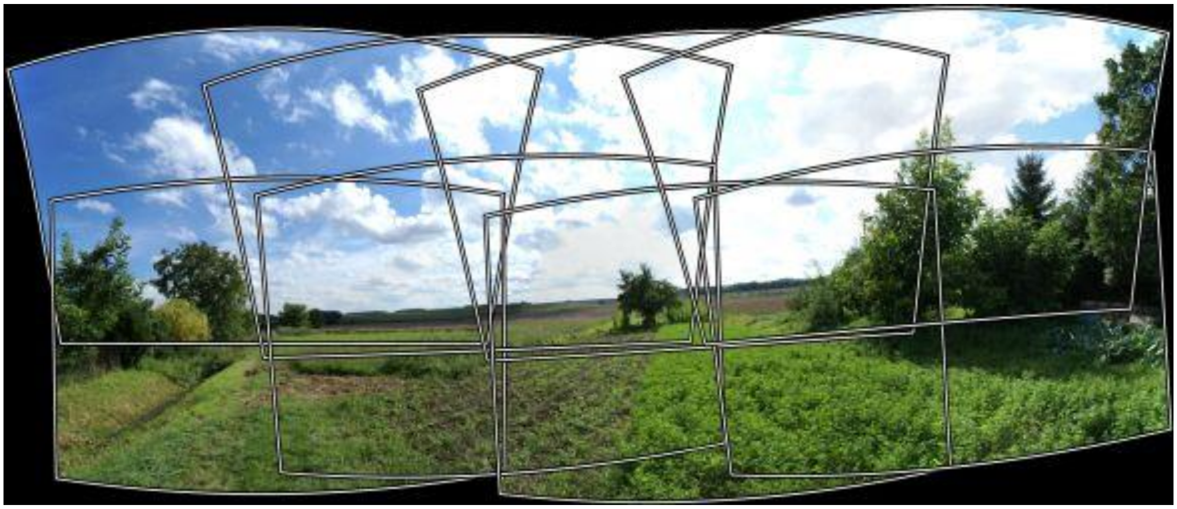
SIFT features, Tie/control points, Take into account: camera perspective, internal (lens) distortions

Take many images (at least 8) from a scene e.g. within the VUB, like the ones shown in the image, and do image stitching automatically based on specific control points.

References

- Mathew Brown, "Automatic Panoramic Image Stitching using Invariant Features"
- Richard Szeliski, "Image Alignment and Stitching: A Tutorial"
- http://nl.mathworks.com/help/images/-automatic-registration.html
- http://www.mathworks.com/help/images/functionlist.html

NOTE: Please create your own **MATLAB** code.



17. Denoising for astrophotography

Description

Denoise astrophotography video

Technologies

Get images from video, apply averaging of frames as denoising technique and remake video. Try the same procedure with and without frame alignment, and compare.

Without alignment:

$$f_{average}^{1} = f^{1}$$

$$f_{average}^{t} = \frac{t-1}{t} f_{average}^{t-1} + \frac{1}{t} f^{t} \qquad t = 2,3,\ldots$$

With alignment:

$$f_{average}^{1} = f^{1}$$

$$f_{average}^{t} = \frac{t-1}{t} f_{average}^{t-1} + \frac{1}{t} Align\left(f^{t}, f_{average}^{t-1}\right) \qquad t = 2,3,\ldots$$

References

- http://www.mathworks.com/help/images/functionlist.html
- Matlab : Image processing toolbox

NOTE: Please create your own MATLAB code.