Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Graphics and Algorithms
Vienna University of Technology
Favoritenstr. 9/186-3
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: {krw}@mail.com
URL: http://www.prip.tuwien.ac.at/

PRIP-TR-140 June 26, 2017

Image Understanding - Final Script SS2017

Birkner Tamás, Chen Wen Chao, Koszticsák Rebeka, Mai Alexandra, Melan Robin, Pucher Daniel, Winkler Andreas
edited by: Walter G. Kropatsch

Abstract

Image Understanding was first organized in form of an "'inverted classroom"' in the academic year 2015/16. The feedback from the students was very positive. Also for the teacher this type of lecture was an extremely positive experience. It was very motivating to see how active all participants contributed to the topics of their choice in Image Understanding and how much they increased their knowledge about the subject, which was clearly visible in the respective discussions after their lecture unit. Beyond the specific topics of image understanding the participants exercised many activities that are essential for their future profession: literature research, to study a selected topic and to bring the material in a pedagogical format to be communicated and discussed with the other students. Finally each participant summarized his topic, played the role of an opponent for another topic and documented the results of the discussion of a third topic. The task of the opponent was to study the subject that was presented by another student and to initiate the discussion by several critical views and questions related to the subject. This script contains the summaries and discussion reports. We thank all contributors for their work in this collection.

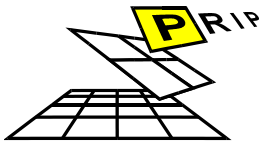



Image Understanding: Schedule SS2017

Find answers to: *What is the role of ... in Image Understanding?*

Date	Title	Pr./Opp./Disc.	unit
7. 3.2017	Organization, understanding images, Applications	krw	A
15. 3.2017	What is Image Understanding?	krw	B
29. 3.2017	Color and Texture	wia/mer/cwc	C
5. 4.2017	(Superpixel-)Segmentation and Grouping	mer/bit/maa	D
10. 5.2017	Vision Models	bit/cwc/kor	E
3. 5.2017	Shapes: Recognition, Models, Generation	cwc/wia/pud	F
31. 5.2017	To See or not to see, Neurophysiology and Illusions	maa/pud/bit	G
7. 6.2017	Scene understanding	pud/kor/wia	H
14. 6.2017	Image Understanding Systems	kor/maa/mer	I

Acronyms follow ...

Assignments 2017

Acro	Matnr.	name	study	topic	opponent	report
bit	1342667	Birkner Tamás	066932	E	D	G
cwc	1129468	Chen Wen Chao	066932	F	E	C
kor	1325492	Koszticsák Rebeka	033532	I	H	E
krw		Kropatsch Walter G.		A		B
maa	1125691	Mai Alexandra	033532	G	I	D
mer	1029201	Melan Robin	066932	D	C	I
pud	1227136	Pucher Daniel	066932	H	G	F
wia	1129264	Winkler Andreas	033532	C	F	H

Contents

Related Books

- Richard Szeliski: Computer Vision: Algorithms and Applications
  Can be found online at http://szeliski.org/Book/
- M. Sonka, V. Hlavac, R. Boyle: Image Processing, Analysis and Machine Vision, 4th edition.
- R. Klette: Concise Computer Vision (2014)
  Can be found online at http://link.springer.com/book/10.1007%2F978-1-4471-6320-6
- David A. Forsyth, Jean Ponce: Computer Vision, A Modern Approach.
  Prentice Hall, 2003. Sample chapters can be found at http://luthuli.cs.uiuc.edu/ daf/CV2E-site/cv2eindex.html

Related Journals

CVIU: Computer Vision and Image Understanding, Elsevier.
Link: http://www.sciencedirect.com/science/journal/10773142

PAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence.
Link: http://www.computer.org/csdl/trans/tp/index.html

PAA: Pattern Analysis and Applications, Springer.

PR: Pattern Recognition, Elsevier.

PRL: Pattern Recognition Letters, Elsevier.

Most important Conferences

ICCV: International Conference on Computer Vision, 2015: Santiago, Chile

ECCV: European Conference on Computer Vision, 2014: Zurich, 2016: Amsterdam

ICPR: International Conference on Pattern Recognition, 2014: Stockholm; 2016: Cancun, Mexico .

CVPR: Conference on Computer Vision and Pattern Recognition, 2015: Boston. 2016: Las Vegas.

Literature for Image Understanding 2017

Complete references are given in Slides of 1st lecture.
Books are available in the PRIP-library, see link below.

C: Color and Texture: Sonka etal, section 2.4; Forsyth etal, chapters 6+9.

D: (Superpixel-)Segmentation and Grouping:
- Aksac2017 (link),
- Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine S□usstrunk. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2274-2282, 2012.
- 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.

E: Vision Models: Forsyth etal chapter 18; Simon J.D. Prince, 2012 (link);

F: Shapes:
Sonka etal chapter 8;
Zygmunt Pizlo, *"3D Shape: Its Unique Place in Visual Perception"*, 2008 (PRIP-Lib).

G: To See or not to see:
R. Blake and R. Sekuler, *"Perception"*, 2006 (PRIP-Lib);
Fermüller etal 2004 (link); http://www.cfar.umd.edu/~fer/optical/

H: Scene understanding:
- Derek Hoiem, James Hays, Jianxiong Xiao, and Aditya Khosla. Scene understanding. International Journal of Computer Vision, 112(2) 2015.
- 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.
- Gregory Zelinsky. Understanding scene understanding. Frontiers in psychology, 4:954, 2013.

I: Image Understanding Systems: Sonka etal chapter 10; Tsotsos92 (Datei)

File 2017

Tsotsos92 (for Topic I) …\References\Tsotsos92.pdf

Links 2017

PRIP library	http://www.prip.tuwien.ac.at/resources/library.php
Simon J.D. Prince, 2012	http://web4.cs.ucl.ac.uk/staff/s.prince/book/book.pdf
Fermüller etal 2004	http://www.cfar.umd.edu/%7Efer/postscript/geo_journal.pdf
Aksac2017	http://www.sciencedirect.com/science/article/pii/S0031320317300110

Color and Texture

Image Understanding SS 2017

Andreas Winkler (1129264)

April 3, 2017

1 Color

1.1 Introduction

Color is a sensation originating from light entering the eye. The perception of color is created from visual receptors reacting to the spectral composition of light rays. The nerve impulses are processed by the human brain to create this sensation known as color [5, chapter 3]. The light rays themselves are not colored. Human color perception is based on the principle of trichromacy. Three different types of color receptors exist within the human eye which react to light of different wave lengths. Color is created from a combination of these impulses. Color perception is a complex process which depends on the context of the observation. Prior knowledge, emotion, spatial relations, contrast, etc. all influence how humans perceive the color of an object.

A surface's perceived color originates from the observed light rays reflected from the surface. The composition of these light rays is dependent on a number of factors. Optics is an important part of physics describing the behavior of light rays. These includes geometric and wave characteristics. The color appearance of light rays in the real world depends on reflection, refraction, absorption, interference, etc. and is only approximated in computer vision applications.

1.2 Color in Image understanding

In human vision, color is used to differentiate between structure-less objects of the same brightness. In image understanding we use color for various tasks such as object detection, image segmentation or classification. In order to understand color in images we need to find an approximate model for the formation of color in an image. Two common light reflection models are Lambert's reflectance model and Shafer's dichromatic reflectance model [6, chapter 3].

Lambert's model assumes that the intensity of reflected light is independent of the viewing direction. This model is only applicable to matte surfaces and completely disregards any specular reflection The dichromatic reflectance model considers Fresnel-

reflection and transmission. The light is described as a combination of diffuse body reflection and directed interface reflection. This model is suitable for specular surfaces.

1.3 Color Spaces

In order to use color information a suitable representation is necessary. For this task various color spaces have been created. The CIE 1931 XYZ color spaces was created to represent every color perceivable by a human. It was created with the use of color matching experiments and formed from the average results of multiple test subjects.
In practice, devices such as displays or printers use a color space created from mixing three primary colors. (e.g. RGB, CMY). These color spaces are only able to represent a subset of the colors within the XYZ color space. When using a specific color space it is important to match the device with the human color perception. This is done experimentally by comparison to reference colors. When device colors are matched colors can be represented in an arbitrary numeric format. Different color spaces used in practice are usually suited for specific applications or highlight specific characteristics of color. E.g. The HSV (Hue, Saturation, Value) color space decorrelates color and brightness and is therefore invariant to shading (under certain assumptions).

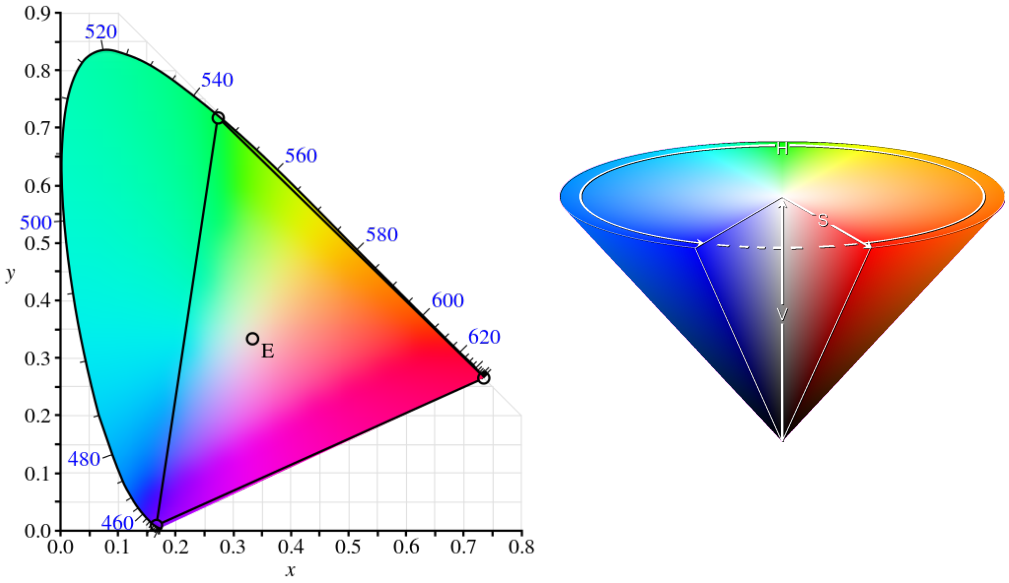


Figure 1: Left: CIE 1931 color model with the CIE RGB color space shown as a subspace. Right: The HSV color cone. Source: Wikipedia.

1.4 Photometric Invariance

There are certain challenges for understanding of color in images. The perceived color of a real world object is not simple the color of its image's pixels [6, chapter 4-7]. There are

various factors influencing the color values of a surfaces in an image: orientation (camera, object), illumination (color, direction, intensity), specularities, shadows, etc. The human visual system is able to filter these factors to perceive the true color and shape of an object. For a computer vision application, these problems need to be addressed. Note: Some of these visual artifacts can be used for certain applications (e.g. object detection via specularities).

Photometric invariance refers to the ability to identify the actual color and shape of objects without these limiting factors. There are various techniques and methods to achieve this.

- *Pixel based methods:* e.g. RGB to HSV conversion. (Hue and Saturation are invariant to illumination changes assuming white light and a dichromatic reflection)
- *Color ratios:* This method assumes that ratios between neighboring pixels stay the same with differing illumination.
- *Derivative based methods:* Methods based on image derivatives (e.g. edge detection) can achieve some level of photometric invariance by filtering out illumination components.
- *Machine learning:* From multiple images of the same object under different illumination a model for the predicted color of the object is created. (example: see Alvarez et al. [1])

1.5 Extraction and representation of color features

There are various ways to describe color features in an image. In order to find a suitable descriptor we look at the requirements for a color descriptor [8].

- Descriptive
- Invariant to illumination
- Unaffected by noise
- Applicable to existing algorithms
- Universally applicable to arbitrary data sets

It is virtually impossible for a descriptor to fulfill all these requirements, therefore certain trade offs need to be made depending on the application. Examples for popular color descriptors are

- RGB histograms
- Hue
- Saturation

- SIFT
- HSV-SIFT
- HueSIFT
- OpponentSIFT
- ColorSIFT
- Color Moments (mean, standard deviation)

All of these color descriptors have different strengths, weaknesses and invariants and are suited for different applications (e.g. object detection).

2 Texture

When color information in images is not descriptive enough or unsuitable for certain tasks, the use of texture might be beneficial. Textures are easily identified by humans but are hard to universally define. A texture is part of an image with certain statistical attributes and similar repeating structures. Whether an image surface is classified as texture mostly depends on scale. A leaf would be considered an object in a close up view, while in an image of a forest it is part of a texture. Generally a group of a large amount of objects that are too small to look at them individually is considered a texture. (e.g. sand, stones, grass, leaves, bricks, fur, etc.)

The self repeating structures within a texture are referred to as texel or textons. Depending on the texture these can have varying size (sand vs. rock), orientation (brick wall vs. pebbles) or regularity (natural vs. artificial).

Psychologists are disputed on the actual purpose of texture in human perception [9]. A common definition describes texture as an attribute of an image region. Texture allows humans to distinguish between objects of the same color and brightness.

2.1 Textures in image understanding

In image understanding textures are mainly used for image segmentation, texture synthesis and 3D reconstruction (shape from texture) [5, chapter 10]. Before any of these tasks can be performed, texture recognition is necessary. Texture-based algorithms are mostly applied in the fields of medicine (e.g. diagnostics) or industry (e.g. quality assurance). In these applications, texture recognition is used to identify certain characteristics of a surface.

2.2 Texture representation

Two fundamentally different ways to describe textures are used in practice: structural and statistical [10, chapter 7].

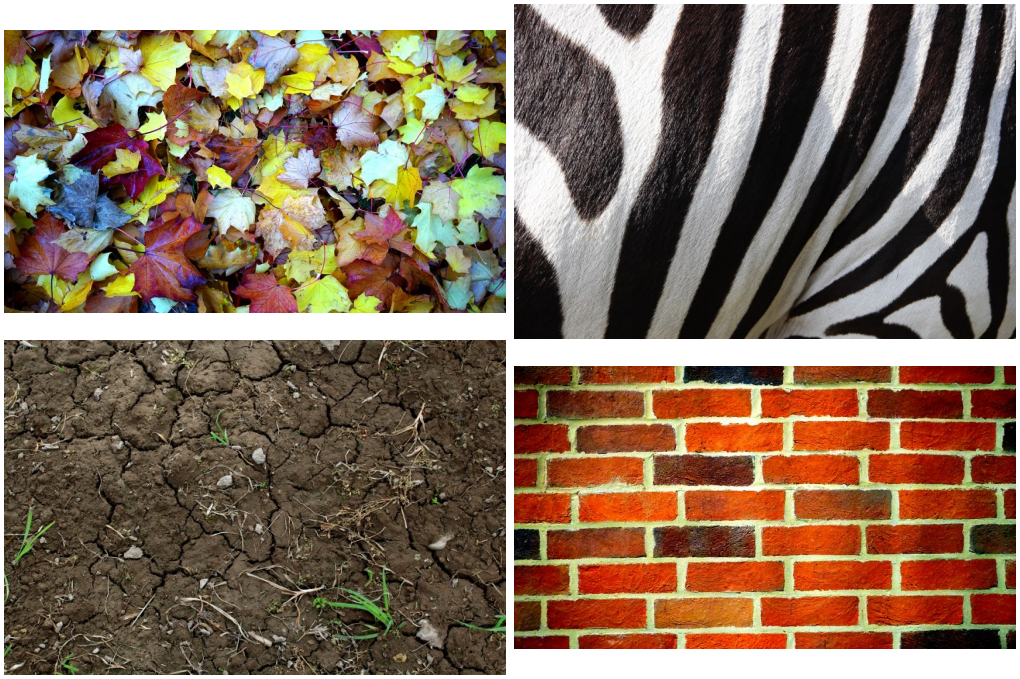


Figure 2: Different types of textures. Source: pexels.com.

Structural textures are described via the appearance and location of their texture elements. This is quite intuitive for humans but very hard to do for algorithms. Whether such a representation is suitable depends on the texture and its texels. If the texels can be clearly identified, Voronoi tessellation can be used to describe the image.

Statistical representations of a texture measure statistical information of color and brightness in the image. While this is less intuitive, it is better suited for algorithmic processing. In practice, statistical algorithms are more common for most applications (e.g. segmentation, classification) and are the focus of this report.

2.2.1 Statistical texture recognition

A large number of different methods and descriptors exist for statistical texture recognition, including [10, chapter 7.3]:

- *Edge orientation and density*: An edge detector is applied on the image. Textures are characterized by the density and orientation of edges within an image area.
- *Local binary pattern*: LBP encode the pixel neighborhoods via an 8bit binary vector (0 = neighbor is lower, 1 = neighbor is greater). The texture is described as a histogram of LBP values.
- *Co-occurrence matrix*: A co-occurrence matrix is a two-dimensional array with the columns and rows representing image values (color or grayscale). The matrix

entries describe the number of occurrences of two image values in a specific spatial relation (e.g. value i appears right of value j)

- *Laws' texture energy measures*: Measures the variation within a fixed window. 9 different 5x5 filter masks are applied on the image. The texture is described as a vector of filter responses. The filter mass are composed of 4 different vectors, which represent different parts of an image.

- *Gabor filter*: Similar to the above method, a texture is represented as an array of Gabor filter responses. A Gabor filter kernel is the product of a Gaussian and an oriented sinusoid. Multiple Gabor filters are used with different orientation and frequency, describing different types of image features.

2.3 Texture segmentation

Texture descriptors such as these can be used to segment an image into regions of uniform texture. The process of texture segmentation does not differ from basic image segmentation. There are two different types of segmentation algorithms: [10, chapter 7.4]
Region based algorithms group and cluster pixels with similar texture attributes
Boundary based algorithms locate and trace edges between regions of different textures.

Simplified Example: A color-invariant texture segmentation algorithm [7].

