Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Graphics and Algorithms
TU Wien
Favoritenstr. 9/186-3
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801 - 18661
Fax: +43 (1) 58801 - 18697
E-mail: sek@prip.tuwien.ac.at
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-136 June 21, 2016

Student Papers Image Understanding Academic Year 2015/2016

Lukas Geyer, Johann Götz, Milena Nowak,
Rebecca Nowak, Michaela Tuscher, Gernot Winkler
edited by: Ines Janusch and Walter G. Kropatsch

Abstract

This technical report presents a collection of selected papers, submitted by students in the course "Image Understanding" (VU 186.846) of the Pattern Recognition and Image Processing group during the academic year 2015/2016.

Table of Contents

Article Page

Report: *Benchmarking* (Author: Milena Nowak) 3
Summary: *Benchmarking* (Author: Johann Götz) 12
Discussion: *Benchmarking* (Author: Johann Götz) 15
Report: *Superpixel Segmentation* (Author: Rebecca Nowak) 17
Summary: *Superpixel Segmentation* (Author: Lukas Geyer) 33
Discussion: *Superpixel Segmentation* (Author: Lukas Geyer) 34
Report: *Superpixel Grouping* (Author: Gernot Winkler) 37
Summary: *Superpixel Grouping* (Author: Milena Nowak) 51
Discussion: *Superpixel Grouping* (Author: Milena Nowak) 53
Report: *Computer Vision Models* (Author: Johann Götz) 57
Summary: *Computer Vision Models* (Author: Gernot Winkler) 69
Discussion: *Computer Vision Models* (Author: Gernot Winkler) 71
Report: *Illusions* (Author: Michaela Tuscher) 74
Summary: *Illusions* (Author: Rebecca Nowak) 88
Discussion: *Illusions* (Author: Rebecca Nowak) 89
Report: *Scene Understanding* (Author: Lukas Geyer) 95
Summary: *Scene Understanding* (Author: Michaela Tuscher) 107
Discussion: *Scene Understanding* (Author: Michaela Tuscher) 107

Report - Benchmarking

Bildverstehen SS 2016

Milena Nowak (0927584)

May 22, 2016

1 Introduction

Originally, "benchmarks" were the markings used by surveyors. The term is now (additionally) widely understood to mean the assessment of relative performance. While methods, goals and processes differ depending on the application, the general principle is used across many fields such as economics, computing and marketing. Goal of benchmarking analyses is to produce comparable and replicable results.

In IT and computing, benchmarking is a standard method for the analysis of both hardware and software. This summary focuses mainly on the analysis of software - or rather algorithms. It is usually impractical to look for "the best algorithm", as different approaches work well under different circumstances. The more reasonable task is to find the best performing algorithm for a specific task. As it is difficult to produce meaningful performance evaluation results based only on analytical predictions, benchmarking is used. Depending on the algorithm and it's proposed application, individual assessments and/or comparisons with State-of-the-Art algorithms or standard algorithms known to solve the problem can be done. Some characteristics and features are best tested using "real" data (data collected from a setting similar to that in which the algorithm is supposed to be applied, e.g. Caltech Pedestrian Detection Dataset [13]), others can be analysed in a more effective manner using "simulated" data (e.g. images manipulated in a controlled way to see the effect of very particular variations).

Typical computer vision tasks like facial-, character- or object recognition, segmentation and tracking obviously require different datasets. One of the main difficulties in choosing a benchmarking suite and/or dataset is the balance between comparability and applicability. The more similar methods an algorithm can be compared to, the more conclusions can be drawn about its performance. If a well-know dataset for the task exists, reporting on results using these images (e.g. the Berkeley Segmentation Dataset [15] for image segmentation) will often be more meaningful to other researchers. If no usable dataset exists for the task at hand it is often advisable to publish the dataset (and possible benchmarking suite) in addition to the results of the method. There is no one

"best" dataset or benchmarking suite. The choice depends not only on the algorithm being tested but also the features the test focuses on. While a "master dataset" for every computer vision task imaginable may be desirable for the sake of comparison, there are also drawbacks. As it is, many researchers tune their methods to perform well on well-known datasets, distorting the outcome and therefore significance of the findings. That is one of the reasons existing datasets are often updated after a while. When comparing results, it is therefore important to check which version of the benchmark/dataset was used by the other researchers when comparing results.

2 Berkeley Segmentation Dataset

The Berkeley Segmentation Dataset [15] is one of the most "famous" benchmarking suits for image segmentation currently available. Figure 1 shows sample images from the database including the visualised results of a segmentation algorithm. It consists of both an image database and a benchmarking suite. It includes comparatively few images (300 or 500, depending on the version), depicting a variation of subjects such as objects, people, sceneries etc. (compare figure)1. All of them are available in colour as well as greyscale. As it includes a benchmarking suite, a ground truth is available. The ground truth was labelled by hand by several individuals. Each image was labelled by more than one person. On the website [2], images can be viewed by "human subject" (all images labelled by one particular individual).

Figure 1: Images from Berkeley Segmentation Dataset with visualised segmentation. [1]

There are also lists of algorithms tested using the benchmarking suite. Figure 2 is a screenshot from the website showing a table listing the best-performing algorithms for individual images. The respective papers are cited, enabling researchers to compare not only the numerical results but also the methods used to obtain them (e.g., the best result for image 1, as well as quite a few others, was achieved by Arbelaez et al. [10])

The advantages of this database are included benchmarks, the fact that it is very widely-used, a lot of additional information and the resulting wealth of material to compare results to. Weaknesses could be the relatively small size of the database as well

	Grayscale			Color		
Rank	ID	Image	Best Algorithm [Score]	ID	Image	Best Algorithm [Score]
1	#5 (42049)		gPb-ucm (gray) [0.93 / 0.92]	#6 (167062)		Arbelaez POCV2006 [0.92 / 0.95]
2	#65 (3096)		gPb-ucm (gray) [0.92 / 0.94]	#5 (42049)		Ren et al. NIPS2012 (color) [0.92 / 0.96]
3	#97 (296059)		gPb-ucm (gray) [0.90 / 0.95]	#45 (62096)		gPb-ucm (color) [0.92 / 0.90]
4	#58 (227092)		gPb-ucm (gray) [0.89 / 0.80]	#17 (101085)		gPb-ucm (color) [0.90 / 0.93]
5	#36 (241004)		Ren et al. NIPS2012 (gray) [0.87 / 0.95]	#36 (241004)		Ren et al. NIPS2012 (color) [0.89 / 0.94]

Figure 2: Best-performing algorithms on images. Screenshot from [2]

as the fact that, given its "fame" some of the results may be too focused on achieving impressive results on these particular images.

3 The San Diego Vision Benchmark Suite

Another "classic", the San Diego Vision Benchmark Suite [19] focuses on hardware tests more than analysis of algorithms. It also consists of a dataset and a benchmarking suite. The benchmarks are available in MATLAB and C. The benchmarks include a variety of applications such as Feature Tracking, Image Segmentation, Support Vector Machines (SVM), Scale Invariant Feature Transform (SIFT), etc. It is aimed at system designers (e.g. researching multicore and parallel architectures) of systems focusing on vision-oriented applications. The images the benchmarks are applied to vary in size and subject (see figure 3). It is maintained and updated - the current version and planned updates can be found on their website [9].

Figure 3: Images from the San Diego Vision Benchmark Suite. Image from: [9]

4 Nistér & Stewénius

Nistér & Stewénius created another combined database and benchmarking suite [17]. Their collection of images is larger, currently including 10200 images. They are all the same size, 640x480 pixels. The task the benchmarking suite was built for is object recognition. Therefore, the images are of objects, each object being represented by four separate images taken from different angels, as shown in figure . As it is a very large database, it is also used for other tests, e.g. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration" by Muja and Lowe [16], who use it to research search problems.
Benchmarks are also implemented in MATLAB. The simplest measure of performance counts how many of the four images are in the top-four of the ranking when looking for a particular object. Different subsets are taken from the database to calculate scores (see figure 5). The training is done using SIFT and can be downloaded from their website [8].

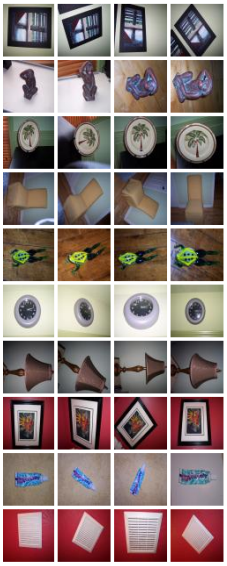


Figure 4: Images from the Nistér & Stewénius Benchmark Suite. [8]

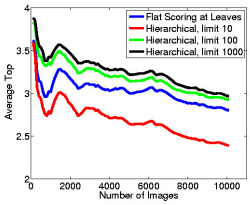


Figure 5: Performance when taking subsets 0:n. [8]

5 Caltech Pedestrian Detection

While the databases discussed previously were all comprised of still images, the Caltech Pedestrian Detection Database [13] is a video database. The application scenary is the detection of humans (pedestrians) in videos. All in all, the material includes 250 000 frames with a resolution of 640x480 and 30Hz. The data was captured by cars driving through the streets. The ground truth was again labelled by hand. It differentiates between occluded, visible and partially occluded pedestrians. They are marked using bounding boxes, see figure . The code used for evaluation and labelling is written in MATLAB and can be downloaded via their website [3].This database is also widely used in its domain, detection of humans in video (see for example Beneson et al. 7).

Figure 6: Frames from the Caltech Pedestrian Detection dataset. [3]

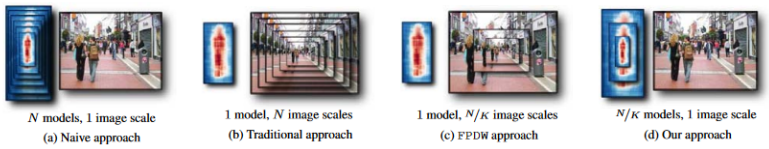


Figure 7: Application example from Benenson et al. Pedestrian Detection. [11]

6 MNIST

MNIST [14] is an older database from 1998 and does not include benchmarking methods. All images are samples of handwiritten digits by 500 different writers. There are 70 000 images in total, divided into a training set of 60 000 and a test set of 10 000. The samples are binary images and normalised to the same size (20x20 pixels) - see figure 8.

On their website [6], they list algorithms that were tested using the database as well as their error rate. They are sorted by category (linear classifiers, SVMs, Neural Nets,...) in a table including performance, preprocessing and reference. The results of the long list of algorithm vary with error rates ranging from 0.23% to 12%. Ciresan et al. "Multi-column deep neural networks for image classification" [12] was one of the examples with the lowest error rates.

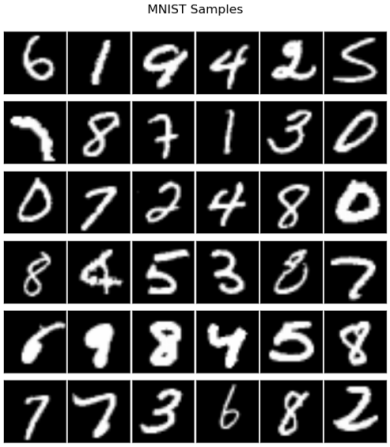


Figure 8: MNIST Dataset Samples. [7]

7 The FERET Database

Like the MNIST database, the FERET Database [18] does not contain benchmarking methods. It is a database of human faces sponsored by the US Department of Defence. There are 14 126 images of 1199 individuals. The data was collected over time, from 1993 to 1998. Many of the images were taken of the same person but some years apart, giving users access to data that few other databases contain.

Their goal was to create a large database independently from the algorithm developers in order to ensure that the results of facial recognition software are meaningful and not tuned to a particular small dataset. They also wanted to collect a set of data that was both large and varied (earlier databases were either a lot smaller, containing a few hundred images at most, or only included full frontal photos).

The program financed by the Department of Defence also included research (they invited researchers to develop facial recognition algorithms using their database). The database was used to evaluate all participating projects as well as algorithms from other organisation to create a referencing framework. Figure 9 shows some sample image from the database. The images (as well as links to other datasets) can be found on the website of the project [4].

Figure 9: Sample images from the FERET Database. [5]

8 Conclusion

The databases referenced in this summary are all well-known and often used. They are, however only a very small sample of existing datasets and benchmarking suits. As mentioned before, the one optimal dataset does not exit - the ideal choice depends on the application. In general it can be concluded that choosing a prominent dataset is a good idea for the sake of comparisons. Using large datasets is also generally desirable as the goal is to show applicability for real scenarios (i.e. the test data should ideally be as varied as reality - though that is obviously an unreachable goal). More important than size or prevalence of a benchmark is its suitability for the algorithm (or more general: application scenario) being tested. It may be necessary or advisable to create your own dataset in some cases, but in many cases, it is unnecessary. There are many more datasets as well as a number of resources attempting to provide an overview over them. The following list provides a starting point for dataset searches and an insight into range of sets available. Aside from such meta-databases, the best places to look for benchmarking material are institutions such as universities and state departments. Most of the databases introduced in the previous sections are from such sources. E.g. both the Department of Defence and Caltech University provide a couple of other datasets.

- `http://riemenschneider.hayko.at/vision/dataset/index.php?sort=ddatechanged`
- `http://www.computervisiononline.com/datasets`
- `http://datasets.visionbib.com/`
- `http://deeplearning.net/datasets/`
- `http://peipa.essex.ac.uk/benchmark/databases/`

* Also, Google may not be your friend but it does contain a lot of information...

References

[1] http://www.timotheecour.com/research.html (Accessed May 18, 2016).

[2] Berkeley segmentation database. https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ (Accessed May 18, 2016).

[3] Caltech pedestrian detection. http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/ (Accessed May 18, 2016).

[4] Feret dataset. http://www.nist.gov/itl/iad/ig/feret.cfm (Accessed May 18, 2016).

[5] Feret dataset samples. http://www.hindawi.com/journals/cmmm/2013/248380/ (Accessed May 18, 2016).

[6] Mnist database. http://yann.lecun.com/exdb/mnist/ (Accessed May 18, 2016).

[7] Mnist dataset samples. http://corpocrat.com/2014/11/09/running-a-neural-network-in-gpu/ (Accessed May 18, 2016).

[8] Nistér & Stewénius Benchmark. http://vis.uky.edu/~stewe/ukbench/ (Accessed May 18, 2016).

[9] San diego vision benchmark suite. http://cseweb.ucsd.edu/groups/bsg/ (Accessed May 18, 2016).

[10] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[11] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. Pedestrian detection at 100 frames per second. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2903–2910. IEEE, 2012.

[12] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[13] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, 2012.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001.

[16] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2:331–340, 2009.

[17] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[18] P Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998.

[19] Sravanthi Kota Venkata, Ikkjin Ahn, Donghwan Jeon, Anshuman Gupta, Christopher Louie, Saturnino Garcia, Serge Belongie, and Michael Bedford Taylor. Sd-vbs: The san diego vision benchmark suite. In *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pages 55–64. IEEE, 2009.

Benchmarking

Benchmarking

- Used to evaluate performance of an algorithm.
- Analytical Analysis often very difficult.
- Using Datasets makes evaluation and comparison easier

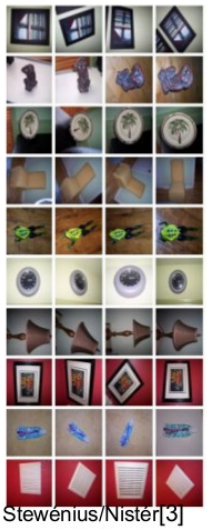
Datasets

- No single best dataset
  - Depends on Application
- The chosen dataset can have huge differences on evaluated performance.
- Sometimes performance evaluation is part of the dataset

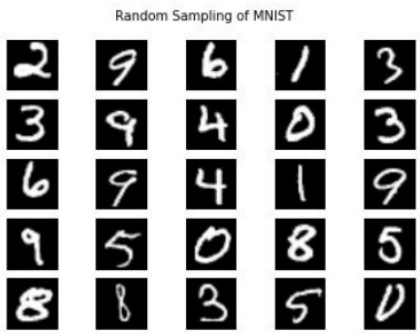
Examples:


Berkley [1]

San Diego Vision [2]


Stewénius/Nistér[3]


Caltech[4]



MNIST [5]

Berkeley

- Used for image segmentation and boundary detection
- Hand-labeled ground truth
- Very popular dataset
- Included Statistical Tests

San Diego Vision Benchmark Suite

- Includes Benchmarks in Matlab and C
- Used for different applications
  - Motion, Tracking
  - Stereo Vision
  - Image Analysis
  - Image Understanding
  - Image Processing and Formation

Nistér & Stewénius

- Used for object recognition
- Includes Benchmarking
- Consists of groups of four images for each object (different viewing angles)
- Performance Evaluation in Matlab
- Includes MSER, SIFT, Visual Words

Caltech Pedestrian Detection

- Used for video analysis
- Has annotated bounding boxes for pedestrians

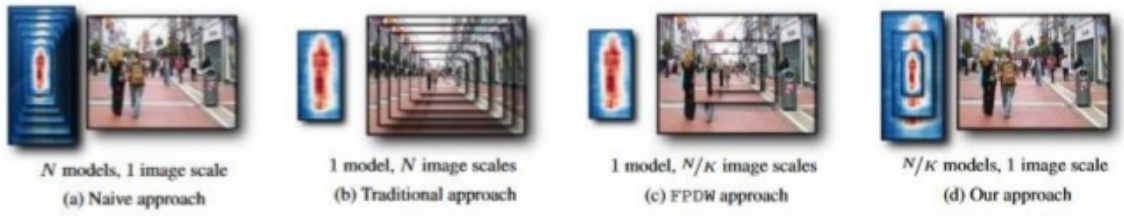


Figure 2: Different approaches to detecting pedestrians at multiple scales.

MNIST

- Used for handwritten character recognition
  - Linear Classifiers, SMVs,..

FERET Database

- Used for face recognition
- Data was collected since the nineties (by Department of Defense)

Semantic 3D

- http://www.semantic3d.net
- Large-Scale point cloud classification benchmark+
- Semantic-8
  - 8 class labes, 2 billion points
- Reduced-8
  - 70 million points

ImageCLEF

- http://www.imageclef.org/
- Scaleable Concept Image Annotation Challenge
  - Held every year
- Image annotation and retrieval tasks
  - Medical Classification
  - Image annotation
  - Handwritten scanned document retrieval

Finding more datasets

- **Keyword, tag**
  - http://riemenschneider.hayko.at/vision/dataset/index.php?sort=ddatechanged
-
- **Title, image, description**
  - http://www.computervisiononline.com/datasets
-
- **By category**
  - http://datasets.visionbib.com/
  - http://deeplearning.net/datasets/
  - http://peipa.essex.ac.uk/benchmark/databases/

Discussion

Straw-man Algorithm

Algorithms usually have a specific application. Which means that there cannot be a “best” algorithm, because the quality of a method heavily depends on its application. Furthermore a best algorithm might mean less competition and fewer new methods being developed.

Black-box / White-box Evaluation

The multitude of available benchmarks resulted in mostly black-box evaluation, which means only input and output are relevant for testing. On the other hand white-box evaluation also examines intermediary results and how the algorithm obtains the end result.

Only looking at the end result may produce algorithms that detect wrongs parts of the image (background instead of foreground and vice versa), but due to the composition of the benchmark set still score favorably.

Special Applications with very few datasets

Very special applications with unusually few datasets may not be possible to solve with machine learning algorithms for this reason (bank robberies,...).

Ground Truth Critique

The way a ground truth sometimes is created (receiving money for a number of annotated images or even crowdsourcing) may result in not ideal situations. Sometimes annotations are incomplete or missing, which could result in a low score for an algorithm, which actually works very well. This means don’t just accept the ground truth, it should be analyzed as well.

Datasets usually big and ungrouped

Grouping dataset further based on some properties would be advantages to developing methods for special applications, that could work for a limited set of data exceptionally well. At the moment there is very few development in that direction and most people focus on results of benchmarks of big datasets.

Artificially increasing Benchmark Scores

It is possible to artificially increase benchmark scores by deliberately omitting parts of the datasets or doing aggressive performance tuning for a specific dataset, which does not increase the overall performance of an algorithm.

[1] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on (Vol. 2, pp. 416-423). IEEE.

[2] Venkata, S. K., Ahn, I., Jeon, D., Gupta, A., Louie, C., Garcia, S., ... & Taylor, M. B. (2009, October). SD-VBS: The San Diego vision benchmark suite. In Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on (pp. 55-64). IEEE.

[3] Nister, D., & Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In Computer vision and pattern recognition, 2006 IEEE computer society conference on (Vol. 2, pp. 2161-2168). IEEE.

[4] Dollar, P., Wojek, C., Schiele, B., & Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(4), 743-761.

[5] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[6] Phillips, P. J., Wechsler, H., Huang, J., & Rauss, P. J. (1998). The FERET database and evaluation procedure for face-recognition algorithms. Image and vision computing, 16(5), 295-306.

Image Understanding

Superpixel Segmentation

Rebecca Nowak (0626227)

Vienna, 21st April 2016

Contents

1 Introduction **3**

2 Normalised Cuts Algorithm **4**

3 Mean Shift **5**

4 SLIC Superpixels **5**
4.1 The SLIC Algorithm 5
4.2 Sample Images Segmented with the SLIC algorithm 6

5 Applications of superpixel segmentation **14**
5.1 Recovering Human Body Configurations: Combining Segmentation and Recognition 14
5.2 Automatic Cloud Detection for All-Sky Images Using Superpixel Segmentation 14
5.3 Robust Superpixel Tracking 14
5.4 Image Classification via Object-Aware Holistic Superpixel Selection . . . 14
5.5 Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera 15
5.6 Complex Networks Driven Salient Region Detection based on Superpixel Segmentation 15

1 Introduction

Pixels are grouped into perceptually meaningful atomic regions (superpixels), which are then used instead of pixels for subsequent image processing tasks like computing image features. [3] One reason for the use of superpixels, is to reduce computational complexity by capturing image redundancy. Another is image segmentation. According to [3] speed, the ability to adhere to image boundaries and impact on segmentation performance are the main criteria for the quality of a superpixel method. Another quality criterion is the flexibility of the algorithm, providing parameters to influence e.g. number or compactness of superpixels.

Figure 1: Images segmented using SLIC into superpixels of size 64, 256, and 1,024 pixels (approximately)[3]

[3] categorises superpixel methods into two approaches: Graph-based algorithms and gradient ascent methods. In a graph-based method, each pixel is treated as a node in a graph. The edge weights between two nodes correspond to their similarity. Superpixels are created by minimising a cost function defined over the graph. Gradient ascent based algorithms start from a rough initial clustering of pixels, and then iteratively refine the clusters until come convergence criterion is met.

2 Normalised Cuts Algorithm

[8] describes the normalised cuts algorithm for superpixel segmentation. The underlying assumption is that perceptual grouping is about extracting the global impressions of a scene. The image is represented by a weighted undirected graph. Each pixel corresponds to a node in the graph. An edge connects each pair of pixels, and the edge weight is the similarity between the two pixels. The goal is to find a partition that maximises the similarity inside a set of nodes, and minimises the similarity between the sets. The criterion to achieve this presented in [8] is the normalised cut (see equation 1. A cut is the dissimilarity between two pieces of a graph, calculated by adding up the weight of the edges that have been removed.

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{1}$$

The optimal bipartitioning of a graph is the one that minimises the cut value. [11] use the minimum cut to partition a graph in to k subgraphs by recursively finding the minimum cuts that bisect the existing segments. The minimum cut criterion favours cutting small sets of isolated nodes. See figure 2 for an example.

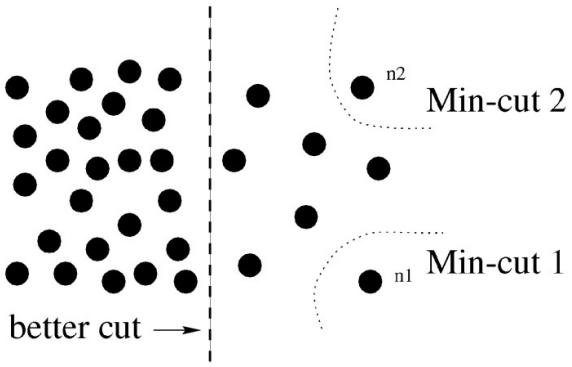


Figure 2: A case where a minimum cut gives a bad partition.[8]

To avoid this bias for partitioning out small sets of points, [8] propose the normalised cut (see equation 2).

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{2}$$

with

$$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t) \tag{3}$$

The cut cost is thus computed as a fraction of the total edge connections to all the nodes in the graph. The normalised cut is used as the partition criterion. This criterion

can be computed efficiently by solving a generalised eigenvalue problem (see [8] for details). According to [3] the normalised cuts algorithm produces very regular, visually pleasing superpixels but has relatively poor boundary adherence.

3 Mean Shift

[2] introduces the mean shift algorithm for superpixel segmentation. Each pixel is assumed to be a sample from an unknown probability density function. To efficiently find the peaks (modes) without computing the complete function explicitly, multiple restart gradient descent is used. The gradient vector is estimated iteratively at a point, from where an uphill step is taken to the next one. Pixels that converge to the same peak define the superpixels. According to [3] the mean shift algorithm produces irregularly shaped superpixels, superpixels have non-uniform size, and the algorithm gives no control over the amount, size or compactness of the superpixels.

4 SLIC Superpixels

Simple linear iterative clustering is an adaptation of k-means for superpixel generation and was introduced by [3].

4.1 The SLIC Algorithm

The image is converted to the CIELAB colour space. One input parameter is the desired number of approximately equally sized superpixels k. k initaial cluster centres on a regular grid. The cluster centres are moved to the lowest gradient position in a 3x3 neighbourhood. This avoids centring a superpixel on a an edge and reduces the chance of noisy pixels. Each pixel associated with its nearest cluster centre according to distance measure D. The distance measure is described in formulas 4, 5 and 6.

$$D = \sqrt{d_c^2 + (\frac{d_s}{S})^2 m^2} \tag{4}$$

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \tag{5}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{6}$$

S is the sampling interval, and therefore the maximum spatial distance expected within a given cluster. The constant m is used to weigh the relative importance between colour similarity and spatial proximity. For each pixel in a 2Sx2S region around each cluster centre, the distance between that pixel and the cluster centre is calculated. If the distance is smaller than the one already assigned to that pixel, the distance and corresponding cluster centre are assigned to the pixel. This is repeated until the residual error is smaller than a certain threshold. Pseudocode for the algorithm can be found in [3]. The

algorithm does not guarantee connected regions, so a clean-up stage is necessary. A connected components algorithm is used to assign single pixels and small regions to the nearest cluster centre.

4.2 Sample Images Segmented with the SLIC algorithm

I tested an implementation of the slic suerpixel algorithm by Peter Kovesi, Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia with images from the berkeley segmentation dataset [5]. The function takes the following input parameters:

- **im** the image to be segmented
- **k** the number of desired superpixels
- **m** weighing factor, large value to enforce superpixels with more regular shapes
- **seRadius** Regions morphologically smaller than this are merged with adjacent regions
- **colopt** mean or median, indicating how the cluster centres should be computed
- **mw** optional median filtering window size

Figures 3 to 8 show the influence of the weighing factor m. A bigger m means more regular superpixels but less boundary adherence. Figures 9 and 10 show a finer segmentation with double the number of superpixels. The effect of salt and pepper noise on the segmentation results are presented in figures 13 and 14. See figures 11 and 12 for a segmentation of the same image with the same parameters but without noise added to the image. The results of adding Gaussian white noise with constant mean and variance to the image before segmentation can be seen in figures 15 and 16. Whereas the salt and pepper noise added to figures 13 and 14 did not change the superpixel boundaries much, the impact of the Gaussian noise added in figures 15 and 16 is significant.

Black lines represent the superpixel boundaries. The boundaries are shown before and after the clean-up stage.

