VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Algorithms used for classifying visual signals using methods for extracting significant features

Algoritmy pro klasifikaci visuálních signálů s využitím technik extrakce významných rysů

Bc. Vojtěch Dorňák

Diploma Thesis

Supervisor: Ing. Lukáš Pospíšil, Ph.D.

Ostrava, 2021

Abstrakt

Tohle je český abstrakt

Klíčová slova

typografie, LaTeX, diplomová práce

Abstract

This is English abstract.

Keywords

typography, LaTeX, master thesis

Acknowledgement

I would like to thank all those who helped me with the work, because without them this work would not have happened.

Contents

List of symbols and abbreviations

SIFT	–	Scale Invariant Feature Transform
SURF	–	Speed Up Robust Features
ORB	–	Oriented Fast and Rotated Brief
SVM	–	Support Vector Machine

Chapter 1

Introduction

The goal of this thesis...

Chapter 2

Image Data Transformation

We can attempt to classify the raw image data; the image being represented as a vector of light intensities at each pixel. However, it might be beneficial to transform the image data into a feature space using a feature extraction technique. The feature extraction techniques can be split into two categories. Local feature extractors and global feature extractors.

Local feature extractors localise such points in an image, which could be found regardless of the image subject posstion. These points are called keyponits. The area around such keypoints is then described using a vector called descriptor. The descriptors are created in a way, which allows for matching similar features.

Global feature extractors transform the whole image into a lower dimensional vector, attempting to retain the most important information.

For feature extraction, we compare three local feature extractors: SIFT, SURF and ORB and a global feature extractor: PCA.

In this section, we describe the four feature extraction algorithms.

2.1 SIFT

SIFT (Scale Invariant Feature Transform) was proposed by David Lowe, and published in the original paper [1] in 1999. SIFT.

Compared to ordinary techniques such as the Harris Corner Detector [2] or Canny edge detector [3], the SIFT identifies the general (not only edges and corners) keypoints. The descriptors, which describe the local image region around the keypoints are scale invariant. Moreover, they are invariant to rotation, illumination and to change in affine transformations. In our text, we introduce the methodology of training the classifier that recognizes the objects in real images (photographs); therefore the properties of SIFT feature vector are essential.

2.1.1 Keypoint Detection

Let us consider an input image $I_{img}(x, y)$. A convolution of the image with a Gaussian kernel

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{2.1}$$

at a scale σ is exploited to get a scale-space representation of the image, so that:

$$L(x, y, \sigma) = G(x, y, \sigma) * I_{img}(x, y). \tag{2.2}$$

In order to detect scale invariant keypoints, scale normalised Laplacian of Gaussian (LoG)

$$\triangle_{norm} L(x, y, \sigma) = \sigma \left(\frac{\partial^2 L(x, y, \sigma)}{\partial x^2} + \frac{\partial^2 L(x, y, \sigma)}{\partial y^2} \right) \tag{2.3}$$

is required [4]. It has been shown [5], that local extrema of LoG provide the most stable image features, compared to many other popular image functions.

Determination of the LoG would be time consuming, therefore, it is approximated by the Difference of Gaussian (DoG)[6].The DoG is computed from two adjacent scales separeated by a constant multiplicative factor $k \in \mathbb{R}$:

$$D(x, y, \sigma) := L(x, y, k\sigma) - L(x, y, \sigma). \tag{2.4}$$

To ensure scale invariance, the extrema are located not only in the domain of one DoG, however it is found across different scales as well. The different DoGs at the scales are created by progressively convolving the original imagewith the Gaussian kernel. The consecutive Gaussian kernels' scale differs by the multiplicative factor k.

At each doubling of the scale k, the resolution of an image can be reduced by factor of 2 for efficiency. Each group of blurred images of the same resolution is called an octave. In each octave, we generate the DoG by subtracting the $L(x, y, \sigma)$ of neighbouring scales. This is called the DoG pyramid and can be seen in Figure 2.1.

Each pixel in the DoG pyramid is compared to the 3×3 pixels in DoGs below and above, and to the 8 surrounding pixels. This is shown in Figure 2.2. The pixel is selected as a keypoint candidate, if it has lower or higher value than all of these 26 pixels.