1. Describe Texture features with color invariant Gabor Filters
2. PCA
3. K-Means clustering

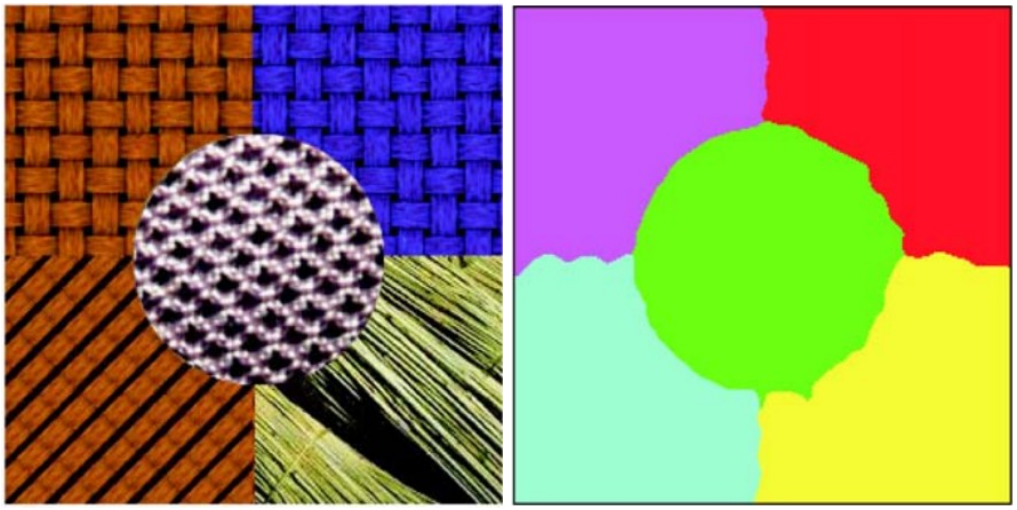


Figure 3: Input image and texture segmentation results for the color invariant algorithm from Hoang et al. [7]

2.4 Texture synthesis

Texture synthesis should not be confused with procedural texturing where textures are created without source material. In texture synthesis, the texture of an input image is extracted and expanded. Commonly this technique is applied in image processing to fill holes in an image or to transfer the texture or material of a surface to another object's structure. Methods for texture synthesis include:

- *Tiling*: The texture image is simply repeated. This is the most basic approach and not suitable for most applications
- *Chaos mosaic*: Randomly place image patches and perform feathering. This produces slight better results.
- *Stochastic methods*: It is assumed the texture originates from some probability density function. We attempt to find this distribution and randomly take color values from this distribution. This disregards any structure in the image.
- *Pixel based methods*: In a scan-line manner the neighborhood for each pixel in the resulting image is compared to pixel-windows in the source image. The source pixel with the most similar neighborhood is used in the result image.
- *Patch based method*s: Blocks of the texture image are copied and merged in the source image. *Image Quilting* [4] places random overlapping blocks and calculates an error metric. Based on the minimal error a boundary between these blocks is chosen (see image 4).

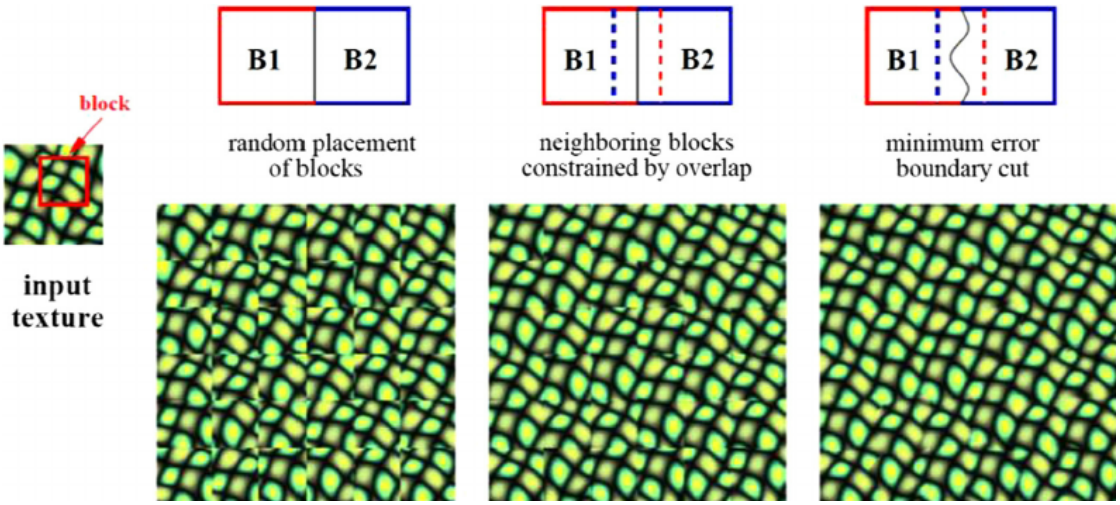


Figure 4: Illustration of the image quilting algorithm by Efros et al. [4]

2.5 Shape from texture

The goal is to reconstruct the shape of an object in an image based on the deformation of its texture [5, chapter 10]. For this task, a structural representation of the texture is preferred as the orientation of individual texels is considered.
If we assume the texels on a surface are of circular shape, they would appear as ellipses on the image plane. From the tilt and slant of the ellipses, the surface orientation can be reconstructed. For arbitrary texels, the orientation of individual texels on the surface needs to be determined first. The general process of a shape from texture algorithm consists of the following steps:

- Identify each texels in the image
- For each texel find the orientation of the texel based on its distortion
- Reconstruct the surface from the orientation of the texels.

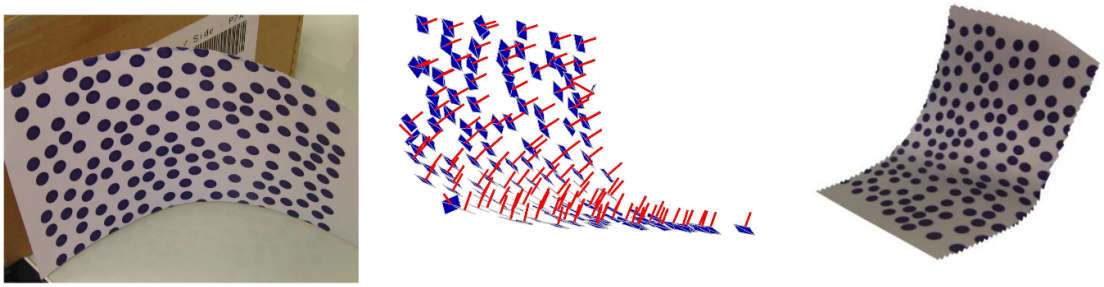


Figure 5: Input image and results from 'Single-view Perspective Shape-from-Texture with Focal Length Estimation' from Collins et al [3].

3 Conclusion

Color and texture have an important role in the human perception of the world. In image understanding a range of challenges need to be addressed to make use of this information. Color and texture descriptors are powerful tools for various image understanding algorithms. Depending on the type and goals of the application the right descriptors need to be chosen.

Steadily new methods for both versatile or specialized color and texture descriptors are proposed. Machine learning is an interesting trend for both color and textures ([11], [2]).

References

[1] Jose M Álvarez, Theo Gevers, and Antonio M López. Learning photometric invariance for object detection. *International Journal of Computer Vision*, 90(1):45–61, 2010.

[2] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015.

[3] Toby Collins, Jean-Denis Durou, Pierre Gurdjos, and Adrien Bartoli. Singleview perspective shape-from-texture with focal length estimation: A piecewise affine approach. *3D data processing visualization and transmission (3DPVT10)*, 2010.

[4] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[5] David A Forsyth and Jean Ponce. A modern approach. *Computer vision: a modern approach*, pages 88–101, 2003.

[6] Theo Gevers, Arjan Gijsenij, Joost Van de Weijer, and Jan-Mark Geusebroek. *Color in computer vision: fundamentals and applications*, volume 23. John Wiley & Sons, 2012.

[7] Minh A Hoang, Jan-Mark Geusebroek, and Arnold WM Smeulders. Color texture measurement and segmentation. *Signal processing*, 85(2):265–275, 2005.

[8] Rahat Khan, Joost Van de Weijer, Fahad Shahbaz Khan, Damien Muselet, Christophe Ducottet, and Cecile Barat. Discriminative color descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2866–2873, 2013.

[9] Michael S Landy and Norma Graham. 73 visual perception of texture. *The visual neurosciences*, 1:1106, 2004.

[10] Linda Shapiro and George C Stockman. Computer vision. 2001. *ed: Prentice Hall*, 2001.

[11] Boyang Su, Jie Shao, Jianying Zhou, Xiaoteng Zhang, and Lin Mei. vehicle color recognition in the surveillance with deep convolutional neural networks . 2015.

Report - Color and Texture

Image Understanding SS 2017

Wen Chao Chen (1129468)

April 4, 2017

1 General dicussion

1.1 Color Symbolismn and Color Harmony

The perception of color and its meaning differs strongly depending on the geography and the cultural differences. Where in one country a certain color represents vitality, in another country the same color might be a symbol of grief or mourning. In this aspect, the harmony people believe certain color combinations provide more so than others do influence our perception of color, e.g. countries of Eastern Europe tend to use a combination of red, blue and white. Regarding this, languages are also be quite influential, as the Inuit for example use several words describing various facets of one color, or traditional chinese only needs one character for 3 different colors. There is also the fact that the distribution of photoreceptor cells is different for each individual person, where people having a deficiency of cones end up with the result of living with color deficiency or color blindness, which means that this group of people in turn perceive color differently. Apart from the human being, it is also unclear, how animals perceive color, if they are able to perceive them, and if so, how? So, the main question remaining is whether color is distinctive provided the situation that the context in which the color has a certain meaning is removed.

1.2 Color Models and Color Invariance

The opponent took an approach to look at the different color models in use, as the practice has shown a wide variance, starting from the renowned RGB color model, over to the HSI color model, and other alternatives, which shall focus on color attributes that are useful for certain fields of practice, as, e.g. the normal RGB color is not ideal for object recognition. However, given the fact, that using a color model already provides us a description of color information as numerals, we can use that information to transform it into any color model and vice versa. The only problem hereby is, that by using numerals to describe color, we have already computed an approximation, therefore lost accurate color information. However, in many practical cases, it is not necessary to have

perfectly accurate color information. In some cases, we even reduce the color space down to one component, because sometimes grey values suffice for the cause.

1.3 Texture

In general, people think of textures as an idle, non-moving layer over an object. Dynamic textures, however, change the appearance of the texture depending on the position and angle of the viewer. Examples of dynamic textures are: Driving in the evening or at night when it is raining, or driving in summer when it is forty degrees celcius outside and the asphalt is heating up, blurring the vision of the one driving. So in general, a dynamic texture is an overlapping of two motions, given a perspective mapping. So how do we detect changes or defects in such textures, e.g. a small hole in a sock? We could try to provide scale-invariance by introducing image pyramids, however, we don't know how many levels we actually need. In that specific case of detecting texture defect in a sock, a solution would be to have human skin underlying the texture.

(Superpixel-) Segmentation and Grouping

Image Understanding SS 2017

Robin Melán (1029201)

April 12, 2017

1 Segmentation Definition

Szeliski [11] defines segmentation as being the task of finding groups of pixels that *go together*. According to this definition the challenge of segmentation is to find natural groupings in the image, segmenting pixels according to their properties. For Jähne [8] an operation is called segmentation when each individual pixel is checked to see whether it belongs to an object of interest or not. This interpretation goes a little further into the actual topic of image understanding, where we are challenged by the fact of separating the image into relevant (object of interest in the foreground) and irrelevant (background) parts. Segmentation is a essential step in image analysis and achieving the goal of image understanding: to *make machines see*.

2 Segmentation Methods

Segmentation methods are distinguished between:

Pixelbased Method take into account the grayscaled value of the pixel and compare it with a threshold

Edgebased Method find edges in the image and follow them to solve discontinuities

Regionbased Method find coherent regions in grayscaled images

Modelbased Method segmenting object of interest with predefined knowledge

Texturebased Method identify certain textures in the image

2.1 Pixelbased Segmentation

Pixelbased algorithms work for **very simple images**, in the best case with a bimodal histogram, having bright objects on a dark background or vise-versa. This methods are easy to implement and give good results on special cases. Disadvantages are, that it is prone to illumination changes in the image, does not necessarily return connected regions, fails with textured objects and the results are strongly dependent on the threshold.

2.1.1 Global Threshold

For the whole image a threshold is defined and compared to every pixel, turning the grayscaled image into a binary. This works well when clear peaks in the histogram are visible and in between the minimum can be selected as threshold. This technique shows problems when dealing with noise and illumination differences in the image leading to false segmentation of the object and background.

2.1.2 Local Threshold

To achieve a better segmentation with illumination changes the local threshold method can be applied. The image is divided into regular regions and for every region a threshold is computed, which is compared with every pixel in that region. This way global illumination changes are suppressed.

2.1.3 Dynamic Threshold

With the dynamic threshold algorithm for every pixel, a threshold is computed in its window dynamically. This returns the best results regarding pixelbased segmentation, but has the worst runtime, since a threshold is computed per pixel and not per grid or image.

2.2 Edgebased Segmentation

In edgebased segmentation algorithms the image is searched for gradient intensity changes. The problem of well known edge detection algorithms, like Sobel, Prewitt, Laplace, etc, is that no continuous, closed edge borders are provided. Therefore so called edge-linking algorithms are necessary to complete and link edges to closed regions.
In the literature there are many different edgebased segmentation algorithms from which two of them are explained in the following.

2.2.1 Watershed Algorithm

The watershed algorithm was first introduced by Vincent and Soille [13]. The procedure is very simple: The image is transformed into a grayscaled image and the object contours are defined where strong intensity changes of the gray values are found (see Fig. 1 [1]).

[1] `https://en.wikipedia.org/wiki/Watershed_(image_processing)`

Figure 1: Explanation of the Watershed Algorithm

The great advantage of this algorithm and the reason why it is still widely used in practice, is that it can be extended easily, returns continuous, closed regions and is fast. Disadvantages are that it returns highly over-segmented images and further reduction of segments is not trivial.

2.2.2 Active Contour Models (Snakes)

Another widely used algorithm is active contour models (ACM) or better known as Snakes in literature. Snakes is defined as a explicit curve described with advanced information on the boundaries of the shape. This predefined information is for example the shape of a spine as can be observed in Fig. 2 from E. Spodarev presentation slides [2].

The ACM algorithm resolves a minimization problem, where the energy function is defined by the internal energy term and the edgebased external term. The first one represents the predefined information and the second term uses the input image. The external term is low when snakes coincide with the actual contour information in the

[2] Jun.-Prof. Dr. Evgueni Spodarev, Bildsegmentierung mit Snakes und aktiven Konturen, Universität Ulm, 2005

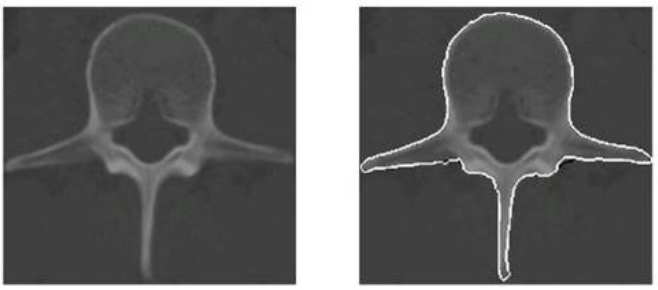


Figure 2: Segment the spine with ACMs.

image. Snakes is initialized outside of the segment, starting with high energy. By minimizing the function iteratively snakes deforms actively towards the desired segment.

By being a deformable model, snakes can adapt to differences, noise and bypass holes. Problems occur when the topology changes or the object has concave curves which can not be sampled correctly by the control points of snakes curve. Regarding the topology changes there is an extension possible with *topological adaptive snakes* found in literature as T-Snakes.

2.3 Regionbased Segmentation

Region-oriented methods rely mainly on the assumption that neighboring pixels within one region have similar values. This procedure works well for images with homogeneous regions.

2.3.1 Region Growing

The basic idea of region growing is to start from an arbitrary pixel, the *seed pixel* to fill a coherent region. The neighboring pixels are compared with the seed point and if the similarity criterion is satisfied they are joined to the region. This is done until all pixels are assigned to a region. Drawback of this method are its inconsistency returning different results when distinct seed pixels are used. Another problem can be if the seed points is starting on an edge.

2.3.2 Region Splitting

Region splitting is a classical top-down approach partitioning the image into disjoint regions. Initially the complete image is viewed as the *area of interest*. Looking at this area to remain one region, all pixels contained in the region have to satisfy some similarity constraint regarding their properties. If this is true a region in the image was found, otherwise the *area of interest* is split into four regions which again are viewed all separately as new *areas of interest*. This splitting of the image can be described using a quadtree.

2.3.3 Split and Merge

Using only *region splitting* in the final segmentation would lead to many neighboring regions that have identical or similar properties. Therefore a merging process is used after each split which compares adjacent regions and merges them if necessary. Algorithms of this nature are called *split and merge* algorithms. So in the quadtree structure each non-terminal node has at most four descendants, although it may have less due to merging.

2.3.4 Pyramid Linking

Another regionbased method is the pyramid linking, which at first computes a Gaussian pyramid. As a next step the connection from children to parents are removed and

assigned new. So on the base level of the pyramid every child is assigned to their original parent or its neighbors depending on its similarity properties. This way some children are associated to a new parent node and some parents in the next level to new parents until to the top of the pyramid. The children mean value sets the new value of the parent in each level. This is done iteratively until some stopping criterion is reached. The nodes on the top level of the pyramid define the segmentation in the image.

2.4 Modelbased Segmentation

Modelbased segmentation algorithms can be used, when preinformation of the shape of the object is known. The difference of these algorithms compared with the previous techniques are that they not only consider local information. Since the human visual system is more complex than that in many applications it is important to consider previous knowledge about the possible shape. In the literature Active Contour Models, which has been explained already in section 2.2 since the curve is deforming according to the gradient information in the external term of the energy function, it is sometimes also categorized as modelbased segmentation, since predefined information contributes to the internal term. Other modelbased segmentation algorithms are explained in the following.

2.4.1 Hough Transformation

The hough transformation is a procedure which works well with geometry figures, like circles, lines, ellipses, etc. which are represented as mathematical equations. This can be observed in the image Fig. 3 from J. Kürbig presentation slides [3] where linepoints and line segments where found and after the hough transformation the lines are completed.

2.4.2 Template Matching

In the template matching procedure masks representing the object or parts of it, are used to find a possible matching in the input image. These masks should be as much as possible invariant to illumination and contrast differences. Although the biggest problem

[3] Jens Kürbig und Martina Sauter, Bildsegmentierung und Computer Vision, Universität Ulm

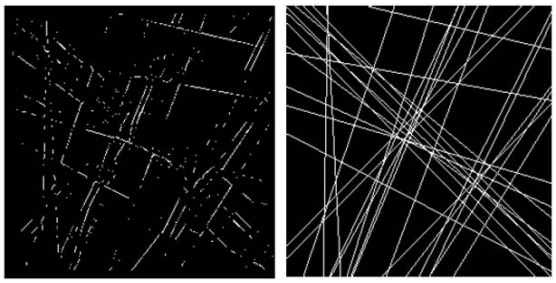


Figure 3: Hough Transformation from J. Kübig presentation slides.

of template matching is its time complexity. More masks means more time it takes to check all possibilities. In practice template matching is not used very often anymore.

2.5 Texturebased Segmentation

Regarding the texturebased segmentation I want to refer to the Andreas Winkler summary about *Color and Texture* where the statistic representation of texture has been discussed and the methods where texture analysis is used. In most of the procedures texturebased segmentation is used to improve and help other methodologies to achieve better results.