Figure 3: k = 200, m = 5, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

Figure 4: k = 200, m = 5, seRadius = 2, colopt = 'mean', mw = 5; after clean-up

Figure 5: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

Figure 6: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; after clean-up

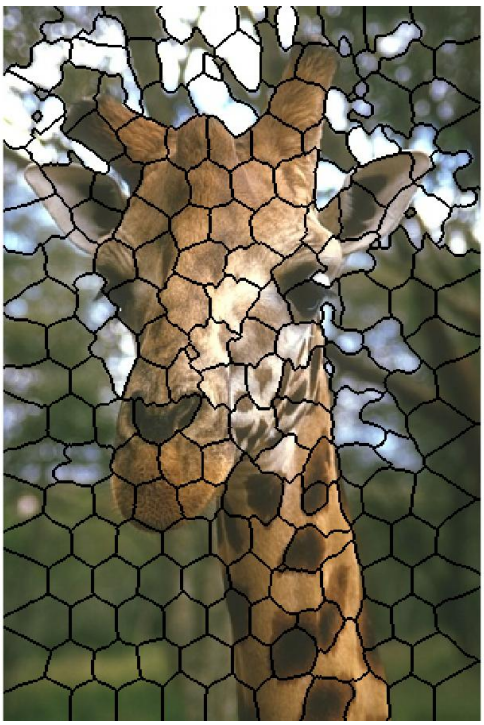


Figure 7: k = 200, m = 40, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

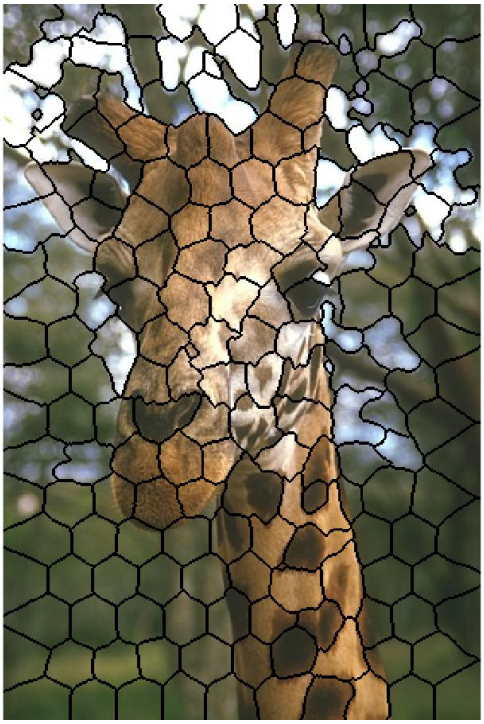


Figure 8: k = 200, m = 40, seRadius = 2, colopt = 'mean', mw = 5; after clean-up

Figure 9: k = 400, m = 10, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

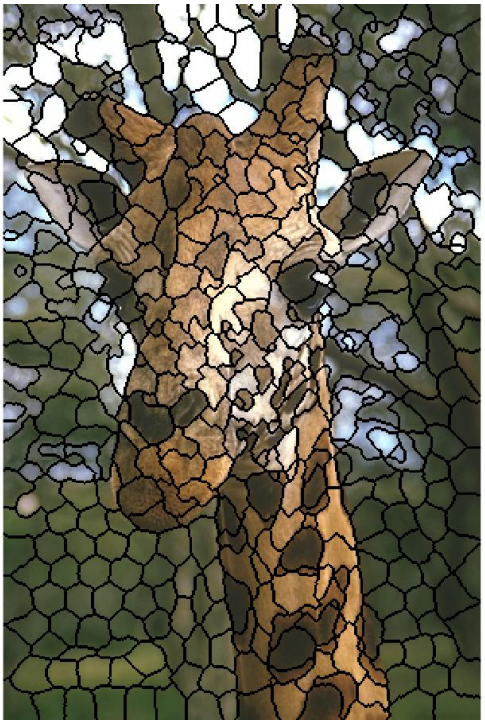


Figure 10: k = 400, m = 10, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

Figure 11: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

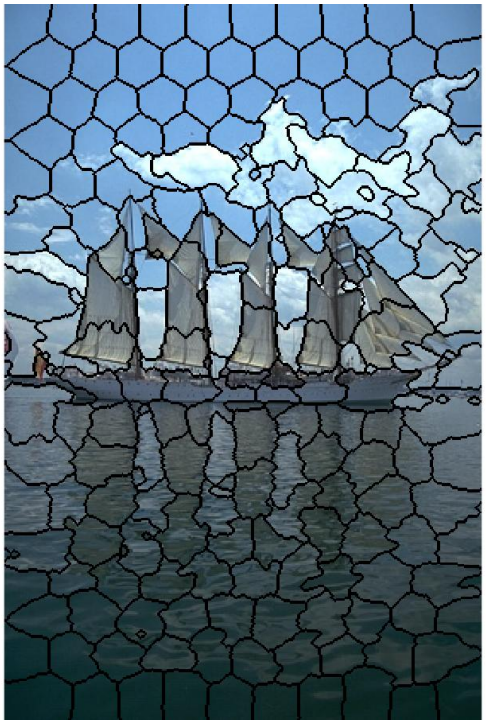


Figure 12: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; after clean-up

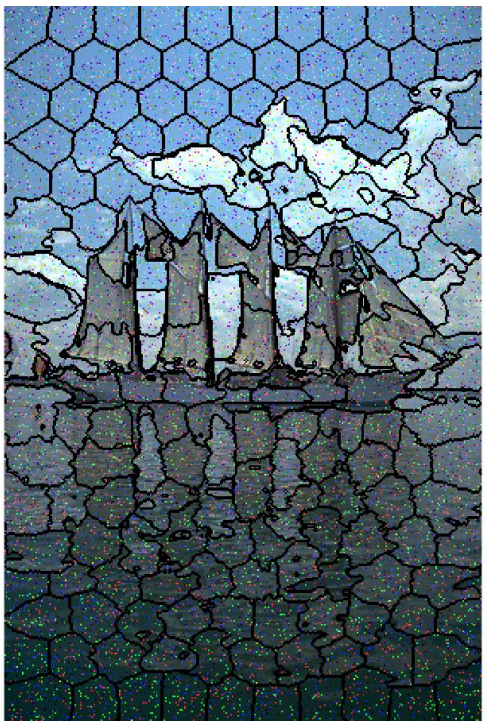


Figure 13: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

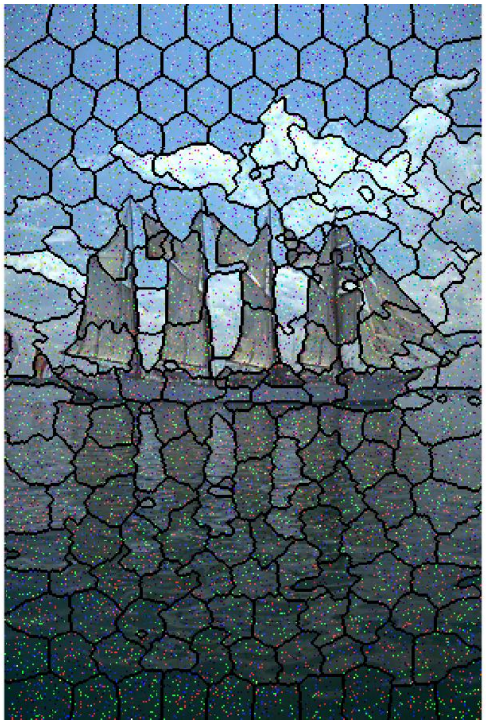


Figure 14: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; after clean-up

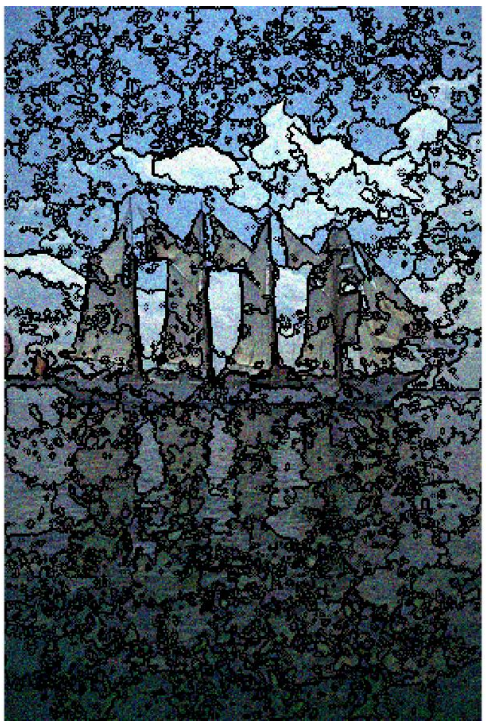


Figure 15: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; before clean-up

Figure 16: k = 200, m = 10, seRadius = 2, colopt = 'mean', mw = 5; after clean-up

5 Applications of superpixel segmentation

Superpixel algorithms are widely used for computer vision tasks like classification, tracking or salient region detection. Some papers are presented below as examples for the application of superpixel segementation algorithms.

5.1 Recovering Human Body Configurations: Combining Segmentation and Recognition

[6] use superpixels as part of their algorithm for recovering human body coniňĄgurations. Images are segmented into superpixels with the normalised cuts algoirthm. In a segmentation with relatively few superpixels per image, salient upper and lower limbs as well as potential head and torso positions are detected. The parts are combined into partial body configurations consisting of a few half-limbs (upper and lower legs or arms) and a torso. Impossible configurations are pruned away by enforcing global constraints such as relative scale and symmetry in clothing. In the final stage, another superpixel segmentation with a higher number of superpixels is used to complete partial configurations by combinatorial search in the space of superpixels to recover full body configurations.

5.2 Automatic Cloud Detection for All-Sky Images Using Superpixel Segmentation

In [4] superpixel segmeentation is used to distinguish between clouds and clear sky in all-sky images. Each pixel needs to be labelled as either cloud or clear sky. The image is segmented into superpixels, then a local threshold value for each superpixel is obtained and a threshold matrix is computed by interpolating the thresholds of all the local thresholds. The threshold matrix and an R-B feature image are compared to achieve the final classification.

5.3 Robust Superpixel Tracking

[12] use superpixels for tracking objects. During the training stage, the segmented superpixels are grouped for constructing a discriminative appearance model to distinguish foreground objects from cluttered backgrounds. For tracking, a confidence map at superpixel level is computed using the appearance model to obtain the most likely target location with maximum a posteriori estimates. The appearance model is constantly updated to account for variation.

5.4 Image Classification via Object-Aware Holistic Superpixel Selection

[10] propose a method to automatiocally select the discriminative superpixels of an image for the purpose of image classification. The interference of a cluttered background is

reduced by only considering selected superpixels.

5.5 Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera

In [9] superpixels are used for gesture recognition with the kinect depth camera. The depth and skeleton information from the kinect are used for hand extraction. The hand shapes (including depth information) are represented with superpixels, which retain the overall shapes and colour. The superpixel earth mover's distance is proposed to measure the dissimilarity between hand gestures.

5.6 Complex Networks Driven Salient Region Detection based on Superpixel Segmentation

[1] use superpixels for salient region detection. The image is decomposed using the SLIC superpixel algorithm. The authors argue that the labxy space used by SLIC is appropriate for their application, that SLIC generates compact and regular sized superpixels and that it removes and isolates the input image from unnecessary noise. Another algorithm considered was the lazy random walks algorithm [7] which promises better performance, but as SLIC has better complexity the authors chose SLIC as the most suitable superpixel algorithm for their use.

References

[1] Alper Aksac, Tansel Ozyer, Reda Alhajj. Complex networks driven salient region detection based on superpixel segmentation. 2016.

[2] Dorin Comaniciu, Peter Meer, and Senior Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[3] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[4] S. Liu, L. Zhang, Z. Zhang, C. Wang, and B. Xiao. Automatic Cloud Detection for All-Sky Images Using Superpixel Segmentation. *IEEE Geoscience and Remote Sensing Letters*, 12:354–358, February 2015.

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[6] G. Mori, Xiaofeng Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–326–II–333 Vol.2, June 2004.

[7] J. Shen, Y. Du, W. Wang, and X. Li. Lazy Random Walks for Superpixel Segmentation. *IEEE Transactions on Image Processing*, 23:1451–1462, April 2014.

[8] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000.

[9] Chong Wang, Zhong Liu, and Shing-Chow Chan. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE Trans. Multimedia*, 17(1):29–39, 2015.

[10] Zilei Wang, Jiashi Feng, Shuicheng Yan, and Hongsheng Xi. Image classification via object-aware holistic superpixel selection. *IEEE Trans. Image Processing*, 22(11):4341–4352, 2013.

[11] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov 1993.

[12] F. Yang, H. Lu, and M. H. Yang. Robust superpixel tracking. *IEEE Transactions on Image Processing*, 23(4):1639–1651, April 2014.

Superpixel - Segmentation

Presentation by Rebecca Nowak, 0626227
on 12th April 2016

Protocol by Lukas Geyer, 1026408

Short summary of the presentation:

In images, neighbouring pixels often have similar color, therefore the image representation as a regular pixelgrid is somewhat redundant. To reduce the complexity for subsequent image processing tasks, one can cluster visually coherent regions to *superpixels*. The segmentation of the image into these superpixels might be performed using a graph-based approach, like the normalized cut (ncut), or by using a gradient-ascent-based approach, as it was done in the mean shift method.

In addition to the ncut and the mean shift algorithm, the presentation also explained the Simple Linear Iterative Clustering (SLIC) algorithm and mentioned a proposal of Microsoft Research Cambridge, concerning resolution-independent superpixels.

ncut: $O(N^{3/2})$; recursively finds cuts in the image's graph representation; results in regular superpixels, but with poor boundary adherence
mean shift: $O(N^2)$; no parameters; starting from each pixel, follow gradient direction, pixels converging to same peak form a superpixel; results in irregularly shaped superpixels of different sizes
SLIC: O(N); start with evenly distributed cluster centers, assign pixels to cluster centers that are nearest in the LABXY-space, relocate cluster centers to mean of their assigned pixels, final cleanup step needed since connected components are not guaranteed

Some possible applications for superpixels are image classification, object tracking, cloud detection, hand gesture recognition, salient region detection, and the segmentation of images depicting humans, followed by pose recognition.

Additionally mentioned details that are not written in the slides: The tested implementation of SLIC showed that the cleanup-phase of the algorithm is the slowest part.

Protocol of the discussion:

Runtime-Complexity: For SLIC, the runtime complexity is given with O(N). The innermost loop actually has a complexity of $O(S^2)$, with S being the maximal side length of a superpixel, so $O(S^2 N)$ might be a more precise declaration - but since S is constant, it can be neglected.
Another point of criticism was, that the given complexity O(N) ignores the fact, that the loop with N iterations is repeated until the error converges - the number of repetitions R is not defined to be bounded by a constant value, so it is questionable to neglect this factor in the complexity. A given complexity of O(R N) might be more adequate. Nevertheless, a few tests of SLIC always led to a fast convergence within $R \approx 8$ iterations. A second argument supporting the annotation of the complexity as just O(N) is, that the iterative nature of the algorithm allows to define a runtime boundary as a second termination criterion. Even if the error did not converge in time, the intermediate result will be valid, although not optimal.
In this context was also mentioned, that a statement about the parallel compexity of an algorithm would also be desirable, especially for image processing algorithms that might be implemented on the GPU.

Quality of results: SLIC was tested with a variety of parameter combinations, with the purpose of finding out how good/bad the results can get. Only the best results were shown in the presentation, but for a good impression of the strengths and weaknesses of an algorithm, it would be desirable to also provide negative examples: cases, in which the algorithm performs poorly. An example for a case, which might cause problems for segmentation algorithms, is a grayscale ramp: there is no exact region border separating the dark from the bright side of the ramp. Especially algorithms that rely on local information will have trouble to draw the separating line in a sensible manner, e.g. in the middle of the grayscale ramp. In the task of cloud detection, such cases might appear frequently.
In addition to providing negative examples, the used parameters should be mentioned next to the image of the algorithm's result.

Possible improvements - image noise: The algorithms dedicated to creating superpixels might be prone to image noise and other disturbance factors like snowfall. SLIC, for example, shifts the cluster centers after every iteration using the mean value of the cluster - the usage of the median instead of the mean value might reduce the algorithm's vulnerability to image noise. This would also allow to

state a maximum tolerance of disturbed pixels inside a superpixel, before the results are influenced.

Possible improvements - granularity: The superpixels can be created in diverse granularity - instead of first specifying a fixed granularity that is most appropriate for the task at hand, and then computing the superpixels just in this level of detail, a hierarchy of superpixels with different granularity could be computed. First, the algorithm creates superpixels of the finest granularity. Then it clusters these superpixels into bigger superpixels, resulting in a more coarse resolution. The superpixels of each level of detail can be repeatedly combined to bigger superpixels, until the image consists of only one superpixel containing all pixels.

Similarities to well-known procedures: In the discussion was pointed out that the SLIC approach has much in common with k-means.
The computation of the exact reconstruction error, needed to evaluate the results of the superpixel-segmentation using the method of Microsoft Research Cambridge, shows similarities to the process of constructing a laplacian pyramid (the deviation of the reconstructed image from the original is computed).

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-theart superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2274–2282,2012.

[2] S. Liu, L. Zhang, Z. Zhang, C. Wang, and B. Xiao. Automatic Cloud Detection for All-Sky Images Using Superpixel Segmentation. IEEE Geoscience and Remote Sensing Letters, 12:354–358, February 2015.

[3] Alper Aksac, Tansel Ozyer, Reda Alhajj. Complex Networks Driven Salient Region Detection based on Superpixel Segmentation. 2016.

[4] J. Shen, Y. Du, W. Wang, and X. Li. Lazy Random Walks for Superpixel Segmentation. IEEE Transactions on Image Processing, 23:1451–1462, April 2014.

[5] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8):888–905, August 2000.

[6] Dorin Comaniciu, Peter Meer, and Senior Member. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24:603–619, 2002.

[7] F. Yang, H. Lu, and M. H. Yang. Robust superpixel tracking. IEEE Transactions on Image Processing, 23(4):1639–1651, April 2014.

[8] G. Mori, Xiaofeng Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–326–II–333 Vol.2, June 2004.

[9] Richard Szeliski. Computer Vision: Algorithms and Applications. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[10] Zilei Wang, Jiashi Feng, Shuicheng Yan, and Hongsheng Xi. Image classification via object-aware holistic superpixel selection. IEEE Trans. Image Processing, 22(11):4341–4352, 2013.

[11] Chong Wang, Zhong Liu, and Shing-Chow Chan. Superpixel-based hand gesture recognition with kinect depth camera. IEEE Trans. Multimedia, 17(1):29–39, 2015.

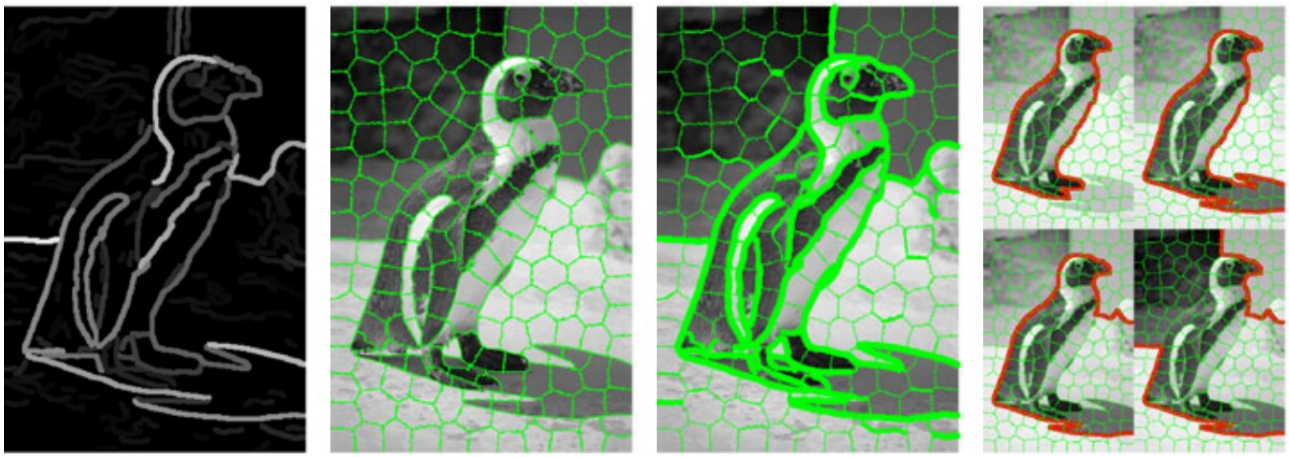
Image Understanding SS16 Winkler Gernot 0929255

Superpixel Grouping

Optimal Contour Closure by Superpixel Grouping [1][2]

The Contour closure algorithm tries to link together fragmented contours to separate an object from it's background.

Basic approach



(a) (b) (c) (d)

As input a contour image is used (a).

As second Input a superpixel segmentation of the image is also needed (b).

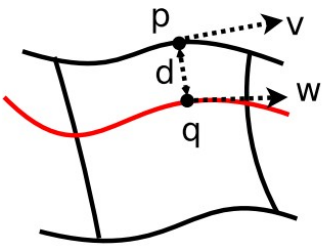
Image (c) shows a new measurement that describes how good the superpixel boundaries match the contours of the image. This is called the gap measurement.

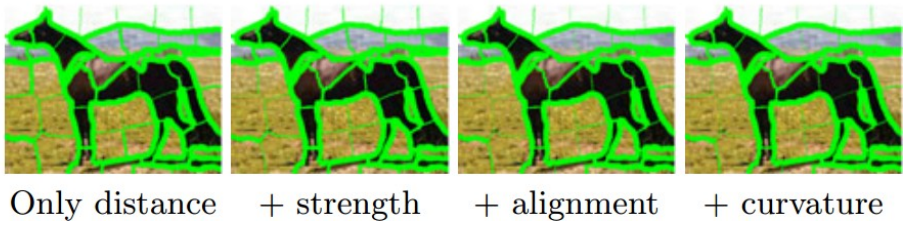
With a globally optimized cost function a „best match“ can be found. Image (d) shows different results with increasing cost (left to right, top to bottom).

Problem forumulation: Search for a subset of superpixels with borders that have strong support of the image contours. It is important to prefer spatial coherent superpixel subsets with the reformulated problem. The goal is therefor to find the maximal set of superpixels which have a high spatial coherence whose boundaries have strong support of the image contours.

How does this work? The collective boundary of a superpixel set must be quantified on how well it fits the image contours. This is called the 'gap measurement'.

To evaluate the 'gap measurement' the following 4 values are used as a feature vector:

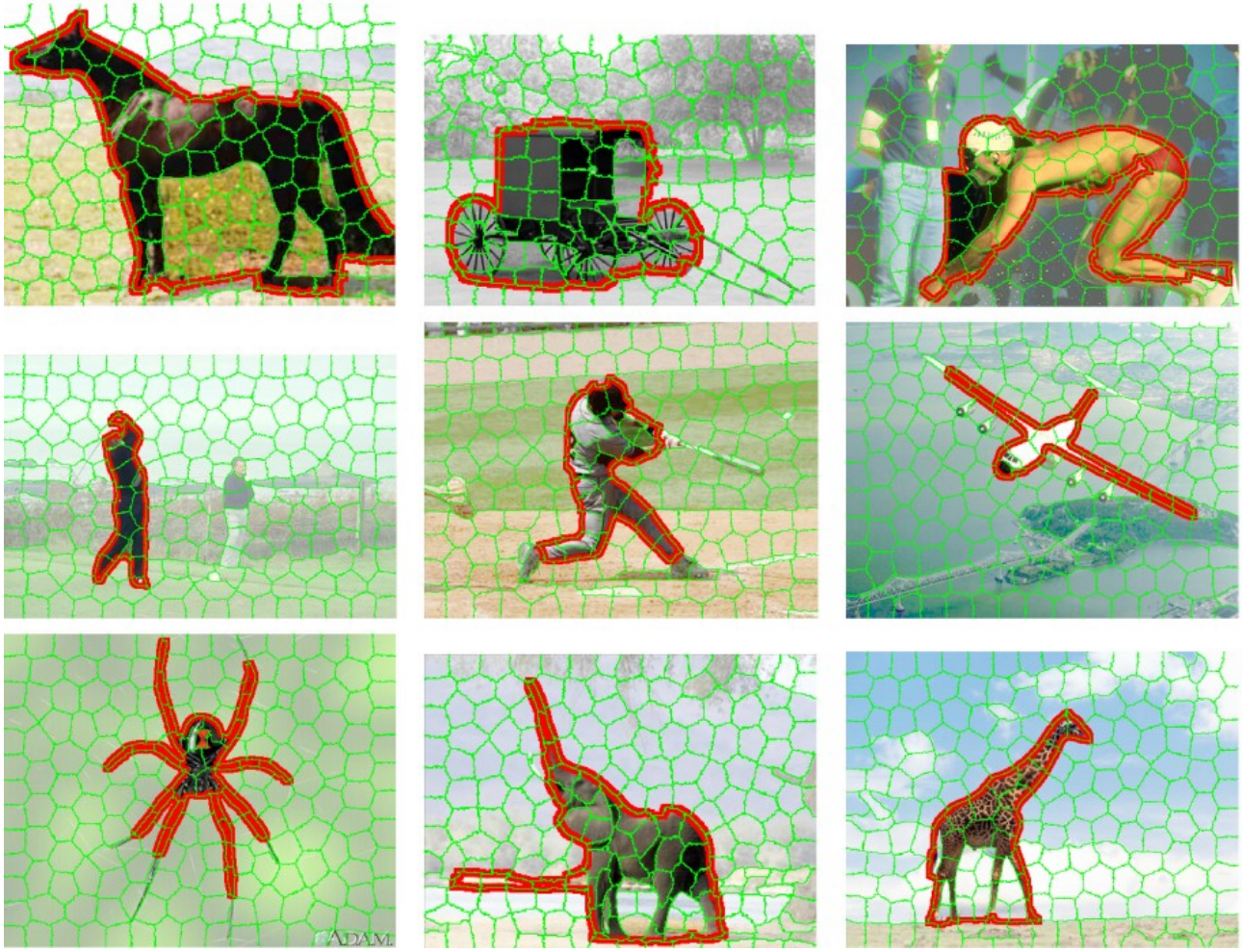




- **Distance**: The distance to the nearest image edge (d in the above image, = distance between p and q)
- **Strength**: The strength of an edge also has an influence, a strong edge provides stronger image evidence.
- **Alignment**: The tangents (v and w) from the superpixel edge (black) and the image edge (red) are compared. The angle between them provides an information on how well those fit together.
- **Curvature**: The squared curvature is also taken into account

Subsets of the features where also evaluated, but the authors of the paper[1] came to the conclusion that these 4 features provided the best results.

Results



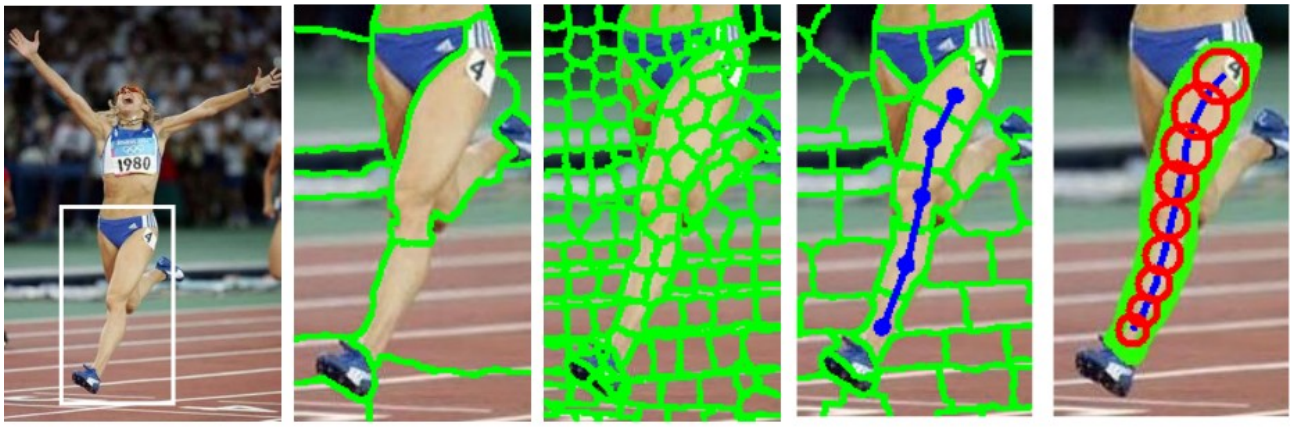
The image above shows a few results of segmentations with the contour closure algorithm.

What can be concluded from those results? First, there where no holes in the segmentation. Also the shadow of an object is classified as strong edge and part of the boundary, as seen in a few of the images (e.g. horse, elefant).

Symetric Part Detection and Grouping [6][2]

The goal of this algorithm is to recover the symmetric part structure of an object by the principle that a skeleton is defined as the locus of medial points, or maximally inscribed disks. A medial point is a point that has more than one closest neighbor on the objects boundary. The medial axis[7] is the set of medial points.

Multiple superpixels segmentations at different resolutions are used and a superpixel represents a medial point hypotheses.

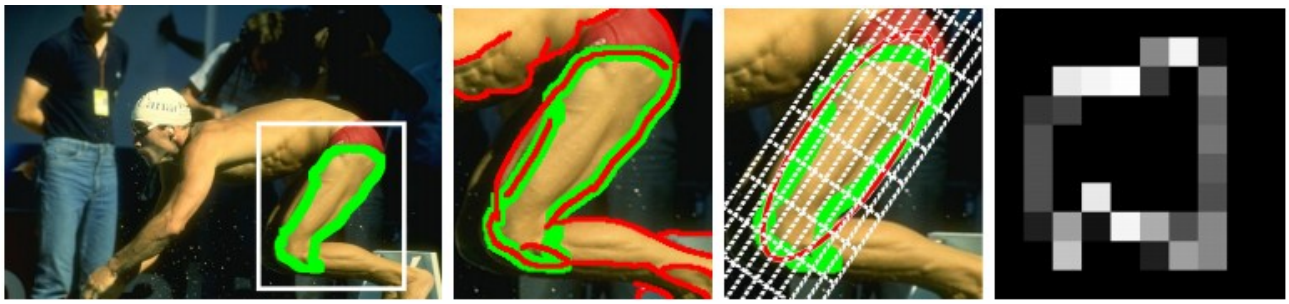


a) shows the original image and the zoom window. In image b,c,d different superpixel resolutions are shown. While b and c have a too low or too high superpixel resolution to capture the object in this case, in image d) some superpixels where identified as medial point and grouped together to form a medial branch (e).

To group superpixels together an affinity is calculated, this affinity composes of two parts. A shape and an appearance component. Both are trained classifiers that learned with manually labeled superpixel pairs.

The appearance component is a 406 dimensional vector, including values like absolute difference and variance in both RGB and HSV color. For the appearance affinity a logistic regressor with L1-regularization is trained.

The shape component is calculated by grouping two superpixels together, transforming them into normalized coordinates to achieve rotation and scale invariance. The shape classifier is then retrieved from a 10x10 2D-histogram. This whole process is explained in the image below:

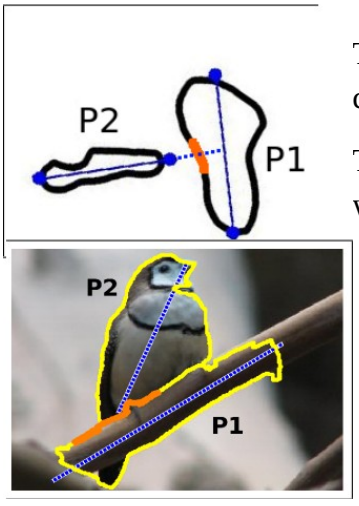


a b c d

Figure 3. Superpixel shape feature: (a) boundary of two adjacent superpixels representing two medial point hypotheses; (b) a blow-up of the two superpixels, in which the boundary of their union (green) defines an underlying image edge distribution (red); (c) the normalized scale- and orientation-invariant coordinate system (grid in white) based on the ellipse (red) fitted to the superpixel union; (d) the shape-context-like feature that projects image edgels, weighted by edge strength, into this coordinate system.