To detect a subpixel locations of extrema, the DoG is interpolated using quadratic Taylor expansion at each keypoint candidate:

$$D(\boldsymbol{x}) = D + \frac{\partial D^T}{\partial \boldsymbol{x}} \boldsymbol{x} + \frac{1}{2} \boldsymbol{x}^T \frac{\partial^2 D}{\partial \boldsymbol{x}^2} \boldsymbol{x}, \tag{2.5}$$

where $D(x, y, \sigma)$ and the partial derivatives are evaluated at the keypoint and $\boldsymbol{x} = (x, y, \sigma)$ is the

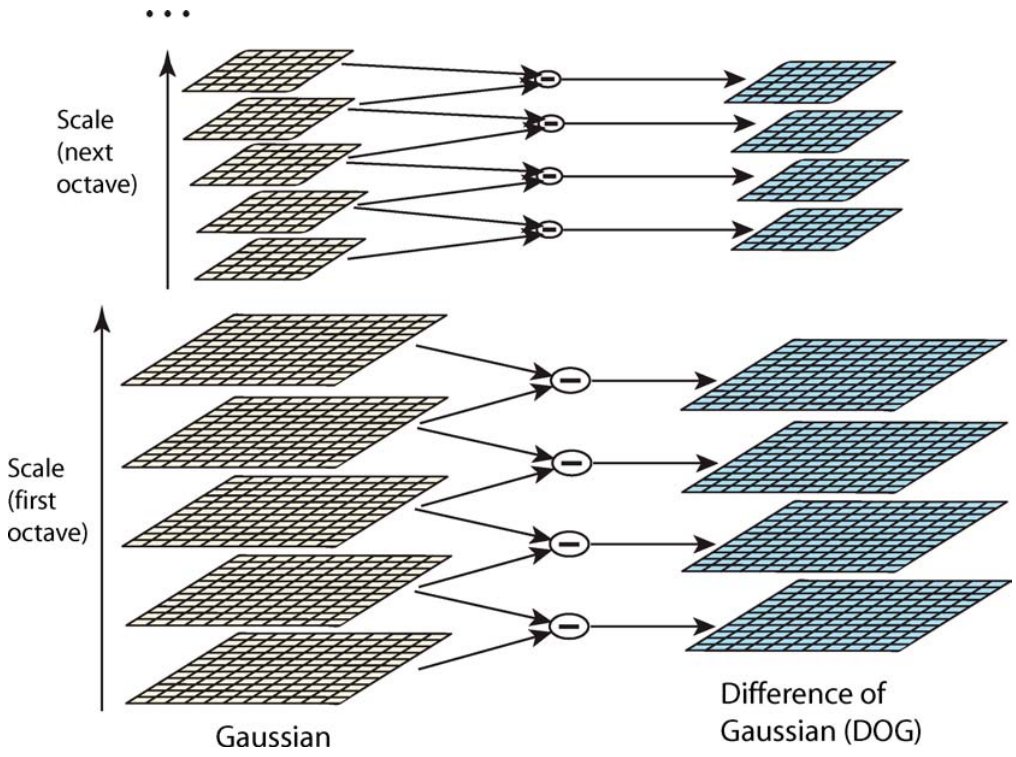

Figure 2.1: DoG pyramid. [6]

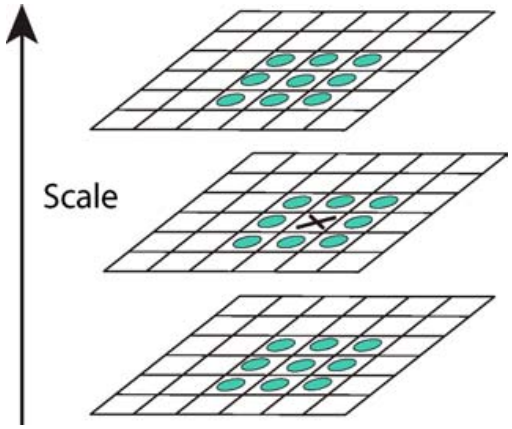

Figure 2.2: Optima of the DoG are selected by the means of comparing pixel to the 26 pixels surrounding it in the scale-space. [6]

offset from the keypoint. Then, extremum $\hat{\boldsymbol{x}}$ is located by setting the gradient of D to be zero vector:

$$\hat{\boldsymbol{x}} = -\frac{\partial^2 D^{-1}}{\partial \boldsymbol{x}^2}\frac{\partial D}{\partial \boldsymbol{x}}. \tag{2.6}$$

If the offset $\hat{\boldsymbol{x}}$ is larger than 0.5 in any dimension, it indicates that the extremum is closer to another point. In this case, the candidate point is changed to the new point and interpolation is performed again.