3 Superpixel Methods

Achanta et al. [2] describes superpixel algorithms as grouping pixels into perceptually meaningful atomic regions which can be used to replace the rigid structure of the pixel grid. According to this definition, the idea of superpixels is to capture the image redundancy, provide convenient primitives from which image features can be extracted and it reduces complexity of subsequent image processing tasks.
The desired properties of superpixel algorithms are most importantly the adherence of boundaries. How well a method adheres the contours can be measured with the quantitative evaluations of the boundary recall and the under-segmentation error. The first evaluation describes the fraction of the ground truth edges that fall within at least two pixels of a superpixel boundary. The second measures the amount of superpixel "leak" for a given ground truth region. The third important quantitative evaluation is speed. When superpixels are used, one of the desired properties are to reduce computational complexity as a pre-processing step. Superpixels should be fast to compute, memory efficient, and simple to use, so that they improve the following steps of a method.
Observe a comparison of all the State-of-the-Art approaches in Fig. 4 where SLIC (Simple Linear Iterative Clustering) algorithm outperforms most of them in all three categorize.

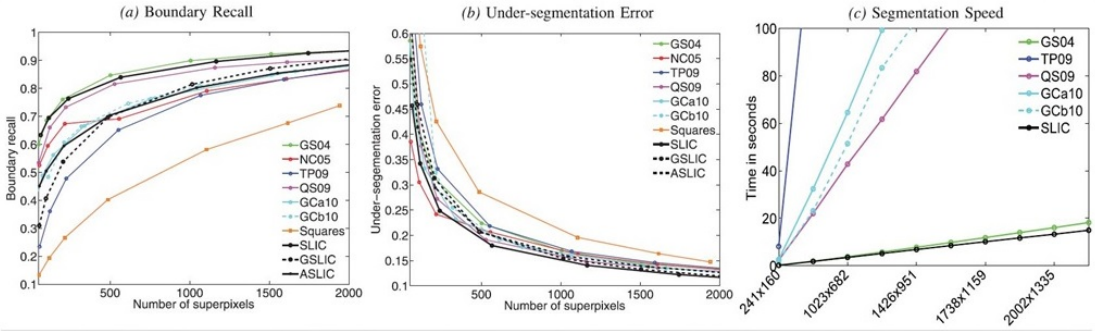


Figure 4: Quantitative evaluation measurements from Achanta et al. [2]: The SLIC (Simple Linear Iterative Clustering) algorithm outperforms most of the other State-of-the-Art approaches in boundary recall, under-segmentation error and speed.

Algorithms for generating superpixels can be broadly categorized as either **graph-based** or **gradient-ascent-based** methods. In graph-based methods each pixel is treated as a node and the similarity between two neighbors define the edge weights. Well known algorithms which were used in the past are: Normalized Cuts Algorithm by Shi and Malik [10], GS04 by Felzenszwalb and Huttenlocher [5], etc. Gradient-ascent-based algorithms start with a rough clustering of pixels and iteratively refine the clusters until some convergence criterion is met to form superpixels. Some examples are Mean Shift by Comaniciu and Meer [4], Quick Shift by Vedaldi and Soatto [12], Watershed Approach by Vincent and Soille [13] (see explanation in section 2.2), SLIC (Simple Linear Iterative Clustering) by Achanta et al. [2], etc. In the following chapter SLIC is explained further since in the current literature most of the methodology found uses this superpixel segmentation algorithm.

3.1 SLIC (Simple Linear Iterative Clustering)

Among all the superpixel algorithms, the simple linear iterative clustering (SLIC) method is widely adopted due to its practicality and performance (see evaluation results in Fig. 4). SLIC is an adaptation of the k-means algorithm, generating compact and regular-sized superpixels by clustering pixels located close to each other based on their color similarity and spatial information. For this it uses a five-dimensional space, namely *labxy*, where *lab* represents pixel color values in the CIELAB color space which is considered both device independent and suitable for color distance calculations, and *xy* represents the coordinates for pixel position. The reason for SLIC having a linear complexity compared with the original k-means algorithm is limiting the size of search region to a constant distance measure, instead of comparing each pixel with every cluster center. Results of SLIC can be observed in Fig. 5, where images were segmented into superpixels of size 64, 256 and 1024 pixels approximately.

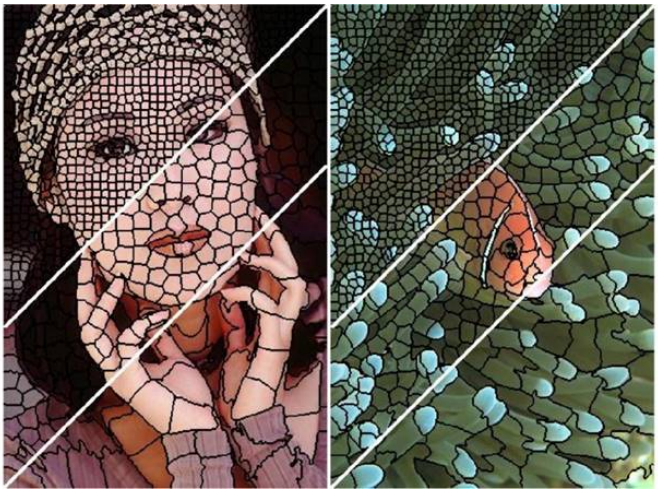


Figure 5: Achanta et al. [2] results of SLIC algorithm with different resolution size of superpixels (64, 256, 1024).

4 Image Understanding Examples with Superpixel Segmentation

When used for segmentation purposes, superpixels increase the speed and improve the quality of the results. Therefore many application and approaches in the literature make use of superpixel segmentation algorithms as a pre-processing step.

4.1 Example: Medicine

In the paper of A. Lucchi and Fua [1], the superpixel segmentation algorithm SLIC is used to achieve a fully automated approach to segment irregular shaped cellular structures like mitochondria from electron microscope data (see Fig. 6). This can be challenging since the texture of mitochondria is easily mistaken with other vesicles or endoplastic reticula and the shape can differ. Therefore A. Lucchi and Fua [1] propose an approach using sophisticated cues that capture the global shape (shape cues that do not require an explicit shape model), texture information, and boundary cues.

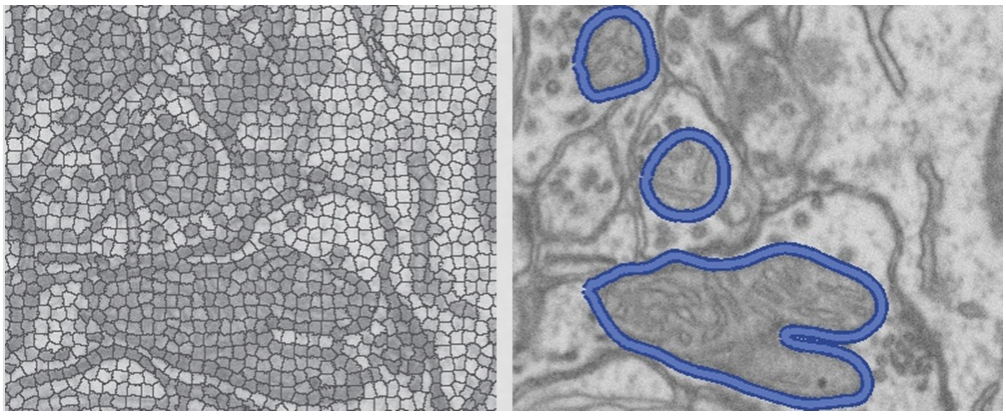


Figure 6: A. Lucchi and Fua [1]

4.2 Example: Meteorology

Liu et al. [9] propose an automatic cloud detection approach for all-sky images using superpixel segmentation. This is an important issue in the meteorology and current approaches where not sufficiently detecting cloud particles and air molecules correct. Most of State-of-the-Art algorithms treat color as the primary characteristic for distinguishing cloud and clear sky. Liu et al. [9] first applies a superpixel segmentation, where every superpixel either contains only cloud or sky segments. For every segment a local threshold is computed with which a threshold matrix is build by interpolation. This matrix is applied on the interpolated red and blue channel of the original image to gain the result (since these channels contain more information regarding cloud and sky dissimilarities). See for a visual description Fig. 7.

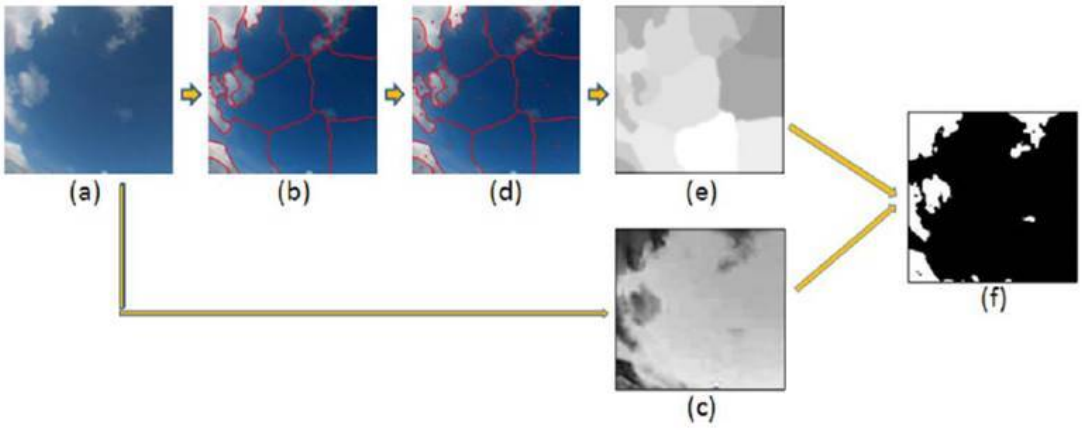


Figure 7: Proposed cloud detection algorithm Liu et al. [9]: (a) Original Image. (b) Superpixel. (c) R-B channel. (d) Location for threshold of all superpixels. The red dots are the location of the local threshold. (e) Threshold matrix. (f) Detection result.

4.3 Example: Human Visual System

Aksac et al. [3] method is used to detect salient region based on superpixel segmentation to imitate the human visual system (HSV). The HVS is capable of easily detecting and separating the important parts of a given image from the remainder. This topic is broadly investigated by researches not only in the area of salient object detection, but also affecting areas like image retrieval, image retargeting – seam carving, image/video compression, image resizing, collage, etc.
In this approach the image is first decomposed into superpixels with SLIC, grouping similar pixels and generating compact regions. Based on the segments similarities between regions salient maps are calculated by benefiting from color, location, histogram, intensity, and area information of each region as well as community identification via complex networks theory in the over-segmented image.
To achieve such good results as can be observed in Fig. 8 without any preinformation on the objects in the image, some assumptions are defined:

A salient object has a high contrast feature compared to its surrounding background. Distribution of colors in the image is rare.

A salient object takes great advantage of closer regions rather than farther ones for its contrast value.

A salient object is commonly located near the image center (human attention firstly focuses on the center area of the image)

A salient object appears in a smaller area compared to background objects

A salient object exhibits a uniform color distribution (overall parts of a salient object are homogeneously highlighted)

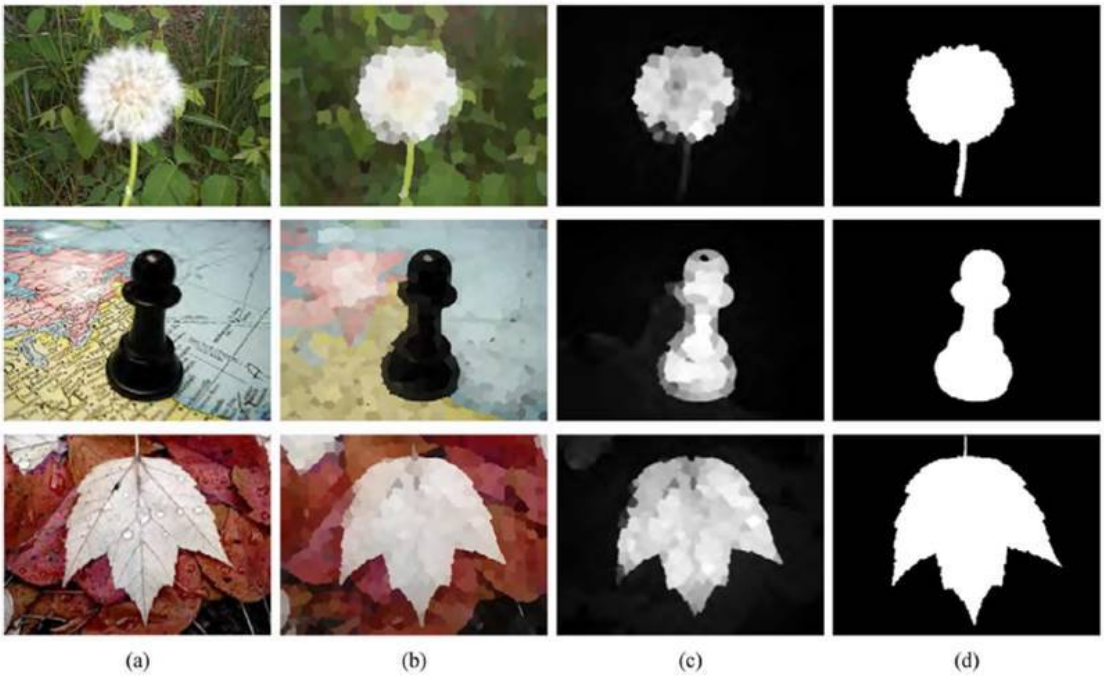


Figure 8: Complex networks driven salient region detection algorithm Aksac et al. [3]: General overview (a) original image (b) superpixel segmentation (c) saliency map (d) ground truth

4.4 Example: Bag of Features

In the paper of Fulkerson et al. [6], they propose a method to identify and localize objects using superpixel segmentation. The superpixels allow Fulkerson et al. [6] to measure feature statistics on a natural adaptive domain rather than a fixed window. They construct a bag-of-features classifier which operates on the regions defined by the superpixels. To achieve better results superpixel neighborhoods are taken into consideration, to provide spatial consistency in the classification as can be observed in Fig. 9. On the one hand, increasing N means adding the histograms of adjacent superpixels, which have more features in common and increase the spatial extent of the region. On the other hand, increasing N also means blurring the boundaries and having more false positives selected as part of the object.

4.5 Example: Multi-class segmentation

The previous idea of 4.4 is manifested in the proposed approach of Gould et al. [7], where they formulate a layered model for object detection and multi-class segmentation. In

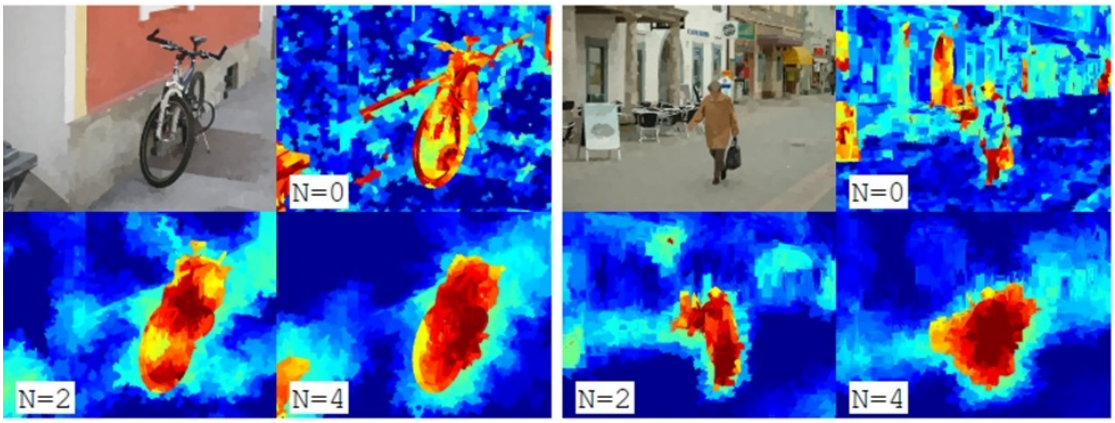


Figure 9: Fulkerson et al. [6]:

this approach, they use the output of a bank of object detectors in order to define shape priors stated as a "soft" masks and estimate appearance, depth-ordering and labeling of pixels in the image (see Fig. 10). Difficulties are observed since the probabilistic mask of an object has to provide different poses of the object (side versus frontal, etc.) and occlusion make finding the object in the image much harder.

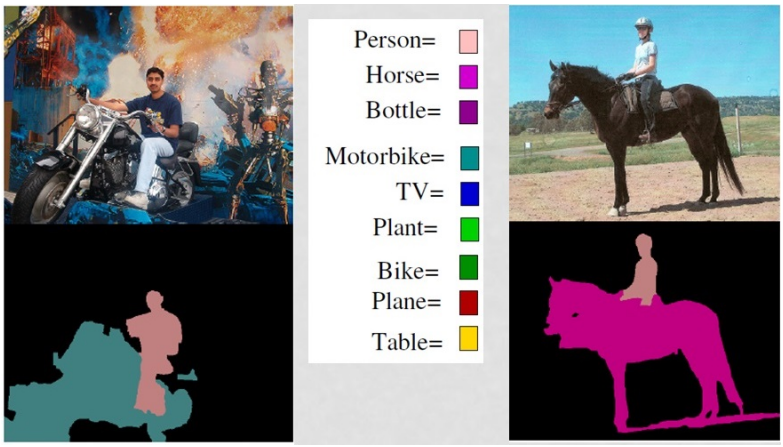


Figure 10: Gould et al. [7]:

4.6 Example: Stereo Matching

Zitnick and Kang [14] suggests to solve the stereo matching problem using superpixels instead of the well known epipolar geometry approach. They compute match values over entire segments rather than single pixels. This way the extracted depth value for one segment is set for all its pixels, which does not need to be correct, but as explained in the paper, as long as the recovered views appear correct, they are visually plausible. Computing match values over entire segments provides robustness to noise and intensity

bias and Zitnick and Kang [14] experiments showed that it works well with occlusions caused by the stereo (see Fig. 11). The stereo matching results are compared with the State-of-the-Art moderate, but the authors show another area where the proposed approach performs well: Image-based Rendering. To achieve photo-realistic rendering of novel views (virtual cameras), the extracted depth map need not be correct, but plausible and artifact-free.

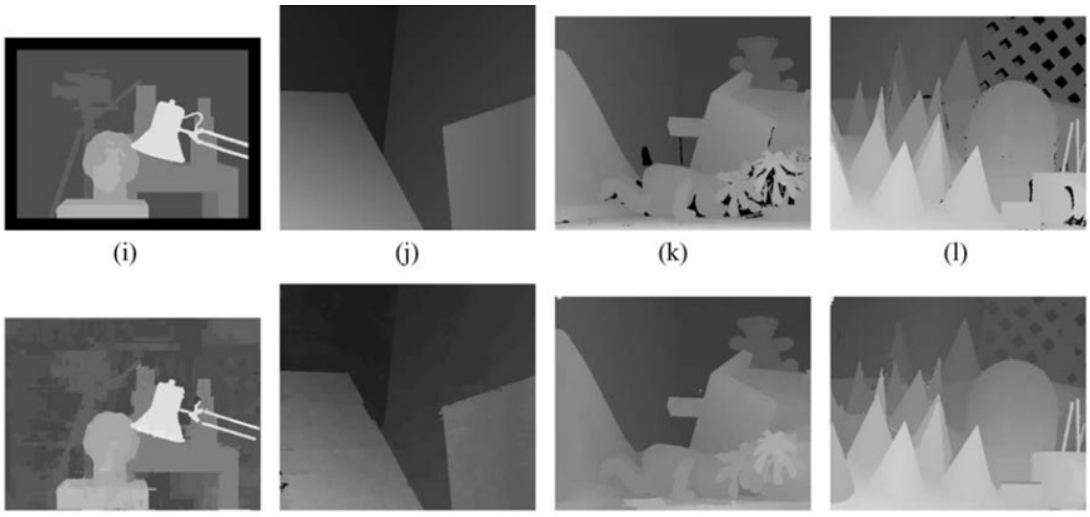


Figure 11: Zitnick and Kang [14]: Computing match values over entire segments rather than single pixels

References

[1] R. Achanta V. Lepetit A. Lucchi, K. Smith and P. Fua. A fully automated approach to segmentation of irregularly shaped cellular structures in em images. *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention*, 30(11):474–486, 2010. ISSN 0162-8828.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, November 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.120.