After finding medial branch clusters at each scale those must be assembled if the are likely to belong to the same object.

It is possible branches are redundant, the probability of redundancy includes the following values: overlap in area, overlap in boundary and appearance similarity.



The attachment of two branches is also determined with a learned classifier and graph clustering.

The image on the left shows the attachment boundary which together with the appearance similarity is used to determine the part affinity.

Results

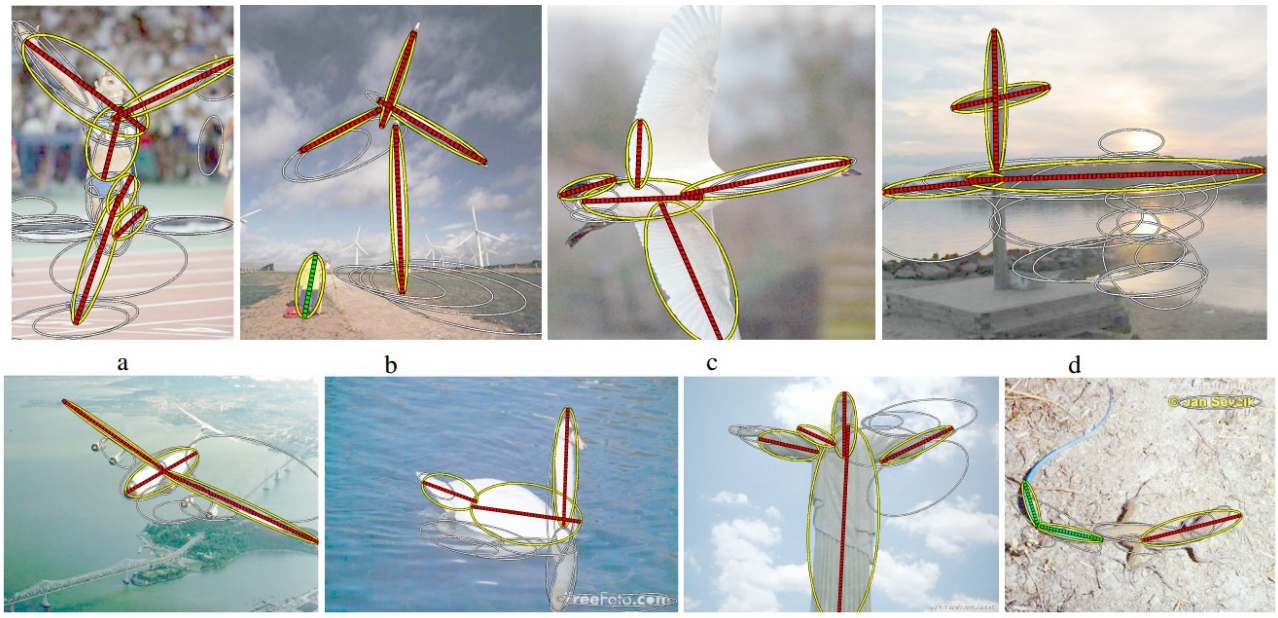


Figure 7. Detected medial parts and their their clusters. In each image, we show the most prominent cluster, showing the medial branch (red dashed) and extent (yellow ellipse) of each abstract part. In some images, a secondary part cluster is shown with green medial branches. All other parts are shown faintly in grey.

Segmentation Using Superpixel: A Graph Partitioning Approach [3]

A main idea is to use visual cues provided by superpixels for effective image segmentation.

This algorithm uses multiple superpixel over-segmentations. Superpixel cues: Superpixels within a superpixel are very likely to belong to a coherent region.

Smoothness cues: neighbouring pixels that are close in feature space tend to belong to the same coherent region.

These cues should be encoded with a bipartite graph.

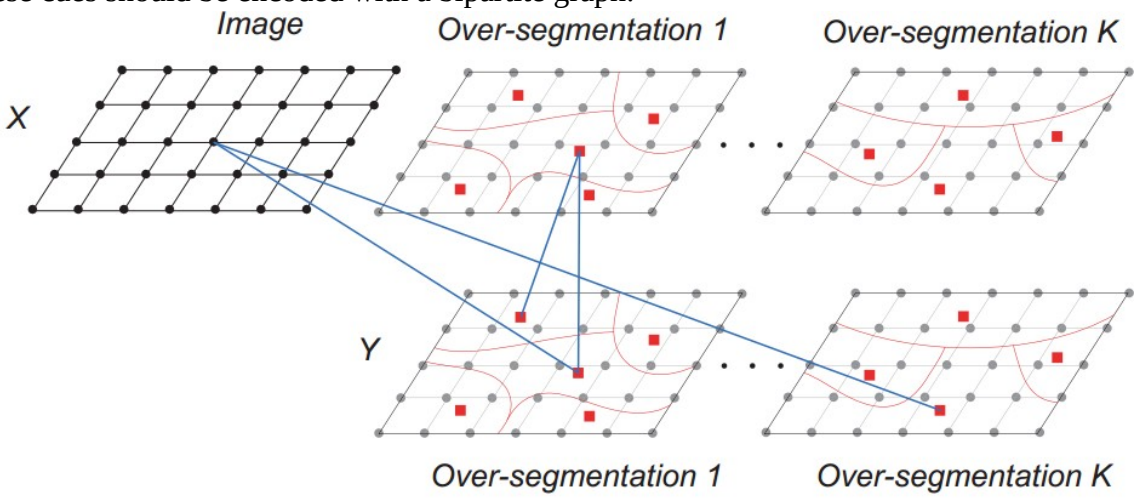


Figure 2. The proposed bipartite graph model with K over-segmentations of an image. A black dot denotes a pixel while a red square denotes a superpixel.

Multiple over-segmentations are used and a pixel is connected to a superpixel if it is contained in it's region. Further superpixels that are close in feature space are connected together, this compensates for smoothness of pixel across superpixels.

Affinity between superpixels

The affinity of two superpixels is used as edge weights for the graph. The so called across-affinity matrix is calculated the following way:

$$b_{ij} = \alpha, \text{ if } \mathbf{x}_i \in \mathbf{y}_j, \mathbf{x}_i \in I, \mathbf{y}_j \in \mathcal{S};$$

$$b_{ij} = e^{-\beta d_{ij}}, \text{ if } \mathbf{x}_i \sim \mathbf{y}_j, \mathbf{x}_i \in \mathcal{S}, \mathbf{y}_j \in \mathcal{S};$$

$$b_{ij} = 0, \text{ otherwise,}$$

b_{ij} is an element of the matrix. I denotes the Image and S the superpixel segmentations. This means the weight between a pixel and a superpixel is set to a constant called α. If two superpixels are adjacent and similar (denoted by the ~ relationship) their weight is calculated with the formula of the second row. β is a constant and d_{ij} denotes a distance in feature space (which can be the euclidean distance of the average color in RGB).

Otherwise there is no connection and therefor a zero weight.

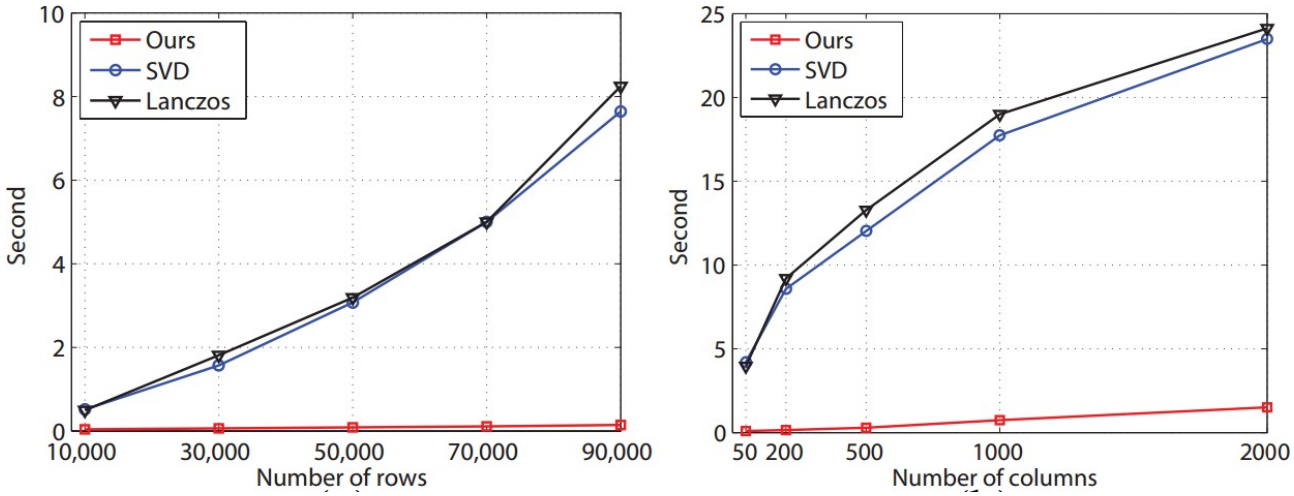
The next step is to partition the graph into k groups. The basic idea is to use **Spectral clustering**.

Spectral clustering is a method to cluster a graph. The Laplacian matrix of a graph is calculated and the eigenvectors of this matrix are clustered into k-clusters.

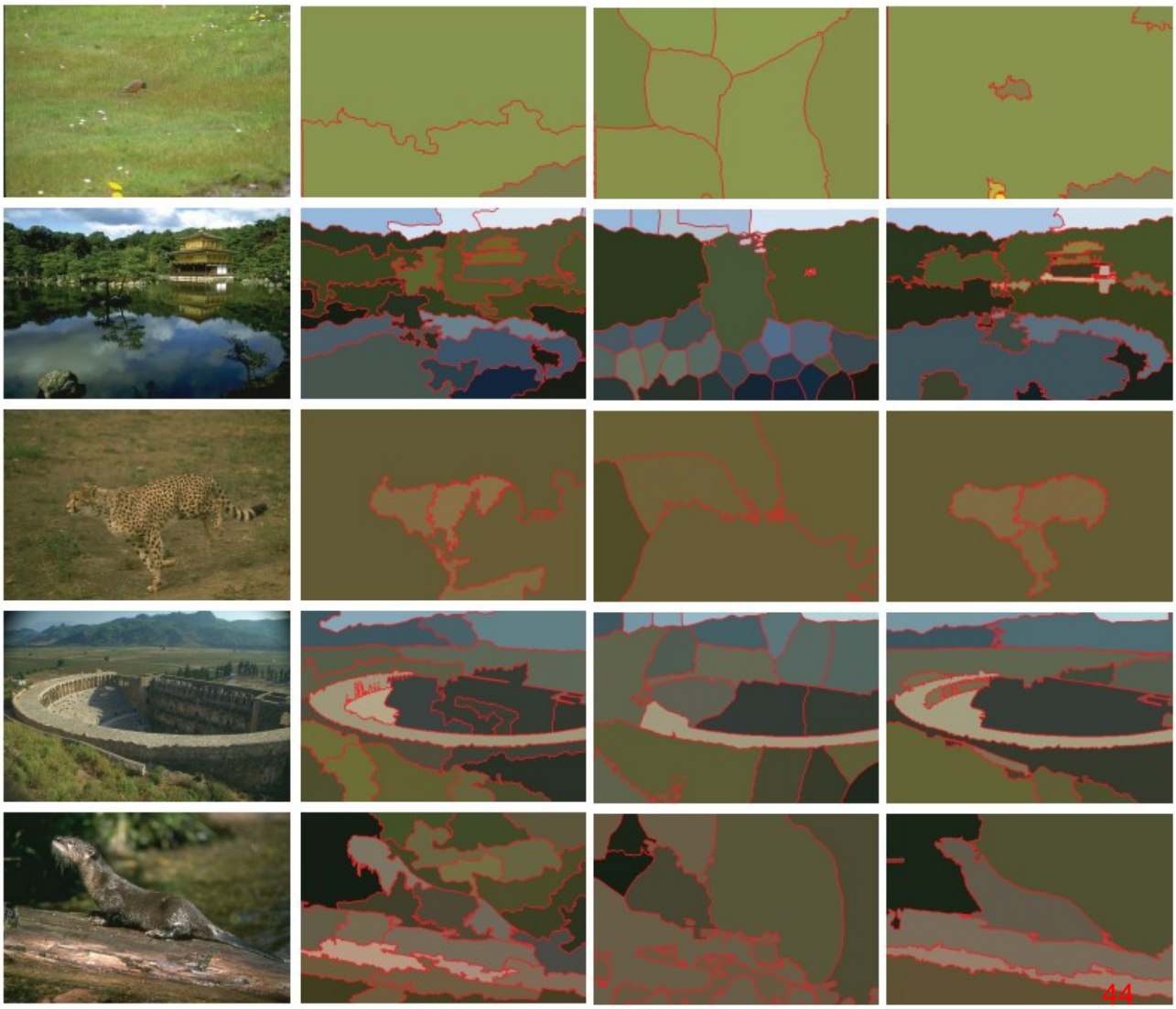
This paper proposes a new method called “transfer cuts” for efficiently calculating the eigenvalues of a bipartite graph for spectral clustering. By exploiting the properties of an unbalanced bipartite graph it is shown that there is an equivalence between the eigenvalues of the original bipartite graph and a much smaller graph only containing superpixels. It is stated that the bottom k eigenvectors from the bipartite graph can be retrieved from the bottom k eigenvectors of the smaller graph. The whole process is very complex and explained in detail in [3], including graph theoretical proofs.

The advantage is a huge performance gain compared to other solvers for the eigenvalues problem like singular value decomposition (SVD) or Lanczos.

The picture above shows a comparison of their performance. Ours stands for the proposed transfer-cuts algorithm[3].



Results:



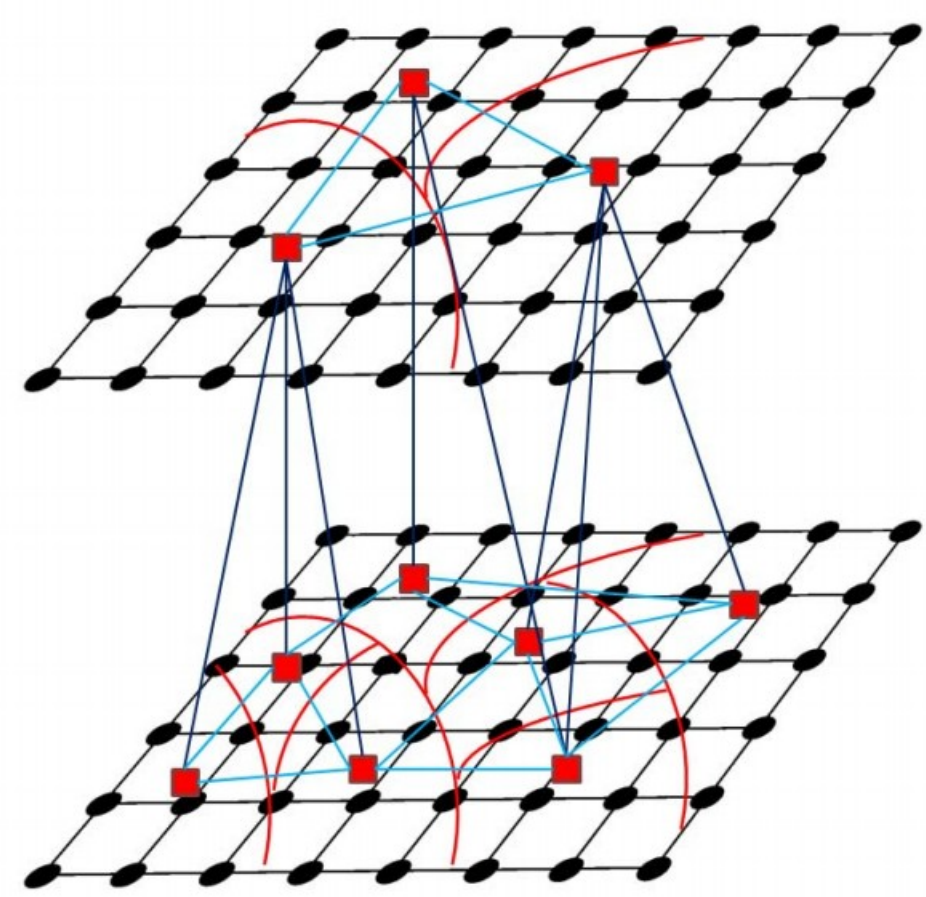
Comparison of the original image (first column), a segmentation using mean shift (second column), segmentation using N-Cuts (third column) and the proposed algorithm (fourth column).

Image Segmentation by bilayer Superpixel Grouping [4]

The Idea of this proposed method is to segment an image by grouping a subset of superpixels that partions a bilayer graph of superpixels while the graph edges encode superpixel similarity.

The graph is constructed from two superpixel segmentations of different resolution. This algorithm uses only two segmentations opposing to [3] which uses many. These two layers will be called U and V.

This graph model is expanded to a hybrid graph model where also information within a layer (U and V) is used. Neighbouring superpixels are connected with edges and their similarity is used as edge weights.



The image above shows this hybrid graph model. The red circles are superpixels and the red quads

their centers (graph vertices), the black dots are pixels.

The basic algorithm workflow for Image segmentation by bilayer superpixel grouping:

1. Partition the Image I into two segmentations U and V with a segmentation algorithm.
2. Construct the Graph
3. calculate the across-affinity matrix W.
4. compute the affinity matrices P and Q.
5. Compute similarity matrix S.
6. compute the Laplacian matrix L.
7. calculate first k eigenvectors
8. Cluster with the help of k-means.

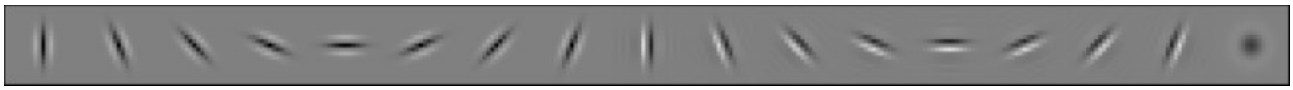
The process of using the eigenvectors and the Laplacian matrix (6-8) is called spectral clustering.

Affinity and Across-Affinity and expanded similarity Matrix

In the above procedure it was mentioned, that the affinity matrices P/Q and the across affinity matrix W are calculated. The affinity matrix encodes the edge weights for the graph within a layer, while the across-affinity matrix encodes the weights between the layers.

The affinity matrices P and Q, P for layer U and Q for layer V are calculated the following way:

For each superpixel a histogram of texton occurrence is calculated. Textons[8] are basic perceptual elements, they are retrieved with convolution with Gaussian derivate filters. The 17 filters are shown in the image below.



The edge weight is now calculated as $\|ti - ti'\|_1$. ti stands for the histogram of texton occurrence within superpixel.

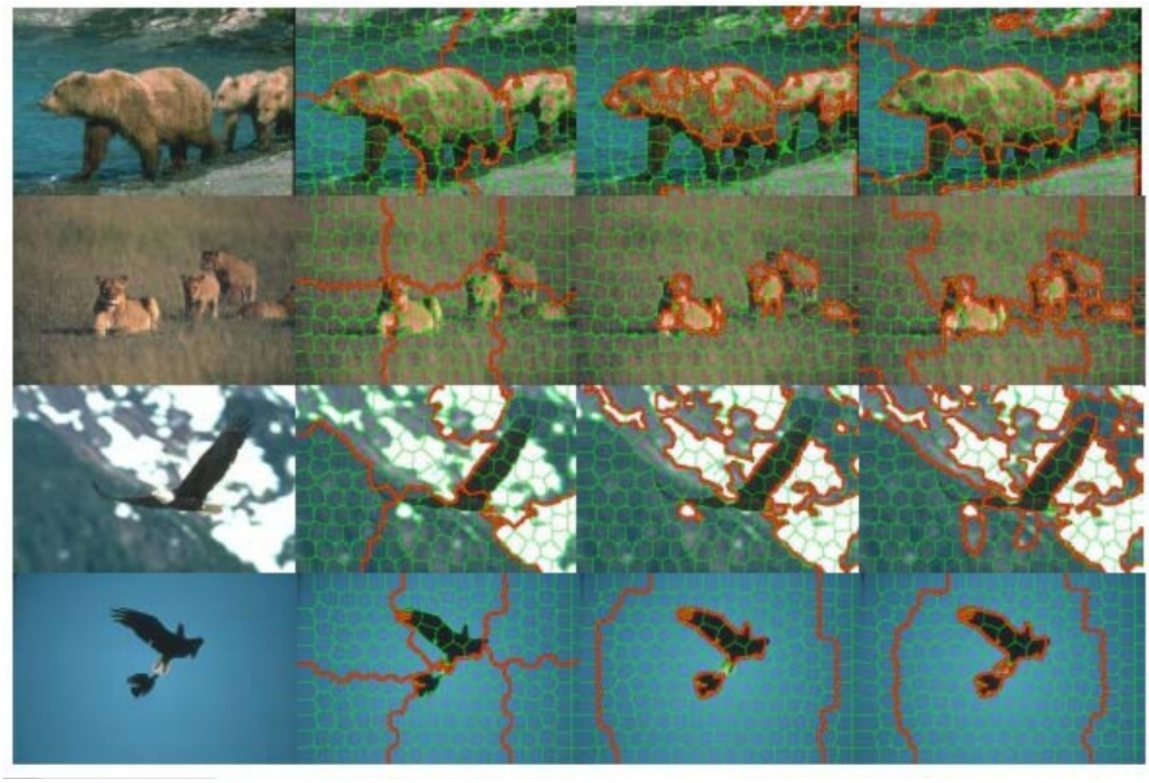
The across affinity matrix is calculated like in [3] but the distance (d_{ij}) between two superpixel is described by a way more complex calculation:

Visual codewords are extracted with SIFT[9] and clustered into 100 clusters (k-means). For each superpixel the histogram of occurrence of the clustered codewords is calculated. The distance is then evaluated as -log of the Kullback Leibler Divergenz (KL)[10] between those two histograms. KL is a similarity measurement between to probability density functions (pdf).

The final step before spectral clustering is to combine these matrices to the expanded similarity matrix in the following form:

$$S=\begin{bmatrix} P & W \\ W^T & Q \end{bmatrix}$$

Results



The first column is the original image.

Column two shows a flat clustering result with local neighborhood information,

Column 3 shows clustering with only a bipartite graph (only with W matrix instead of expanded similarity matrix S)

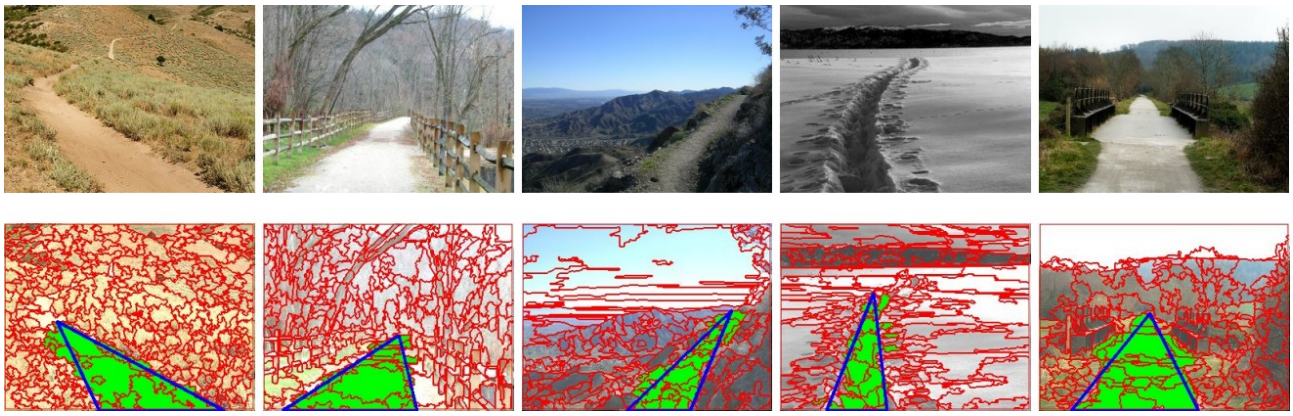
Column 4 shows the result of the hybrid graph clustering.

Shape-guided Superpixel grouping for Trail detection and Tracking[5]

This is an application oriented algorithm for tracking a trail for autonomous vehicles. The algorithm consists of two parts:

- Finding a trail in a single image with no information
- Keeping a trail over a sequence of multiple images

Image Understanding SS16 Winkler Gernot 0929255



For finding a Trail the following assumptions are made

- There is only one trail
- The shape of the trail is similar to a triangle
- The base of the triangle is aligned to the bottom of the image

To create trail hypotheses a superpixel at the bottom of the image is taken. The next superpixel is added depending on similarity (Euclidean distance in RGB space). This step is terminated early when the dissimilarity becomes to high.

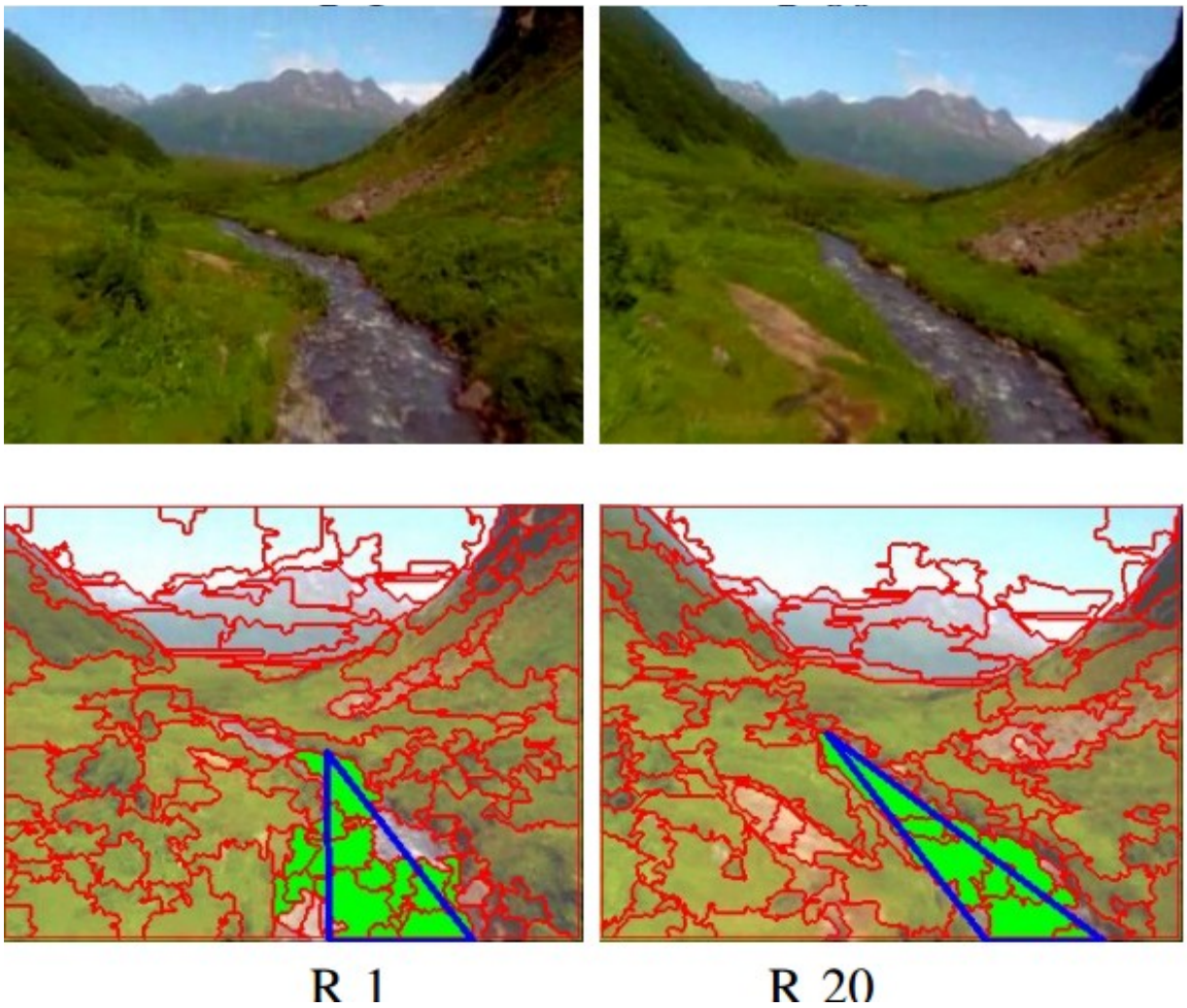
For a fuller exploration of the space of hypothetical groupings a randomization is used instead of a strict deterministic selection. Therefore a next member probability is calculated.

To evaluate the quality of a hypothetical trail a trail likelihood function is calculated which consists of 3 parts:

1. **Shape**. How well does the shape fit a triangle
2. **Appearance**. How consistent is the internal appearance in contrast to the surroundings.
3. **Deformation**. How much deformation is between the best fitting triangle and the most likely triangle of a trail.

For the deformation the most likely triangle was learned out of manually labeled test data.

Trail following



Only for the first frame a full evaluation is needed, since it's very likely that the trail has not changed a lot in the next image.

For this reason the triangle from the last frame is used as starting point and possible triangles around the last one are random sampled. The one with the best appearance likelihood is chosen as the next trail. This process is a lot cheaper than the full trail evaluation.

References

Papers of the main algorithms

[1] Levinshtein, Alex, Cristian Sminchisescu, and Sven Dickinson. "Optimal contour closure by superpixel grouping." *Computer Vision–ECCV 2010*. Springer Berlin Heidelberg, 2010. 480-493.

[2] Dickinson, Sven J., Alex Levinshtein, and Cristian Sminchisescu. "Perceptual grouping using superpixels." *Pattern Recognition*. Springer Berlin Heidelberg, 2012. 13-22.

[3] Li, Zhenguo, Xiao-Ming Wu, and Shih-Fu Chang. "Segmentation using superpixels: A bipartite graph partitioning approach." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE*

Conference on. IEEE, 2012.