The function value at the subpixel extremum ,$D(\hat{\boldsymbol{x}})$, is used for rejecting extrema with low contrast. In our experiments, we use the OpenCV SIFT implementation default value $\frac{0.04}{3}$[7].

The DoG has a strong response along edges, even if the location along the edge is poorly determined. These candidate keypoints have principal curvature perpendicular to the edge much larger than the principal curvature along it. The principal curvatures can be computed from a 2×2 Hessian matrix at the location of the keypoint candidate

$$\boldsymbol{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \tag{2.7}$$

where the partial derivates are estimated by taking the differences of neighboring sample points. The eigenvalues of $\boldsymbol{H}$ are proportional to the principal curvatures.

The computation of eigenvalues can be avoided, as only the ratio of the eigenvalues is required. Let us denote the larger eigenvalue as α and the smaller one as β. The trace of $\boldsymbol{H}$ is defined as

$$Tr(\boldsymbol{H}) = D_{xx} + D_{yy} = \alpha + \beta, \tag{2.8}$$

and the determinant as

$$Det(\boldsymbol{H}) = D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta. \tag{2.9}$$

Let r be the ratio of the eigenvalues, such that

$$r = \frac{\alpha}{\beta}. \tag{2.10}$$

Then,

$$\frac{Tr(H)^2}{Det(\boldsymbol{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}. \tag{2.11}$$

Therefore, to inspect the ratio of eigenvalues, we can check

$$\frac{Tr(H)^2}{Det(\boldsymbol{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}. \tag{2.12}$$

The candidate keypoints with this ratio below a treshold r is discarded. In our experiments, we use the default value of OpenCV SIFT implementation: 10.

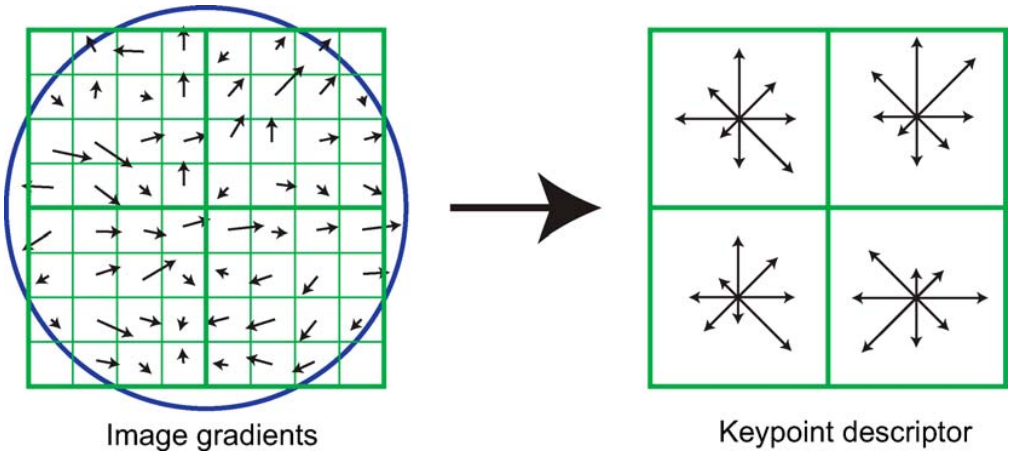

Figure 2.3: Building of the sift-descriptor. [6]

2.1.2 Keypoint Description

We want to create a descriptor for each of our keypoints. These descriptors must be almost identical across different scale, rotation, illumination and other transformations.

In order to ensure descriptor invariance to a rotation, we first determine the keypoint orientation. First, we calculate the gradient magnitude

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \tag{2.13}$$

and orientation

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right), \tag{2.14}$$

of each point within a region around the keypoint. From these, an orientation histogram with 36 bins is created. The largest peak is selected as a keypoint orientation. If there are other peaks more than 80% of the largest peak, new keypoints are created at the same location with the other peaks as their orientation.