[3] Alper Aksac, Tansel Ozyer, and Reda Alhajj. Complex networks driven salient region detection based on superpixel segmentation. *Pattern Recognition*, 66:268–279, June 2017. ISSN 0031-3203. doi: 10.1016/j.patcog.2017.01.010. URL `http://www.sciencedirect.com/science/article/pii/S0031320317300110`.

[4] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002. ISSN 0162-8828. doi: 10.1109/34.1000236. URL `http://dx.doi.org/10.1109/34.1000236`.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, September 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000022288.19776.77. URL `http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77`.

[6] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th International Conference on Computer Vision*, pages 670–677, September 2009. doi: 10.1109/ICCV.2009.5459175.

[7] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-Class Segmentation with Relative Location Prior. *International Journal of Computer Vision*, 80(3):300–316, December 2008. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-008-0140-x. URL `https://link.springer.com/article/10.1007/s11263-008-0140-x`.

[8] B. Jähne. *Digital Image Processing*. Number Bd. 1 in Digital Image Processing. Springer Berlin Heidelberg, 2005. ISBN 9783540240358. URL `https://books.google.at/books?id=qUeecNvfn0oC`.

[9] 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(2):354–358, February 2015. ISSN 1545-598X. doi: 10.1109/LGRS.2014.2341291.

[10] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000. ISSN 0162-8828. doi: 10.1109/34.868688. URL `http://dx.doi.org/10.1109/34.868688`.

[11] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. ISBN 978-1-84882-934-3.

[12] Andrea Vedaldi and Stefano Soatto. *Quick Shift and Kernel Methods for Mode Seeking*, pages 705–718. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-88693-8. doi: 10.1007/978-3-540-88693-8_52. URL `http://dx.doi.org/10.1007/978-3-540-88693-8_52`.

[13] Luc Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13 (6):583–598, June 1991. ISSN 0162-8828. doi: 10.1109/34.87344. URL `http://dx.doi.org/10.1109/34.87344`.

[14] C. Lawrence Zitnick and Sing Bing Kang. Stereo for Image-Based Rendering using Image Over-Segmentation. *International Journal of Computer Vision*, 75(1):49–65, October 2007. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-006-0018-8. URL `https://link.springer.com/article/10.1007/s11263-006-0018-8`.

Report
Superpixel- Segmentation and Grouping
Image Understanding SS17

Alexandra Mai
April 7, 2017

General remarks of the presentation by the opponent:

Positive:
- good talk
- huge variety of paper presented
- slides of Application with visually representative information (pictures)

Hints for improvement:
- not only segmentation should be described, but also saliency
- the object and goals of segmentation could be explained more explicitly:
    - Goals: Object recognition, measurements, image synthesis
- an example for grouping in respect to oversegmentation would have been nice

After the general remarks the opponent together with the audience thought about questions which will be categorized in the following 4 areas:
Machine learning, the human vision system, superpixel shapes and the comparison between SLIC and graphbased methods.

Machine Learning

Nowadays in different fields of computer science the interest in machine learning algorithms has risen. Therefore the opponent asked the question whether the audience think it is also important for (superpixel) segmentation. After a short discussion, it was agreed that on the one hand superpixels can improve the performance of e.g. a SVM, however the overall training process is very costly and can be unpractical.

As an example of the difficulty of machine learning for segmentation a CNN was mentioned for segmenting the surrounding of an (partly) automatically driven car. The segmentation results strongly depend on the training sets and furthermore while driving wrong segmentation/classifications can have horrible consequences. After the learning process, it is not possible to have a look inside the segmentation and determination process to see how, and based on what parameters, decisions are made (which is similar to the neuronal network of the human brain).

In comparison to CNN, ‘normal’ pyramide linking was mentioned as a method which is always traceable to its primary level by e.g. adding weights together.

Human Vision System

Another stimulus for the discussion was the human eye, which was mentioned in the presentation as the reason why objects in the center are more salient than other ones. The human eye has a high resolution in the center (fovea) which gets blurry outwards, however for the eye itself the “important things” do not have to be in the center (cutoff scenes are normal). Therefore it was agreed that it would be better to state: Humans *like* to have important things centered and complete in an image, to focus better on them.

Superpixel Shapes

The sizes of the different superpixels depend on the method used, however the importance of the same sizes/shapes among each other was an important question for the audience. In general if all superpixels have the same size and shape it is (visually) easier to compare them. A shortcoming of superpixels are thin and long structures as they are either lost or, if there are several thin structures, they could be summed up as one mass.

SLIC vs Graphbased Methods

The last big topic discussed in the lecture was the differences between SLIC and graphbased Methods. One major difference is the complexity between both approaches as SLIC has a linear (O(n)) and graphbased methods a logarithmic complexity (O(n*log n)). While the discussion the importance of direct access to pixels arose, as for a segmentation of e.g. animated images the assigning of superpixels among each other is important. Graphbased methods treat the superpixels as nodes which makes it difficult to directly access pixels leading to longer computation times. In comparison SLIC has a very similar structure to raster where geometrical transformations are easy to compute (example of moving clouds was given).

Hints:

While the discussion two major tips for further presentations or future scientific work were mentioned by Mr. Kropatsch and Mr. Chen.

1) Technical reports should not be used for citations, as most of them are not proofread.
2) Statistics should be scrutinized critically, especially when there is only few data in it which is presented as the ground truth (e.g. presented table from opponent: only a few databases tested with very low resolutions - hard for general comparisons) .

Report
Vision Models

Rebeka Koszticsák
1325492

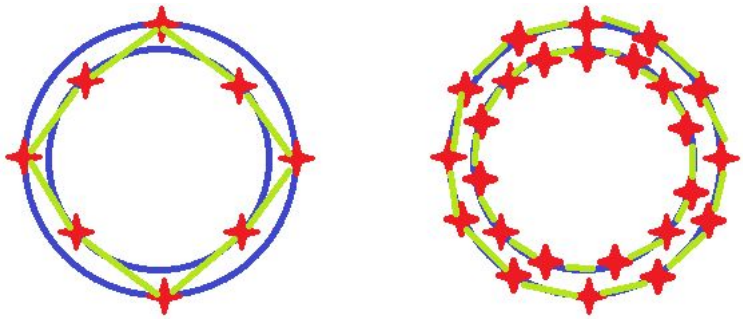
Presentation
The goal of vision models is to understand the actual world represented by the observations. A sequence of observations, measurements etc. defines a vision model, which describes (a state of) the real world. The attributes of vision models make it easier to process objects (algorithms which describe the vision model itself) and makes them more robust against changes (for example rotation) and measurement inaccuracies (probability distribution, where the actual object can/should be).

There are two different types of vision models. Discriminative models describe the possible world based on the observations, while generative models describe possible observations based on the real world. With the help of the Bayesian Method it is possible to calculate one from the other.

There are several model classes, which can be grouped by their purpose or their working methods. Regression models are based on non-linear transformations (for example kernelization) which are able to reduce the amount of the parameters. Graphical models represent the world states as directed (Bayesian) or as non-directed (Markov) graph structures, where the edges symbolise conditional dependencies between world states or events. Temporal models are built for the prediction of possible future events based on the tracking of the object parameters.They are based on the Hidden-Markov-Model (temporal evolution of the world) and on the Bayes Theorem (the change of the parameter depends on the current state) due to their reliance on probability and non-independent event sequences. Examples for temporal models are the (Unscented) Kálmán Filter and Particle Filters. Shape models segment objects from their background. There are several models to represent visual words, one of them is the Bag-of-Words method. It collects similarities between two objects by utilizing techniques like SIFT, but ignores the spatial information.

Discussion
On a markov graph it is possible to define markov blankets for each node, which group the current node, its parents, its children and the parents of its children together. This grouping influences the transmission probabilities between the nodes, depending on the type of the graph, directed or non-directed, the amount of the grouped nodes in the blanket can change

The Bag-of-Words, and other pooling models could be improved, so that they consider the spatial information. The model should not only save feature points but also their coordinates and the size of the reference image. After all in some cases it would be equivalent with graph structures. The feature points depend on the resolution, the method and the quality (for example the sampling frequency) of the measurement. The nearest neighbor algorithm is not accurate enough, therefore a more accurate algorithm is needed to define the connections between the feature points.



Another improvement could be scale invariance, with the help of opponent SIFT. This algorithm extracts gradients of histograms and is robust against some deformations but not against distortions.
The model should also be viewport invariant.

The objects are influenced by their surrounding, so the visual context should be considered by building a visual model, that would better simulate the human vision system, which is strongly influenced by the context. The regional CNN takes a predefined ROI as input, and it enlarges the input area for the next loop with the output of the previous iteration.

CNNs provide better features, but are not perfect, and supervision for the best solution. For example the new iteration sometimes override the previous one, and classifies the original objects as something else. In many cases a trained CNN is used, which can provide good results, even while nobody knows how or why it is working. Therefore it is hard to improve. In many cases strict and not necessarily rational restrictions are used, in order to get proper solutions. (For example the family membership experiment: a CNN will tell which persons are related. But it is required that the face of the mother and father are similar, and at most one grand parent pair is present) On the other hand the first input data is defined by a human. The algorithm is not able to find important regions, the startpoint is predefined. All in all CNNs are not able to provide the perfect solution for a complex problem. It can be used as a starting point to get a proper basis which can then be built upon.

Shapes

Image Understanding SS 2017

Wen Chao Chen (1129468)

June 6, 2017

1 Introduction

The general course of actions to re-identify a certain object for a human is to first take a look at the respective object, use characterizing properties to describe and memorize it. Finally, to use that description in order to recognize or reconstruct the object. One of the characterizing properties is called *shape*. The shape describes how a specific object looks like based on the external boundary. From a computer's point of view, given the image of a triangle, the shape is a description of the triangle's region bounded by the outline of three vertices and three edges. As for human vision, it can only be assumed how our brain decomposes a seen object and extracts a description of the shape. Formally, a shape is defined by the $n-1$ dimensional boundary of an n dimensional object. Below are three examples:

- Cube: surface of the cube
- Piece of paper: rectangle surrounding paper
- Line: endpoints of the line

A crucial question in understanding human vision is how the brain handles visual input, whether it is a top down or a bottom up process. Bottom up processing implies that perception starts at the retina and is carried all the way to the visual cortex. The eyes have to see something in order for the neurons in the brain to be activated. Thus, object properties such as size or shape are taken into consideration for visual perception. On the other hand, top down processing is exactly the other way round. Top down implies that perception is based on prior knowledge and past experience. In this case visual perception is primarily based on contextual information and not what the eyes see. The things that are seen serve a interpretative role given the pre-assumed context.

For top down processing, the most often referenced example is likely to be the Selective Attention Test introduced by Simons and Chabris [1]. In the first run, the audience is asked to count a basketball is passed during a short video clip. So the prior knowledge

is to focus on the basketball based on its location in each frame and its shape. However, then the question arises whether the observer has seen a gorilla and generally, most people put all their attention on the ball, such that the context skipped out of vision and focus. It is only in the second run, when the clip is replayed that people were aware of the presence of a gorilla, because they have already seen the video once and in addition to that, have gained more prior knowledge. An example for bottom up processing would be the sight of a mosquito, which from the decision making of the visual cortex is perceived as a nuisance, or a threat.

According to Kveraga et al. [2] however, such an example also shows that visual perception involves a certain amount of both processing types. Take for example a situation were an adventurous person walks in the jungle and discovers a paw print. Up to that point, it is bottom up processing. As soon as the paw print has been discovered, Kveraga et al. believes that the processing switches to top down, as the seen paw print triggers associations with contextual information provided by prior knowledge. Hence, the question is not which type of processing is used, but how the two types are proportioned.

Another question regarding the human vision is whether the human eyes perceive the real world as two or three dimensional. Since the human retina lies flat, it is assumed that humans see two dimensional. However, objects in the real world are typically three dimensional and humans are still able to distinguish them, even if the orientation of the objects change. The ability to perceive and distinguish an object from different angles is called *shape constancy*. If humans only see two dimensional, how can it be possible for them to distinguish between three dimensional objects? How is it possible for humans to have depth perception, meanwhile being bad at estimating distances? The three dimensional information must be somehow transmitted through visual perception. In this regard, it should be mentioned that the concepts of stereo vision does not apply to humans. The theory is to find corresponding points between two pictures that were taken from slightly different angles, calculating the disparity and finally merging those two pictures into one. The main reason, why this concept does not apply to humans, is because the disparity is not defined absolutely in the brain.

Therefore, the main motivation of using shapes is to have computers imitate our human visual perception and image understanding. The objective is to generate a model that describes the shape of an object in a way such that the computer is able to recognize it based on the shape description. The focus of this paper is on 2D images. The following sections give an overview of the two main types of shape models and their respective descriptors. Section two presents selected shape models and descriptors, section three emphasizes on the invariance and robustness problem of descriptors, while section four concludes this paper.

2 Shape Models

Shape models are mainly categorized into two types, the region-based models and the contour-based models. The region-based models focus on the surface, the internal area

of an object. Contour-based models on the other hand are based on the boundaries of the object.

2.1 Region-based Models And Descriptors

In general, region-based models are related to *region segmentation*. So the first step is always *region identification* using methods known from fields such as *connected component labelling* and *colouring*. Methods and respective implementation algorithms introduced by Sonka et al. [3] are *Neighbourhood region identification with 4 or 8-connectivity*, *region identification in run-length encoded data* and *Quadtree region identification* (see figure 1). Once the regions are determined, one possible approach is to use mathematical and heuristical descriptors such as *region area*, *Euler's number*, *height and width*, *eccentricity*, *elongatedness*, *rectangularity*, *direction* or *compactness*. A problem with this approach is that objects can not be reconstructed, because the actual form of the object is not described. Two alternative approaches, *region skeletons* and *region decomposition* address this issue and are solutions based on graph representations. To add further to that, in case of region decomposition, the subregion can be represented as neighborhood graph. A node would denote a subregion and subregions would be connected by edges. Then, for each subregion aforementioned mathematical, heuristic descriptors could be computed for even more detailed description. This way, the actual shape of the object and the attributes of the shape can be preserved.

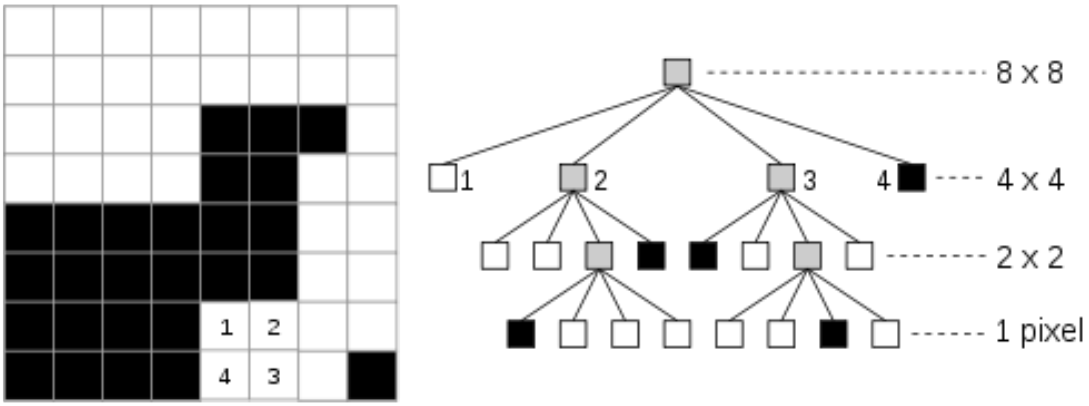


Figure 1: A Quadtree region representation [4].

2.2 Contour-based Models And Descriptors

Contour-based models describe the borders of an object shape or a region with geometric properties. One simple approach to describe the border of an object is to use a *Chain code*, e.g. with 4 or 8-connectivity, based on *Freeman's code*. The issue with this simple representation is that it is not scale invariant, because we only know direction changes of

the boundary, but not the length. Movement in one direction is considered to be of unit length, or $\sqrt{2}$ for diagonal movement in the 8-connectivity case. Neither is it rotation invariant, because the direction changes are only valid if the projection of the object shape remains unchanged. For the border representation Sonka et al. [3] emphasizes on the fact that the scale invariance problem is a general problem when using geometric properties. However, the robustness of some descriptors can be increased through modification. Hereby, the geometric descriptors listed by Sonka et al. are *boundary length*, *curvature*, *bending energy*, *signature* and *chord distribution*. The curvature, in fact, can be used to determine boundary vertices using the *tolerance interval approach* with *recursive boundary splitting*. An alternative possibility is to define constant curvature and use those to describe the boundary as chain code, as illustrated in figure 2. The benefit of using curvatures is also that it is not affected by projective transforms. As far as the boundary length is concerned, the issue is that the length is not dependent on the actual length of the boundary, but on the number of direction changes in the code chain. The bending energy is also not a good representation, since computation is the result of sum of squares. This makes reconstruction of an object shape impossible. From the list of geometric descriptors, only the signature and the chord distribution descriptors are considered robust. Sonka et al. also mentions neural networks as an additional approach for robust shape recognition approach. The neural network approach may achieve good result, but it first of all trained to recognize very specific shapes. Secondly, since nobody is able to answer the question what exactly is going on during the whole recognition procedure, it is hard to say how accurate the recognition is.

3 Invariance Problems and Shape Invariants

When generating a model, one aspect to consider is the ability to reconstruct the shape of an object based on the description. Ideally, a description is not only robust to translation, rotation, and scale transformations, but is also working for different resolutions. In that case, one single shape descriptor could describe every single object.

One big issue when working with digital 2D images is the resolution, because high resolutions may incorporate noise, while small, but important details are lost in low resolutions. Thus, the shape of an object might look very different comparing images of various resolutions. This is bad, especially when small details are essential. Let take a pin tumbler lock key as an example, which is designed to fit one particular lock. It therefore has an unique combination of a tip, some cuts and a shoulder. However, if the resolution is low and those details are distorted or lost, the description would not be suitable to reconstruct or recognize the shape of the key.