[4] Yang, Michael Ying. "Image segmentation by bilayer superpixel grouping." *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*. IEEE, 2013.

[5] Rasmussen, Christopher, and Donald Scott. "Shape-guided superpixel grouping for trail detection and tracking." Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE, 2008.

[6] Levinshtein, Alex, Sven J. Dickinson, and Cristian Sminchisescu. "Multiscale symmetric part detection and grouping." ICCV. 2009.

Secondary References

[7] *A transformation for extracting new descriptors of shape* H Blum, Models for the perception of speech and visual form, 1967

[8] Julesz, Bela. "Textons, the elements of texture perception, and their interactions." *Nature* 290.5802 (1981): 91-97.

[9] D.G. Lowe, “Distinctive image features from scaleinvariant keypoints,” IJCV, vol. 60, no. 2, pp. 91–110, 2004.

[10] Kullback, Solomon, and Richard A. Leibler. "On information and sufficiency." *The annals of mathematical statistics* 22.1 (1951): 79-86.

Protocol: Superpixel Grouping 26.04.

Image Understanding SS 2016

Presentation - Gernot Winkler (0929255)
Protocol - Milena Nowak (0927584)

May 1, 2016

1 Summary of presentation

The presentation "Superpixel Grouping" continued the previous weeks talk on "Superpixel Segmentation". It focused on the application of superpixel segmentations for image segmentation tasks, focusing on contour finding and image recognition. Six algorithms were introduced and their approaches explained.

1.1 Contour Closure

The algorithm [3] uses information A feature vector with four features is calculated (distance to edge, strength of nearest edge, alignment between tangentes superpixel boundary - to image edge, square curvature of superpixel edge points) and a global cost function is optimised in order to find the best contour. As figure 1 shows, the method's success varies depending on the object.

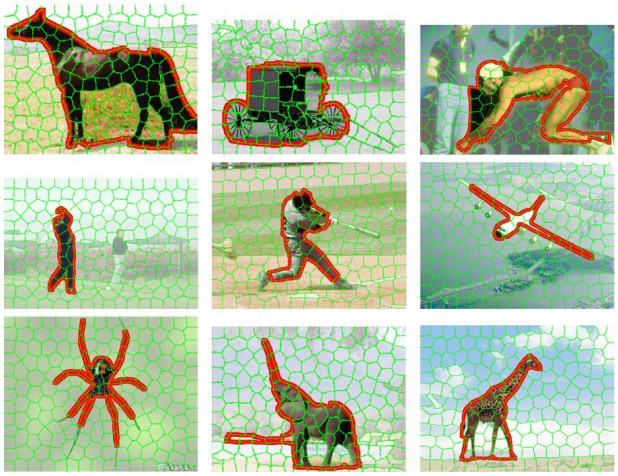


Figure 1: Results of experiments [3].

1.2 Symmetric Part Detection and Grouping

The second paper [4] presented in the talk describes another approach to object recognition or "medial part detection". Clusters of medial points are calculated by combining neighbouring superpixels. In order to determine shape affinity, a normalised scale- and orientation-invariant coordinate system is fitted to the combined superpixels. The clusters form medial branches (skeletons), that are in turn grouped together if they belong to the same object. Figure 2 shows both successful and problematic application scenarios.

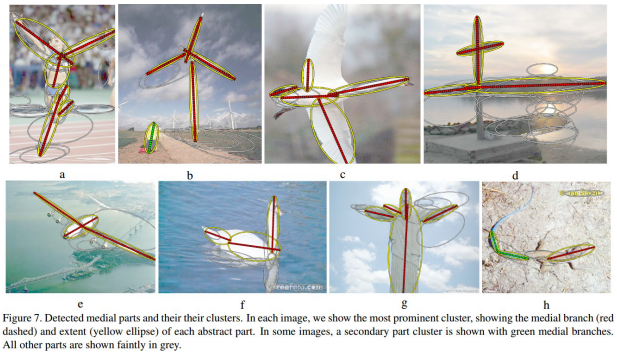


Figure 7. Detected medial parts and their their clusters. In each image, we show the most prominent cluster, showing the medial branch (red dashed) and extent (yellow ellipse) of each abstract part. In some images, a secondary part cluster is shown with green medial branches. All other parts are shown faintly in grey.

Figure 2: Results of experiments [4].

1.3 Segmentation with Graph Partitioning

This paper [5] uses bipartite graphs and over-segmentation. It is more complex than other algorithms for similar tasks but achieves better performance. A cross-affinity matrix is calculated for multi-layer superpixels and clusters calculated. Figure 3 shows results of the algorithm compared to other (state-of-the-art) algorithms.

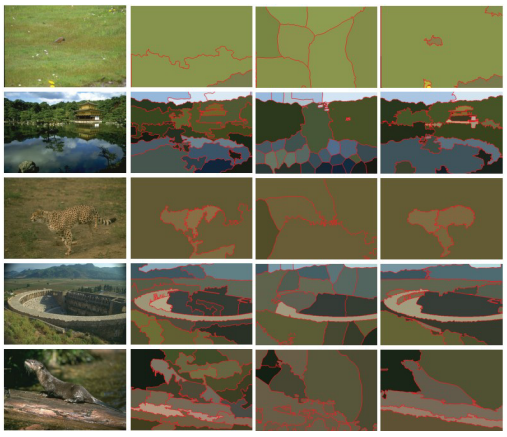


Figure 3: Results of experiments [5].

1.4 Bilayer Superpixel Grouping

[8] is another graph based approach to superpixel grouping based segmentation. Different cues from bilayer superpixels are integrated. The edges of bilayer graphs encode similarity of superpixels. For the actual segmentation, spectral clustering is used.

1.5 Trail detection and Tracking

The last paper presented in the talk [7] used video input to recognise and follow trails and other paths (of any kind; see figure 4). A triangle (the path from street-perspective) is found and tracked over following frames. The triangle is built from the bottom edge and based on the similarity of colours. Therefore, it performs less well on paths with an irregular structure.

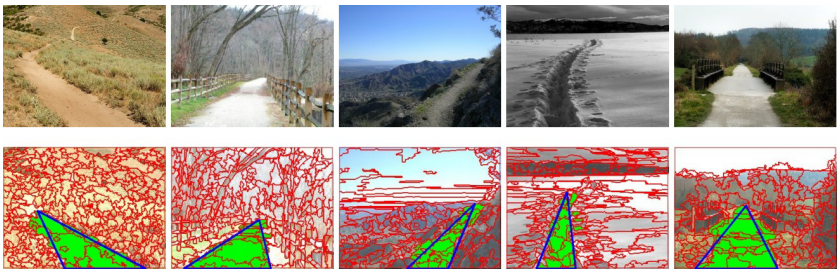


Figure 4: Results of experiments [5].

2 Discussion

2.1 The term "affinity"

There was a discussion about the term "affinity", its definition and whether it was or could be used as a synonym for "similarity" in relation to the paper on segmentation with graph partitioning [4]. The conclusion was that the authors used the term "cross affinity" when talking about the values calculated between layers, "affinity" for the results within layers and "similarity" when both values were combined. In general, it was suggested that "affinity" could be defined in relation to the term "affine transformation" - something that is similar in form but not necessarily in appearance.

2.2 The shadow problem

In particular in relation to figure 1, the failure to properly separate objects and their shadows as well as possible solutions and workarounds were discussed. It was suggested that rather than favouring compact objects, more importance could be given to the homogeneity within the area. Another suggestion was to look at the shadow separately. For humans, it is not difficult to differentiate between shadow and object, as we have implicit knowledge about lighting conditions. There are image forensic methods (e.g.

using highlights) and shape from shading methods that could be used. As most objects are standing on the ground (rather than flying), an even simpler method could be employed for many cases - lighting from above as well as shadow and object being connected can be assumed. Assuming lighting from above, the bottom edge is the one most likely to be a shadow rather than a true edge. This and other (e.g. general shapes of humans or animals) structural information could also be figured into an algorithm. Another proposition was to combine the algorithms [3] and [4]. The usefulness of all these suggestions depends on the dataset at hand. In this context, the 1982 paper on low level segmentation strategies [2] was mentioned - concluding that bottom-up vision does not work.

2.3 Superpixel edges and segmentation

Most superpixel segmentation grouping algorithms assume that superpixel edges correspond to edges within the image. Therefore, the better the segmentation, the better subsequent steps work. As soon as superpixels exist, their geometrical information can be used for grouping. It was suggested that this could be done on multiple levels ("resolutions").

2.4 Symmetric Part Detection and Grouping

The results and errors shown in figure 2 were discussed. A possible improvement for the calculation of the invariant coordinate system was suggested: as ellipses are used, elliptical coordinates could be used (unlike in a square coordinate system, object and coordinate tangents would be parallel when using elliptical coordinates). The topic is currently being researched in the Pattern Recognition and Image Processing Group (Aysylu Gabdulkhakova). Image h in figure 2 shows another case for which the algorithm is not ideal: objects with strong curves. Another paper published by the PRIP group [6] deals with cases of "worm-shaped" objects using medial axes and circle radii (see figure 5). It was concluded that [4] is a necessary method, though it only works well for certain objects, e.g. people and other objects with limb-like structures (in contrast to [3], which works well on compact objects).

2.5 Trail detection and tracking

Possible issues or problems for this algorithm [7] were also discussed. A suggestion for the problem of shadows (of trees etc.) was to use the HSV colour space rather than RGB. It was concluded that curves in the path are probably not as problematic as they might look at first glance, provided that the camera is at street level. While curves make it impossible to find an exact triangular shape, the algorithm should still work as long as a vanishing point exists.

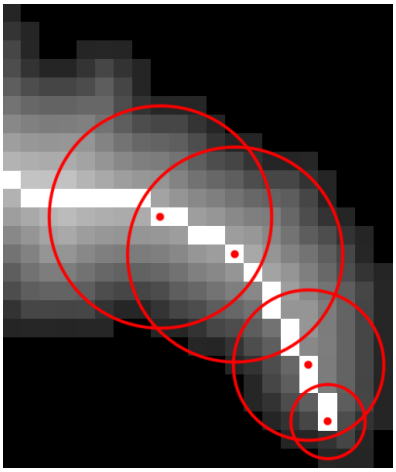


Figure 5: Part of the distance transform of a worm with circles drawn for four points on the skeleton. [6].

2.6 Segmentation with Graph Partitioning

In connection with this algorithm [5], the paper by Cho and Meer [1] on consensus regions was mentioned. They use a set of different segmentations (pyramid-based segmentation) and base the final result on the consensus among them (using aregion adjacency graph).

2.7 Life lessons

- If a paper contains bad results still show them (comparison: under which circumstances does/doesn't the algorithm work)
- Expect difficult papers in this field (both in terms of language and mathematical proofs)
- It makes sense to learn the basics properly (algorithms like SIFT keep turning up everywhere)
- Graphs are important but there is no unified system to deal with them (everyone writes their own software)
- Find more details on application scenarios and tests on researcher's web pages

References

[1] Kyujin Cho and Peter Meer. Image segmentation from consensus information. *Computer Vision and Image Understanding*, 68(1):72–89, 1997.

[2] Martin D Levine and AM Nazif. An experimental rule-based system for testing low level segmentation strategies. *Multicomputers and Image Processing Algorithms and Programs*, pages 149–160, 1982.

[3] Alex Levinshtein, Cristian Sminchisescu, and Sven Dickinson. Optimal contour closure by superpixel grouping. In *Computer Vision–ECCV 2010*, pages 480–493. Springer, 2010.

[4] Alex Levinshtein, Cristian Sminchisescu, and Sven Dickinson. Multiscale symmetric part detection and grouping. *International journal of computer vision*, 104(2):117–134, 2013.

[5] Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 789–796. IEEE, 2012.

[6] Daniel Pucher, Walter G Kropatsch, Nicole M Artner, Stephanie Bannister, and Kristin Tessmar-Raible. 2d tracking of platynereis dumerilii worms during spawning.

[7] Christopher Rasmussen and Donald Scott. Shape-guided superpixel grouping for trail detection and tracking. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 4092–4097. IEEE, 2008.

[8] Michael Ying Yang. Image segmentation by bilayer superpixel grouping. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 552–556. IEEE, 2013.

Computer Vision Models

Johann Götz

June 20, 2016

1 Probability

The first chapter will give a very brief overview of probability theory. Those few ideas, when combined, can provide a very powerful tool for modelling uncertainty.

1.1 Random Variable

A random variable is a variable which denotes a quantity that is subject to variations due to chance. It might represent the outcome of an experiment or the result of measuring a fluctuating property. Observing several such instances may result in a different outcome each time and this information is described by the probability distribution $\Pr(x)$ [1].

A random variable could be discrete, which mean it can take value from a predefined set and it can also be visualized by a histogram. A continuous random variable takes real numbers as its values. In this case the probability density represents the tendency of the random random variable to take a specific value. It is important to note that the integral of the probability density function always sums to one [1].

1.2 Joint Probability

The joint probability written as $\Pr(x, y)$ describes the propensity of the combination of two random variables. In can simply be read as the probability of x and y. The joint probability is not limited to two random variables and can be used for an arbitrary number and multidimensional variables [1].

1.3 Marginalization

Marginalization is used to recover the probability distribution of a single variable from the joint distribution using the equations [1]

$$\Pr(x) = \int \Pr(x, y)dy \tag{1}$$

$$\Pr(y) = \int \Pr(x, y)dx \tag{2}$$

This can also be used to recover the joint probability of a subset of variables

$$\Pr(x, y) = \sum_{w} \int Pr(w, x, y, z)dz,$$

where w is discrete and z is continuous [1].

1.4 Conditional Probability

The conditional probability of x given $y = y*$ describes the chance of random variable x to take specific values if y is fixed to value y*. It is written as $\Pr(x|y = y*)$ and can be constructed from the joint distribution as figure 1 shows. The calculation is written as: [1]

$$\Pr(x|y = y*) = \frac{\Pr(x, y = y*)}{\int \Pr(x, y = y*)dx} = \frac{\Pr(x, y = y*)}{\Pr(y = y*)},$$

which is usually simplified to

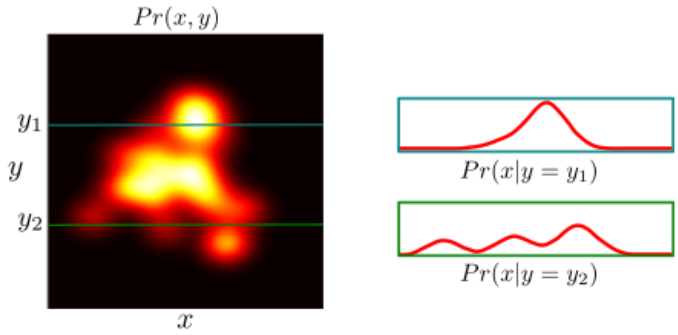


Figure 1: Joint pdf of x and y and two conditional probability distributions $\Pr(x|y = y_1)$ and $\Pr(x|y = y_2)$ on the right side. They are constructed by taken a slice from the joint distribution. [1]

$$\Pr(x|y) = \frac{\Pr(x, y)}{\Pr(y)}$$

1.5 Bayes' Rules

Resulting from the conditional probability the Bayes' rule can be written as:

$$\Pr(y|x) = \frac{\Pr(x|y)\Pr(y)}{\Pr(x)} \tag{3}$$
$$= \frac{\Pr(x|y)\Pr(y)}{\int \Pr(x, y)dy} \tag{4}$$
$$= \frac{\Pr(x|y)\Pr(y)}{\int \Pr(x|y)\Pr(y)dy} \tag{5}$$

In this equation each term has a name. $\Pr(y|x)$ is referred to as the posterior, the term $\Pr(y)$ is the prior [1]. The terms $\Pr(x|y)$ and $\Pr(x)$ are called likelihood and evidence respectively [1].

1.6 Independence

Independence simply means that from knowing the value of one variable the value of another variable cannot be determined. This can be written as: [1]

$$\Pr(x|y) = \Pr(x) \tag{6}$$
$$\Pr(y|x) = \Pr(y) \tag{7}$$

Consequently this means that the joint probability can be written as: [1]

$$\Pr(x, y) = \Pr(x|y)\Pr(y) = \Pr(x)\Pr(y)$$

2 Models in Computer Vision

Solving problems in computer vision means taking an input data $\mathbf{x}$ and trying to deduce the world state $\mathbf{w}$ from it. In general there can be multiple world states that are compatible with a given image data. This is due to noisy data and uncertainty in the contents of the visual data. The same object can have vastly different appearances in the different images and different may look very similar in different conditions [1].

In the case of these problems a possible way to solve them is to use the *posterior probability distribution* $\Pr(\mathbf{w}|\mathbf{x})$. Usually it is not computationally possible to calculate the posterior, in this case using the world state $\hat{\mathbf{w}}$ at the peak of the posterior is the best that can be done [1].

2.1 Components in Computer Vision Models

Three components are necessary to solve a computer vision problem

- A model which uses image data $\mathbf{x}$ to infer the world state $\mathbf{w}$ [1].
- A learning algorithm that uses a training set $\{\mathbf{x_i}, \mathbf{w_i}\}$ to tune parameters $\mathbf{\Theta}$ [1].
- An inference algorithm to calculate the posterior $\Pr(\mathbf{w}|\mathbf{x}, \mathbf{\Theta})$ of new image data $\mathbf{x}$ [1].

2.2 Types of Models

There are two types of Models:

- Discriminative models which model the world state on the data $\Pr(\mathbf{w}|\mathbf{x})$. In this case a distribution of the world state $\Pr(\mathbf{x})$ is chosen with parameters, which are a function of $\mathbf{x}$ and it also relies on a set of parameters $\mathbf{\Theta}$. Therefore the posterior distribution is written as $\Pr(\mathbf{w}|\mathbf{x}, \mathbf{\Theta})$. The objective is to use the training set and fit the parameters $\mathbf{\Theta}$ to it, which can be achieved with Maximum Likelihood (ML), Maximum A Posteriori (MAP), or Bayesian Approaches. Afterwards the posterior distribution $\Pr(\mathbf{x}|\mathbf{x}, \mathbf{\Theta})$ can simply be evaluated [1].
- Generative models which model the data on the world state $\Pr(\mathbf{x}|\mathbf{w})$. For these models a fitting distribution for the data $\Pr(x)$ is used. This distribution depends on the world state and a set of parameters and is therefore written as $\Pr(\mathbf{x}|\mathbf{w}, \mathbf{\Theta})$ and called likelihood. A training set is used to fit the parameters Θ to. For new data the posterior distribution is calculated using Bayes' rule:

$$\Pr(\mathbf{w}|\mathbf{x}) = \frac{\Pr(\mathbf{x}|\mathbf{w})\Pr(\mathbf{w})}{\int \Pr(\mathbf{x}|\mathbf{w})\Pr(\mathbf{w})d\mathbf{w}}$$

[1]

3 Models for Shape

An easy way to describe a shape is to define an algebraic expression for its contour. Simple shapes like circles, ellipses, parabola, and hyperbola can be defined with the following expression:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \alpha & \beta & \gamma \\ \beta & \delta & \epsilon \\ \gamma & \epsilon & \xi \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

As this approach is very limited therefore the following chapters focus on different methods.

3.1 Active Contour Models

With active contour models it is assumed that the topology is known and that it is smooth [1]. This approach tries to find a closed contour defined by N landmark points $\mathbf{W} = [\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_N}]$, which are initially not known [1]. The objective is to find a set of point that will best describe the contour.

If the landmark mark points are in close proximity of edges, the likelihood $\Pr(\mathbf{x}|\mathbf{W})$ should be very high and low otherwise [1]. Therefore a good function for the likelihood is

$$\Pr(\mathbf{x}|\mathbf{W}) \propto \prod_{n=1}^{N} exp\left[-(dist(\mathbf{x}, \mathbf{w_n}))^2\right]$$

[1]

The function $dist(\mathbf{x}, \mathbf{w})$ evaluates the distance to the nearest edge in the image. This means the likelihood increases the closer a landmark point lies on an edge.

Additional constraints are needed to avoid that the landmarks points are attracted to any edge in the image and to make sure they form a contour. A possibility to do this is to choose the a prior like: [1]

$$\Pr(\mathbf{W}) \propto \prod_{n=1}^{N} exp[\alpha \text{ space}(\mathbf{w}, n) + \beta \text{ curve}(\mathbf{w}, n)]$$

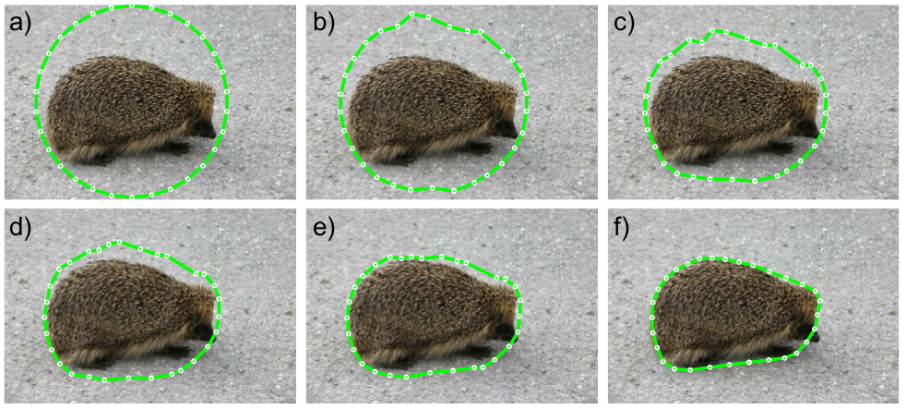


Figure 2: Active Contour Method. A series of landmark points define the contour. a-f) The optimization moves the point closer to the edges of the image with each step. The optimization function is chosen in such a way to achieve this goal, while keeping the distances between objects the same and the curvature low. [1]

The first term punishes uneven point distribution as the evaluation of the function $curve(\mathbf{w}, n)$ results in a high value if the spacing of the contour point $\mathbf{w_n}$ to its neighbours close to the average of all points [1].

The second term punishes a high curvature. It returns larger values for smaller curvatures. Finally both parameters α and β give a weighting for both functions.

3.1.1 Inference

To calculate the contour points on new image data, maximum posteriori criterion is used.

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}[\Pr(\mathbf{W}|\mathbf{x})] = \underset{\mathbf{W}}{\operatorname{argmax}}[\Pr(\mathbf{x}|\mathbf{W})\Pr(\mathbf{W})] \tag{8}$$

$$\underset{\mathbf{W}}{\operatorname{argmax}}[\log\Pr(\mathbf{x}|\mathbf{W}) + \log\Pr(\mathbf{W})] \tag{9}$$

A nonlinear optimization technique must be applied here, because the objective function cannot be evaluated in closed form [1]. Additionally the optimization may be stuck in a local optimum, in such a case the evaluation may be restarted from a different starting position.

Figure 2 shows process of optimization as the landmark points move closer to the object. The final set of points fits tightly around the object.

3.1.2 Problems

This method has several limitation:

- The location of the object in the image has to be known or selected by user input [1]
- This technique works on specific objects only, knowing the object class is not enough [1]
- Projections of 3D surfaces cannot be understood with the active contour models [1]
- Articulated objects cannot be modeled using this approach [1]

3.2 Shape Template Models

Shape templates already start with the object's shape and try to find the object inside the image and any transformations that are necessary. This means finding the parameters Ψ of the transformation are the objective [1].

Landmark points $\mathbf{W} = \{\mathbf{W_n}\}_{n=1}^{N}$ are used to set the shape. This shape will be mapped onto the image by mean of a 2D transformation $(trans[w|\Psi])$, where Ψ represents all parameters of the transformation [1]. The likelihood is defined as

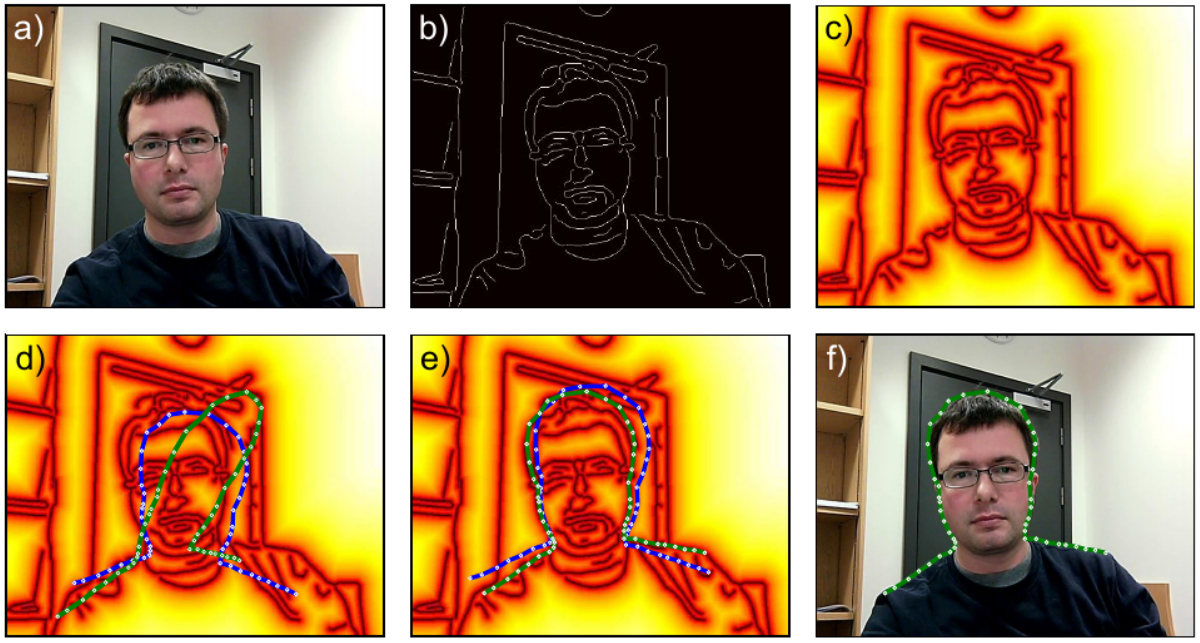


Figure 3: With shape templates the unknown affine transformation which maps the shape onto the image will be calculated. a) Original image. b) Canny edge detector c) Distance transform d) Fitting a shape template. The initialization (blue curve) is random. In this example the optimization (green curve) was stuck in a local optimum. e) With a different initialization the the true optimum can be found. f) Final Result. [1]

$$\Pr(\mathbf{x}|\mathbf{W},\mathbf{\Psi}) \propto \prod_{n=1}^{N} exp[-dist(\mathbf{x},\mathbf{trans}[\mathbf{w_n},\mathbf{\Psi}])^2],$$

where the $dist[\mathbf{x},\mathbf{w}]$ function, returns the distance of the landmark points to the nearest edge in the image.

3.2.1 Inference

The transformation parameters can be found with a maximum likelihood approach: [1]

$$\hat{\mathbf{\Psi}} = \underset{\Psi}{\operatorname{argmax}}[L] = \underset{\Psi}{\operatorname{argmax}}[log(\Pr(\mathbf{x}|\mathbf{W},\mathbf{\Psi}))] \tag{10}$$

$$\underset{\Psi}{\operatorname{argmax}} \left[\sum_{n=1}^{N} -dist(x, trans(\mathbf{w_n},\mathbf{\Psi}))^2\right] \tag{11}$$

The solution requires nonlinear optimization [1].

Figure 3 shows the entire fitting procedure. The procedure might not find the true position of the object and it might has to be restarted from a different starting position or rely on user input.

3.3 Statistical Shape Models

This method tries to model the differences inside the same class of object. As with the two previous methods the shaped is represented by a set of N landmark points $\{\mathbf{W_n}_{n=1}^{N}\}$ [1]. The likelihood relies again on the distance transform: [1]

$$\Pr(\mathbf{x_i}|\mathbf{w_i}) \propto \prod_{n=1}^{N} exp[-dist(\mathbf{x_i},\mathbf{trans}[\mathbf{w_{in}},\mathbf{\Psi_i}])^2]$$

Where $\mathbf{w_{in}}$ represents the n^{th} landmark point in the i^{th} training image and dist evaluates the distance to the nearest edge [1]. Subsequently the prior is modeled as a normal distribution

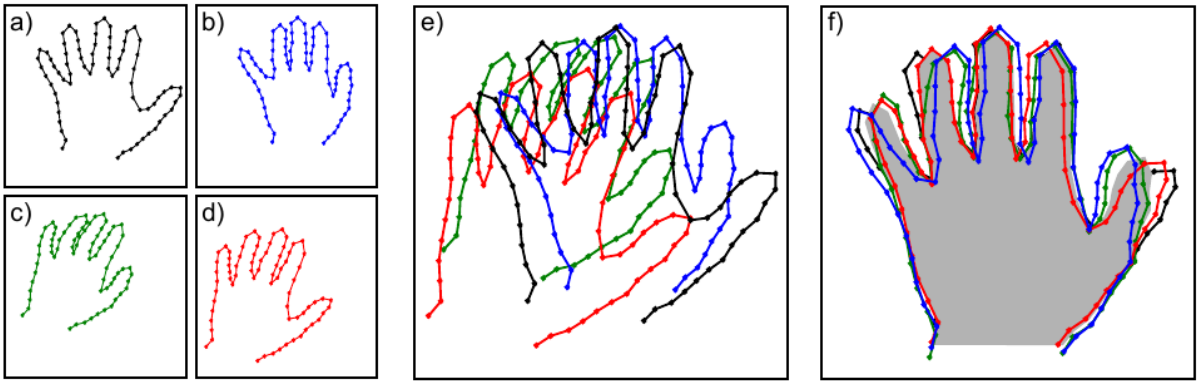


Figure 4: Generalized Procrustes analysis. a-d) Training shapes e) Shows that the alignment is not accurate. f) The generalized Procrustes analysis tries to achieve the simultaneous alignment of all training shapes. The gray area represents the mean shape. [1]

$$\Pr(\mathbf{w_i}) = Norm_{w_i}[\mu, \mathbf{\Sigma}],$$

with the average shape μ and covariance Σ [1].