For the descriptor, a window 16×16 pixels around the keypoint, rotated by the orientation of the keypoint, is used. In this window, the gradient magnitude and the orientation is computed for each point. The 16×16 window is divided into 16 (4×4) sub-windows. For each sub-window, an 8 bin gradient orientation historam, weighted by gradient magnitudes, is created. This can be seen for smaller 8×8 window in Figure 2.3. These histograms then form a descriptor vector.

2.2 SURF

SURF (Speeded Up Robust Features) is a local feature transform algorithm proposed by Herbert Bay, Tinne Tuytelaars, and Luc Van Gool in 2006[8]. Compared to SIFT[1], the authors claim the

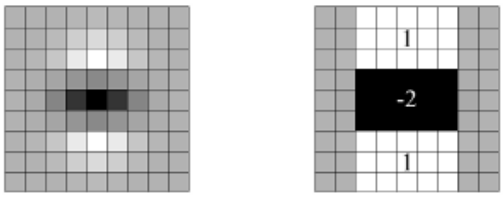

Figure 2.4: Gaussian second order derivative in the y-direction and its approximation using box filters [8]

SURF detector and descriptor, to be faster and more robusit against various image transformations.

2.2.1 Keypoint Detection

The SURF keypoint detection is based on determinant of the Hessian matrix. Let us consider an input image $I_{img}(x, y)$ and the scale space representation of the image

$$L(x, y, \sigma) = G(x, y, \sigma) * I_{img}(x, y), \tag{2.15}$$

where $G(x, y, \sigma)$ is the Gaussian kernel described in (2.1).

The Hessian matrix at scale σ is defined as

$$\mathcal{H}(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \tag{2.16}$$

where $L_{xx}(x, y, \sigma)$, $L_{xy}(x, y, \sigma)$ and $L_{yy}(x, y, \sigma)$ are the second order derivatives of the scale space representation of the image at a point (x, y).

As the property of Gaussian filter, that no new structures can appear while going to a lower resolution has been shown not to apply in 2D case [4], the SURF authors choose to approximate the second order derivatives of Gaussian filter with box filters (shown in Figure 2.4). These allow for the use of integral images, which reduce the computational complexity.

Let us denote the approximations by D_{xx}, D_{xy}, and D_{yy}. The relative weights in the calculation of determinant of Hessian need to be weighted with by 0.9, which yields

$$\det(\mathcal{H}) = D_{xx} * D_{yy} - (0.9 * D_{xy})^2. \tag{2.17}$$

Due to the use of box filters and integral images, any size of the box filter can be applied to the original image at the same speed directly. Therefore, the scale space is created by the use of up-scaled filters of size 9×9, 15×15, 21×21, 27×27, etc. For each octave, the difference between filter sizes is doubled (from 6 to 12 to 24).

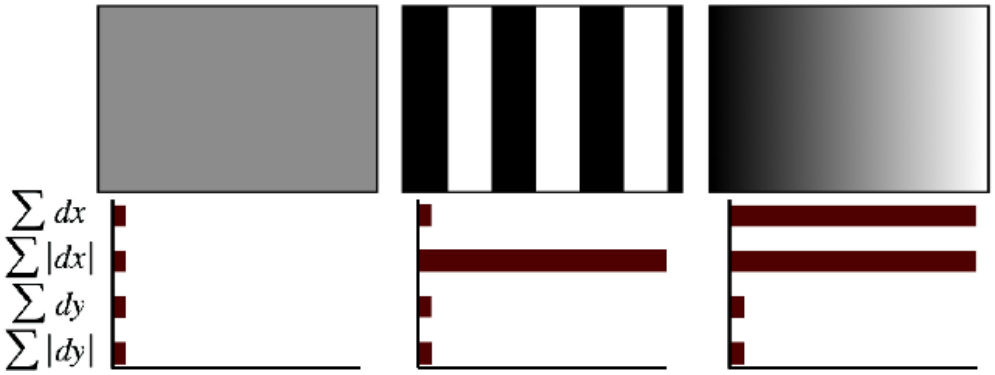

Figure 2.5: Behaviour of SURF descriptor for different image patterns[8]

As the box filter layout remains the same, the corresponding Gaussian filter scales accordingle. The 9×9 box filter corresponds to a Gaussian filter with the scale $\sigma = 1.2$. From this, we can calculate the corresponding Gaussian filter scale for each box filter size.