Another problem is projection. Even for humans, it can be quite difficult to recognize an object, if seen from an unusual perspective. Take for instance a horse. Looking at it and associating it with the shape of a horse by human eye is much easier when the horse is seen from the side rather than from the front, because from prior knowledge, most of us have the side view of a horse engraved into our minds. The same applies to computers. They get images as input, that show an object from a specific angle and

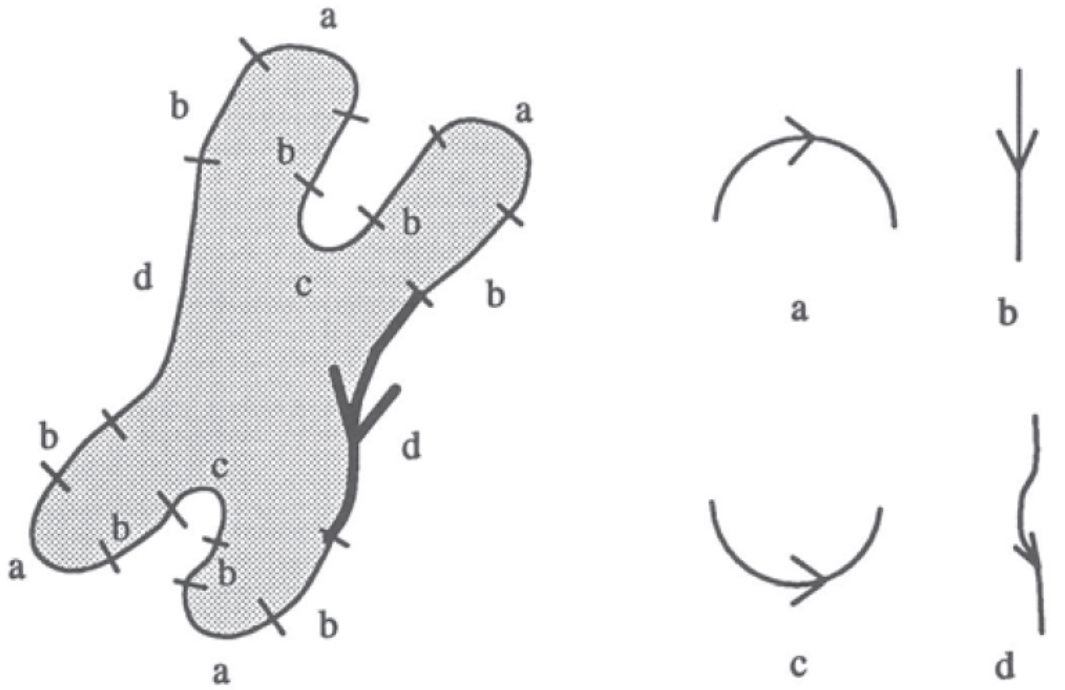


Figure 2: Chain of boundary segments by Sonka et al. [3].

position. From this perspective, a description of the shape is created. However, since the computer only has knowledge about this one specific projection, if the image input is the same object seen from another perspective, the description of the shape would most likely no longer apply. Therefore, a description invariant to translation, rotation and scale would be ideal.

Sonka et al. introduces three shape invariants, namely *Cross ratio*, *Systems of lines or points*, and *Plane conics* and also points out to secondary literature.

4 Conclusion

Describing the shape of an object in a way such that computers are able to recognize or reconstruct the object like humans are able to is by no means an easy task. The typical way of approaching shape models is to use a description suited for a particular problem. This kind of problem-oriented approach is somehow also the way how humans learn to recognize object. The difference is though, that humans and computers see perceive things differently. Our imagination and image understanding is much more complex than that of the computer.

That is also the reason why there are so many shape representations, and research is continuously ongoing, with invariance as a big focus point [5, 6, 7]. Especially in respect to 3D, the development of generating shape representation has been steadily ongoing

[8, 9, 10, 11].

References

[1] Daniel Simons and Christopher Chabris. Selective attention test. `https://www.youtube.com/watch?v=vJG698U2Mvo`, 1999. [Online; accessed 31-May-2017].

[2] Kestutis Kveraga, Avniel Singh Ghuman, Karim S. Kassam, Elissa A. Aminoff, Matti S. Hämäläinen, Maximilien Chaumon, and Moshe Bar. Early onset of neural synchronization in the contextual associations network. *Proceedings of the National Academy of Sciences*, 108(8):3389–3394, 2011.

[3] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Shape representation and description*, pages 192–254. Springer US, Boston, MA, 1993.

[4] Wikimedia Commons. Quad tree bitmap, 2008. [Online; accessed 06-June-2017].

[5] C. Y. Tsai, H. C. Liao, and Y. C. Feng. A novel translation, rotation, and scale-invariant shape description method for real-time speed-limit sign recognition. In *Proceedings of the 2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, pages 486–488, Nov 2016.

[6] A. L. Codizar and G. Solano. Plant leaf recognition by venation and shape using artificial neural networks. In *Proceedings of the 2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, pages 1–4, July 2016.

[7] M. Meng, H. Drira, M. Daoudi, and J. Boonaert. Human object interaction recognition using rate-invariant shape analysis of inter joint distances trajectories. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 999–1004, June 2016.

[8] Zygmunt Pizlo. *3D Shape: Its Unique Place in Visual Perception*. MIT Press, 2010.

[9] Chunyuan Li and A. Ben Hamza. A multiresolution descriptor for deformable 3d shape retrieval. *The Visual Computer*, 29(6):513–524, 2013.

[10] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[11] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Report - Shapes

Image Understanding SS 2017

Daniel Pucher (1227136)

May 10, 2017

1 What is the relation between human visual perception and image understanding?

Humans ...
... see images as a portrayal of the reality, with the difference of depth perception.
... especially recognise edges.
... have prior knowledge of objects.
... are bad when it comes to estimating distances.

Stereo vision (finding correspondences, calculating disparity, ...) is a nice mathematical theory, but not really applicable to humans. What if no correspondences can be found? Vision still works for humans, which suggests that the disparity is not defined absolutely in the brain.

Learning process as in image understanding is also present in children.

Do humans see 2D or 3D? The retina is flat which suggests only 2D. But 2D information is not really needed, since most objects in the real world are 3D. But is the 3D information not only a distance information?

Shape Constancy – 3D shape usually stays the same and can be easily distinguished by humans.

Experiment with synthetic satellite images: The lightning direction was changed from south to north which also made the shadows to change and made the recognition task much more difficult for humans.

Similarities to neural networks? Humans don't learn shape descriptors like neural networks do.

2 What is a shape?

It is the $n-1$ dimensional boundary of an n dimensional object. Examples:

The shape of a cube is the surface of the cube.
The shape of a piece of paper is the surrounding rectangle.
The shape of a line? Both endpoints.

3 Do we see bottom up or top down?

Bottom up: Neurons in the brain are activated after we see with our eyes.
Top down: Kontextual information is used before we see with our eyes.

Examples for top down:
"Moonwalking Bear" (https://www.youtube.com/watch?v=Ahg6qcgoay4)
"Dalmatian Dog" (See Figure 1)

Figure 1: Dalmatian optical illusion [1].

These and other examples suggest, that there is a certain "top down" amount involved. The question that remains is, how much? Kveraga et. al. [2] raised the question whether contextual information is activated early enough to facilitate the perception of individual objects, which also suggests a top down approach. They conclude that there experiments indicate *"that contextual information is activated early during object recognition rather than solely as a late postperceptual process. Such rapid activation of contextual associations and the ensuing facilitation of recognition make sense if one thinks of the evolutionary pressures faced by most organisms. For example, seeing a paw print or scat of a predator and rapidly activating the context associated with this image could afford the prey animal enough time for an escape, conferring it a significant evolutionary advantage over time."*

4 Applications beyond object recognition?

Human interaction recognition: The problem here is primarily deformation.

References

[1] R.L. Gregory. *The Intelligent Eye*. World university library. Weidenfeld & Nicolson, 1970.

[2] Kestutis Kveraga, Avniel Singh Ghuman, Karim S Kassam, Elissa A Aminoff, Matti S Hämäläinen, Maximilien Chaumon, and Moshe Bar. Early onset of neural synchronization in the contextual associations network. *Proceedings of the National Academy of Sciences*, 108(8):3389–3394, 2011.

To see or not to see - Optical Illusions

Image Understanding SS 2017

Alexandra Mai (1125691)

June 2, 2017

Optical Illusion can affect all aspects of the visual process and can be described as perceived images which are different from reality. The illusions are caused by the limitations of the visual apparatus and its processing. Therefore optical illusions are used to explore the mechanisms of perception and the linked neuronal activities of the brain and the eyes. Until know many illusions can't be fully explained and are still subject of further research.

In the following five chapters the illusions are categorized, similar to Bach et al [10], into luminance and contrast, angle and size, movement, color and cognitive and impossible shape illusions.

1 Luminance and Contrast Illusion

Illusions caused by different types of light sources and brightness scales are often explained by lateral inhibition in combination with cortical processes [11]. In 1870 the Herman grid was first investigated by Ludimar Hermann. The optical illusion can be seen in figure 1. When looking at the crossing of two white edges they seem normal, however the rest of the intersections seem to have grey dots. A possible way to interrupt the illusion was found by Geier et al[11], by slightly manipulating the shape of the black squares (see Figure 1 right side).

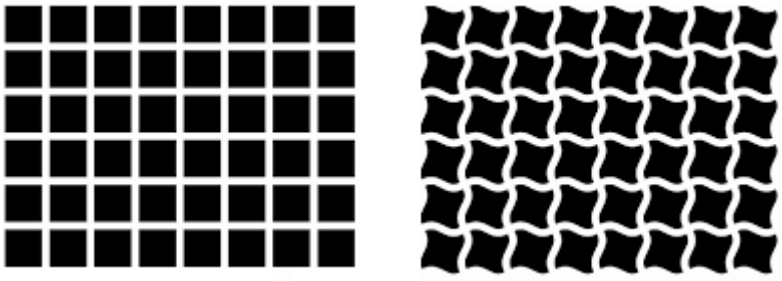


Figure 1: **Left side**: Hermann grid optical illusion
Right side: variation of Herman grid wihtout optical illusion [2].

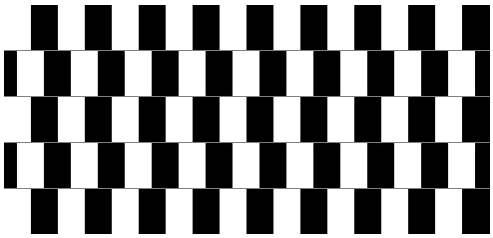


Figure 2: Cafe wall illusion [6]

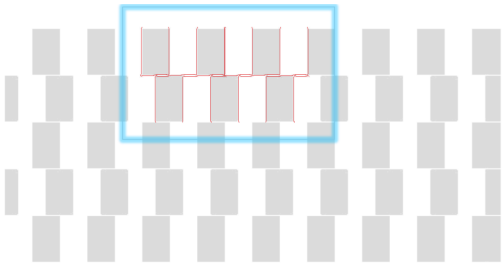


Figure 3: Cafe wall illusion explained with smoothing and edge detection

Another well known contrast illusion is the cafe wall illusion (see Figure 2). For our brain the lines seem to be asked when looking just straight on the picture. In 2004 Fermüller et. al [6] this illusion was explained with the help of smoothing and edge detection. In figure 3 it is visible that the edges are bulged on the horizontal line (grey line near the black tiles are more similar than to the white tiles), which causes the optical illusion of aslope lines.

2 Angle and Size Illusion

The size and angle illusions are often caused by the addition of the assumed distances by the human visual system [10]. Two examples of those two kind of illusions can be seen in figure 4 and 5. The Zöllner illusion describes the effect of intersections between long and short lines at certain angles and their appearance of divergence. A general explanation for this effect is the assumption of the visual system that small angles are near the obliques [6].

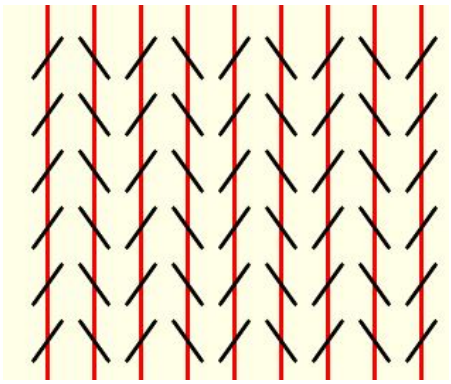


Figure 4: Zöllners illusion [9]

The desk size illusion tricks our trained estimation of object sizes as it heavily depends on the assumed distances. The two shown tables would not have the same size in real world, as the left one would be much longer with a smaller width in comparison to the

right one. However, when only the size of the two table tops are compared both have the same parallelogram area (see Figure 5 right side).

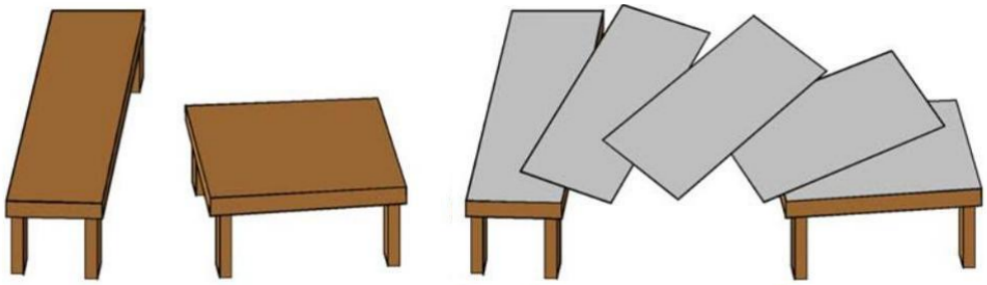


Figure 5: Optical illusion of desk surface size [6]

3 Movement Illusion

Movement Illusion can be seen in two different fields, on the one hand by motion pictures and moving objects and on the other hand by single pictures. Both of them are caused by under sampling when looking at them. A wide spread explanation are that the visual system takes discrete "'snapshots"' of the world. Kline et al [12] argued however that many movement illusions wont appear when a mirror is used and both sides are watched.

The rotating snake illusion is a caused by the peripheral drifts of the human eye. Intensified by the color constellation the circles seem to move in different directions while not looking directly at them.

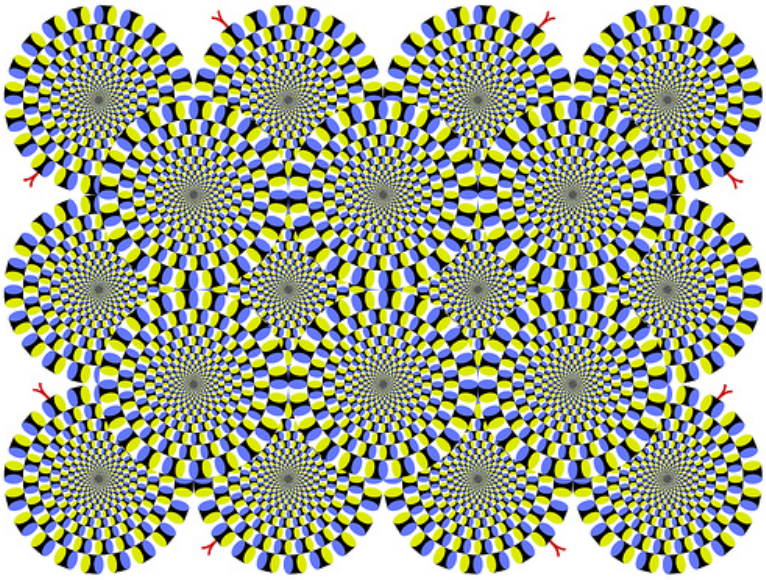


Figure 6: Rotating snake illusion [7]

4 Color Illusion

Color is an electromagnetic phenomenon, depending on light reflections and absorption. Therefore it can be described as an illusion itself which is very subjective for each individual [1]. The visual system inhibits the neighbours of a colour and becomes to the same color less and to a different color more sensitive. An example of a color illusion can be seen in figure 7. Although the light circles appear to have two different colors (white and yellow) they are certainly the same (white).

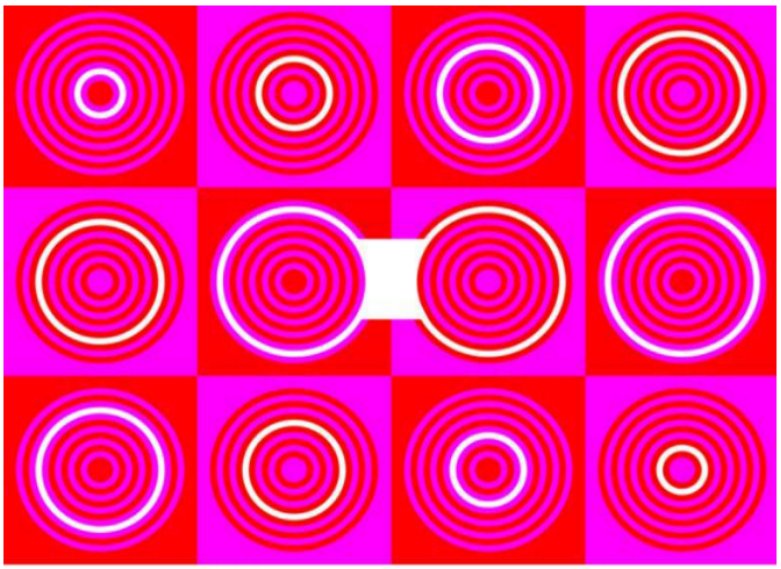


Figure 7: Color illusion made by Kitaoka [4]

5 Cognitive and Impossible Shape Illusion

Cognitive illusions are caused by the perceived reality (memories on how things should look like) [13]. Most of the cognitive illusions strongly depend on the setting where they appear (also if there is a specific question asked), on the mood of the observer and on the (recent) past. An example of a cognitive illusion can be seen in figure 8. Depending on the surrounding and for example recent incidents with either young or old people, the observer first sees an old or an young women. Another example demonstrating the focus depending, cognitive illusion is the "'Monkey Business Illusion"' [5]. In this video two teams are passing a basketball within their own team. The task is to count the passes of the white basketball team. While the process a monkey walks through the scene, a player of the black team goes away and the curtains change their color. It shows very illustratively that although the human eyes can see everything the brain only process certain stimuli.

Figure 8: Cognitive shape illusion: old and young women [8]

Impossible shapes may seem at the first look as normal figures or objects, however when taking a closer look they can't be real [10]. The lines are connected, however the shape itself can not exist. In figure 9 two of those impossible shapes are visible.