3.3.1 Learning

The training set is used evaluate the parameters $\mathbf{\Theta} = \mu, \mathbf{\Sigma}$. As the data in the training set is already transformed

$$\mathbf{w}'_{in} = \mathbf{trans}(\mathbf{w_{in}}, \mathbf{\Psi_i}),$$

it must be aligned using using the inverse transformation

$$\mathbf{w_{in}} = \mathbf{trans}(\mathbf{w'_{in}}, \mathbf{\Psi_i^-}),$$

before the parameters μ and $\mathbf{\Sigma}$ can be estimated [1].

The method to achieve this is called *generalized Procrustes analysis* and figure 4 visualized the technique [1].

3.3.2 Inference

The simplest way to approach a new image is to use brute force optimization to guess the landmark points $\mathbf{w} = \{\mathbf{w_n}\}_{n=1}^{N}$.

$$\hat{\mathbf{w}} = \underset{w}{\operatorname{argmax}} \left[\max_{\Psi} \left[\sum_{n=1}^{N} -(dist[\mathbf{x_i}, \mathbf{trans}[\mathbf{w_n}, \mathbf{\Psi}]])^2 + \log[Norm_w[\mu, \mathbf{\Sigma}]] \right] \right]$$

The large number of variables (2N for N landmark points) make optimizations very costly. Additionally a large number of training examples are required to accurately evaluate the covariance and a number of variables could just represent noise in the ground truth [1].

3.4 Subspace Shape Models

These models make the assumption that all shape vectors $\{\mathbf{w_i}_{i=1}^{I}\}$ lie in proximity of a K-dimensional linear subspace and they are represented as

$$\mathbf{w_i} = \mu + \mathbf{\Phi h_i} + \epsilon_i,$$

where μ is the mean shape, Φ is a matrix with the K basis functions that span the subspace, in its columns, and ϵ_i represents a noise term with spherical covariance $\sigma^2\mathbf{I}$ [1]. The vector $\mathbf{h_i}$ contains the weights [1].

Probabilistic principal component analysis (PPCA) will applied here. First the above equation can be rewritten as:

$$\Pr(\mathbf{w_i}|\mathbf{h_i}, \mu, \mathbf{\Phi}, \sigma^2) = Norm_{w_i}[\mu + \mathbf{\Phi h_i}, \sigma^2\mathbf{I}]$$

Additionally a prior over variable $\mathbf{h_i}$ is chosen:

$$\Pr(\mathbf{h_i}) = Norm_{h_i}[\mathbf{0}, \mathbf{1}]$$

Using marginalization of the joint distribution $\Pr(\mathbf{w_i}, \mathbf{h_i})$, the prior density can be calculated:

$$\Pr(w_i) = \int \Pr(\mathbf{w_i}|\mathbf{h_i}) \Pr(\mathbf{h_i}) d\mathbf{h_i} \tag{12}$$

$$\int Norm_{w_i}[\mu + \mathbf{\Phi h_i}, \sigma^2\mathbf{I}] Norm_{h_i}[\mathbf{0}, \mathbf{1}] d\mathbf{h_i} \tag{13}$$

$$Norm_{w_i}[\mu, \mathbf{\Phi\Phi^T} + \sigma^2\mathbf{I}] \tag{14}$$

3.4.1 Learning

The ground truth contains a list of position of landmark points $\{\mathbf{w_i}\}_{i=1}^{I}$ [1]. The goal is to guess the parameters $\mu, \mathbf{\Phi}$ and σ^2 of the PPCA model. Initially the mean μ is set to the mean of the training data:

$$\mu = \frac{\sum_{i=1}^{I} \mathbf{w_i}}{\mathbf{I}}$$

Subsequently a matrix $\mathbf{W} = [\mathbf{w_1} - \mu, \mathbf{m_2} - \mu, \ldots, \mathbf{w_I} - \mu]$ is formed and the singular value decomposition of $\mathbf{WW^T}$ is calculated: [1]

$$\mathbf{WW^T} = \mathbf{UL^2U^T},$$

where $\mathbf{U}$ is the orthogonal matrix and $\mathbf{L^2}$ is a diagonal matrix [1]. The parameters are then computed by: [1]

$$\hat{\sigma}^2 = \frac{1}{D-K} \sum_{j=K+1}^{D} L_{jj}^2 \tag{15}$$

$$\hat{\mathbf{\Phi}} = \mathbf{U_K}(\mathbf{L_K^2} - \hat{\sigma}^2\mathbf{I})^{\frac{1}{2}}, \tag{16}$$

where $\mathbf{U_K}$ is matrix $\mathbf{U}$, but retains only the first K columns and the same for matrix $\mathbf{L}$ and $\mathbf{L_K}$ [1].

Figure 5 shows that principal components can sometimes have very simple interpretations [1]. The first principal component in this image captures the opening and closing of the fingers.

3.4.2 Inference

Applying the model to new data is done by changing the weight $\mathbf{h}$ according to the data. The following function can be used:

$$\hat{\mathbf{h}} = \underset{h}{\operatorname{argmax}} \left[\max_{\Psi} \left[\sum_{n=1}^{N} \left(-\frac{(dist[\mathbf{x_i}, \mathbf{trans}[\mu_\mathbf{n} + \mathbf{\Phi_n h}, \mathbf{\Psi}]])^2}{\sigma^2} \right) + \log[Norm[\mathbf{0}, \mathbf{1}]] \right] \right]$$

There are different ways to approach the optimization of this model. In this example the iterative closest point method is discussed briefly: [1]

- Compute the current landmark points $\mathbf{w} = \mu + \mathbf{\Phi h}$
- Transform each point into the image: $\mathbf{w'_n} = \mathbf{trans}[\mathbf{w_n}, \mathbf{\Psi}]$
- Associate each transformed point with the closest edge point $\mathbf{y_n}$ in the image
- Compute the transformation parameters Ψ for the best mapping
- Transform each point again with the updated parameters Ψ

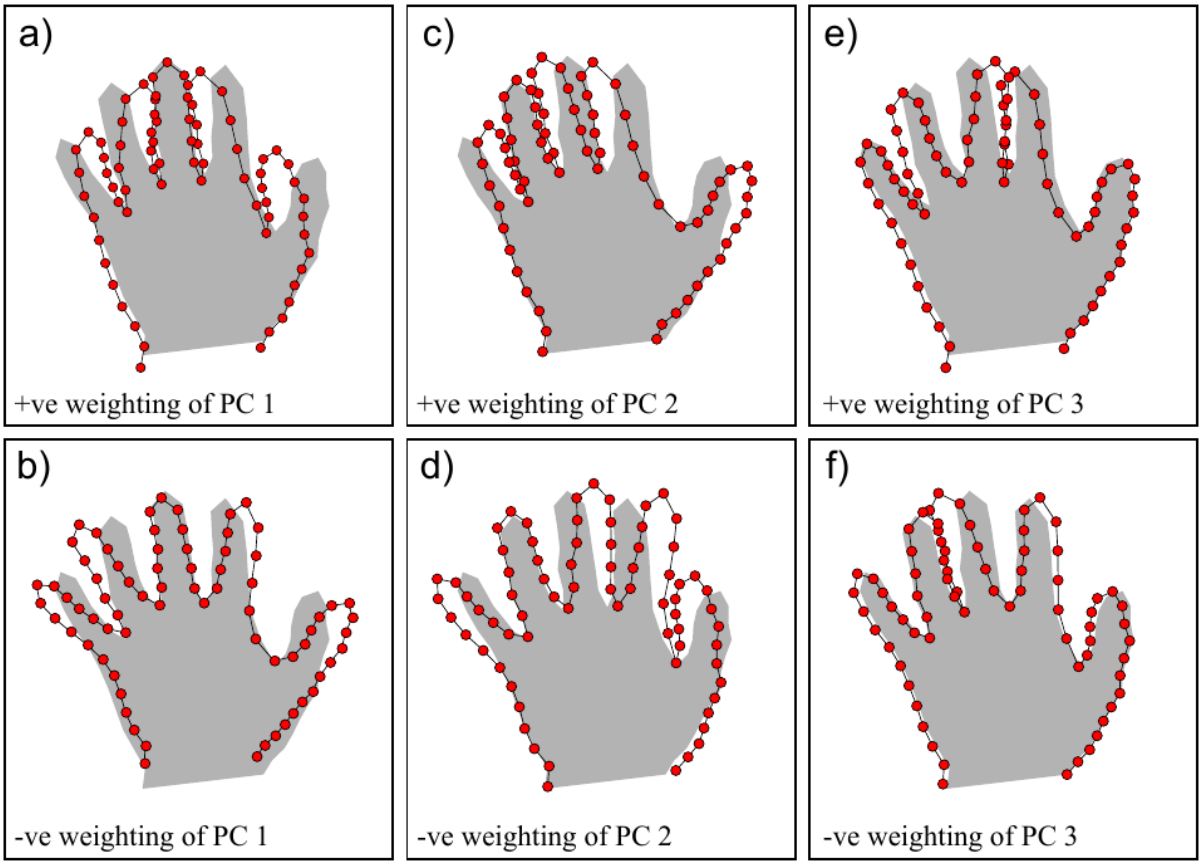


Figure 5: Principal components. a-b) Result of changing the first principal component. In panel (a) a multiple λ of the first principal component Φ_i is added to the mean vector and subtracted in panel (b). The gray area shows the mean. Panels (c-d) and (e-f) show the effect of changing principal component two and three. [1]

- Evaluate the closest edge points $\{\mathbf{y_n}\}_{n=1}^{N}$ again
- Update variables $\mathbf{h}$

Updating $\mathbf{h}$ is done with the following function:

$$\hat{\mathbf{h}} = \underset{h}{\text{argmax}} \left[\sum_{n=1}^{N} \log[\Pr(\mathbf{y_n}|\mathbf{h}), \Psi] + \log[\Pr(\mathbf{h})] \right] \tag{17}$$

$$= \underset{h}{\text{argmax}} \left[\sum_{y=1}^{N} -\frac{(\mathbf{y_n} - \mathbf{trans}[\mu_\mathbf{n} + \phi_\mathbf{n}\mathbf{h}, \mathbf{\Psi}])^2}{\sigma^2} - \log[\mathbf{h^T h}] \right] \tag{18}$$

Which can then be computed in closed form with the following equation:

$$\hat{\mathbf{h}} = \left(\sigma^2 \mathbf{I} + \sum_{n=1}^{N} \mathbf{\Phi_n^T A^T A \Phi_n} \right)^{-1} \sum_{n=1}^{N} \mathbf{A \Phi_n} (\mathbf{y_n} - \mathbf{A}\mu - \mathbf{b})$$

4 Models for Style and Identity

These models take identity (i.e., whose face it is in the case of face recognition) and style (i.e., in what angle or lighting the image was taken) into account. Everything else that makes up the final image has to fit into a generic noise term [1].

4.1 Factor analysis

The i^{th} data example $\mathbf{x_i}$ can be written as

$$\mathbf{x_i} = \mu + \mathbf{\Phi h_i} + \epsilon_\mathbf{i},$$

where μ is the mean and the matrix Φ contains K basis vectors in its columns [1]. The noise term $\epsilon_\mathbf{i}$ is normally distributed with diagonal covariance Σ [1].

$$\Pr(\mathbf{x_i}|\mathbf{h_i}) = Norm_{x_i}[\mu + \mathbf{\Phi h_i}, \mathbf{\Sigma}] \tag{19}$$

$$\Pr(\mathbf{h_i}) = Norm_{h_i}[\mathbf{0}, \mathbf{1}] \tag{20}$$

Using marginalization the likelihood can be calculated as

$$\Pr(\mathbf{x_i} = \int \Pr(\mathbf{x_i}, \mathbf{h_i}) dh_i = \int \Pr(\mathbf{x_i}|\mathbf{h_i} \Pr(\mathbf{h_i}) d\mathbf{h_i}) \qquad = Norm_{x_i}[\mu, \mathbf{\Phi\Phi^T} + \mathbf{\Sigma}]. \tag{21}$$

Learning from training data is done using the expectation maximization algorithm. In the E-step the posterior distribution $\Pr(\mathbf{h_i}|\mathbf{x_i})$ over each variable $\mathbf{h_i}$ is computed: [1]

$$\Pr(\mathbf{h_i}|\mathbf{x_i}) = Norm_{\mathbf{h_i}}[(\mathbf{\Phi^T \Sigma^{-1} \Phi} + \mathbf{I})^{-1} \mathbf{\Phi^T \Sigma^{-1}} (\mathbf{x_i} - \mu), (\mathbf{\Phi^T \Sigma^{-1} \Phi} + \mathbf{I})^{-1}]$$

In the M-step the parameters are updated as: [1]

$$\hat{\mu} = \frac{\sum_{i=1}^{I} \mathbf{x_i}}{\mathbf{I}} \tag{22}$$

$$\hat{\mathbf{\Phi}} = \left(\sum_{i=1}^{I} (\mathbf{x_i} - \hat{\mu}) E[\mathbf{h_i}]^T \right) \left(\sum_{i=1}^{I} E[\mathbf{h_i h_i^T}] \right)^{-1} \tag{23}$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{I} \sum_{i=1}^{I} diag[(\mathbf{x_i}\hat{\mu})(\mathbf{x_i} - \hat{\mu})^T - \hat{\mathbf{\Phi}} E[\mathbf{h_i}](\mathbf{x_i} - \hat{\mu})^T]. \tag{24}$$

More information can be found in [1] section 7.6.

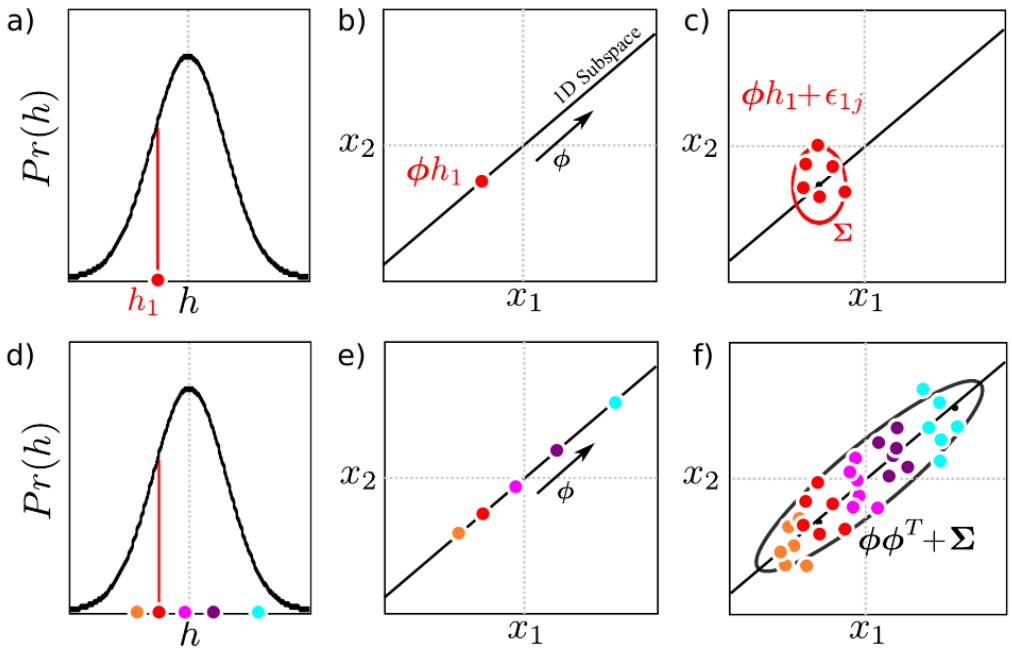


Figure 6: Sampling from factor analyzer. The assumption is made that the mean μ is zero. a) The first step to chose a variable $\mathbf{h_i}$ from the normally distributed prior. In this example h_i is one-dimensional. b) Multiply the weights by Φ, which creates a point on the subspace. c) Subsequently the noise term ϵ_i is added. Finally a mean team μ is added, which is not shown in this visualization. d-f) Repetition of this process. The final covariance is $\mathbf{\Phi\Phi^T} + \mathbf{\Sigma}$. [1]

4.2 Subspace Identity Models

The factor analysis model is extended by adding identity information. $\mathbf{x_{ij}}$ denotes the j^{th} data example from the i^{th} identity and it can be represented as:

$$\mathbf{x_{ij}} = \mu + \mathbf{\Phi h_i} + \epsilon_{\mathbf{ij}}.$$

It is important to note that all data from the same identity is formed by the same linear combination $\mathbf{h_i}$ of the the basis vectors $\Phi_1 \dots \Phi_K$. In probabilistic form this is written as: [1]

$$\Pr(\mathbf{h_i}) = Norm_{h_i}[\mathbf{0}, \mathbf{1}] \tag{25}$$
$$\Pr(\mathbf{x_{ij}}|\mathbf{h_i}) = Norm_{x_{ij}}[\mu + \mathbf{\Phi h_i}, \mathbf{\Sigma}] \tag{26}$$

The density for a data point is:

$$\Pr(x_{ij}) = Norm_{x_{ij}}[\mu, \mathbf{\Phi\Phi^T} + \mathbf{\Sigma}]$$

Sampling from subspace identity model is visualized in figure 6.

4.2.1 Learning

The expected maximization algorithm can be used to estimate the parameters $\mathbf{\Theta} = \{\mu, \mathbf{\Phi}, \mathbf{\Sigma}\}$. In the E-step the posterior probability over each variable $\mathbf{h_i}$ given all the data $x_{i\cdot} = \{x_{ij}\}_{j=1}^{J}$ is calculated: [1]

$$\Pr(\mathbf{h_i}|\mathbf{x_{i\cdot}}) = \frac{\prod_{j=1}^{J} \Pr(x_{ij}|\mathbf{h_i}) \Pr(\mathbf{h_i})}{\int \prod_{j=1}^{J} \Pr(x_{ij}|\mathbf{h_i}) \Pr(\mathbf{h_i}) d\mathbf{h_i}} \tag{27}$$

$$= Norm_{\mathbf{h_i}}\left[(J\mathbf{\Phi^T\Sigma^{-1}\Phi} + \mathbf{I})^{-1}\mathbf{\Phi^T\Sigma^{-1}} \sum_{j=1}^{J}(\mathbf{x_{ij}} - \mu), (J\mathbf{\Phi^T\Sigma^{-1}\Phi} + \mathbf{I})^{-1}\right] \tag{28}$$

From this the expectation can be extracted: [1]

$$E[\mathbf{h_i}] = (J\mathbf{\Phi^T\Sigma^{-}1\Phi} + \mathbf{I})^{-1}\mathbf{\Phi^T\Sigma^{-1}}\sum_{\mathbf{j=1}}^{\mathbf{J}}(\mathbf{x_{ij}} - \mu) \tag{29}$$

$$E[\mathbf{h_i h_i^T}] = (J\mathbf{\Phi^T\Sigma^{-1}\Phi} + \mathbf{I})^{-1} + E[\mathbf{h_i}]E[\mathbf{h_i}]^T \tag{30}$$

In the M-step the parameters are updated.

$$\hat{\mu} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\mathbf{x_{ij}}}{IJ} \tag{31}$$

$$\mathbf{\hat{Phi}} = \left(\sum_{i=1}^{I}\sum_{j=1}^{J}(\mathbf{x_{ij}} - \mu)E[\mathbf{h_i}^T]\right)\left(\sum_{i=1}^{I}JE[\mathbf{h_i h_i^T}]\right)^{-1} \tag{32}$$

$$\Sigma = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}diag\left[(\mathbf{x_{ij}} - \hat{\mu})(\mathbf{x_{ij}} - \hat{\mu})^T - \hat{\Phi}E[\mathbf{h_i}](\mathbf{x_{ij}} - \hat{\mu})^T\right] \tag{33}$$

4.2.2 Inference

The state of the world $w \in \{0, 1\}$ is defined to have two possibilities, $w = 0$ denotes that two data examples x_1 and x_2 have different identities and $w = 1$ for them being the same identity. Therefore the posterior can be stated as:

$$\Pr(w = 1|x_1, x_2) = \frac{\Pr(x_1, x_2|w = 1)\Pr(w = 1)}{\sum_{n_0}^{1}\Pr(x_1, x_2|w = n)\Pr(w = n)}$$

For computation of the posterior the prior probabilities $\Pr(w = 0)$ and $\Pr(w = 1)$ are needed. Unless more information is available, those can be set to o.5. Additionally the likelihoods $\Pr(x_1, x_2|w = 0)$ and $\Pr(x_1, x_2|w = 1)$ are also required.

In the case that both data samples have different identities, the equation looks like this:

$$\begin{bmatrix}\mathbf{x_1}\\ \mathbf{x_2}\end{bmatrix} = \begin{bmatrix}\mu\\ \mu\end{bmatrix} + \begin{bmatrix}\mathbf{\Phi} & 0\\ 0 & \mathbf{\Phi}\end{bmatrix}\begin{bmatrix}\mathbf{h_1}\\ \mathbf{h_2}\end{bmatrix} + \begin{bmatrix}\epsilon_\mathbf{1}\\ \epsilon_\mathbf{2}\end{bmatrix}$$

It is important to note that this has the same form as the factor analyzer discussed above

$$\mathbf{x'} = \mu' + \mathbf{\Phi'h'} + \epsilon'$$

The can be expressed as

$$\Pr(\mathbf{x'}|\mathbf{h'} = Norm_{\mathbf{x'}})[\mu' + \mathbf{\Phi'h'}, \mathbf{\Sigma'}] \tag{34}$$

$$\Pr(\mathbf{h'}) = Norm_{\mathbf{h'}}[\mathbf{0}, \mathbf{1}] \tag{35}$$

where Σ' is

$$\Sigma' = \begin{bmatrix}\mathbf{\Sigma} & 0\\ 0 & \mathbf{\Sigma}\end{bmatrix}$$

[1]

Using marginalization the likelihood can be computed as follow [1].

$$\Pr(x_1, x_2|w = 0) = \int \Pr(x'|h')\Pr(h')dh' \tag{36}$$

$$= Norm_{x'}[\mu', \mathbf{\Sigma'\Sigma'^T} + \mathbf{\Sigma'}] \tag{37}$$

The other case ($w = 1$) can be calculated using the same method, the only difference is that the compound generative equation can be written as follow [1]

$$\begin{bmatrix}\mathbf{x_1}\\ \mathbf{x_2}\end{bmatrix} = \begin{bmatrix}\mu\\ \mu\end{bmatrix} + \begin{bmatrix}\mathbf{\Phi}\\ \mathbf{\Phi}\end{bmatrix}\mathbf{h_{12}} + \begin{bmatrix}\epsilon_\mathbf{1}\\ \epsilon_\mathbf{2}\end{bmatrix}$$

5 Conclusion

The few computer vision models discussed here give a brief overview of statistical computer vision models. Computer vision model not discussed here include articulated models, non-linear identity models, multi-linear models, temporal model, and the huge topic of non statistical computer vision models.

References

[1] **Simon J.D. Prince**, "Computer Vision: models, learning and inference", 2012

[2] **Gower, J.C., & Dijksterhuis, G.B.** "Procrustes Problems", Oxford University Press 2004

[3] **Tipping, Michael E., and Christopher M. Bishop** "Probabilistic principal component analysis" in *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999): 611-622.

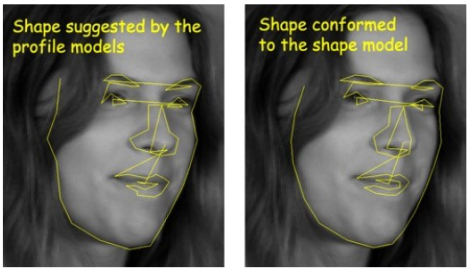
Protocol: Image Understanding 3.5. 2016

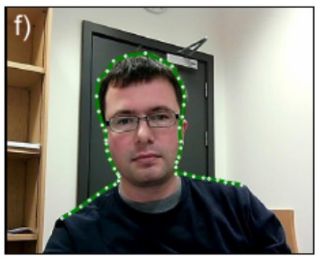
Computer Vision Models

Presentation: Johann Götz 0626963

Protocol: Gernot Winkler 0929255

Presentation Summary:





Computer Vision Models

- Locate known object inside an image
- Independent of Scale, Rotation, Illumination

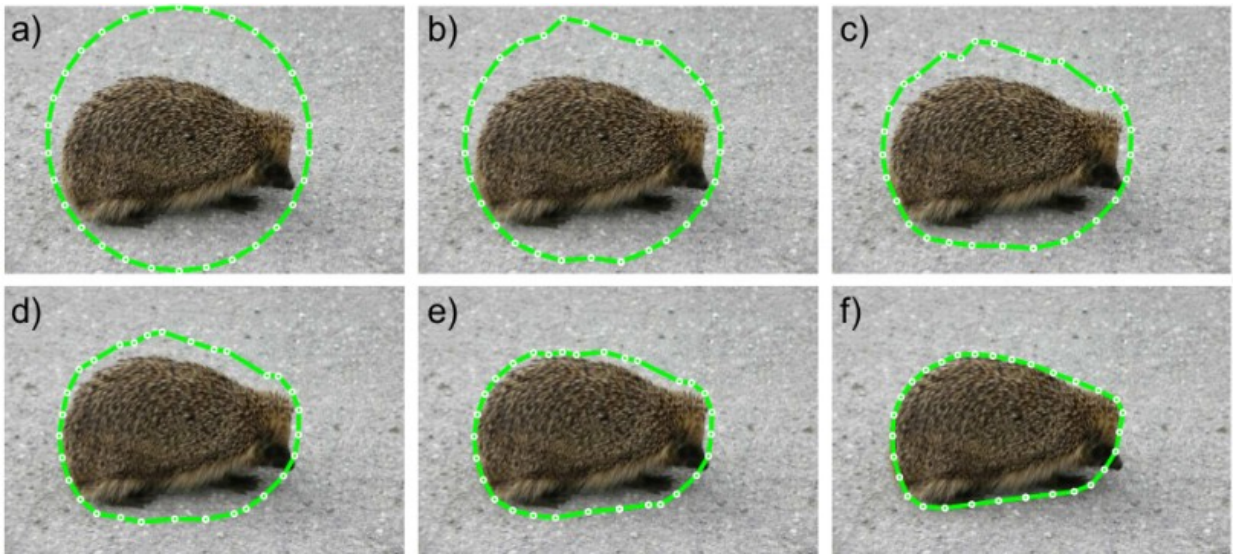
Probability

The introduction for the presentation was a reminder about statistics, since it is important for the presented models. This included: Joint Probability, Marginalization, Conditional Probability, the Rule of Bayes and Independence between probabilities.

Afterward four methods were presented in detail: Active Contour Models, Shape Template Models, Statistical Shape Models and Subspace Shape Models.

Active Contour Models

The first presented model were Active Contour Models which work with some landmark points that should move to the boundary of an object by calculating a likelihood.

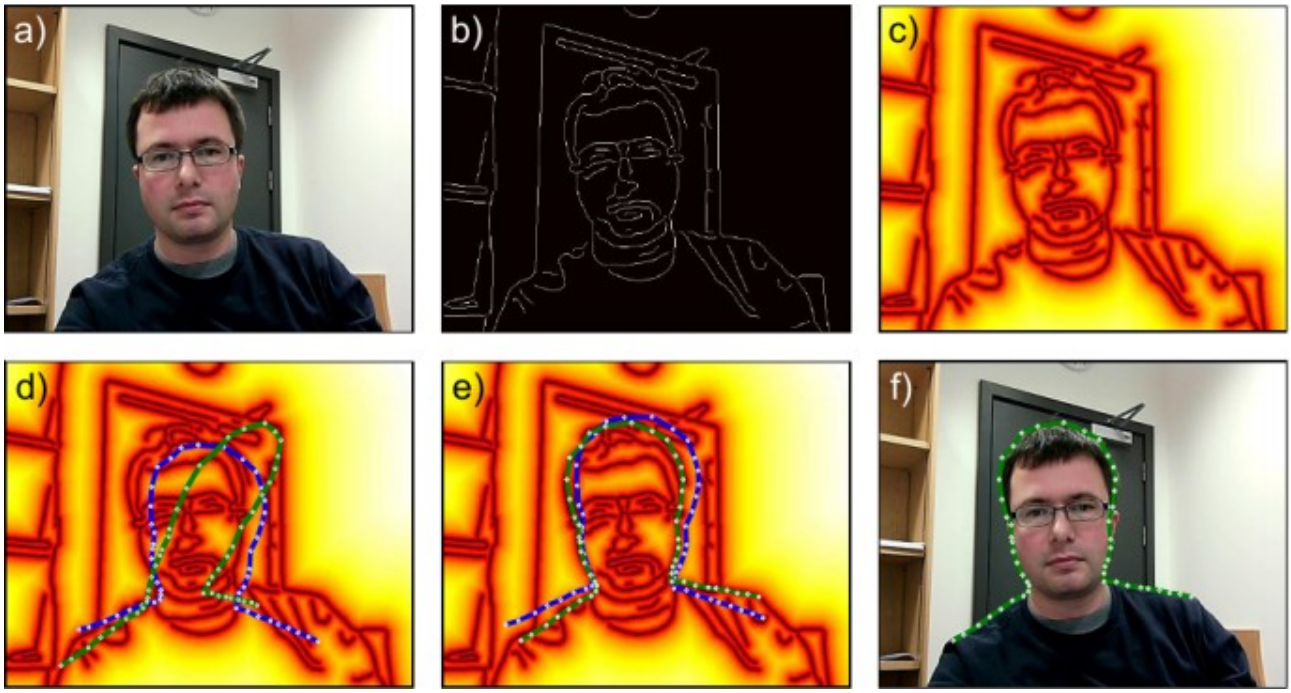


[2]

The limitations of this algorithm are that it only works on specific objects, it's 2D only and articulated objects cannot be modeled by it.

Shape Templates

Assumption that the shape is exactly know. Map the landmark points to the shape.



[2]

Statistical Shape Models

Work with set of N 2D Landmark points and model them as normal distribution.

Learning: estimate parameters μ, σ. Update Transformation, Update mean template.