To select the keypoints, non-maximum suppression in the $3 \times 3 \times 3$ neighbourhood of each point is applied. The maxima of the determinant of the Hessian matrix are then interpolated the same way as in the SIFT algorithm (using quadratic Taylor expansion).

2.2.2 Keypoint Description

First, we need to ensure descriptor's invariance to rotation. For the keypoint, we calculate the Haar-wavelet responses in x and y direction in a circular neighbourhood of $6s$, where s is the Gaussian filter scale at which the keypoint was found. Integral images are used for speeding up the process.

The wavelet responses are then weighted with a Gaussian($\sigma = 2.5s$) centered at the keypoint. The weighted responses are represented as vectors in a space with the x response being a vector along the x axis, and the y response being a vector along the y axis. All the vectors within a sliding window of $\frac{\pi}{3}$ are summed, and the longest of the vector is selected as the keypoint orientation.

The descriptor is constructed from a square window the size of $20s$ around a keypoint. The window is then rotated along the keypoint orientation. This window is then divided into 16 (4×4) square sub-windows. In each sub-window, the features are calculated using 5×5 regulary spaced sample points. For these, we calculate the Haar wavelet responses in horizontal and vertical direction, where "horizontal" and "vertical" are defined in relation to the keypoint orientation. The responses are weighted with a Gaussian($\sigma = 3.3s$), centered at the keypoint.

Let us call the horizontal responses d_x and the vertical responses d_y. Over each sub-window, we denote the sum of the responses $\sum d_x$ and $\sum d_y$, and the sum of absolute values of the responses $\sum |d_x|$ and $\sum |d_y|$. The behaviour of these values for different image patterns can be seen in Figure 2.5. Combining $\sum d_x$, $\sum d_y$, $\sum |d_x|$, and $\sum |d_y|$ for the each of the 16 sub-window into a vector, we get a descriptor of length 64.

2.3 ORB

ORB (Oriented FAST and Rotated BRIEF) is a local feature extractor proposed by Ethan Rublee, Vincent Rabaud, Kurt Konolige and Gary Bradski in 2011 [9].

The algorithm builds on a FAST (Features from Accelerated Segment Test) corner detetector[10] and a BRIEF (Binary Robust Independent Elementary Features) descriptor[11].

2.3.1 Keypoint Detection

The ORB algorithm uses FAST-9 variant of the FAST corner detector. This detector compares each pixel intensity (denoted as I_p) in an image with the intensities of pixels in circle of radius 9 around the pixel.

Let $n \in \mathbb{N}$ and threshold $t \in \mathbb{R}^+$ be given. Let S be a set of pixels in a circle of radius 9 around the examined pixel. Let us denote I_x as the intensity of a pixel x. The examined pixel is selected as a corner, if there exists a set of n continguous pixels $S_n \in S$, where $\forall x \in S_n : I_x + t < I_p$, or $\forall x \in S_n : I_x - t > I_p$.

The selected corners, i.e. keypoints are then ordered according to a Harris corner measure[2]. For N keypoints, the threshold is selected low enough to get more than N keypoints, then the best N keypoints (according to Harris corner measure) are selected.

To find multi-scale features, the scale pyramid of an image is generated and keypoints are generated at each level in the pyramid.

2.3.2 Keypoint Description

To ensure the descriptor's invariance to rotation, we need to determine the keypoint orientation. For this, the intensity centroid[12] is used. The intensity centroid expects the intensity of a keypoint to be offset from its center. The vector from the center to the centroid is used for the keypoint orientation.

Given an image $I(x, y)$, the centroid is found using a moment of a patch

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y). \tag{2.18}$$

The centroid is then located as

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right), \tag{2.19}$$

from which we can obtain the keypoint orientation

$$\theta = \text{atan2}(m_{01}, m_{10}), \tag{2.20}$$

where atan2 is the quarant-aware version of arctan.

As an ORB descriptor, a variation on BRIEF descriptor is used. The BRIEF descriptor is a vector of binary values. This allows for fast matching using a Hamming distance.