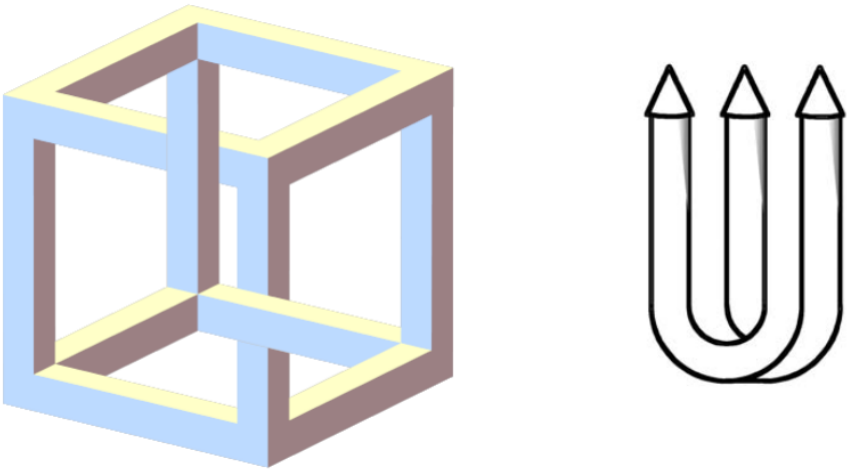


Figure 9: Impossible shape illusion: impossible cube [3]

References

[1] Color of the mind. http://www.archimedes-lab.org/color_optical_illusions.html. Visited on 26.05.2017.

[2] Hermann grid. https://www.researchgate.net/figure/5246149_fig1_Figure-2-The-classical-Hermann-grid-above-and-the-sinusoid-grid-below-In-the-case. Visited on 28.05.2017.

[3] Impossible cube. https://en.wikipedia.org/wiki/Impossible_cube. Visited on 26.05.2017.

[4] Kitaoka a. optical illusions. http://www.ritsumei.ac.jp/~akitaoka/index-e.html. Visited on 29.05.2017.

[5] The monkey business llusion. https://www.youtube.com/watch?v=IGQmdoK_ZfY&t=28s. Visited on 30.05.2017.

[6] Optical illusions. http://www.cfar.umd.edu/~fer/optical/index.html. Visited on 27.05.2017.

[7] Optical illusions: the illusion of movement. https://able2know.org/topic/121703-1. Visited on 27.05.2017.

[8] Pictogrephic ambiguity. http://www.grand-illusions.com/opticalillusions/woman/. Visited on 26.05.2017.

[9] Zöllner illusion. http://www.cut-the-knot.org/Curriculum/Geometry/Zollner.shtml. Visited on 25.05.2017.

[10] Michael Bach and CM Poloschek. Optical illusions. *Advances in Clinical Neuroscience and Rehabilitation*, 6(2):20–21, 2006.

[11] János Geier, László Bernáth, Mariann Hudák, and László Séra. Straightness as the main factor of the hermann grid illusion. *Perception*, 37(5):651–665, 2008.

[12] Keith Kline, Alex O Holcombe, and David M Eagleman. Illusory motion reversal is caused by rivalry, not by perceptual snapshots of the visual field. *Vision research*, 44(23):2653–2658, 2004.

[13] Rüdiger F Pohl. Cognitive illusions. *Cognitive Illusions: Intriguing Phenomena in Judgement, Thinking and Memory*, page 3, 2016.

Scene Understanding

Image Understanding SS 2017

Daniel Pucher (1227136)

June 18, 2017

1 Understanding Scene Understanding

Zelinsky et. al. [6] ask the question what it means to "understand" a scene. As an example taken from their work, take a very brief look at the picture in Figure 1. What do you see? Research has shown that the extraction of the gist or substance of a scene is very quickly. Furthermore also some categories of objects, especially people and animals, as well as actions can be detected very quickly. For the scene shown in Figure 1 the classification from a very brief glance and combining all pieces of information might lead to - *a women's track meet.*

But scene understanding does not stop at the gist and in the example scene from Figure 1 also other things were happening. For example, one runner had fallen and the others were trying to avoid tripping over her. And finally, all of the runners had one prosthetic leg. With this additional information the scene tells a story about a special race for women amputees. This example illustrates the fact that scene understanding exists on a continuum. At one end is a very fast and seemingly effortless extraction of the scene's gist or substance, which is often just its category name. At the other end is the slower and often harder attachment of deeper meaning to the scene.

The authors define scene understanding as: What is the scene about? What is the *story* that it is trying to tell? Furthermore, they compare the nature of a scene with the nature of a story. Every story must begin with a context, this is the *gist* of a scene. After that come the characters which are the *objects* of a scene. Then the relationships between the characters are specified, both in respect to each other and the context. These relationships define the *actions* of a scene. Some actions can only be understood in the collective which is the *event* of the scene. Finally, a person should be drawn into a good story. What are the characters thinking and feeling? This is *understanding* a scene.

The example scene can be described as a *women's track meet* or a scene about a *special race for women amputees, where one runner fell and two others were trying to avoid her.* The question is, which one is correct? The answer is, that both interpretations are correct since scene understanding is an interpretation, so it is whatever a person tells you it is. This introduces unique challenges.

Figure 1: What is this scene? [6].

For example, if scene understanding is an interpretation, how can it be evaluated? Or in other words, what is the ground truth for scene understanding? The authors note, that establishing the ground truth at the gist level is easy to deal with. For this reason, research has focused disproportionately on gist level interpretations. To get the ground truth for more elaborated interpretations, scene description tasks can be used. Their, subjects are asked to describe a scene they just saw as if they were talking to another person.

Another challenge is, that the knowledge of all elements in a scene does not tell the whole story. If the objects and actions "man", "wall" and "jumping" are successfully detected in a scene the interpretation might be "man jumping over wall" - but *why* is the man jumping over the wall?

2 Applications

2.1 Indoor Scene Understanding

Choi et. al. [2] propose a method that automatically learns interactions among scene elements in order to understand a scene. Scene elements are the scene type, the objects in 3D space and the spatial layout of the room. In the physical world, these elements are closely linked. A scene's type, such as dining room or bedroom, influences the presence of objects, like dining table or bed, and vice versa (*scene-object interaction*). The 3D geometric faces such as walls, floor and ceiling constrain the placement of individual objects in the image, and vice versa (*object-layout interaction*). The presence of an object suggests that other objects are around (e.g. as dining table suggest that there are chairs around).

Scene interpretation is performed within a hierarchical interaction model, fusing together object detection, layout estimation and scene classification. In a given image, object detections are hypothesized, layout hypotheses are generated that are geometrically consistent with these object hypotheses, and scene class hypotheses are generated by a scene classifier. Then, these hypotheses are put into an image parsing formulation in which a parse graph is constructed for the image (See Figure 2b). At the root of the parse graph is the scene type and layout. The leaves are the individual detections of objects. In between is the core of their system, the novel 3D Geometric Phrases (3DGP) (See Figure 2c).

The 3DGP are trained and encode the geometric and semantic relationships between objects which frequently occur in spatially consistent configurations. In addition, they are defined in 3D which makes them rotation and viewpoint invariant, and therefore more robust than 2D models. By modeling the relationships between groups of objects, 3DGPs allow strongly detected objects to provide contextual evidence to boost the weaker detections of their partners. This is especially beneficial in situations where the appearance of an object is highly variable or the object is occluded. The 3DGP are trained using a new learning scheme presented by the authors.

The high level goal of their system is to take a single image of an indoor scene and classify its scene semantics (such as room type), spatial layout, objects and object relationships in a unified manner. Given a new image, the parse graph must estimate the scene semantics, layout, objects and 3DGPs which makes the space of possible graphs quiet large. To efficiently search this space during inference, the authors present a novel search algorithm which will not be discussed here.

Putting it all together, their model captures rich contextual relationships and provides scene interpretations from a single image in which (i) objects and space interact in a physically valid way, (ii) objects occur in an appropriate scene type, (iii) the object set is self-consistent and (iv) configurations of objects are automatically discovered (See Figure 2d, e).

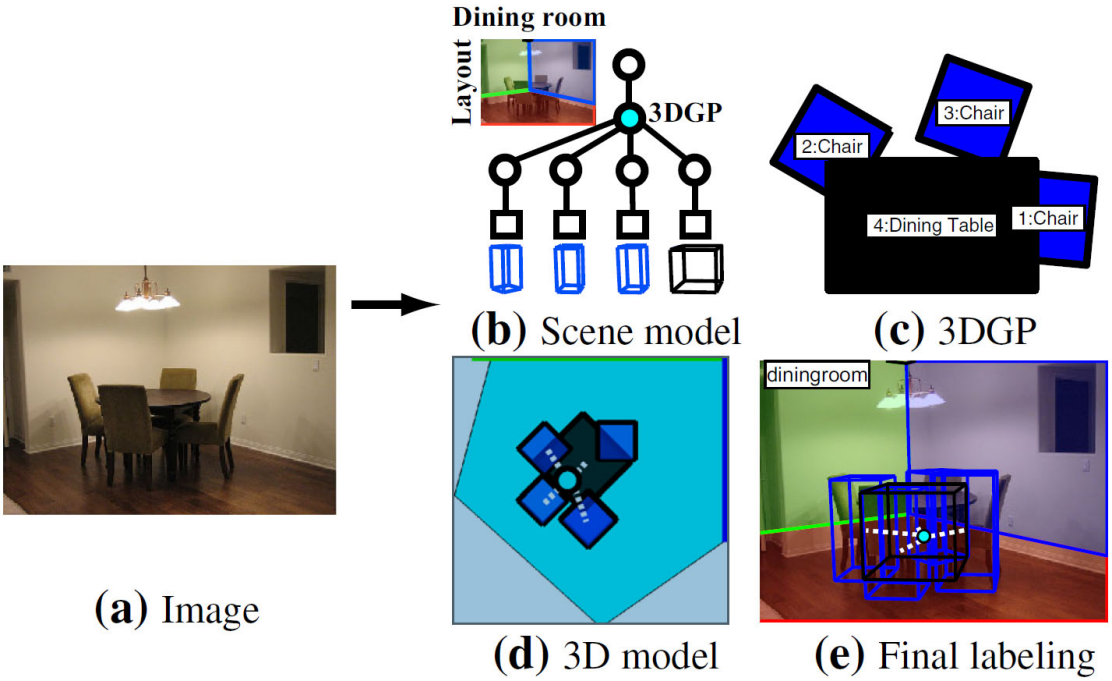


Figure 2: Taken from Choi et. al. [2]. Their unified model combines object detection, layout estimation and scene classification. A single input image (a) is described by a scene model (b), with the scene type and layout at the root, and objects as leaves. The middle nodes are 3D Geometric Phrases (c) describing the 3D relationships among objects (d). Scene understanding means finding the correct parse graph, producing a final labeling (e) of the objects in 3D (bounding cubes), the object groups (dashed white lines), the room layout, and the scene type.

2.2 Labeling Complete Surfaces

Guo et. al. [3] propose a simple and general approach to infer labels of occluded background regions since scene understanding requires reasoning about both what can be seen and what is occluded (See Figure 3). The goal of their work is to label both visible and occluded regions into background categories. For example, a car pixel should be labelled depending on what is behind it. The background categories are defined by hand. For the StreetScenes dataset, one of the datasets they perform their experiments on, the background categories include "building", "road", "sidewalk", "sky", "store" and "tree".

Their method incorporates three basic types of information. First, they classify visible background regions. For example, it is more likely for an occluded patch to be road if it is surrounded by road. For the labeling of the visible part of the scene, they use an off-the-shelf image labeling algorithm and pre-trained object detectors.

Second, they classify visible foreground regions and apply object detectors to find common objects, such as cars and pedestrians. The location of these foreground objects can help to predict the background regions since cars are often on the road, and people are often on sidewalks, for example. After obtaining object bounding boxes and label confidences for the visible regions, they predict the labels of occluded regions with a contextual classifier. Therefore the image is first over-segmented into superpixels, which are then grouped into multiple segmentations. Cues based on color, texture, edge, and vanishing point are then computed for each superpixel. A boosted decision tree classifier combines the prediction and estimates the likelihood of each possible label for each pixel, providing a confidence map for each label. Finally, a shape prior is computed for each label by matching polygons based on current label confidences.

Third, they include global scene priors and region shape priors from training images. Intuitively, the overall pattern of labels should be similar to other images observed in the training set, and the pattern of a particular type of label is likely to match some training image quite closely. They use this intuition by finding polygons in the training set that match their current label predictions. These polygons provide a scene prior (because the training image that they come from should have similar labels overall) and a shape prior (because the transferred region maintains its shape). The transferred regions can be used to refine their per-pixel background labels, and the set of transferred regions provide alternative hypotheses about the hidden portions of the scene.

The most confident set of retrieved polygons provides the best guess of the configuration of the scene. Similarly, the set of second most confident polygons provides an alternative "guess" (See last two columns in Figure 4). The polygon prediction gives a shape prior for background surfaces. Furthermore, the polygons provide a more structured representation than pixels, enabling a connected set of building pixels to be represented as two separate buildings or a set of road pixels as a single road. The final pixel prediction includes predictions for visible surfaces and the transferred polygons to provide the final complete background labeling.

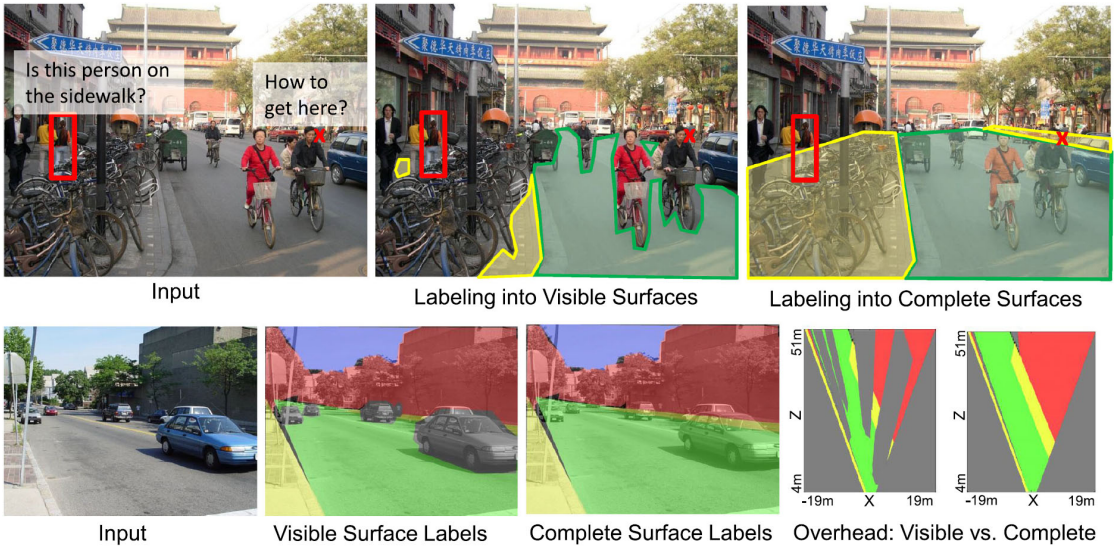


Figure 3: Taken from Guo et. al. [3]. Scene parsing is often viewed as a problem of labeling pixels into visible categories. But these representations leave much of the underlying scene unknowable. For example, because the woman (top row) is occluded, we cannot determine what she is standing on without inferring that the bicycles are occluding the sidewalk. Likewise, finding paths through cluttered scenes is nearly impossible without reasoning about the underlying surfaces. Below, they project the ground into an overhead view (yellow = sidewalk; green = road; red = blocked by building or trees; gray = unknown). Without more complete estimates of the background, huge portions of the scene are left unknown.

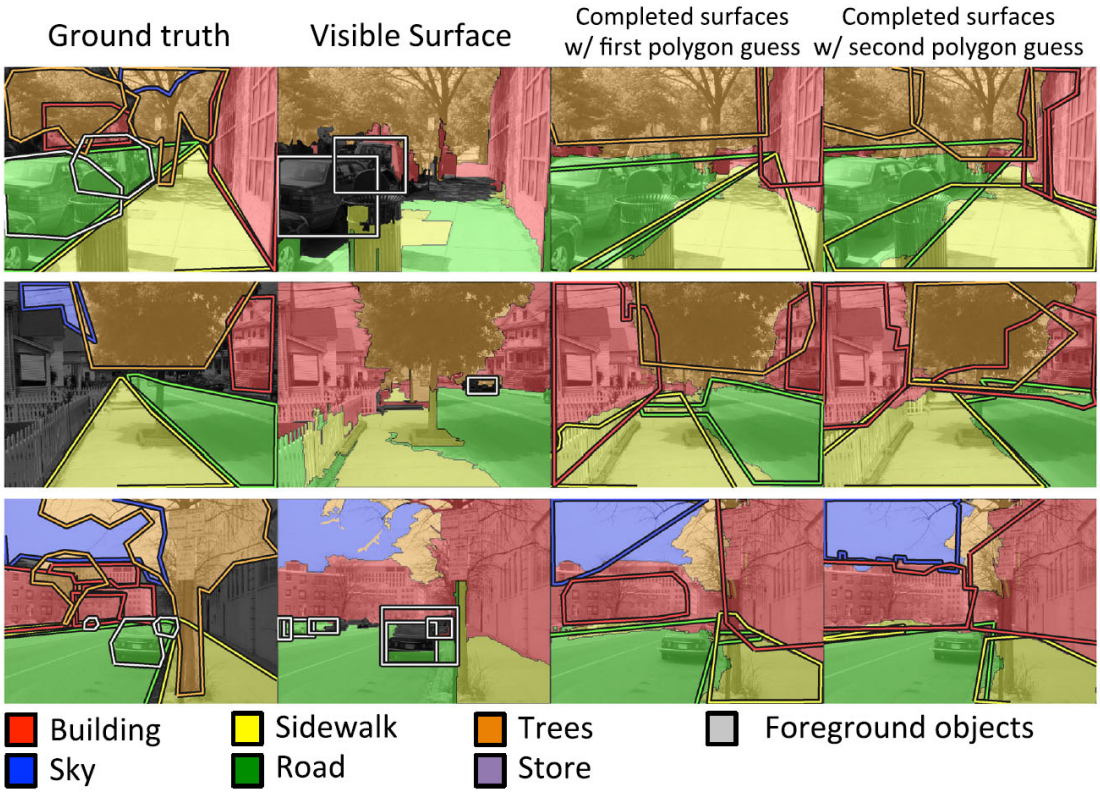


Figure 4: Taken from Guo et. al. [3]. Qualitative results on street scenes. Left to right: ground truth, labeling into visible surfaces and detected objects, labeling of completed surfaces with first polygon guess, same labeling with second polygon guess. In each image, the region colors indicate pixel labels.