Problems: A lot of variables to optimize. Many variables could just describe noise.

Subspace Shape Models

Assume shape vectors lie close to a K-dimensional linear subspace. Calculation with PCA, Factor Analysis, Probabilistic PCA.

Models for Style and Identity

Create models that explain Identity, style and remaining variation.

Subspace Identity Models

Subspace Identity Models are an extension of Factor Analysis. Use Multiple data examples with the same identity. Limitations are model of covariance is only a diagonal matrix, non-Gaussian densities can't be modeled.

Other Models

Finally a few Models where mentioned that also exist but where not covered in this presentation:

- Articulated Models
- Non-linear identity Models
- Multi-linear Models
- Temporal Models

Discussion

Math at TU

It was mentioned that there were many formulas that were shown in the presentation. A lot of statistics is used. It was questioned if the students did learn enough about these mathematics, but they agreed that all that topics were covered in math courses. One point that was mentioned as problematic is the strict distinction between Mathematics and Informatics. Math courses are all held by Mathematics faculty members. It might be easier for Informatics students to understand if Math courses would be partially held by members of the Informatics faculty.

Discussion about Models

The presentation focused a lot on statistical models. There also exist more models, like structural models or grammars (L-systems). Also Gaussian Mixture Models (GMM) were mentioned as widespread model in computer vision.
Statistical Models are complicated when they contain normal distributions. Also real models are not always normal distributed.

Algorithm vs Model

Models offer a more general approach while algorithms are often easier too understand and better tailored at a specific problems with no overhead. An advantage of models is that they are not dependent on implementation specific details. A lot of papers in the last years use models.

Hidden Markov Models (HMM)

Hidden Markov models are a combination of a structural and statistical model. They consist of a graph with transition probabilities.

Active Contour Models and shape Templates

If it is unknown where the object is, they have problem. The starting position is very important. Manual selection of the starting point is sometimes needed. It is difficult to know the direction in which the line should move. There also exist 3D versions that work on 3D data (retrieved from stereo vision or kinect for example).

It is also possible to include eyes, mouth and nose with shape templates. There are models that use more than 1 line or also structural methods could be used for this case. Also multiple persons in one image doesn't work.

Active contours have problems with background edges. It works best if there is a homogenous background.

Different features for different races?

„All Asians look the same“.

In each ethical group there might be a different set of optimal features to discriminate two persons. Are there regional specific data sets?

Where could research be done?

What extensions to existing models are possible?. What features could be included into the models. Example: The image of the hedgehog from the active contours chapter. The hedgehog's nose could not be detected by the algorithm. But including such details into the model would make it more complex and optimization more expensive.

How important is Invariance? Adding invariance to models could expand the cases where it is applicable. There are models that can detect faces from different viewing detections. Those are

Multi-linear models, for example bilinear Subspaces.

Problems with Argmax

A lot of the formulas mentioned in the presentation use argmax, which is easily written in theory but might cause some problems in practice. The maximum argument is not always deterministic. For example medial points could cause such a problem, because they have more than 1 closest point on the boundary. This could create an oscillating behavior where a function jumps between two maxima.

References

[1] Kendall, D. G. (1984) Shape manifolds, Procrustean metrics, and complex projective spaces. Bulletin of the London Mathematical Society 16 (2): 81–121

[2] Simon J.D. Prince, “Computer Vision: models, learning and inference, 2012”

[3] Tipping, Michael E., and Christopher M. Bishop. "Probabilistic principal component analysis." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61.3 (1999): 611-622.

Illusions

186.846 Image Understanding SS16

Michaela Tuscher, 0827032
June 5, 2016

Some facts about vision

In the beginning of vision research, the researchers often underestimated the complexity of vision and perception. One of their assumptions was that it would be easy to construct a computer which can see and perceive the world like we do, but very hard to construct one which can defeat the world champion of chess. This assumption proved to be wrong.
Even now many questions concerning our visual system and how it exactly works are still unsolved.

Also, Al Seckel who researches optical illusions and their effects on our perception, states in [1] that the commonly used analogy of the camera and our visual system is not really fitting because a camera records information and our brain interprets it. So there are many different interpretations for one image and visual information is therefore ambiguous. However, most of the time the correct interpretation is chosen automatically. Still there are some cases where the wrong interpretation is chosen and this might then be an optical illusion.

Illusions

Illusions are often studied to get new insights into our visual system, because they lead to knowledge about how we perceive the world. In most cases we are aware of the fact that we see an illusion because our eyes give us contrary information than our mind, which leads to the feeling that something is wrong with the image.

There are different types of illusions. On the following pages some of them are listed, the categorization follows [1], as well as the explanations for the illusions.

Brightness- and Contrast Illusions

In images, the brightness we perceive is very dependent on context and surroundings of an area. Generally, dark surfaces appear darker if they are surrounded by brighter areas and bright surfaces appear brighter if they are surrounded by darker areas.
A very famous illusion of this category is Adelson's illusion. It was constructed by Ted Adelson, a vision researcher at MIT and can be seen in Figure 1. In this image there are a chess board and a cylinder. In fact, the brighter surfaces in the shadow of the cylinder and the darker surfaces outside of the shadow have the same brightness. However, because the darker surface outside the shadow is surrounded by brighter ones it appears darker and the brighter shadowed surface is surrounded by darker ones it appears brighter. Also since there is a shadow and we know from our experience that shadowed surfaces are darker than the surroundings, we automatically try to compensate the

shadows and calculate the brightness without shadow, so we think it has to be even brighter.

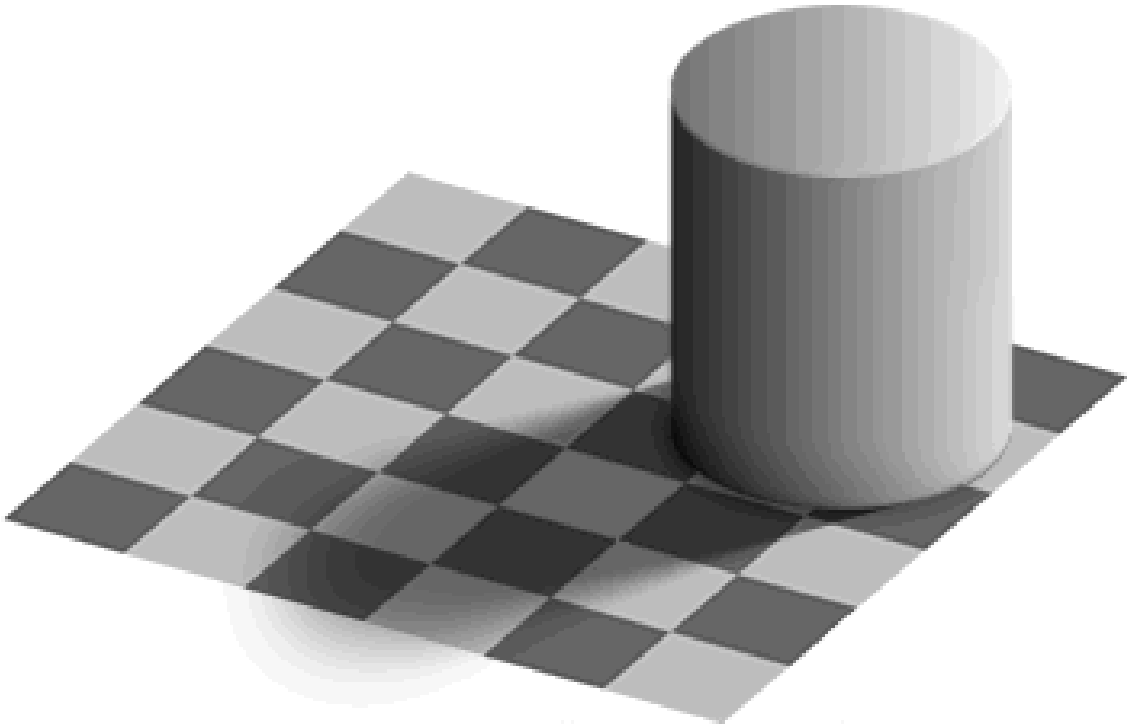


Figure 1: Adelson's Illusion. The image is taken from: http://web.mit.edu/~bcs/images/people/adelsonresearch.gif

Scintillating Illusions

To scintillate means to flicker or to blink. These illusions create the semblance that certain parts of the image are flickering. They were first described 1844 by the British physicist Sir David Brewster and later often used by Op art[1] artists. Like with the previous described Brightness- and Contrast Illusions the cause for the scintillating here is also the contrast, in this case between foreground and background. The cause was first described by the German physiologist Ludimar Hermann, therefore the Hermann Grid was named after him. In the Hermann Grid there appear to be scintillating points at the intersection of the grid lines. Later, even stronger illusions based on the Hermann Grid where found and called Scintillating Grids. The effect here is stronger because of the dots at the intersections. Such a grid can be seen in Figure 2.
Scintillating Grids exist in different kinds of forms an colours. It works best in black and white or with complementary colours or coloured background with gray lines or coloured lines and black background. The strength of the effect depends on the width of the lines and the number of intersections. If the grid is rotated by 45°, the effect is weaker.

1 Derived from "optical art", an art style that uses geometric and abstract forms and also optical illusions

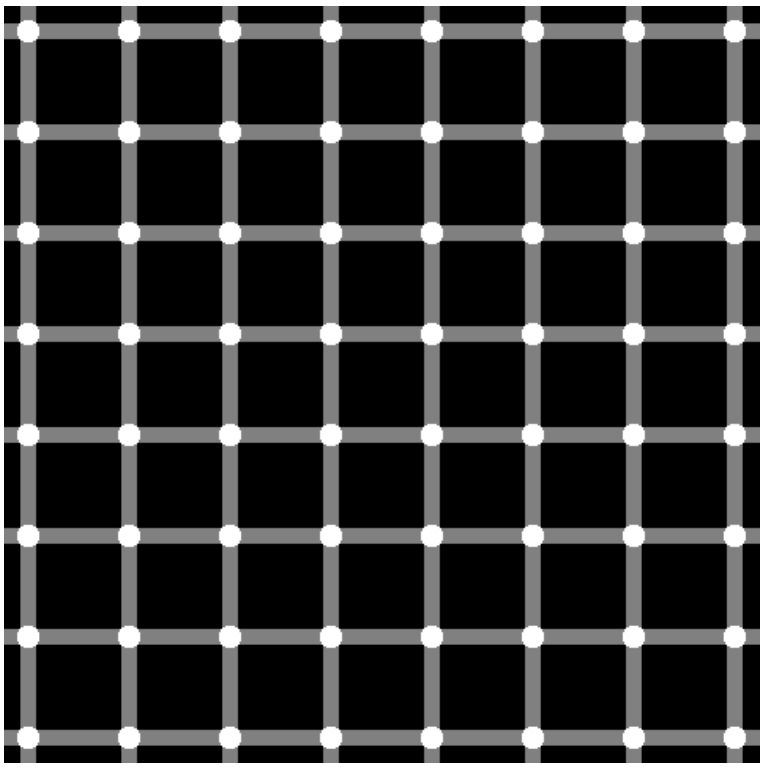


Figure 2: Scintillating Grid. The image is taken from [1].

Parallelity Deceptions

They were discovered by weavers in the 19th century. Often they are generated using spirals, twisted ropes or subtle placed triangles causing parallel lines to look bent or not parallel at all.
One of the most famous illusions of that category is the Café Wall Illusion, see Figure 3. It was discovered on the wall of a British café and therefore got its name. Even though all the lines in this image are parallel, they look bent. The cause for this deception are the shifted black and white squares.
Another interesting illusion are Wilcox' Circles, created by the 12-year old James Wilcox, see Figure 4. In this image, there are perfect circles which look like they are distorted. Here the cause are the twisted lines and triangles at certain positions.

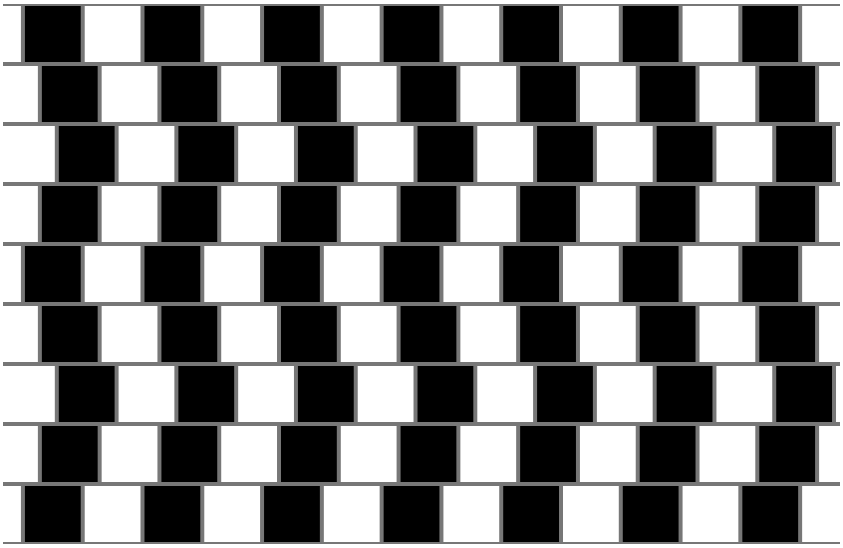


Figure 3: Café Wall Illusion. The image is taken from: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d2/Caf%C3%A9_wall.svg/840px-Caf%C3%A9_wall.svg.png

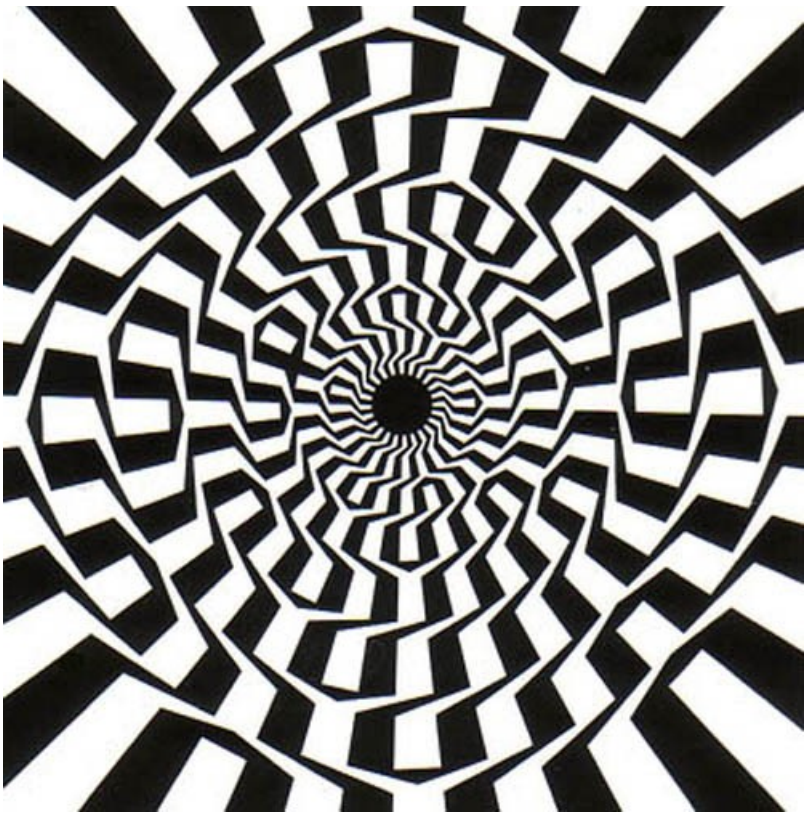


Figure 4: Wilcox' Circles. The image is taken from [1].

Object/Background-Illusions

The differentiation between object and background is very important for humans. A set of rules helps us to decide what is the important object and the less important background of a scene. The differentiation can be made more difficult by manipulating these rules and the object contours. These illusions are popular since the end of the 19th century and are often used for entertainment e.g. puzzles, postcards or posters.

A famous example is the Face/Vase Illusion in Figure 5. There are many different versions of this illusion, also with real images. The image shows either a vase in black or two faces looking at each other in white. The viewer cannot really decide if the important thing on the image are the two faces or the vase.

A similar example is the image with the cat in Figure 6. When looking closely, one can see that the nose of the cat is actually a mouse. This image was created by the digital artist Alice Clarke after a drawing of the British artist Peter Brooks.

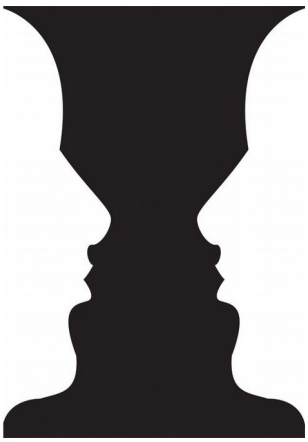


Figure 5: Face/Vase Illusion. Image taken from: https://richardwiseman.files.wordpress.com/2011/03/vase.jpg?w=620

Figure 6: Cat or mouse? Image taken from [1].

Wrong estimations

These are the oldest type of deceptions. As the name suggests, this happens when estimations about areas and dimensions are made. They were discovered at the end of the 19th century, but even now there is still no theory which can fully describe these deceptions.
A very famous illusion in this category is the Müller-Lyer Illusion, which can be seen in Figure 7. Even though all the arrows have the same length, the length of the line with the outward pointed arrows is underestimated whereas the other one with the inward pointed arrows is overestimated.
Another interesting illusion is Sander's Parallelogram, see Figure 8. The two diagonals are of the same length. However the right one seems to be shorter, because of the fact that the parallelogram on the right side has a smaller area than the one on the left side and we compare them to each other and get to the wrong conclusion, that the right diagonal has to be shorter.
The Ebbinghaus Illusion is also an illusion where areas are compared, see Figure 9. Both orange circles have the same size. However we compare the left one with the bigger ones surrounding it so it seems smaller and likewise the right one is compared to the smaller ones surrounding it which makes it seem bigger. If the circles are moved apart, the illusion gets weaker.

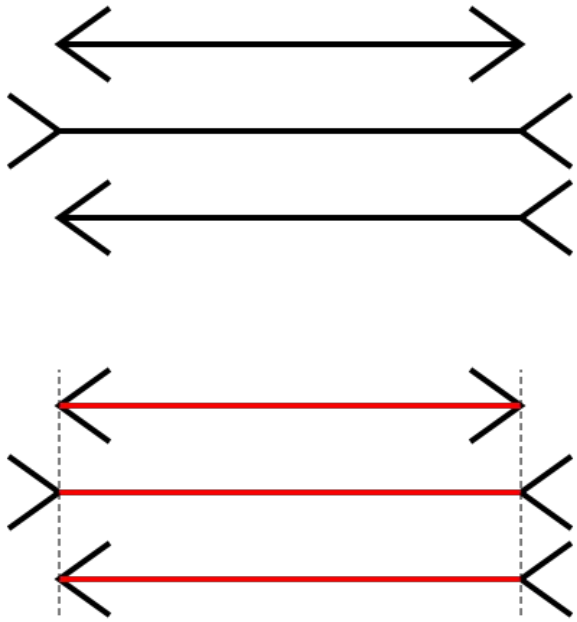


Figure 7: Müller-Lyer Illusion. The image is taken from: https://upload.wikimedia.org/wikipedia/commons/thumb/f/fe/M%C3%BCller-Lyer_illusion.svg/420px-M%C3%BCller-Lyer_illusion.svg.png

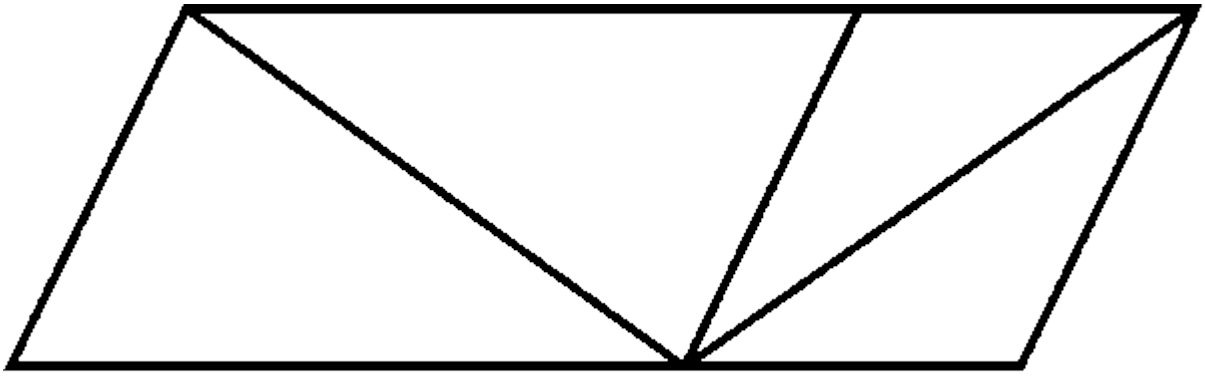


Figure 8: Sander's Parallelogram. Image taken from: http://brisray.com/optill/sander.gif

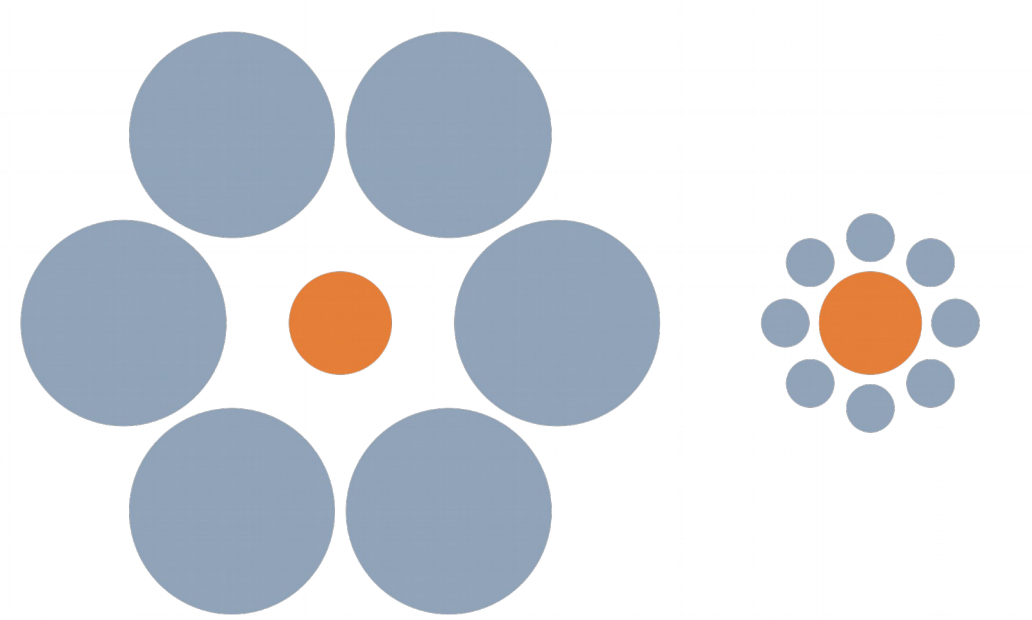


Figure 9: Ebbinghaus Illusion. Image taken from: https://upload.wikimedia.org/wikipedia/commons/thumb/b/bc/Mond-vergleich.svg/2000px-Mond-vergleich.svg.png

Color Deceptions

These illusions are related to Brightness- and Contrast Illusions. Color perception is very complex. In general colour is perceived when light falls on a surface, gets reflected from there and falls into our eyes. As well as with Brightness- and Contrast Illusions the context and surrounding of a surface plays a great role in colour perception, because surrounding colours can influence the perceived colour of a surface. Another important aspect which influences the perceived colour is lighting.

A similar illusion as Adelson's Illusion is Purves' and Lotto's Rubik's Cube, which can be seen in Figure 10. It was created by the neuro scientists Dale Purves and R. Beau Lotto from Duke University and is a good example for the fact that colour perception is also based on experience and context. The brown squares in the middle of the surfaces have the same colour. Again we have the same effect with the shadow as in Adelson's illusion, which leads us to believe that the shadowed brown surface is brighter. Also the brown square on the top surface is surrounded by brighter squares. Therefore it seems all in all darker than the brown square on the shadowed surface.

An illusion with a different effect is Kitaoka's Flowers, see Figure 11. It was created by the Japanese vision researcher and Op art artist Akiyoshi Kitaoka. In this image the red squares have the same shade of red, although the ones with the blue grid look more purple and the ones with yellow grid more orange. This is called chromatic colour assimilation.

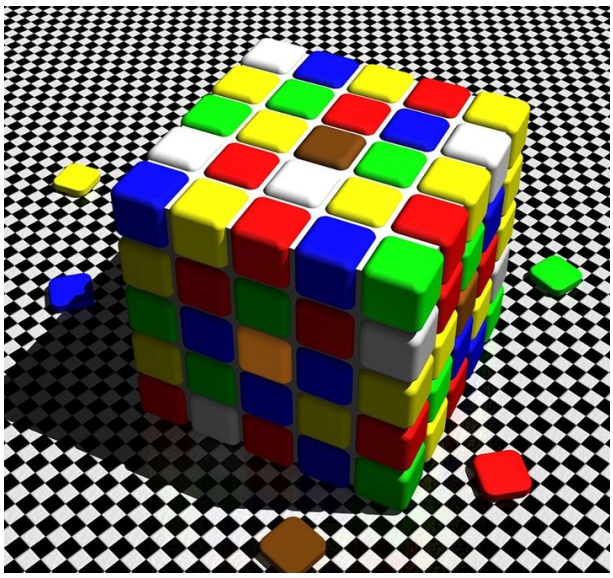


Figure 10: Purve's and Lotto's Rubik's Cube. Image taken form: https://qph.is.quoracdn.net/main-qimg-187717d8c7b0f066ceea649dc4df6469?convert_to_webp=true

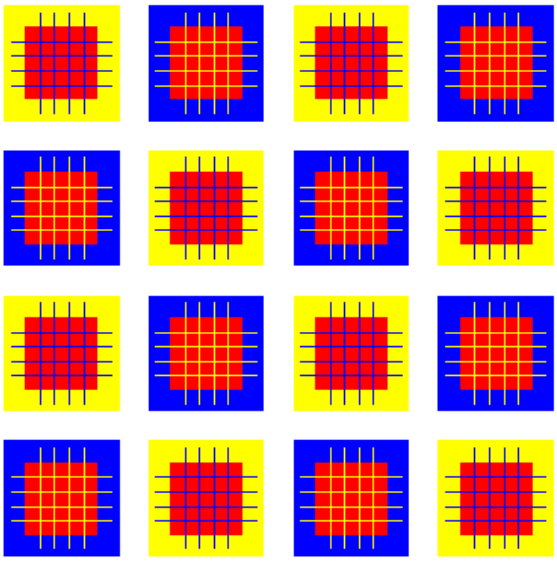


Figure 11: Kitaoka's Flowers. The image is taken from [1].

Stereo Illusions

These illusions are very popular and also known as “The magic eye”. Our depth perception is based on the two stereo images which fall on the retina and the differences between them. Stereo illusions create the impression of depth with one or two 2D images.
The most popular images in this category are stereograms. There are different types of stereograms. Examples are two slightly different images of a scene next to each other, or only one image with seemingly random dots or patterns with hidden depth cues. The former where very popular in 19th century and often used for entertainment. An example for the latter can be seen in Figure 12.

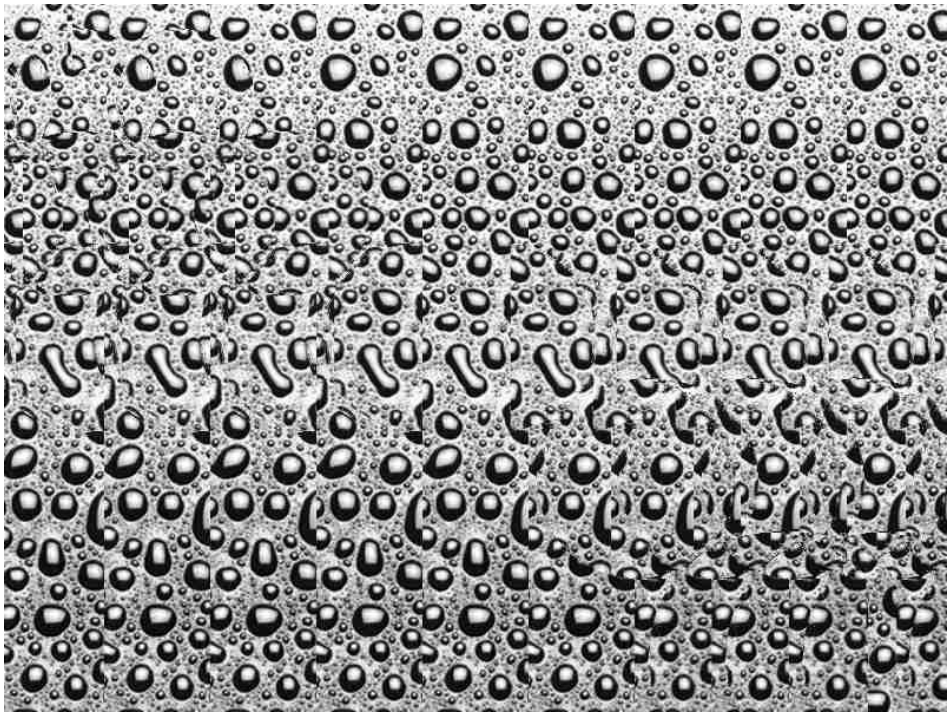


Figure 12: Stereogram with random pattern. The image is taken from: http://img.webme.com/pic/s/stereogramm/tropfen.jpg

Movement Illusions

The detection of movement is a very important task of our visual system. For example it is very important for us to detect things which are moving towards us so we can evade them. Our eyes focus automatically on moving things and our visual system uses complex processes to detect movement. Movement illusions are often created by Op art artists. Those images stimulate multiple parts of the brain which are important for the detection and perception of movement and trick us into seeing movement where there is none.
A famous illusion in this category is Kitaoka's Rotating Snakes, see Figure 13. The circles seem to rotate though in reality they do not move. When fixating one point, the movement stops. With increased eye movement, the perceived movement gets stronger. This kind of illusion is also called Peripheral Drift Illusion. Kitaoka describes the effect in more detail in [3]. By using stepwise luminance profiles and fragmented or curved edges, the effect gets stronger. This can be seen when comparing the images in Figures 14 and 15. Four different colours or luminances are used to create the effect.