The descriptor values is computed by comparing random pairs of pixel intensities in a patch. The binary vector is created from the responses on a patch $\boldsymbol{p}$ of test

$$\tau(\boldsymbol{p}; x, y) = \begin{cases} 1 & \text{if } \boldsymbol{p}(x) < \boldsymbol{p}(y) \\ 0 & \text{otherwise} \end{cases}, \tag{2.21}$$

where $\boldsymbol{p}(x)$ is the pixel intensity of a patch $\boldsymbol{p}$ at a point x, x and y. The pairs of points x and y are from a random predetermined set

$$\boldsymbol{S} = \begin{pmatrix} x_1 \dots x_n \\ y_1 \dots y_n \end{pmatrix} \tag{2.22}$$

where n is the size of our descriptor. The BRIEF descriptor is then defined as a vector of n binary tests:

$$f_n(\boldsymbol{p}) \coloneqq \sum_{i=1}^{n} 2^{i-1} \tau(\boldsymbol{p}; x_i, y_i) \tag{2.23}$$

The steered version of a BRIEF descriptor, according to the orientation θ of the keypoint, can be created by using a coresponding rotation matrix $\boldsymbol{R}_\theta$:

$$\boldsymbol{S}_{\boldsymbol{\theta}} = \boldsymbol{R}_\theta \boldsymbol{S}. \tag{2.24}$$

The sets of pairs are precomputed in a lookup table for discretized θ in icrements of $\frac{2\pi}{30}$ to improve speed. In the end, the steered BRIEF descriptor becomes

$$g_n(\boldsymbol{p}, \theta) \coloneqq f_n(\boldsymbol{p}) | (x_i, y_i) \in \boldsymbol{S}_\theta. \tag{2.25}$$

Chapter 3

Classifiers

For data classification we are using two classification techniques, Bayes classifier and SVM. The goal of these techniques is to find a mapping from a feature space into a space of labels.

3.1 Bayes Classifier

This classifier is suitable for classifying data, represented by a stochastic vector. The BoVW data can be easily transformed into such vector. Instead of each component of the BoVW vector representing the amount of keypoints in respective category, the component in our new vector represents the probabilty of keypoints belonging in respective category.

Let us denote the stochastic data vector $\Pi_{xt} \in mathbbR^{K_x}, t = 1, \ldots, T$, where T is the amount of samples. Let the vector $\Pi yt \in \mathbb{R}^{K_y}$ be vector of probabilities, with which Πxt belongs to each category, and K_y the amount of categories.

Given a stochastic data vector Π_x, we can describe the transformation $\mathbb{R}^{K_x} \to \mathbb{R}^{K_y}$ using a matrix $\Delta \in \mathbb{R}^{K_y, K_x}$:

$$\Delta = \begin{bmatrix} P(\Pi_y^1|\Pi_x^1) & P(\Pi_y^1|\Pi_x^2) & \ldots & P(\Pi_y^1|\Pi_x^{K_x}) \\ P(\Pi_y^2|\Pi_x^1) & P(\Pi_y^2|\Pi_x^2) & \ldots & P(\Pi_y^2|\Pi_x^{K_x}) \\ \vdots & \ddots & & \\ P(\Pi_y^{K_y}|\Pi_x^1) & P(\Pi_y^{K_y}|\Pi_x^2) & \ldots & P(\Pi_y^{K_y}|\Pi_x^{K_x}) \end{bmatrix}, \tag{3.1}$$

where Π_x^n is the n-th element of Π_x, similar for Π_y^n, and the matrix Δ is a left stochastic matrix.

The search for the optimal Δ^* can be written as

$$\Delta^* = \underset{\Delta \in \Omega_\Delta}{\arg\min} \sum_{t=1}^{T} \mathrm{dist}(\Pi_{yt}, \Delta \Pi_{xt}), \tag{3.2}$$

where Ω_Δ is a set of left stochastic matrices. The $\text{dist}(\Pi_{yt}, \Delta\Pi_{xt})$ is calculated as Kullback-Leiber divergence[13]:

$$\text{dist}(\Pi_{yt}, \Delta\Pi_{xt}) = -\sum_{i=1}^{K_y} \Pi_{yt}^i \ln \frac{(\Delta\Pi_{xt})_i}{\Pi_{yt}^i} = -\sum_{i=1}^{K_y} \Pi_{yt}^i (\ln(\Delta\Pi_{xt})_i - \ln \Pi_{yt}^i). \tag{3.3}$$