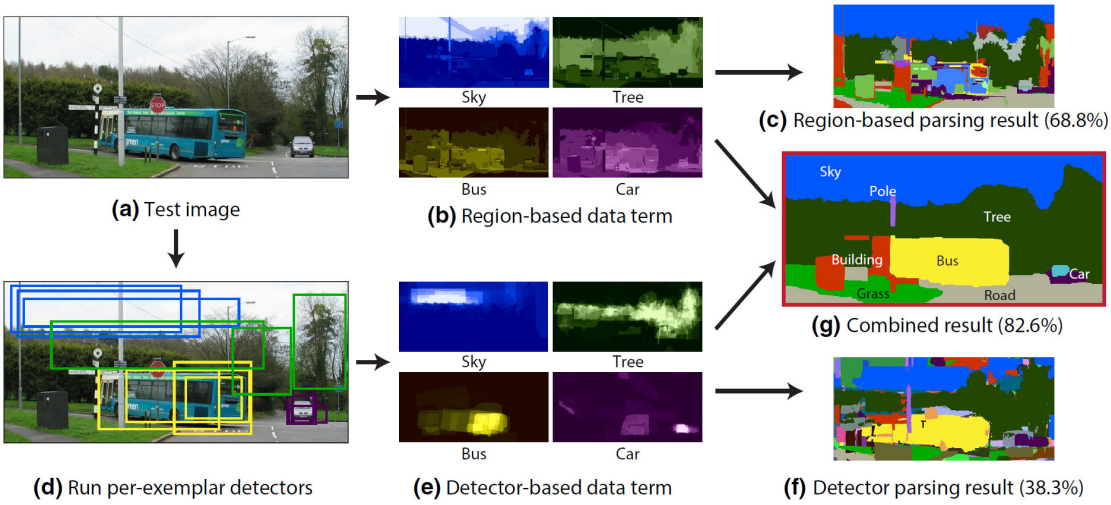


Figure 5: Taken from Tighe et. al. [5]. Overview of their region- and detector-based pixel labeling approach. The test image (a) contains a bus which falls into the "thing" class. Their region-based parsing system computes class likelihoods b based on superpixel features, and it correctly identifies "stuff" regions like sky, road, and trees (b), but is not able to get the bus (c). To find "things" like bus and car, they run per-exemplar detectors on the test image (d) and transfer masks corresponding to detected training exemplars (e). Since the detectors are not well suited for "stuff", the result of detector-based parsing (f) is poor. However, combining region-based and detection-based data terms (g) gives the highest accuracy of all and correctly labels most of the bus and part of the car.

2.3 Scene Parsing

Tighe et. al. [5] describe a system for interpreting a scene by assigning a semantic label at every pixel and inferring the spatial extent of individual object instances together with their occlusion relationships. Their goal is to achieve broad coverage which means the ability to recognize hundreds of classes that commonly occur in everyday outdoor and indoor environments. The non-uniform statistics of object classes in realistic scene images pose a major challenge. When looking at large and diverse scene datasets, just a handful of classes constitute a large portion of all image pixels while a much larger number of classes occupy a small percentage of image pixels. The more frequent classes are for the most part "stuff" such as sky, roads, trees and buildings and have no consistent shape but fairly consistent texture. That way they are acceptably handled by image parsing systems based on pixel- or region-level features. On the contrary, the less frequent classes are mainly "things" such as trucks, dogs, vases, etc. and are better handled by overall shape which makes it necessary to include detectors that model the object shape.

Therefore the authors propose a novel image parsing approach that labels every pixel with its class by combining region- and detector-based cues. An overview of their system is shown in Figure 5.

Given a test image, the following sequence of steps is applied in order to produce a dense pixel-level labeling:

1. Obtain a region-based data term

   They obtain the region-based cues from a superpixel-based parsing system they have developed earlier. Given a query image, the system first uses global image descriptors to identify a retrieval set of training images similar to the query. Then the query is segmented into regions or superpixels. Each region is matched to regions in the retrieval set based on 20 features representing position, shape, color, and texture. These matches are used to produce a log-likelihood ratio score which is used to define the *region-based data term*.

2. Obtain a detector-based data term

   For the detector-based cues, they rely on the framework of per-exemplar detectors or exemplar-SVMs proposed by Malisiewicz et al. [4]. Exemplar-SVMs associate each detection with visually similar training exemplars, allowing for direct transfer of meta-data such as segmentation, geometry or 3D model (See Figure 6 for an example). A per-exemplar detector is trained for each labeled object instance in the dataset.

   Although it may seem intuitive to only train detectors for "thing" categories, the authors train them for all categories, since their experiments demonstrated that this yields the best results for the combined region- and detector-based system.

   At test time, given an image that needs to be parsed, they first obtain a retrieval set of globally similar training images. Then they run the detectors associated with the first k instances of each class in that retrieval set. Next, they take all detections above a given threshold and project the associated object mask into the detected bounding box for each detection. The sum of all detection masks from a class weighted by their detection score gives the *detector-based data term*.

3. Combine the two data terms into a single unary potential with the help of a learned support vector machine.

4. Compute the final pixel labeling by optimizing an objective function.

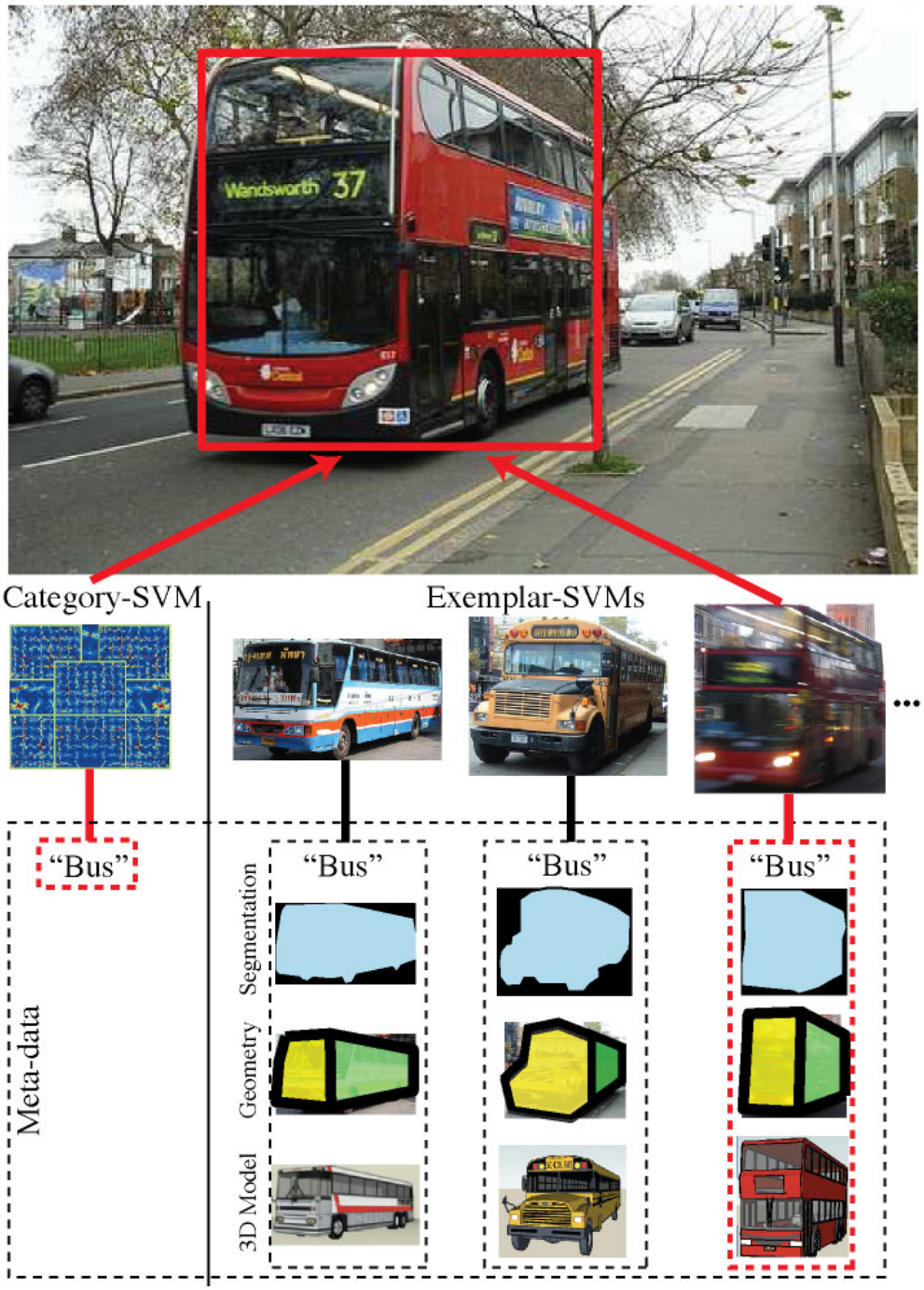


Figure 6: Taken from Malisiewicz et al. [4] showing a classic object category detector compared with a group of exemplar detectors. The output of a typical object detector is just a bounding box and a category label (left) whereas their group of exemplar- SVMs is able to associate each detection with a visually similar training exemplar (right).

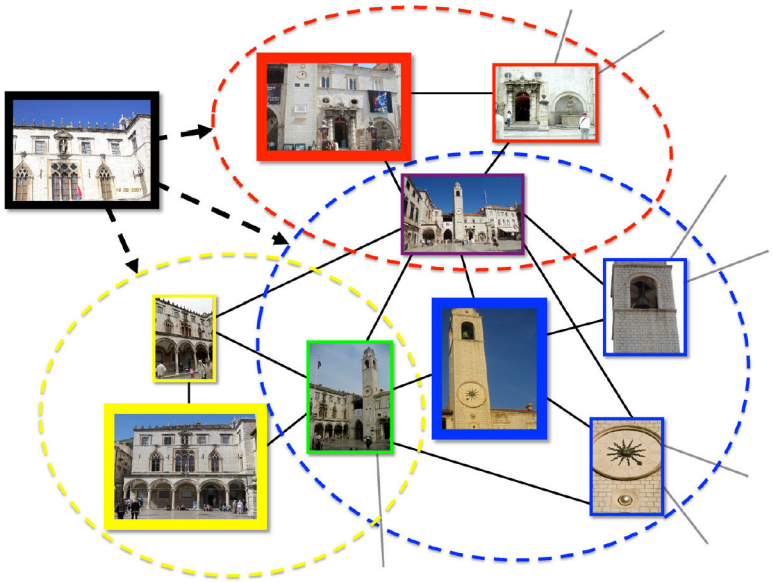


Figure 7: Taken from Cao et. al. [1]. A region of an example image graph with three neighborhoods defined by representative images A, B and C. Nodes in this graph are images, and edges connect visually overlapping images. The method uses the graph to find a set of representative neighborhoods (inside the dotted circles above) that cover the graph, and learns a local distance function for each neighborhood. These distance functions are used to connect a new query image (left) to the rest of the graph and hence recognize its location.

2.4 Location Recognition

Cao et. al. [1] tackle the task of recognizing the location of a query image by matching it to an image database represented as a graph. By exploiting the structure of the graph they are able to improve location recognition methods based on bag-of-words. Graphs represent visual overlaps between images where nodes correspond to images and edges to overlapping, geometrically consistent image pairs.

Given an image graph, the goal is to "plug" a query image in to the graph in the right place, which effectively means to recognizing its location. The idea is that the structure inherent in these graphs encodes much richer information than the set of database images alone. This structural information is used in a location recognition framework, in which a query image is taken, similar images in the database are retrieved and a detailed matching is performed in order to verify each retrieved image until a match is found.

As their main contribution, the authors introduce two new ways to exploit the graph's structure in recognition. First, they build local models of what it means to be similar to each subgraph (or neighborhood) of the graph (See Figure 7). Second, they use the connectivity structure of the graph to encourage diversity in the set of results, using a probabilistic algorithm to retrieve a shortlist of similar images that are more likely to have at least one match.

The image graph is constructed using a standard image matching pipeline. Features are extracted from each image and a pairwise feature matching is performed on a set of candidate image pairs using bag-of-words image similarity. The question is then, how to use the information encoded in the image graph to recognize the location of a query image.

One key piece of information is a notion of similarity, which is provided by the connectivity of the image graph. According to some desired distance metric, it is known which images are expected to be similar (connected pairs) and which are not (disconnected pairs). Therefore, a natural way to approach the problem is to learn a distance metric between pairs of images. The authors have considered several possible ways to learn such a distance metric based on the image graph. The two extremes would be to either learn one a single, global image distance metric for a specific image graph or to learn a local distance metric for each image in the database. Practice showed that they achieve better performance with an approach that balances the two, where the graph is divided into a set of overlapping, representative subgraphs and a separate distance metric is learned for each of these representative subgraphs (or neighborhoods). At query time, these learned distance metrics are used to determine to which neighborhood a query image belongs. To summarize, their approach consists of the following steps:

At Training Time

1. Compute a covering of the graph with a set of overlapping subgraphs.
2. Learn and calibrate a distance metric for each subgraph.

At Query Time

1. Use the models in Step 2 to compute the distance from a query image to each database image, and generate a ranked shortlist of possible image matches.
2. Perform detailed matching and geometric verification with the top database images in the shortlist, until a successful true image match is found.
3. Optionally use this match to further refine the query image location, e.g., through pose estimation.

Figure 8 shows two example query images comparing the approach proposed by the authors and bag-of-words retrieval.

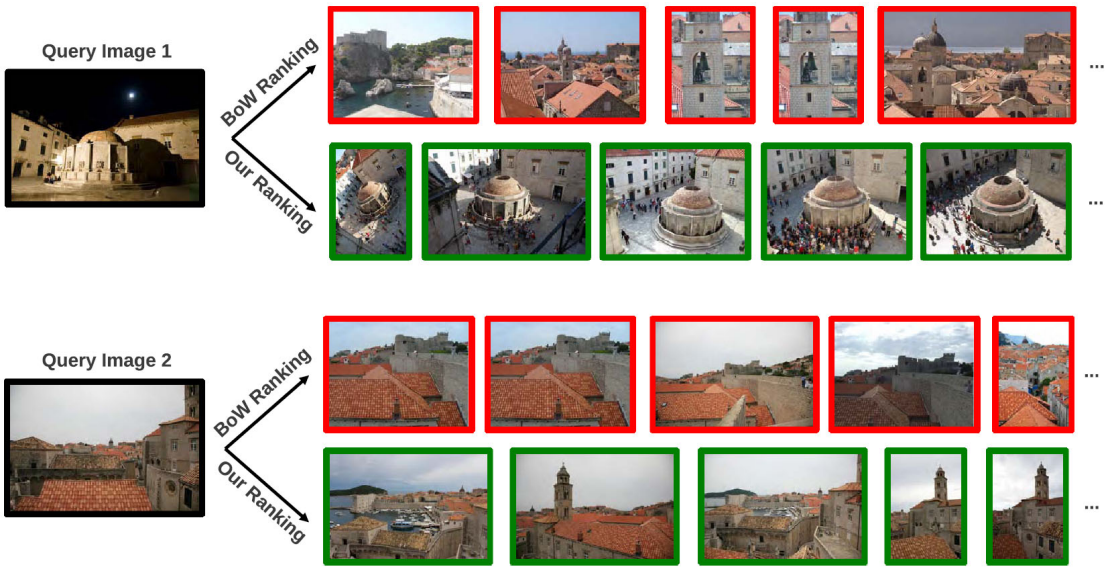


Figure 8: Taken from Cao et. al. [1]. Two example query images and their top 5 ranked results using learned similarities and raw bag-of-words retrieval. For each result, a green border indicates a correct match, and a red border indicates an incorrect match. The two example query images on the left are difficult for bag-of-word retrieval techniques, due to drastically different lighting conditions (query image 1) and confusing features (rooftops in query image 2).

References

[1] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 700–707, 2013.

[2] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Indoor scene understanding with geometric and semantic contexts. *International Journal of Computer Vision*, 112(2):204–220, 2015.

[3] Ruiqi Guo and Derek Hoiem. Labeling complete surfaces in scene understanding. *International Journal of Computer Vision*, 112(2):172–187, 2015.

[4] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 89–96. IEEE, 2011.

[5] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instance inference using regions and per-exemplar detectors. *International Journal of Computer Vision*, 112(2):150–171, 2015.

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

Discussion: Scene Understanding

Image Understanding SS 2017

Andreas Winkler

Discussion: "Labeling of Complete Surfaces"

It was asked if depth information is used for the scene recognition in this application. It was clarified that depth is tracked to some extents, but it was not clearly resolved. We assumed that depth information was manually recorded in this example

In the paper, unoccluded connected surfaces in the scene (e.g. street or sidewalk) are referred to as complete Surfaces. It was asked what the exact definition of this is. The term is not clearly defined in the paper but it was clarified, that the algorithm cares about background, not about objects. Objects are taken as cues but their regions are not important.
The labeling of the complete surfaces is only an intermediate result to understanding of the sene.

It was later asked, if this method is suitable to detect objects. We assume that this method is not suitable for such tasks. For instance it would be difficult to find the border between trees.

How is the background in the paper defined? Depending on the dataset the researches defined certain objects as background or foreground.

Opponent question: What is the ground truth and how to train an application?

In the talk we heard that the ground truth for a scene is a matter of interpretation. A remark was made, that this is not true for a lot of objects, where the ground truth would be pretty clear (e.g. a chair). Manually annotated images in large databases should have a clear ground truth.

It was argued that even for obvious objects such as cars, there would always be exceptions that are unclear.
Another counter argument was made, that manual annotations are not reliable. In some applications, people are paid to manually annotate objects, and they might not correctly perform this task.

This brought us to the conclusion that the ground truth of a scene is a matter of the viewer and his task or strategy. What is important in a scene depends on what a viewer expects to see.
This brought us back to the video of last week, where we had the task to count the number of times a ball is passed, and missed other actions that were goint on. This makes it pretty clear that the strategy and task which is involved when looking at a scene also directly changes the ground truth.

A further problem that was brought up is the difficulty of actually storing the ground truth of

a scene. Even if we annotade every object in a scene, there might be a task that needs different information (e.g. Find every space in the room where we could put a jacket)

The next question from the opponent was about the differences of indoor and outdoor scenes. Are indoor-scene-understanding methods also suitable for outdoor scenes? Where is the difference?

It was argued that indoor scenes have much more constraints. For instance a room is usually shaped like a box and has only 6 different sides. A room in a house has most likely some specific purpose. In outdoor scenes, these constraints do not exist

This also means that annotation of training data is much more difficult. For outdoor scenes the number of different categories of possible scenes is much higher than for indoor scenes by a large factor.
In general, indoor scene algorithms should also work for outdoor scenes, but it would be much more difficult

Opponent question: What is the difference between scene and image understanding?

"Image understanding integrates explicit models of a visual problem domain with ... methods for extracting features from images ... and methods for matching features with domain modules using a control structure."

What is the relation between scene description and the reason of watching?

It was argued that scene understanding applications do not try to determine the reason of things happening in an image. Predictions about some objects are made, but in general it is asked what is there and not why it is there.

Discussion: Scene vs. Image:

A scene exists in reality, whereas an image is a 2D representation of a scene. Scene understanding attempts to understand reality from an image. Image understanding tries to understand what is present in the image, not in reality, whereas scene understanding goes one step further.

Discussion: Purposive Vision:

This approach was popular in the past. First the task is determined, then the data is filtered for actually relevant information. This is also how human visual perception works. When there is a specific task, we filter out other information.

Discussion: Scene parsing:

In one of the papers in the presentation, the parsing of an image was mentioned. The term parsing is used when a compiler processes a programming language, when input is interpreted by the rules of a grammar.
How is an image in that paper parsed? Apparently it is only processed per pixel.