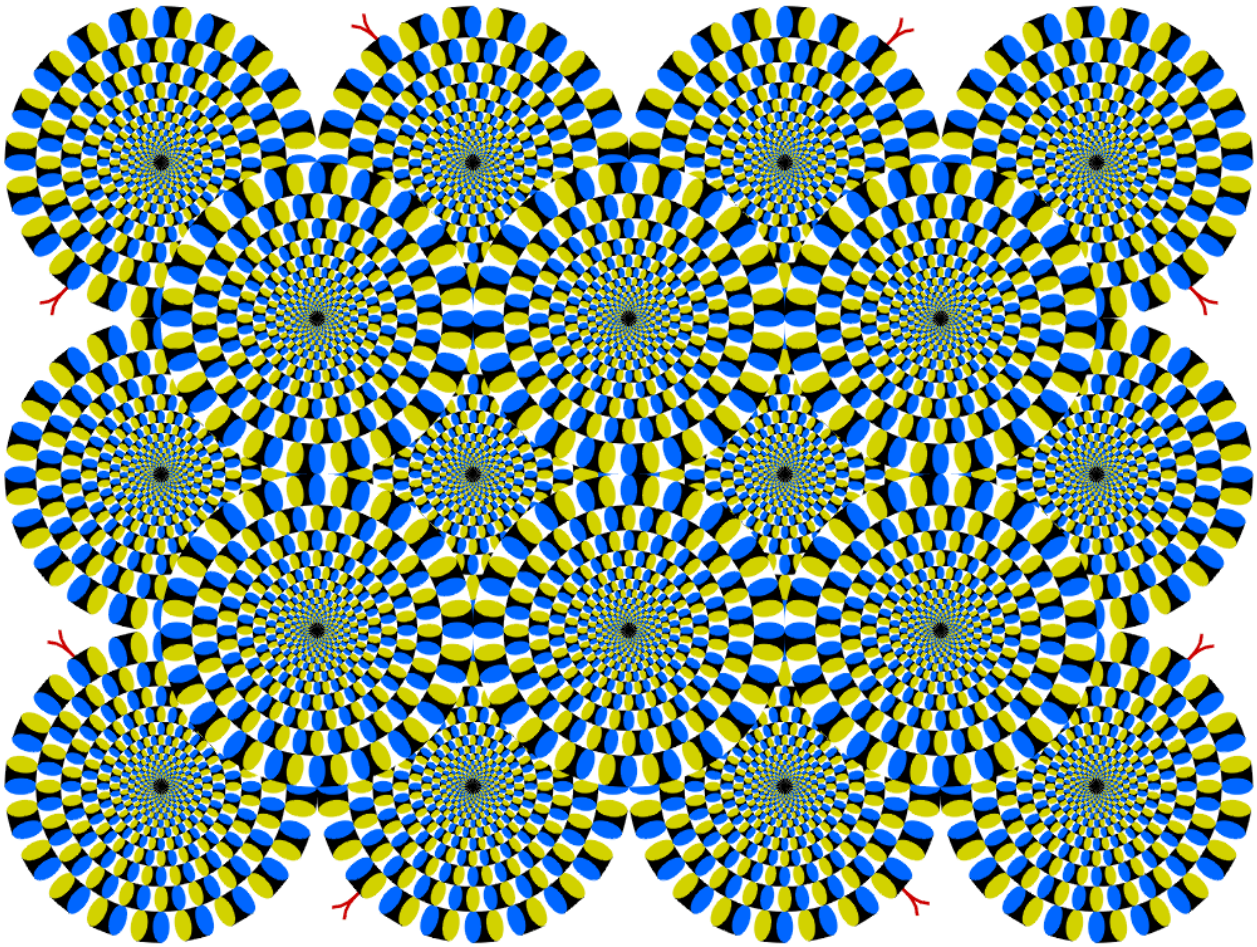


Figure 13: Kitaoka's Rotating Snakes. The image is taken from: http://www.ritsumei.ac.jp/~akitaoka/rotsnake.gif

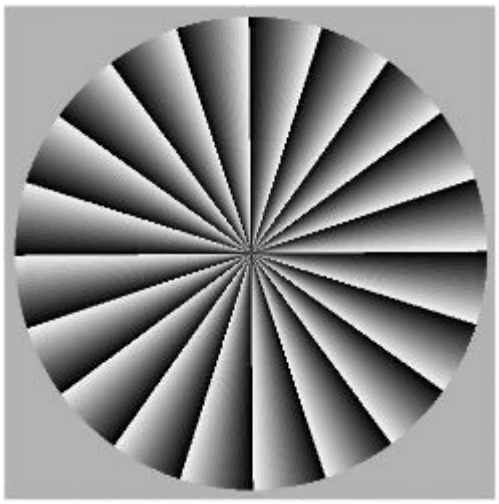


Figure 14: A circle with luminance gradient. Image taken from [3].

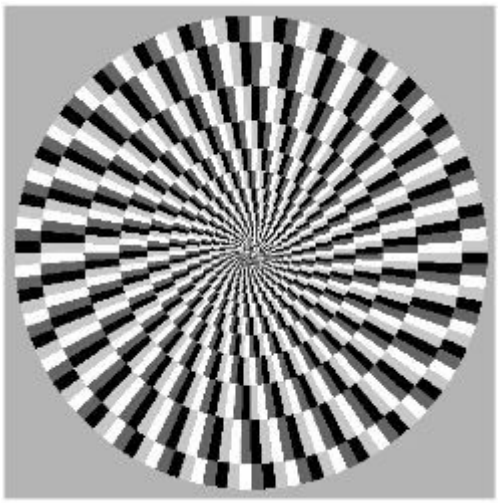


Figure 15: A circle with stepwise luminance profiles and curved edges. Image taken from [3].

Impossible Figures

These illusions are created by two-dimensional perspective images with contradictory depth cues. Mostly when looking only at parts of the image it seems to be just a normal perspective image, but when an overall view is done it creates the impression that there is something wrong. Such images are very important and often researched, because they give insights in how we create three-dimensional perception out of two-dimensional images.
Some artists specialized in drawing impossible figures. Among them are William Hogarth, Oscar Reutersvärd and M. C. Escher. An image of M. C. Escher can be seen in Figure 16. With his famous copper engraving, Hogarth wanted to make fun of artists which use wrong perspectives and also he wanted to point out the power of perspective in images, see Figure 17.

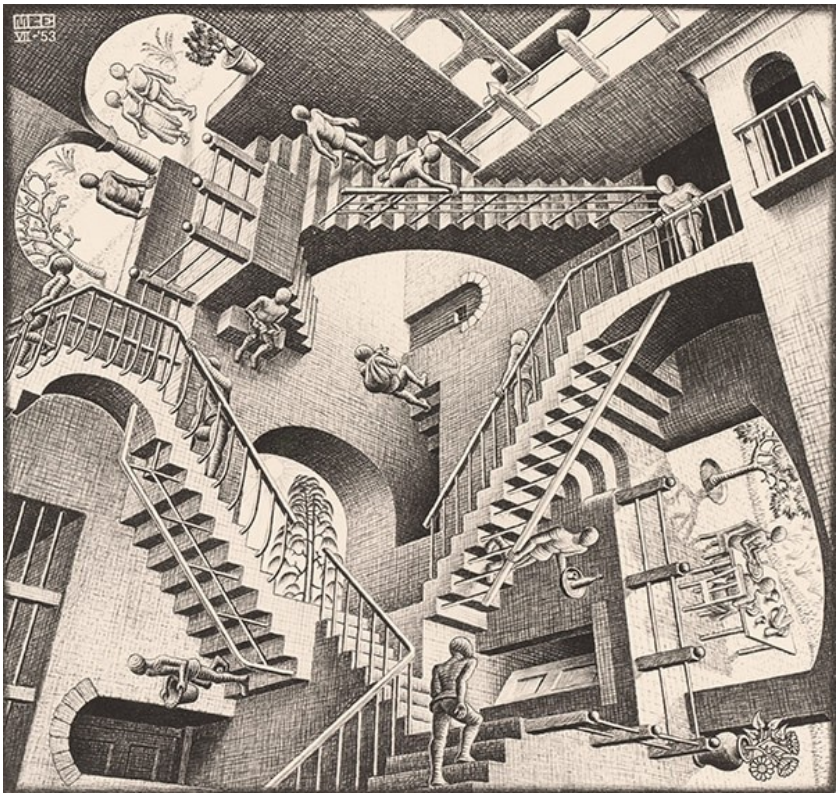


Figure 16: M. C. Escher's Relativity. Image taken from: http://mcescher.com/wp-content/uploads/2013/10/LW389-MC-Escher-Relativity-19531.jpg

Figure 17: Hogarth's copper engraving. Image taken from [1].

Shadow Illusions

Shadows are very important for how we perceive a scene and have a great influence in it. From our experience we know that light mostly comes from above. According to this we interpret the shadows in a scene and derive from them information about the light source. In art shadows are often used to give paintings a three-dimensional impression. However, when shadows are at the wrong place, this leads to wrong impressions of the scene.

The American children's book author and photographer Walter Wick created a shadow illusion with circles and shadows in them where the circles appear either convex or concave depending on how the image is rotated. Walter Wick's illusion can be seen in Figure 18 and rotated about 180° in Figure 19.

Another illusion which shows different interpretations of a scene depending on the placement of the shadows is Kesten's Ball and Shadow Illusion, see Figure 20. From our experience we know that if an object lies on a surface, its shadow touches the object and normally if a shadow does not touch the corresponding object, the object flies above the surface. Therefore in the upper image the balls appear to be lying on the chess board whereas in the lower image they appear to be flying above the chess board though the positions of the balls are identical in both images and only the positions of the shadows differ. If there were no shadows in the image, it would not be possible to determine the positions of the balls.

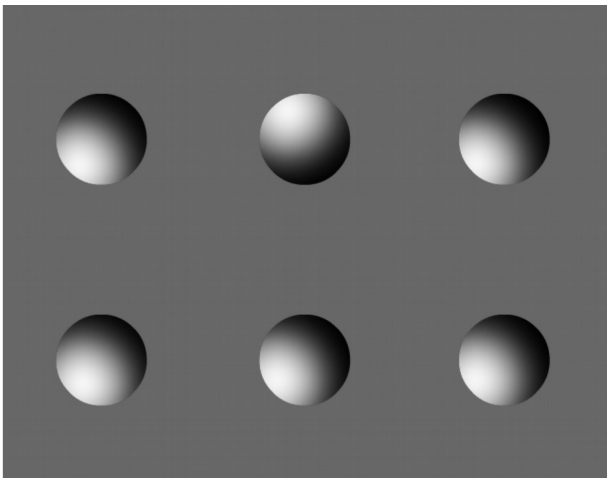


Figure 18: Wick's Shadow Illusion. Image taken from [1].

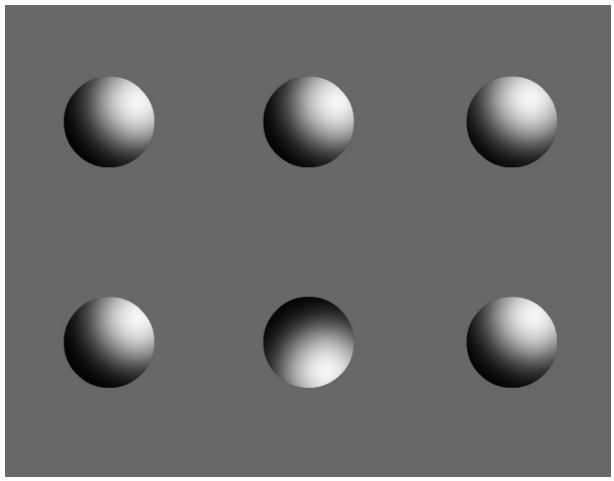


Figure 19: Wick's Illusion rotated about 180°.

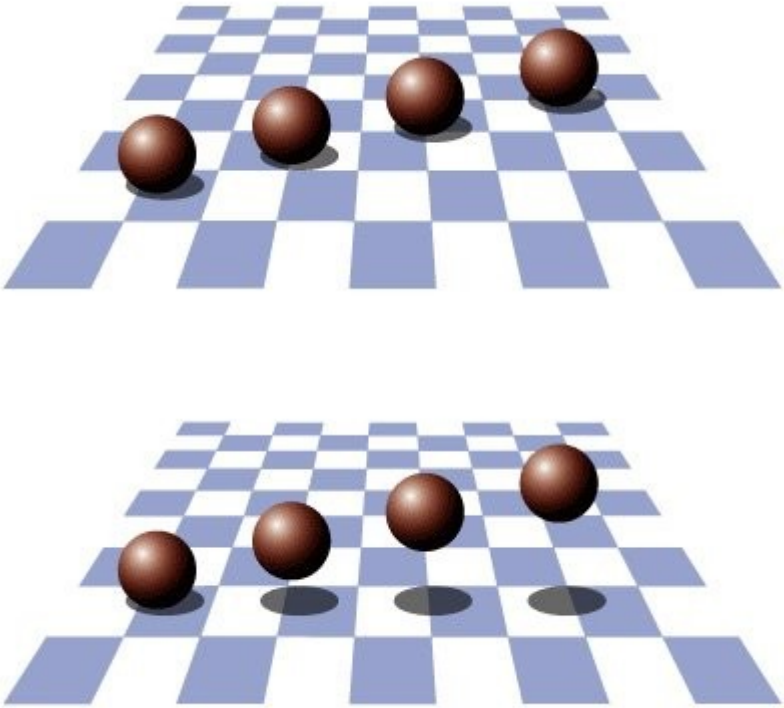


Figure 20: Kesten's Ball and Shadow Illusion. Image taken from [1].

Perspective Illusions

Perspective is very important for estimating distances and the size of objects. It was discovered at the beginning of the 15th century, it was then also used by artists to give the paintings a more realistic impression. To be exact it is an illusion if a painting has a realistic three-dimensional perspective, because it is only two-dimensional. However also illusions can be created through perspective where the perspective does not match the reality.
The Stanford-Psychologist Roger Shepard created some perspective illusions. Among them are the Tabletop Illusion where through perspective the tabletops appear to have different shapes and sizes even though shape and size are the same (see Figure 21) and also Terra Subterranea where there are

two figures of the same size which appear to have different sizes (see Figure 22). Latter is also created by smart use of perspective. Our experience tells us, that normally in an image objects which are at the lower border are closer to the viewer and the closer objects are to the horizon the farther away they are and normally the farther away objects are, the smaller they are. Since we therefore believe that the right figure is farther away, we think it has to be bigger in comparison to the other one.

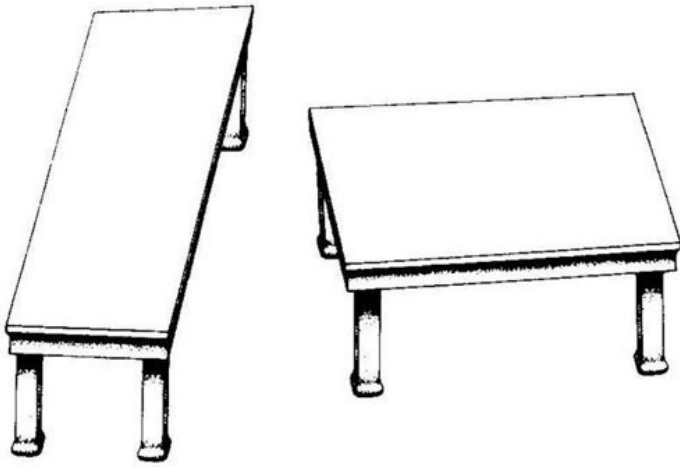


Figure 21: Shepard's Tabletop Illusion. Image taken from [1].

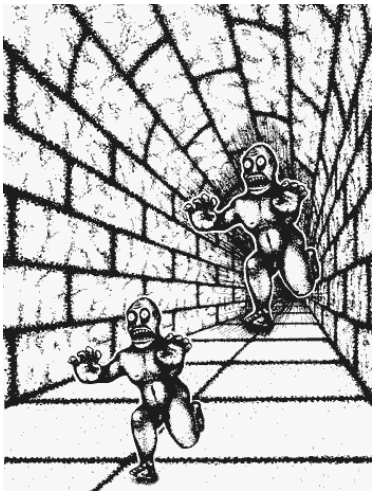


Figure 22: Shepard's Terra Subterranea. Image taken from [1].

Change of Meaning Illusions

These illusions are similar to the Object/Background-Illusions. It is very important for us to identify the important object and the background. Also the meaning of a scene is crucial. Some images have multiple interpretations. They are popular for over 100 years. There are different interpretations of those images until we decide what is the figure and what is the ground. Afterwards it can be very difficult to see the other interpretations.
A good example is Shepard's Illusion with a candle. It is very similar to the Face/Vase Illusion. Either it shows one frontal face and a candle in front of it or two faces in profile which look at each other and the candle in between. It can be seen on Figure 23.
Another famous example is the Old/Young Woman created by the psychologist Edwin Boring, see Figure 24. It shows either an old or a young woman. From the moment we decide for example where the eye is, the other parts of the face get assigned accordingly.

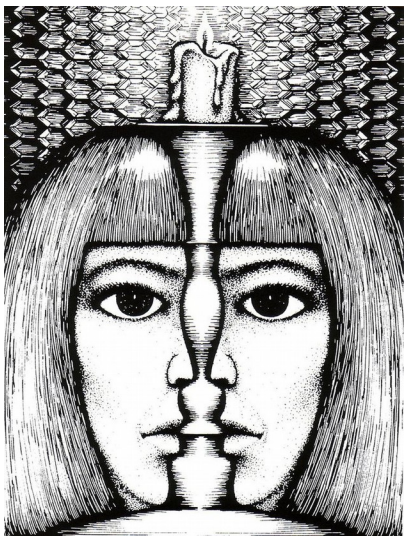


Figure 23: Shepard's Illusion. Image taken from [1].

Figure 24: Borings's Old/Young Woman. Image taken from [1].

Illusions and Faces

A very important part of our visual system is face recognition. For this process different regions of the brain are used. It works different than object- or image recognition. A proof for this is that we can still recognize an object on an image which is upside down, but with faces this gets more difficult. We are used to perceiving the world upright. Therefore our face recognition is trained on faces of people standing upright. However, there was an experiment with monkeys which showed that they were able to recognize faces upright and upside down because of the fact that they are often hanging upside down a tree and therefore their face recognition was also trained with upside down faces.
For humans it is not that easy. Mostly when we look at faces we do not look at the whole face but only at parts of it. The illusion with the upside down face shows this. At first sight the upside down face looks perfectly normal to us, because we only look at single parts of the face like the eyes and not at the whole face. Therefore we do not see that the face, nose and so on are upside down but the mouth and eyes are upright. Also we are not able to recognize facial expressions on upside down faces. Of course this leads to a very strange looking face when the image is turned around. This is shown on Figures 25 and 26.

Figure 25: Image of an upside down face where eyes and mouth are still upright. Taken from: http://www.edinburghnews.scotsman.com/webimage/1.4040976.1456914540!/image/88927743 1.jpg_gen/derivatives/landscape_300/889277431.jpg

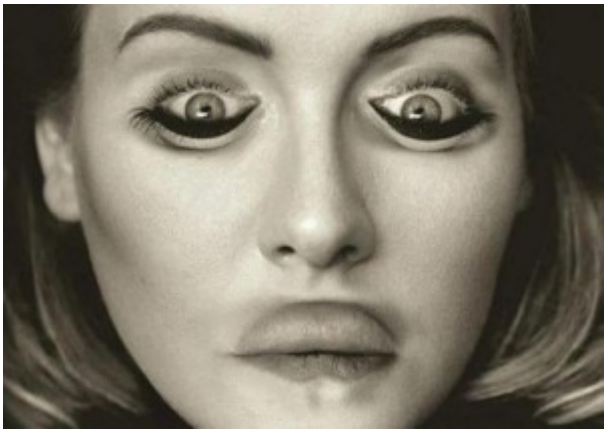


Figure 26: The same image as in Figure 25, rotated about 180°.

Headfirst Illusions

As stated before we are used to perceiving the world upright and even if we tilt the head, our image of the world stays stable. In this category there are images which show different things when turned upside down, but since we perceive the world upright, we do not see them because we do not anticipate that the image will show something different when turned upside down.
An illusion of the Italian artist Giuseppe Arcimboldo shows either a bowl with vegetables or a face when turned upside down, see Figures 27 and 28. It is also an example for a collage where different objects are combined to form one bigger object. In this case, a face is formed by using different vegetables.

Figure 27: Arcimboldo's Vegetable Gardener. Image taken from [1].

Figure 28: The same image as in Figure 27, rotated by 180°.

References

[1] Seckel, Al; Optische Illusionen. Premio, 2008

[2] http://www.ritsumei.ac.jp/~akitaoka/index-e.html (8.5.2016)

[3] Kitaoka A., Ashida H; Phenomenal characteristics of the peripheral drift illusion. VISION Vol.15, No.4, 261–262, 2003

Image Understanding

Protocol - Illusions

Rebecca Nowak (0626227)

Vienna, 12th May 2016

1 Summary

The complexity of human vision has been vastly underestimated by researchers in the past. There are still parts of the visual system we do not understand. For a long time the camera was assumed to be an analogy of the visual system. This is being questioned, as a big part of the visual system relies on interpretation. Illusions are often studied to gain insights into the visual system. The following illusions were mentioned in the presentation:

- Brightness/Contrast Illusions
- Scintillating illusions
- Parallelity Deceptions
- Object/Background-Illusions
- Wrong Wstimations
- Stereo Illusions
- Movement Illusions
- Impossible Figures
- Shadow Illusions
- Perspective Illusions
- Change of Meaning Illusions

2 Opponent

The Sun-in-cloud illusion shown in figure 1 makes the sun appear to go through clouds.[8]

Figure 1: Sun-in-cloud illusion[8]

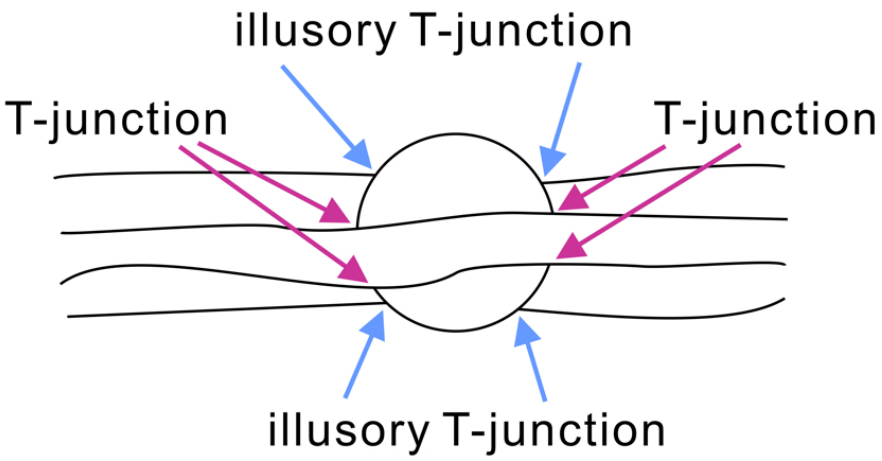


Figure 2: Sun-in-cloud illusion - explanation[8]

A well known illusion is the road mirage.

Figure 3: Mirage[3]

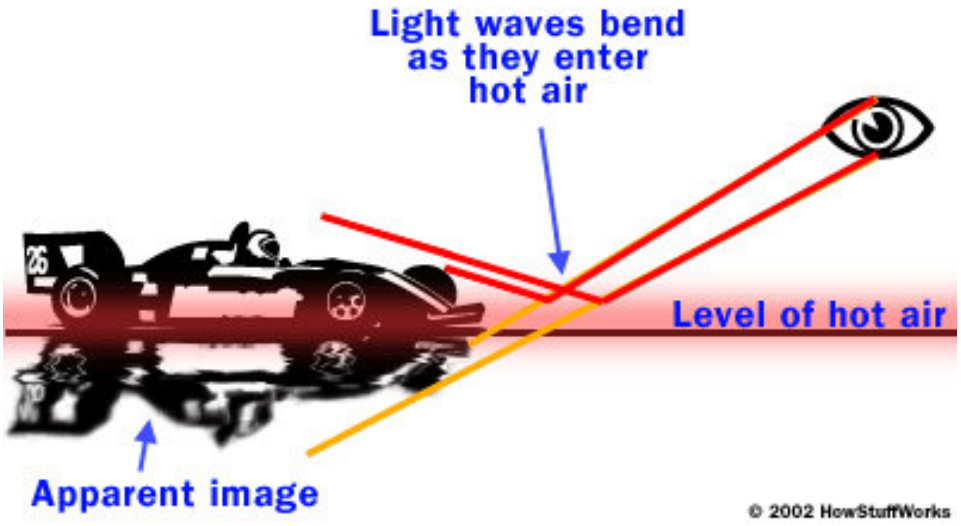


Figure 4: Mirage - explanation[1]

Combinations of incomplete figures can seem like visible contours even when the contours do not actually exist. This effect is called subjective contours.

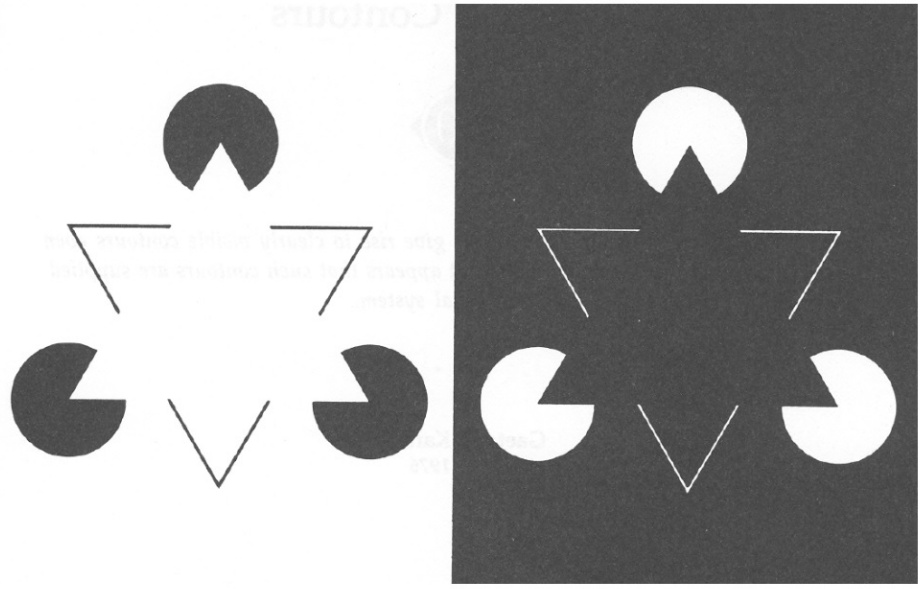
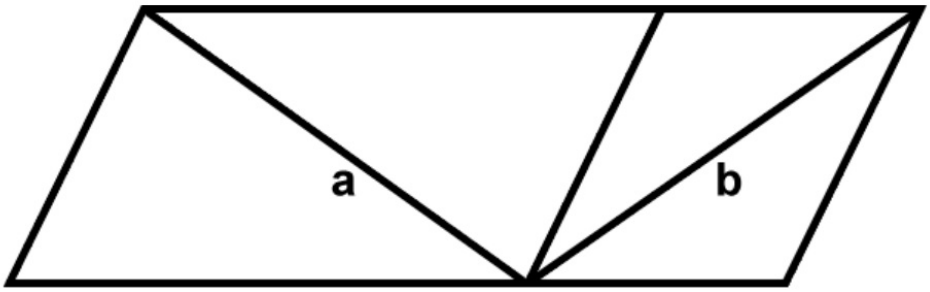


Figure 5: Subjective contours.[2]

Segall et al. describe the influence of culture on visual perception in [4]. An example of this is the fact, that the illusion in figure 6 (Sander's parallelogram) has a stronger effect on people from western cultures.[4]



Sander's parallelogram
Sander, F. *Neue Psychol. Studien* 1931, 8, 311

Figure 6: Sander's parallelogram [4]

[5] describe differences between how deep neural networks and humans recognize objects. Images that are completely unrecognisable to humans may be recognised by a DNN with a very high confidence as a familiar object.