For the purpose of optimization, the term $\ln \Pi_{yt}^i$ is constant, therefore it can be ignored:

$$\text{dist}(\Pi_{yt}, \Delta\Pi_{xt}) \propto -\sum_{i=1}^{K_y} \Pi_{yt}^i \ln(\Delta\Pi_{xt})_i. \tag{3.4}$$

This problem is hard to minimize analytically, therefore, Jensen's inequality is used:

$$-\sum_{i=1}^{K_y} \Pi_{yt}^i \ln(\Delta\Pi_{xt})_i \leq -\sum_{i=1}^{K_y} \Pi_{yt}^i (\sum_{j=1}^{K_x} \Pi_{xt}^j \ln(\Delta_{ij})) = -\sum_{i=1}^{K_y} \sum_{j=1}^{K_x} \Pi_{yt}^i \Pi_{xt}^j \ln \Delta_{ij}. \tag{3.5}$$

From this, we get an optimization problem

$$\Delta^* = \underset{\Delta \in \Omega_\Delta}{\arg\min} -\sum_{t=1}^{T} \sum_{i=1}^{K_y} \sum_{j=1}^{K_x} \Pi_{yt}^i \Pi_{xt}^j \ln \Delta_{ij}, \tag{3.6}$$

where

$$\Omega_{Delta} = \{\Delta \in [0,1], \forall j \in \{1, 2, \ldots, K_x\} : \sum_{i=1}^{K_y} \Delta_{ij} = 1\}, \tag{3.7}$$

which can be solved analytically:

$$L(\Delta, \lambda) = -\sum_{t=1}^{T} \sum_{i=1}^{K_y} \sum_{j=1}^{K_x} \Pi_{yt}^i \Pi_{xt}^j \ln \Delta_{ij} + \sum_{j=1}^{K_x} \lambda_j (\sum_{i=1}^{K_y} \Delta_{ij} - 1) \tag{3.8}$$

$$\nabla_{\Delta_{\hat{i}\hat{j}}} L(\Delta_{\hat{i}\hat{j}}, \lambda) = -\frac{1}{\Delta_{\hat{i}\hat{j}}} \sum_{t=1}^{T} \Pi_{yt}^{\hat{i}} \Pi_{xt}^{\hat{j}} + \lambda_{\hat{j}} = 0 \tag{3.9}$$

Chapter 4

Datasets

In this chapter, we take a look at the datasets we use, to test our pipeline.

Chapter 5

Results

In this chapter, we take a look at the different results of our classification.

Chapter 6

Conclusion

In conclusion, nothing works.

Bibliography

1. LOWE, David G. Object recognition from local scale-invariant features. In: *Proceedings of the seventh IEEE international conference on computer vision*. 1999, vol. 2, pp. 1150–1157.
2. CHRIS HARRIS, Mike Stephens. A COMBINED CORNER AND EDGE DETECTOR. *Plessey Research Roke Manor*. 1988.
3. CANNY, John F. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986.
4. KOENDERINK, Jan J. The structure of images. *Biological cybernetics*. 1984, vol. 50, no. 5, pp. 363–370.
5. MIKOLAJCZYK, Krystian. *Detection of local features invariant to affines transformations*. 2002. PhD thesis. Institut National Polytechnique de Grenoble - INPG.
6. LOWE, David G. Distinctive Image Featuresfrom Scale-Invariant Keypoints. *International Journal of Computer Vision*. 2004.
7. BRADSKI, Gary. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.
8. BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. Surf: Speeded up robust features. In: *European conference on computer vision*. 2006, pp. 404–417.
9. RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary. ORB: An efficient alternative to SIFT or SURF. In: *2011 International conference on computer vision*. 2011, pp. 2564–2571.
10. ROSTEN, Edward; DRUMMOND, Tom. Machine learning for high-speed corner detection. In: *European conference on computer vision*. 2006, pp. 430–443.
11. CALONDER, Michael; LEPETIT, Vincent; STRECHA, Christoph; FUA, Pascal. Brief: Binary robust independent elementary features. In: *European conference on computer vision*. 2010, pp. 778–792.
12. ROSIN, Paul L. Measuring corner properties. *Computer Vision and Image Understanding*. 1999, vol. 73, no. 2, pp. 291–307.

13. KULLBACK, Solomon; LEIBLER, Richard A. On information and sufficiency. *The annals of mathematical statistics*. 1951, vol. 22, no. 1, pp. 79–86.