How could parsing of scenes work? There are some relations between objects in a scene. E.g. a Chair is often in close proximity of a Table. There are also some objects that usually do not occur in the same scene (e.g. a Gorilla in a basketball game). We speculated if this means that human perception would work like a grammr that does not have a rule for certain situations like this one.

For the parsing of complex scenes in general, it would be infeasable to look at all objects individually. It would be important to determine a starting point first.
In general the complexity of a scene can be very high. Therefore the interpretation of a scene is not a trivial task, especially for outdoor scenes.

Image Understanding Systems

Image Understanding SS 2017

Rebeka Koszticsak

(1325492)

Introduction

Image understanding is “the transformation of visual images … into descriptions of the world that can interface with other thought processes and elicit appropriate action”[1]. With other words, the aim of Image Understanding Systems (IUS) is to design an algorithm that integrates explicit models of a visual problem with feature extraction and feature matching algorithms. The computer needs to be able to build an inner image model that represents the concept of the image and the world behind it. The visual content of the image is, however, not self-evident. There is usually a reason, why the image is viewed. The reason for what the image is viewed, if given, should be considered upon the interpretation of the visual information. The output of an IUS is the final inner model which describes the image, but its form depends on the problem. It might be a simple yes/no answer, or something more complex like a 3D shape description depending on what kind of output the problem solution expects.

At human visual perception the familiar objects are searched and identified first. To imitate this general and modifiable knowledge a database is set up, where the inner models of the IUS are stored, and which can be expanded if new information arrives. To find the best inner model a control mechanism is needed, which supervises the image identification process. If a model is chosen, just like in the human system, they are tested and matched against the input data. In case of failure the model is discarded or modified until the best matching interpretation is found.

Tasks of an IUS

The task of an IUS is to eventually find a meaningful interpretation of the input visual data. The steps towards this goal are the construction of image models based on the input data. It is important that the model stays consistent in all processing levels, and the description of these levels constantly match the preceding model’s.

The pixel information is extracted and interpreted at the beginning. Pixels are classified according to their relationship-parameters. It can be intensity, change of intensity, similarity in changes, simple similarity over time or any other property that may provide a useful solution. To interpret the resulting primitives (e.g. edges, regions, vectors) some higher level information is needed, such as image processing information, information about the world or information about the present objects, since the pixel data alone is not meaningful enough. This information about surfaces, volumes, boundaries, shadows etc. is finally bounded into unique physical entities like 3D objects and movements?.

To be able to match the image information against the model it needs to be transformed from image-centred to world-centered representation, in order to different representation of the same object, perspective-size, orientation, occlusion etc. cannot infer the identification.

The identification of the object depends not only on the object properties but also on its context (i.e. where these properties are observed), thus also on the relationship between the objects. Furthermore, according to the context and the problem some parts of the objects need to get described in detail and its properties need to be considered by the interpretation, other parts, however, are allowed to be ignored for further processes.Functional Requirements of an IUS

There are some components and concepts which need to be present in every IUS to provide full functionality. These requirements can be classified into two classes, representational and inference/control.These are described in the following.

Representational Concepts

Usually there is a three-level representation structure. The low-level contains the primitives like edges, textures or regions , the intermediate-level represents boundaries, surfaces, volumes etc. and at the high-level the objects, scenes, events and such are stored.

An IUS works with prototypical models, where the definitions, attributes and relationships are also kept general. This way, the amount of models can be reduced, and “related” real-world items can be identified as related without storing extra information (for instance,. if the concept of a ball is stored, every ball will be identified as ball regardless of its color or texture).

In order to keep the models organized and to track the relations between them, some organization concepts are needed. An aggregation represents a “Part-Of” relationship, where the parents are a more abstract version of their children or the children are a decomposition of their parents depending on the direction of the relation. A specialization describes an “Is-A” relation, where the child is a specialization of the parent, so it has at least the properties of the parent and it also expands it. The instantiation is an “Instance-Of” relationship, where the child is an instance of the parent.

During the interpretation spatial and temporal knowledge are used, where the range of the spatial and temporal attributes are considered. The spatial knowledge stands for

information about the location in space, about the spatial relationships between the objects, and also about the shape and the geometry of the object. The temporal knowledge denotes timestamps, duration, intervall, speed and temporal relationships. The range of these attributes demonstrates that the scale of these properties can change, but they still stand for the same or similar objects (e.g. the duration of a lecture at a university and that of in a school).

During the search for hypotheses the similarities and the differences between the set of possible models and the database are compared. It is not necessary to choose more very similar models and match them individually against the data, if the test of a model fails there is higher chance that a different model can be correct than the other similar possibilities. But a small set of neighboring models is essential. In case a small property gives an error, this faulty model can indicate the appropriate direction of search for the right hypothesis.

Inference and Control Concepts

The interpretation of an image is not a predefined fact that can be unambiguously found on the input image. The relevant information is extracted from the input image, and based on these new facts are derived (eg. if the sky is pink, it is sunset). An optimal IUS effectively chooses the best algorithm to extract meaningful data from the input and finds new facts that are logically competent with the information stored in the original image.

Because of a potentially extensive database , a hypothesis activation algorithm is used, in order to create a relatively short list of possible models. The goal-directed activation method aims to find the most specific model for the problem, whereas the model- and data-directed activation method approaches its tasks in inverse directions. Model-based activation considers the parent correct in case the child passes the test. Data-based activation, however, looks for the parent that the child could be part of. Failure-directed activation tests several similar models simultaneously, and if one fails, it searches for alternative models, where the attribute that caused the problem during the test, has a different (and more accurate) value. Finally, it compares all plausible models by matching it against the original data, and then sends the best ones for further investigation.

Usually, the amount of chosen models is vast, therefore it is impossible to individually verify all of them. Consequently, if a relatively good model is found, the attention of the investigation is focused on it and its neighbours. In case no solution was found, the IUS changes its course to explore other, less similar models, too.

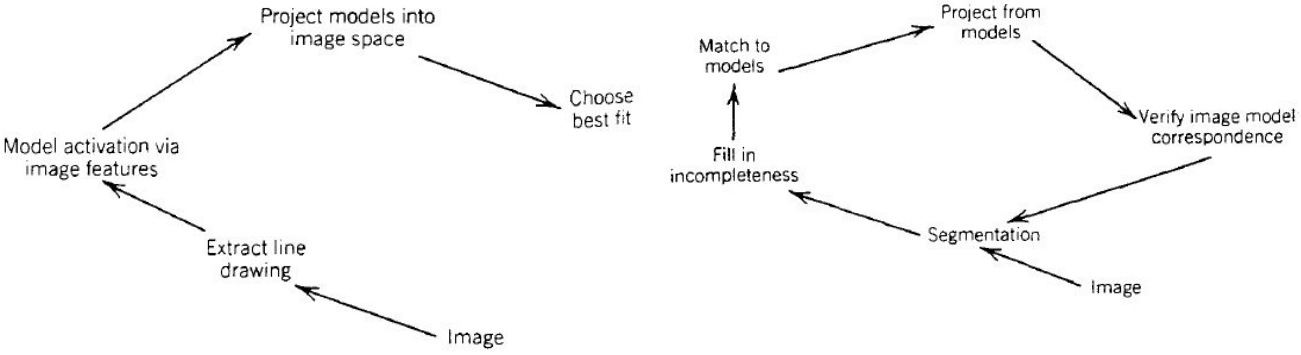
The IUS has to “imagine” what could be going on on the scene, and match the objects with its internal models. For this it is essential to apply an appropriate projection, since the original representation 2D data, needs to be associated with its 3D inner models and to the expectations of the system.

Since the input data is normally incomplete, and it can be disturbed by noise or by other components, it is impossible to derive an unambiguous conclusion from the input. Therefore, in order to conclude with a reasonable result, some evidencing methods must be

applied. Additionally, to overcome the accuracy bias that is common to all algorithms, the IUS establishes a strength of belief parameter based on the plausibility of both the input data and of the applied algorithms.

Control Strategies

In case of image understanding, even if the input data and the computing algorithms are perfect there is always a certain degree of uncertainty and more possibilities for the solution. A cyclic system structure can prevent these pitfalls by giving some feedback about the previous interpretation and then using it to refine the consequent models.



The structure of the first IUSs [2] [3]

Parallel and Serial Control

To mimic the human perception, IUS can use both parallel and serial control strategies. In humans the low level tasks process is parallel, while higher-level exercises, such as the actual interpretation, happen serially. This strategy can be applied to computer systems. Namely, after the image primitives went through a simple parallel processing, later phases (where the components are dependent on each other, e.g. essentially at interpretation) are processed serially.

Hierarchical Control

In case of hierarchical control the modules are able to communicate only in one direction. The hierarchical structure can be used in order to organize the different image-processing steps, or to represent the image in different levels. In the information seeking process a recognition cone is usually used, where the intermediate representation communicates only with its neighbors, and is not able to get further data from any other levels.

The structure can be bottom-up, top-down or combined. In case of a bottom-up architecture the traditional image processing steps are performed (image segmentation, object description, object recognition). The top-down approach however begins with hypothesis construction, where sub-hypotheses are constructed until they can be easily tested. Based on the results the original hypothesis is updated. The combined model is the

most efficient. It uses high-level information by low-level processing (eg.: in case of car identification only rectangles are evaluated).

Non-Hierarchical Control

At non-hierarchical control structures every module is allowed to communicate with any other there is no predefined path of communication. The tasks are decomposed into subtasks, which are processed by expert algorithm of the theme (“daemons”).

There are several control strategies that use non-hierarchical control, for example the blackboard, beam and rule-based approaches. The beam architecture works with a search-tree, where the models that provide the best matches and their near-miss neighbors are considered. In the rule-based architecture the rules are facts about the visual informations and it prescribes the next task if the rule is triggered.

Reasoning and Uncertainty

As mentioned before, in case of image understanding there will always be some uncertainty in the system due to incomplete input data and non-perfectly-accurate image processing algorithms. The IUS needs to be a bit more “creative” and predict some information that is not explicitly found in the input data. That is why some sort of reasoning is needed, that the solution still be consistent, even though some parts of it are uncertain.

Discrete or probabilistic relaxation labeling provides either unambiguous solutions or more possible results with a degree of certainty. In case of evidential reasoning the Bayes Theorem, the more specific Dempster-Shafer Theory is applied to consider the level of uncertainty in the calculations. Furthermore, the Lattice Theory can be used, where the evidences are treated as a vote for a lattice point, and in the end the higher nodes are the most likely solutions. Another possible approach is planning, where either an explicit prediction window is used, or task-paths to the goal are constructed. As an example for explicit prediction window, after edge reduction, pattern recognition is performed, and the recognized pattern is projected to the original image to verify the solution, while at the latter, the most likely ones are actually processed.

Representation Formalisms

Evidently there is a huge amount of models and data to be processed and considered. To allow its efficient operation, an inner data representation structure in IUS is included, where the data, its attributes and the relationships between them are represented.

Spatial representation aims to describe the shape of an object. There are several approaches, how this goal could be reached, none of them unambiguous. The progression of representation has a three level structure, where the 2D (object primitives), a 2(1/2)D (orientation, depth, contours) and a 3D (shape, spatial structure) mesh is stored. The intermediate-level-representation works with several levels, where one level stands for a

certain attribute of the object (surface discontinuity, range, orientation of the surface etc.). Semantic networks work with a graph representation, where the nodes are the objects and the edges are the relationships between them. Another possibility is the use of frames, where each frame is a prototypical representation of an object. In order to keep the overview and note the relations between them they can be organized by using the classical aggregation/specialization/instance construction structures. Heuristics and rules are also representation formalisms, which can be used in more special cases.

Applications

Active contour

Active contours can be used to analyze dynamic image data. After defining the start contour line, even if the image behind it changes, the line will be able to follow the shape of the moving object.

The idea behind snakes[4] is energy-minimization. There are inner (the shape of the snake) and outer (from the image and high-level informations) forces, and the goal of the snake is to keep them as small as possible. If the line is near to the actual feature, based on the above goal, the snake is automatically adjusted to the contour line. The equation is however unstable, that is why a more robust snake growing[5] could be a better choice. The algorithm splits the starting snakes into several smaller ones, of which the ones with minimal energy are allowed to grow while the rest is eliminated. Both algorithm are computationally challenging, so the balloon[6] approach is implemented that terminates faster. It can be used for closed or near closed contours, and the basic idea is to compute a third force in addition to the original two that comes from the inner area of the contour, similar to how a balloon is blown up.

Contextual Image Classification

Contextual image segmentation is used to classify image regions, considering not only the pixel data but also its context, its neighbours. With this method reasonable results can be provided even in case of noisy input or at absence of data regions.

Depending on when the context itself should be considered there are three processing possibilities: post-processing, pre-processing and the combination of the two. In case of post-processing the already labeled data is considered. There are several approaches how they could be processed; a simple median may be calculated to eliminate the noise, or a feature vector from every pixel and its neighbours can be constructed, having the final labels then calculated based on these vectors. At pre-processing the context is already integrated right before the labeling. Based on the context, some pixel areas are merged, or an other approach is to compute further values, not only the median, that consider the neighborhood (average, variance, texture descriptor etc.). The most effective approach however is the one that combines the two previous methods using the Bayes Theorem[7]. The feature vector of each pixel is calculated and the pixels are labelled

according to them. The algorithm is recursive, so because of constraint propagation a bigger neighborhood can be calculated without more effort.

Scene labelling

If the objects are described, it does not mean that they are also recognized. With scene labeling the objects can be interpreted according to their properties and relationships with their neighbours. The following algorithms work with a predefined set of labels that store the object properties, and with a set of relationships between these labels.

The discrete relaxation[8] assigns all labels to all objects and afterwards eliminates the impossible ones. Probabilistic relaxation[9] is a more robust method since it provides a solution to semi-segmented and also to “impossible” data. An object can have more labels, and every label has a probability of how likely they actually are. This approach might lead to locally impossible labeling, but in exchange the global solution can be even better than that of by the discrete algorithm. A further possibility is the use of an interpretation tree[10], where the nodes represent the labels, and the three has as many levels as many objects are present on the image. Starting by the root, a depth-first search is performed, and a possible object is assigned to every label. If the algorithm comes to an impossible solution, it returns to the previous level, and modifies the object assignment until a consistent solution is found.

Semantic Image Segmentation

Semantic image segmentation works on the segmented image, where smaller, similar regions are already marked. The disadvantage of the region segmentation algorithms is that over segmentation is likely to happen, and that there is no way known to stop the algorithm in time in every general cases. The following algorithms tries to eliminate these over-segmented regions, and merge them back using a region adjacency graph, where the nodes represent the regions and the edges represent the common borders between them.

Semantic region growing[11] combine the low- and high-level information to merge neighbouring areas. At the beginning some primitive attributes, like color or intensity, are considered. Afterwards, high-level knowledge is used to eliminate smaller areas and merge them into one single unit. The disadvantage of this method is that the order of merges makes a difference and that it propagates previous and current calculation errors. A better solution is provided by a genetic algorithm[12], where new populations are generated and tested from the original segmented data. Even if the current population provides reasonable interpretation, the algorithm does not terminate, and it is constantly looking for better solutions based on the original data.

References

[1] Shapiro, Stuart C. *ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE SECOND EDITION*. John, 1992.

[2] Roberts, MACHINE PERCEPTION OF THREE-DIMENSIONAL SOLIDS, *Optical and*

Electro-Optical Information Processing (1965): 159-197

[3] Falk, Gilbert. "Interpretation of imperfect line data as a three-dimensional scene." *Artificial intelligence* 3 (1972): 101-144.

[4] Kass, Michael, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models." *International journal of computer vision* 1.4 (1988): 321-331.

[5] Berger, M-O., and Roger Mohr. "Towards autonomy in active contour models." *Pattern Recognition, 1990. Proceedings., 10th International Conference on*. Vol. 1. IEEE, 1990.

[6] Cohen, Laurent D. "On active contour models and balloons." *CVGIP: Image understanding* 53.2 (1991): 211-218.

[7] Kittler, Josef, and Janos Föglein. "Contextual classification of multispectral pixel data." *Image and Vision Computing* 2.1 (1984): 13-29.

[8] Hancock, Edwin R., and Josef Kittler. "Discrete relaxation." *Pattern recognition* 23.7 (1990): 711-733.

[9] Rosenfeld, Azriel, Robert A. Hummel, and Steven W. Zucker. "Scene labeling by relaxation operations." *IEEE Transactions on Systems, Man, and Cybernetics* 6.6 (1976): 420-433.

[10] Grimson, W. Eric L., and Tomas Lozano-Perez. "Localizing overlapping parts by searching the interpretation tree." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 4 (1987): 469-482.

[11] Feldman, Jerome A., and Yoram Yakimowsky. "Decision theory and artificial intelligence: I. A semantics-based region analyzer." *Artificial Intelligence* 5.4 (1975): 349-371.

[12] Sonka, Milan, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

Image Understanding Systems - Discussion Report

Image Understanding SS 2017

Robin Melán (1029201)

June 15, 2017

1 What is first: Artificial Intelligence or Image Understanding?

In the beginning of our discussion the question arose: Do we need Artificial Intelligence (AI) for Image Understanding (IU) or contrariwise? On the one hand, observing this chicken-egg problem from one perspective is to let a AI system learn from data, so a AI system interprets images, which will be the output and with that it gains this *knowledge* and is therefore always in a learning process too. On the other hand IU could be applied on images and this output could be used in the following step as training data for a AI system.
Looking at both perspectives the question emerged: what the definition of AI is.

1.1 What is AI?

According to wikipedia: *Artificial intelligence is intelligence exhibited by machines. In computer science, the field of AI research defines itself as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of success at some goal. Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".*
The first part of the definition is broadly based and in our discussion we concentrated on defining AI in correlation with IU, which goes into the direction of the second part of the definition. A colleague tried to describe it as two circles/categories AI and IU which intersect but do not completely merge together, leaving out e.g. Image Processing as one subdivision of IU. Another colleague tried to define a methodology as AI if it contains some sort of feedback, from which it can grow and continue to learn.

2 Pros & Cons

The intended purpose of AI is to make an intelligent machine that initially thinks as good as a human, but eventually much better, even to the degree of far superior. So in certain areas e.g. medical department, etc. helping doctors to detect for example tumors faster/better, AI can be an advantage and of assistance, but as our discussion continued more and more concerns arose:

- Who is responsible when a AI system fails or decides wrong?
- How did the system came to that specific conclusion?
- Is the system's conclusion for a human being reproducible?
- Do we want to replace a human being with a AI machine completely e.g. in the medical department?

We conclude in this matter, that humanity in general trusts too much and expects from a computer a 100% reliability in their produced solutions. Therefore it is important to sensitize human beings on AI systems and their possibility of also producing inaccurate and spurious results.

3 Image Aesthetic

Due to the short time left at the end of the lecture we could only touch the last subject of the opponent, which was the topic of Image Aesthetic. Aesthetic evaluation of images has attracted a lot of research interests recently, observing illumination in the image, shape, golden ration, etc. The problem here is that the definition if a image is aesthetic or not, is always in the eye of the beholder and therefore very subjective. People could also argue if the content is important or not for defining a image as being aesthetic.