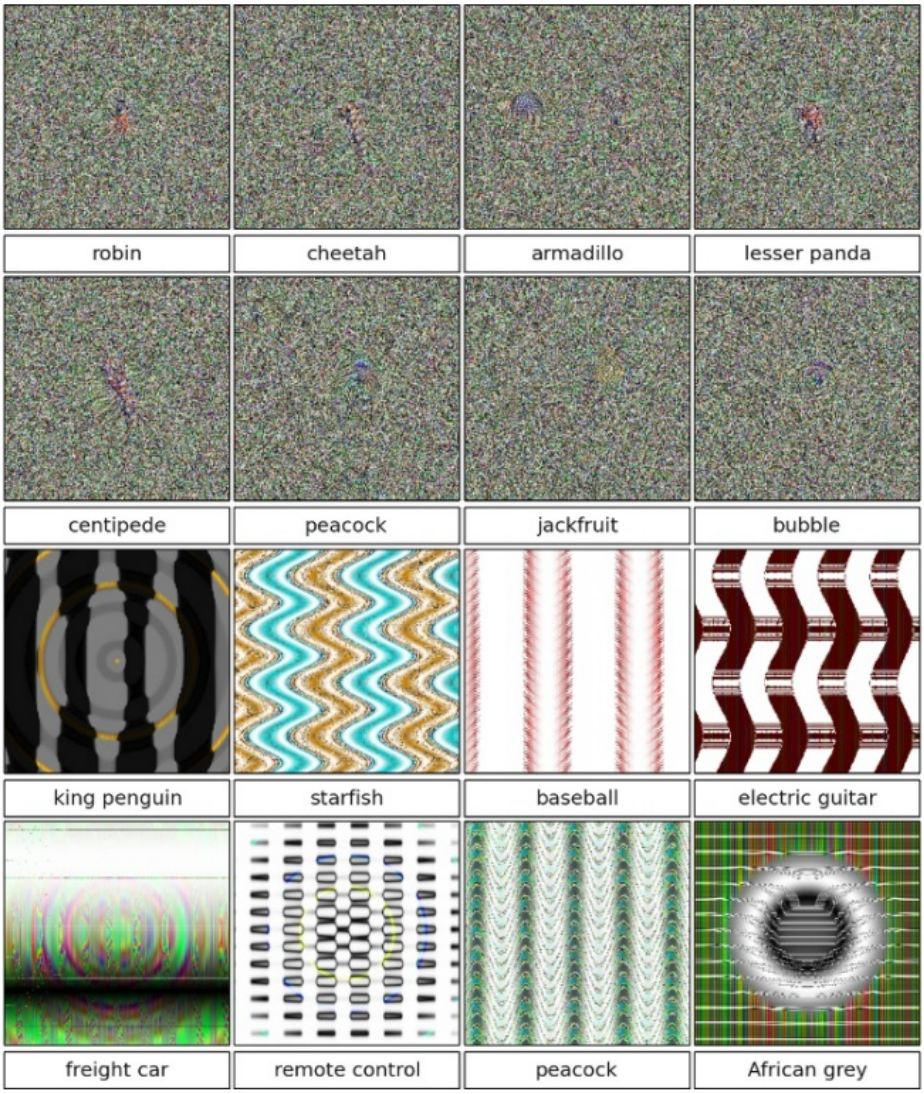


Figure 1. Evolved images that are unrecognizable to humans, but that state-of-the-art DNNs trained on ImageNet believe with $\geq$ 99.6% certainty to be a familiar object. This result highlights differences between how DNNs and humans recognize objects. Images are either directly (*top*) or indirectly (*bottom*) encoded.

Figure 7: Deep neural networks and human perception. [5]

Another example can be found in [6]. Changing an image in a way that is imperceptible by humans can cause a deep neural network to label an object as something else entirely.

(a) (b)

Figure 6: Adversarial examples for QuocNet [10]. A binary car classifier was trained on top of the last layer features without fine-tuning. The randomly chosen examples on the left are recognized correctly as cars, while the images in the middle are not recognized. The rightmost column is the magnified absolute value of the difference between the two images.

Figure 8: Deep neural networks and human perception. [6]

3 Discussion

What can we learn from illusions for image understanding? For example, we could build a detector for edges that the human visual system detects even though they are not there (like the ones in figure 5).

The illusion mentioned in the presentation as "Illusions and Faces" is also known as the thatcher effect, because the paper describing the effect used a photo of Margaret Thatcher. [7] During the discussion, the question arose whether a face recognition algorithm would recognize an upside down face. The conclusion was that it would depend on the algorithm and the training set.

Neural networks are being widely used for computer vision tasks. The danger of neural networks is that we cannot look inside, we have to experiment to find out what it does and does not recognise.

The webpage www.cfar.umd.edu/~fer/optical presents a new theory about visual illusions. Many of the well known geometric optical illusions can be predicted with the theory. It is based on the principle of uncertainty of visual processes. This principle states, that our visual system makes estimates, and these estimates do not correspond to the true value. Most of the time the error is too small to be perceivable, but in certain patterns where the error is repeated it becomes noticeable.

References

[1] Tom Harris. Making a mirage. http://science.howstuffworks.com/mirage2.htm, 2002. [Online; accessed 11-May-2016].

[2] Gaetano Kanizsa. Subjective contours. *Scientific American*, 234(4):48–52, 1976.

[3] Akiyoshi Kitaoka. Physical illusion. http://www.psy.ritsumei.ac.jp/~akitaoka/physicalillusione.html, 2010. [Online; accessed 11-May-2016].

[4] Melville J. Herskovits Marshall H. Segall, Donald T. Campbell. The influence of culture on visual perception. 1966.

[5] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014.

[6] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[7] Peter Thompson. Margaret thatcher: a new illusion. http://www-users.york.ac.uk/~pt2/thatcher1980.pdf, 1980.

[8] Kunio Watanabe. Sun-in-cloud illusion. http://www.psy.ritsumei.ac.jp/~akitaoka/kumokugurie.html, 2016. [Online; accessed 11-May-2016].

Scene Understanding

Geyer Lukas

May 29, 2016

1 Introduction – What is scene understanding?

1.1 Psychological viewpoint

The steps that a human takes to understand a scene are estimated to be in the following order [9]:

Gist Only within a split second, the human brain is able to classify the scene which is presented to it. This happens even before all the separate objects in the scene are analyzed. It represents the first impression and might be refined or altered later on. For example, regarding Figure 1, the first glance reveals that it shows some sort of sport event.

Objects In the second step, (a selection of) the objects in the scene are focused and classified. In the example image, one might subsequently concentrate on the three depicted persons. This might result in noticing that they are all female and each of them has a prosthetic leg, which leads to further insight into the content of the scene. The first impression, that it depicts a sport event, is now refined with the assumption, that it is a sport event for women with physical disabilities.

Actions (relationships) Now is the time to analyze the actions that are happening in the scene, or also the relationships between the found objects: e.g. if something is placed on top of something else, or if an object is held by somebody. In Figure 1, the actions are for example that the leftmost person lies on the back, and the other two are running/jumping.

Events (collective actions) The actions are associated to form actions. The actions in the example image leave the impression, that one of the runners fell down while the other two try to dodge her.

Character's feeling/perspective In a final step, a human looking at the scene might also try to interpret, what the depicted persons are currently feeling and thinking.

Figure 1: A picture of a sport event. Image taken from [9].

1.2 Definition from a computer vision viewpoint

In an editorial for a chapter dedicated to scene understanding, published in a journal in 2015, the ultimate goal of scene understanding is described as follows:

> "Scene understanding is the ability to visually analyze a scene to answer questions such as:
>
> - What is happening?
> - Why is it happening?
> - What will happen next?
> - What should I do?"
>
> Hoiem, Derek et al. in *Guest editorial: Scene understanding* [4]

While these are quite ambitious goals, the subsequent description of the papers contained in the journal shows, that the main focus of the current research still lies on the low-level aspect of scene understanding: all the papers confine themselves to estimate the position and nature of the scene content. The following scene interpretation is not covered by these works. Accordingly, this report will also focus onto the description of low-level scene understanding.

2 Some of the goals in scene understanding

While the final goal in each scene understanding task is, to acquire some sort of semantic meaning about the scene, the results of the intermediate steps can differ. Two possible intermediate goals [2] are listed below:

Figure 2: A scene labeling. On the left is a pixel-level scene labeling. In the middle is the input image with bounding boxes around each detected object instance of the class *car*. On the right is a combination of the pixel-level labeling and the object detector result, which might contribute to achieve instance-level labeling. Image taken from [5].

2.1 Scene recognition

Scene recognition is not the same as scene understanding, but a subdiscipline. It is comparable to the extraction of the *gist* of a scene, as described in Section 1.1. With scene recognition, one tries to make assumptions about the scene class without first detecting the objects. A possible approach would be to create a set of different Textons (small texture components), search for them in a training set of images depicting different scenes, and create a prototype histogram of the texton distribution for each scene type. To classify a new image, the texton histogram is created and compared to the prototype histograms [8].

2.2 Scene labeling

In scene labeling, the contents of the scene are labeled to describe the object classes that are visible. In case of a *pixel-level* labeling, one determines for each pixel, to which object class it belongs (see Figure 2, on the left). If multiple objects of the same class overlap each other in the scene picture, then the area depicting these objects merges into one big blob labeled as the object class. Pixel-level labeling is therefore not suited to count object instances, unless they do not overlap. This problem can be avoided by using *instance-level* labeling, where each object instance gets its own label (see Figure 2, on the right).

3 Benchmarking Datasets

There exist a variety of datasets that can be used to evaluate different scene understanding algorithms. Three examples are:

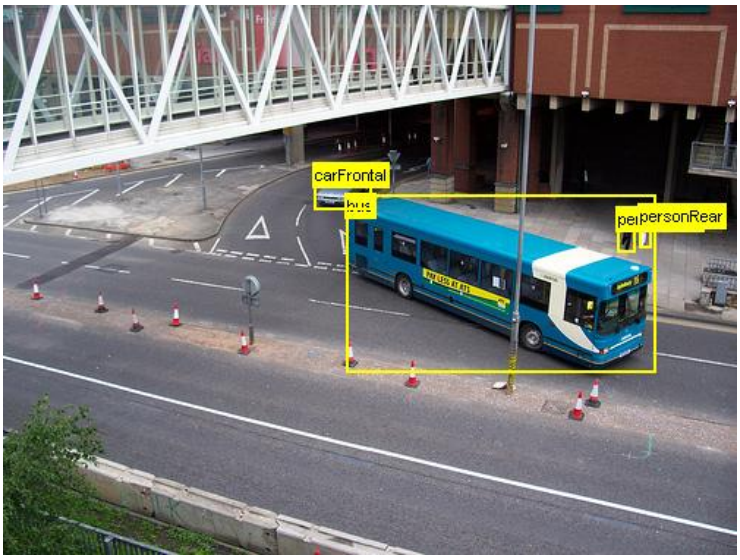


Figure 3: One of the pictures in the PASCAL VOC dataset with bounding boxes around objects as ground truth. Image taken from [3].

PASCAL VOC The PASCAL visual object class challenge [3] ran from 2005 to 2012, where each year, the dataset was updated with new images. The ground truth contains bounding boxes around regions of interest, e.g. object instances of 20 classes (see Figure 3). Since 2009, they also added segmentations to their ground truth.

CamVid The cambridge video dataset [1] contains 10 minutes of video material, recorded from a car driving through cambridge. Over 700 frames are segmented as the ground truth (see Figure 4).

Cityscapes The cityscapes dataset [2] will be released in 2016 and contains a much higher amount of segmented images than PASCAL VOC and CamVid (see Figure 5). In addition to pixel-level labeling, it also provides instance-level labeling.

4 The usage of context

To classify multiple objects in a scene, a first approach might be to regard every object separately and try to determine its class. While this can be sufficient in some cases, this approach ignores a huge source of additional information: the context in which the object appears. An example that shows the importance of context can be seen in Figure 6.

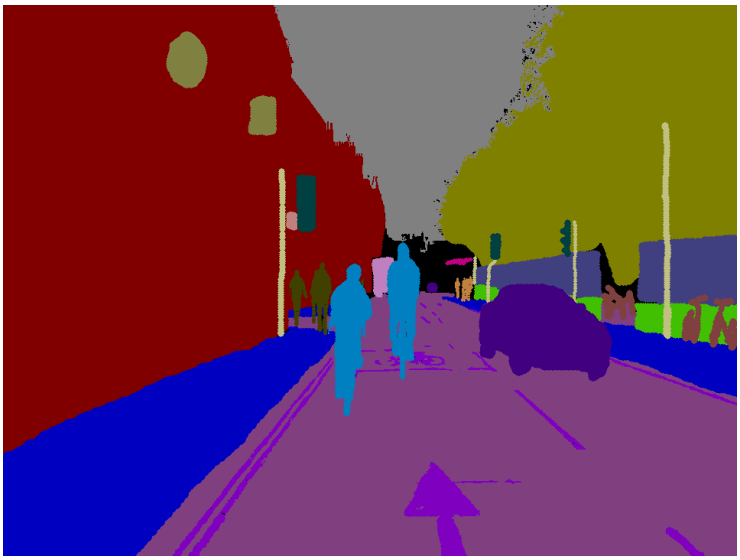


Figure 4: A frame of the CamVid dataset that has been segmented to provide a ground truth. Image taken from [1].

The context information, that can be used to enhance classification results, can be [7]:

Size The size of the object, compared to other objects in the scene, can contain valuable information. For example, if one object, detected as *human*, is much larger than other instances of the class *human*, the probability that it is nevertheless classified as *human* could be decreased (see Figure 7).

Scene type As desrcibed in Section 2.1, it is possible to make assumptions about the scene class without detecting and interpreting single scene elements. Therefore, the obtained scene class can be already available during classification of scene regions. Each scene class has a probability distribution of the labels, that might appear in the scene. By using this probability, the classification can be enhanced (see Figure 8).

Object-scene relationships The position of an object in the scene can provide useful information. Beds are more likely placed next to walls than in the middle of the room, and most objects are placed on the floor rather than flying around.

Object-object relationships In a similar way, a comparison between objects can lead to new insights. The size was already listed separately, but also the position of an object relative to other objects is important. A small furniture

Figure 5: The cityscapes dataset provides a huge amount of city images. The ground truth is a segmentation – either on pixel-level or on instance-level. Image taken from [2].

Figure 6: The context decides if the glyph in the middle is interpreted as an uppercase B or the number 13. Image taken from [8].

Figure 7: The size difference of the two highlighted object instances resembling a human suggests, that they might actually not both belong to the class *human*. The size difference could also be caused by perspective projection, if the object sizes are compared in their two-dimensional representation.

next to a bed is probably a nightstand, and objects distributed around a table are most likely chairs.

5 Papers contributing to scene understanding

The following introduction of some scene understanding algorithms concentrates onto the way of introducing context into the classification. The specific algorithms used to classify objects are not described in this report.

5.1 Combining object detectors and CRFs

Segmentation algorithms usually perform pixel-labeling, which results in segmented images, where the number of object instances can not be retrieved (see Section 2.2). These segmentation algorithms are often based on conditional random fields (CRF). To include the object quantity in the output, Ladickỳ et al. [5] combined the CRF approach with an object detector, which also improves the segmentation, as can be seen in Figure 9.

A CRF is a computer vision model. It is comparable to hidden markov models (HMM), but more powerful. The probability $p(y|D)$ describing how likely e.g. a labeling y is for a given input image D, can be expressed with:

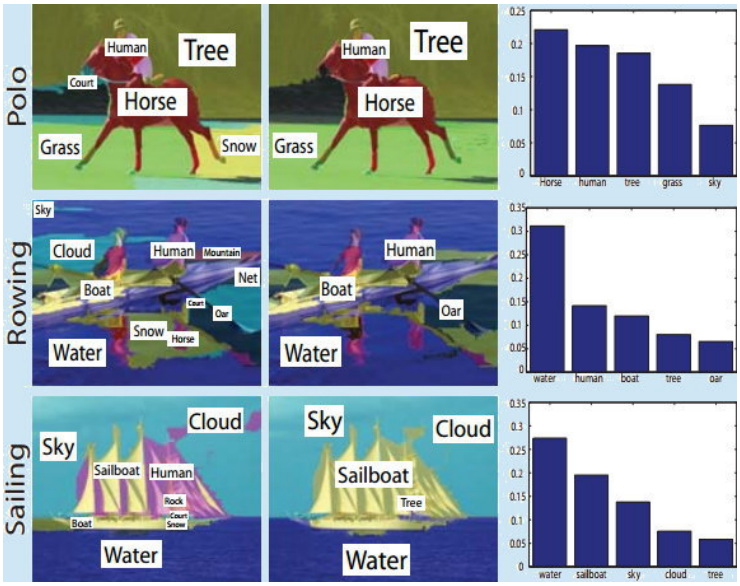


Figure 8: The labeling process can be supported by using information about the scene type. The left column contains image labelings that ignore the scene type, while the middle column uses this context information. In the right column, the five most probable labels of the scene types *polo*, *rowing* and *sailing* are shown. Regarding the polo scene, the low probability of the occurrence of snow, compared to the high probability of grass, changes the wrong labeling *snow* into the right label *grass*. Image taken from [6].

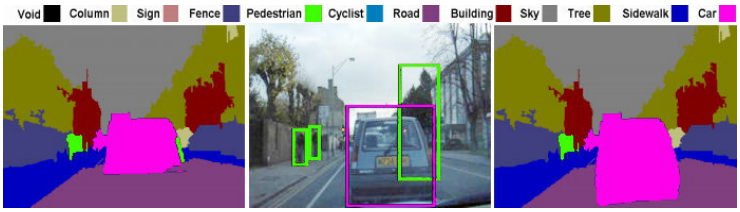


Figure 9: On the left is the result of a CRF-based segmentation. The lower half of the car is falsely classified as *road*. In the middle, an object detector is applied onto the image. The car is detected correctly, but there is one false positive in the *pedestrian*-class. The right image shows the result of a combination of the CRF output and the detected objects. The car is now complete, while the misdetected pedestrian was rejected. Image taken from [5].

$$p(y|D) = \frac{1}{Z} exp\left(-\sum_{c \in C} \psi_c(y_c)\right)$$

The Gibbs energy can be expressed by:

$$E(y) = -\log p(y|D) - \log Z = \sum_{c \in C} \psi_c(y_c)$$

The probability $p(y|D)$ of the labeling y can be maximized by minimizing the energy function.

If $E_{pix}(y)$ is the energy function of the CRF that performs the segmentation, the object detector result can be incorporated into the CRF by adding further potential functions $\psi(y)$:

$$E(y) = E_{pix}(y) + \sum_{d \in D} \psi_d(y_d, H_d, l_d)$$

These additional potential functions encourage consistency within the detected object – if the object detection result is accepted, ideally all the object pixels should get the corresponding label.

5.2 Use of RGB-D images

This approach by Lin et al. [7] is focused on indoor applications. The input image is first segmented into candidate regions using constrained parametric min-cut. The best candidate regions (ranked by an objectness score) are then each enclosed with a three-dimensional bounding box, which is restricted to be parallel to the floor.

The objects are again classified using a CRF. The energy function is minimized for all the object labels $y_i \in \{0, 1, \ldots, C\}$ as well as for the scene type $s \in \{1, \ldots, S\}$ at the same time. So the result is not only a labeling for the objects,

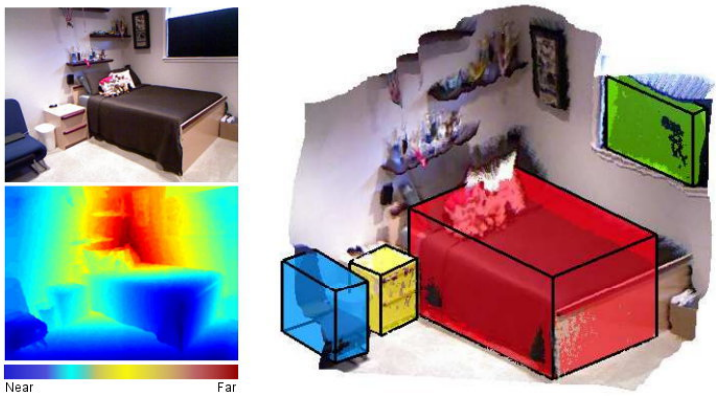


Figure 10: Indoor scene understanding with RGB-D input. At the left is the input: on top is the color image, at the bottom is the depth image. At the right is the scene with four detected objects enclosed by bounding boxes parallel to the floor. Image taken from [7].

but also states which scene is visible. Note that the object label might also become 0, which means that this detected object is rejected because it was deemed to be a false positive.

The energy function has the following form:

$$E(y) = w_s\psi_s(s)+\sum_{t\in U} w_t \sum_{i=1}^{m} \psi_t(y_i)+\sum_{p\in A} w_p \sum_{(i,i')\ P_p} \phi_p(y_i, y_{i'})+\sum_{m\in B} w_m \sum_{i} \phi_m(s, y_i)$$

The feature functions encode the following (contextual) requirements:

$\psi_s(s)$ states how likely the scene type s is, using the result of a previously applied scene recognition algorithm.

$\psi_t(y_i)$ gives the potential, that object i is of class y_i. This term is based on an object classification result that regards each object separately.

$\phi_p(y_i, y_{i'})$ encodes the context between objects, as described in Section 4.

$\phi_m(s, y_i)$ is the term for the context to the scene, as described in Section 4.

5.3 3D object representations

An object in the scene is often represented by two- or three-dimensional bounding boxes – often they are even restricted to be axis-aligned. This representation heavily reduces the spatial precision of the objects. Zia et al. [10] use wireframe representations of objects to increase the spatial precision (see Figure 11). They focused on the detection of cars.

To prepare the object model, a wireframe model has to be generated manually. It is compared to a set of different 3D CAD models showing different car types. With a principle component analysis (PCA) the directions of the main variation

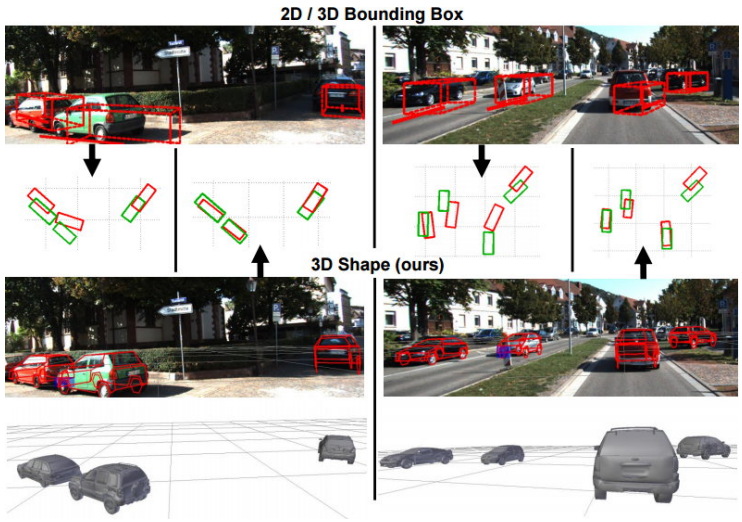


Figure 11: Using wireframe representations of the objects, the positioning could be improved. Image taken from [10].

are detected, so that the wireframe model can be made sensibly deformable with just a few parameters.

For the detection of the car positions, the cars are first located with 2D bounding boxes. The ground plane is derived, and the wireframe models are projected into the scene, using the 2D bounding boxes and the restriction, that the cars are all placed on the ground plane. For the fine-tuning, an objective function is optimized.

6 Conclusion

Scene understanding is a combination of many different techniques from the computer vision research field. To improve the separate results, the context can be taken into account.

References

[1] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *Computer Vision–ECCV*, pages 44–57. Springer, 2008.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *arXiv preprint arXiv:1604.01685*, 2016.

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[4] Derek Hoiem, James Hays, Jianxiong Xiao, and Aditya Khosla. Guest editorial: Scene understanding. *International Journal of Computer Vision*, 112(2):131–132, 2015.

[5] L'ubor Ladickỳ, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip HS Torr. What, where and how many? combining object detectors and crfs. In *Computer Vision–ECCV*, pages 424–437. Springer, 2010.

[6] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Conference on Computer Vision and Pattern Recognition*, pages 2036–2043. IEEE, 2009.

[7] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1417–1424, 2013.

[8] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[9] Gregory Zelinsky. Understanding scene understanding. *Frontiers in psychology*, 4:954, 2013.

[10] M Zeeshan Zia, Michael Stark, and Konrad Schindler. Towards scene understanding with detailed 3d object representations. *International Journal of Computer Vision*, 112(2):188–203, 2015.

Protocol for Scene Understanding 24.5.2016

Presentation: Lukas Geyer (1026408)

Protocol: Michaela Tuscher (0827032)

Short summary of the presentation

Scene Understanding tries to answer questions such as what is happening in a scene, why it is happening, what will happen next and what sould be done. In order to answer these questions following tasks can be executed: image segmentation, image annotation, object detection, object recognition and 3D scene recovery. Also the context is crucial for Scene Understanding. Examples for contextual relationships in scenes can be the size of objects compared to others, the scene type and derived from this the probability of the occurence of certain objects in a scene, the relationship of object positions in a scene and to other objects.

The presented papers contribute to Scene Understanding using Conditional Random Fields, RGB-D images and 3D object representations.

Different datasets with annotations for Scene Understanding tasks are available.

Questions

What is “gist”?
Even if you look at the picture just for 1 second, you can say it is a sports event.

Context between objects, slide 23: Are all the objects compared to all the other objects?
In general yes, but mostly the objects which lie in a neighbourhood are compared to each other.

What is the coloured image on slide 21?
It is the depth image.
It would be best to annotate such things on slides, in this case if possible also with a colour bar which shows which colour depicts closer regions or farther away regions.

Discussion

The methods of the mentioned papers are very restrictive and are only applicable in special cases.

Concerning slide 3:
The question is if this is the same for every human, e.g. if every human would describe the scene

identical or are there other possibilities to interpret this scene? Certainly there are differences between the impressions of different people. Very likely it depends on experience and what somebody knows from the past. For the algorithm it is important what shall be detected or what the algorithm refers to.


[1]

On the other hand a human has to react fast in some situations, e.g. if he sees a bear he has to decide what to do as fast as possible. A human not only gets the information from seeing but uses all of his senses. It is not a bottom-up approach where we build the scene from many details but a popup effect where a small feature suddenly draws our attention towards it. So various stages can be skipped to get to a fast result, which is important in some situations.

Pixel labelling:

Maybe a drawback of this is the merging of some areas. With which granularity does it happen? Why did it detect only one window? Maybe the others were too small or this window has a nice rectangle shape and was therefore detected.

[3]

Datasets and ground truth:

The question arose what is the ground truth of a photo? For a photo there are many ground truths depending on the granularity, e.g. it is possible that one wants to find duct covers on the street but they are not labelled in the ground truth. The problem is, that the ground truth decides about the rating of an algorithm. So if the ground truth does not include e.g. certain labels it can be necessary to create your own ground truth or to adapt the assessment criteria and test all the other algorithms again with the adapted criteria for comparison purposes.

Especially with video sequences it can be a great problem.

Two problems which can arise with ground truths are the quality of the ground truth, e.g. how accurate it is and the discussed problem that for some purposes other kinds of ground truths are needed. (See also the discussion in the protocol of the Benchmarking-presentation.)

Relation object/context:

Maybe it can be a problem that there is a reciprocity in the relation between object and context which can lead to discrepancies, because sometimes an object can only be identified because of the context of the scene but it is also possible that the context can only be identified because of the objects in a scene. On slide 8 the algorithm would likely not terminate and switch between detecting a “B” or a “13”.

[8]

Complexity of optimisation, formula on slide 23:

Convolutional neural networks can learn weights and so on from training and they return the minimas. It can happen that there is more than one extremum which is equally good. This can lead to a situation described above (“B” or “13”), but it depends on what shall be done. E.g. if a number is searched, it would return 13, if a letter is searched it would return B.

$$E(\mathbf{y}) = w_s \psi_s(s) + \sum_{t \in U} w_t \sum_{i=1}^{m} \psi_t(y_i) + \sum_{p \in A} w_p \sum_{(i,i') \sim P_p} \varphi_p(y_i, y_{i'}) + \sum_{m \in B} w_m \sum_i \varphi_m(s, y_i)$$

Why is the statue in slide 12 not a person?
It is decided based on size and colour. One can also utilize perspective, because things which are nearer to the horizon of an image are farther away and therefore should be smaller, but this isn't the case here. But what if someone jumps very high and therefore is closer to the horizon? Also the bigger human-like figures could be really humans which are looking at a model of something e.g. a model railway with small models of humans.

In trying to find out what the real people are in this image one can take into account the ground and the walls and the positions of the people compared to them.
As a human when trying to find the real people in the scene, it depends very much on the material, e.g. we see that the statues are made of stone, but it would be harder to identify the humans if the statues were made of wax. This shows we decide on the basis of many criterias.

Depth-data:
Most of the algorithms don't utilize depth-data and they only detect what they know e.g. cars. To get depth data it is not necessary to have 2 cameras. There are different “shape from...”-approaches and also a vanishing point is sufficient to determine e.g. the speed of a car (markers on highways are useful for this purposes too).

Autonomous cars:
They cause less accidents than cars which are steered by humans. However laws and the legal situation are a problem. The steering wheel has to be mechanically connected. However, it is possible with planes, so why not with cars?

Object representations:
Most papers work with axially parallel bounding boxes. In this case problems occur if there are elongate objects which are not axially parallel, e.g. humans. In 3D depth or time can be set as axis.
How would you model a Ferris wheel? It would be easy to model the cabins but not so easy to model the relationship between them and the wheel, also to model how many cabins there are on the wheel.
Maybe for some purposes it would be better to use ellipsoids. For example for modelling human body parts ellipsoids are very useful because they also indicate a direction.

Digital images:
Digital images are not fully rotational invariant.

References of the presentation:

[1] Zelinsky, Gregory. "Understanding scene understanding." *Frontiers in psychology* 4 (2013): 954.
[2] Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." *arXiv preprint arXiv:1604.01685* (2016).
[3] Ladicky, Ľubor, et al. "What, where and how many? combining object detectors and CRFs." *Computer Vision–ECCV 2010*. Springer Berlin Heidelberg, 2010. 424-437.
[4] Lin, Dahua, Sanja Fidler, and Raquel Urtasun. "Holistic scene understanding for 3D object detection with RGBD cameras." *Proceedings of the IEEE International Conference on Computer Vision*. 2013.
[5] Gupta, Saurabh, et al. "Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation." *International Journal of Computer Vision* 112.2 (2015): 133-149.
[6] Zia, M. Zeeshan, Michael Stark, and Konrad Schindler. "Towards scene understanding with detailed 3d object representations." *International Journal of Computer Vision* 112.2 (2015): 188-203.
[7] Li, Li-Jia, Richard Socher, and Li Fei-Fei. "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework." *IEEE Conference on Computer Vision and Pattern Recognition*. 2009.
[8] Szeliski, Richard. Computer vision: algorithms and applications. *Springer Science & Business Media*, 2010.
[9] Everingham, Mark, et al. "The pascal visual object classes (VOC) challenge." *International journal of computer vision* 88.2 (2010): 303-338.
[10] Brostow, Gabriel J., et al. "Segmentation and recognition using structure from motion point clouds." Computer Vision–ECCV. Springer Berlin Heidelberg, 2008. 44-57.
[11] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." IEEE Conference on Computer Vision and Pattern Recognition, 2009.