Algoritmy digitální kartografie a GIS

Úloha č. 2:

Generalizace budov

Bc. Taťána Bláhová, Bc. Tomáš Krauz, Bc. Adéla Kučerová

8. února 2023

Obsah

1 Zadání

Úloha č. 2: Generalizace budov

Vstup: množina budov $B = \{B_i\}_{i=}^{n}$, *budova* $B_i = \{P_{i,j}\}_{j=1}^{m}$.

Výstup: $G(B_i)$.

Ze souboru načtěte vstupní data představovaná lomovými body budov. Pro tyto účely použijte vhodnou datovou sadu, např. ZABAGED.

Pro každou budovu určete její hlavní směry metodami:

- Minimum Area Enclosing Rectangle,
- Wall Average.

U první metody použijte některý z algoritmů pro konstrukci konvexní obálky. Budovu nahraďte obdélníkem se středem v jejím těžišti orientovaným v obou hlavních směrech, jeho plocha bude stejná jako plocha budovy. Výsledky generalizace vhodně vizualizujte.

Odhadněte efektivitu obou metod, vzájemně je porovnejte a zhodnoťte. Pokuste se identifikovat, pro které tvary budov dávají metody nevhodné výsledky, a pro které naopak poskytují vhodnou aproximaci.

Hodnocení:

Krok	Hodnocení
Generalizace budov metodami Minimum Area Enclosing Rectangle a Wall Average	15b
Generalizace budov metodou Longest Edge.	*+5b*
Generalizace budov metodou Weighted Bisector.	*+8b*
Implementace další metody konstrukce konvexní obálky.	*+5b*
Ošetření singulárního případu u při generování konvexní obálky.	*+2b*
Max celkem:	**35b**

Obrázek 1.1: Zadání úlohy

2 Bonusové úlohy

1. *Generalizace budov metodou Longest Edge. [+5b]*

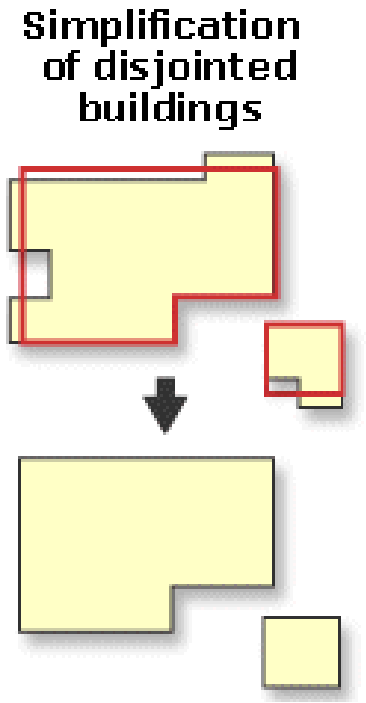
3 Popis a rozbor problému

Na vstupu je množina polygonů (budov) $B = \{B_i\}_{i=1}^{n}$, kde budova $B_i = \{p[x, y]_{i,j}\}_{j=i}^{m}$. Pro každou z budov je hledána její generalizace do úrovně LoD0. Generalizace budovy je zjednodušení tvaru polygonu za účelem redukce množství dat a je provedena pomocí konvexní obálky jako odhad tvaru prostorového jevu.

Konvexní obálkou konečné množiny B_i se rozumí konvexní mnohoúhelník P s nejmenší plochou. Množina P je konvexní pokud spojnice libovolných dvou bodů množiny leží zcela uvnitř množiny P.

Pro generalizaci je možno využít metody jako Minimum Area Enclosing Rectangle, Longest Edge nebo Wall Average. Metody jsou popsány v další kapitole.

Před generalizací může mít budova nepravidelný tvar, po zjednodušení zůstane čtyřúhelník, který symbolicky nahradí původní polygon budovy.



Obrázek 3.1: Generalizace nespojených polygonů [1]

4 Popis použitých algoritmů

4.1 Hledání konvexní obálky

K hledání konvexní obálky byla použita metoda **Jarvis scan**.

Jako předzpracování je nalezen pivot *q* jako $q = min_{\forall p_i \in S}(y_i)$. Takto nalezený bod *q* je přidán do konvexní obálky H. Následně je vybrán bod p_{j-1} pro vytvoření přímky rovnoběžné s osou X. Přímka je daná body q a p_{j-1}.

Pomocí cyklu je přidán do konvexní obálky bod s maximálním úhlem $\angle(p_{j-1}, p_j, p_{j+1})$.

Postup je popsán pomocí vzorců na stránkách [2] .

Jarvis scan - implementace :

1. Nalezení pivota q, $q = min(y_i)$,
2. Přidání $q \rightarrow H$,
3. Inicializace $p_{j-1} \in X, p_j = q, p_{j+1} = p_{j-1}$,
4. Opakování, dokud $p_{j+1} \neq q$:
5. Nalezení $p_{j+1} = argmax_{\forall p_i \in P} \angle(p_{j-1}, p_j, pi)$
6. Přidání $p_{j+1} \rightarrow H$
7. $p_{j-1} = p_j; p_j = p_{j+1}$.

4.2 Generalizace budov

4.2.1 Metoda Minimum Area Enclosing Rectangle

Pomocí této metody je zjištěn hlavní směr budovy. Směr je určen jako směr delší ze stran ohraničujícího obdélníku s minimální plochou.

Minimum Area Enclosing Rectangle - implementace :

1. Nalezení H = CH(S)
2. Inicializace R = MMB(S), $\underline{A}$ = A(MMB(S))
3. Opakování pro každou hranu *e* obálky H:
4. Spočtení směrnice σ hrany hrany *e*,
5. Otočení S o úhel -σ : $S_r = R(-\sigma)S$
6. Nalezení MMB(S_r) a určení A(MMB(S_r))
7. Pokud $A < \underline{A}$:
8. $\underline{A}$ = A, $\underline{MMB}$=MMB, $\underline{\sigma}$= σ

9. R = R($\underline{\sigma}$)MMB

4.2.2 Metoda Longest Edge

Jedná se o detekci hlavního směru budovy, tj. nejdelší hrana polygonu, který budovu reprezentuje. Druhý hlavní směr je na ni kolmý.

Neplatí, že hlavní strana reprezentuje hlavní směr.

Metoda nedosahuje nejlepších výsledků při netypických tvarech polygonu.

Longest Edge - implementace :

1. Inicializace vektoru dvojic délka hrany a směrnice přímky
2. Opakování pro všechny body polygonu:
3. $$s_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \tag{4.1}$$
4. $$\sigma_j = \arctan 2 \frac{y_{j+1-y_j}}{x_{j+1} - x_j} \tag{4.2}$$
5. Seřazení dvojic podle velikosti s_j
6. Uložení $\sigma_{s,max}$, tj. poslední prvek ve vektoru dvojic
7. Vytvoření enclosing rectangle (ohraničující obdélník)

4.2.3 Metoda Wall Average

Všem stranám polygonu (budovy) je spočtena směrnice σ_i, na kterou je aplikována metoda $mod(\frac{\Phi}{2})$ pro nalezení zbytku po dělení. Výsledný směr natočení polygonu je pak dán váženým průměrem těchto zbytků, kde roli váhy zastupuje délka příslušné strany.

Metoda je komplexní a citlivá na úhly různé od úhlů pravých.

Wall Average - implementace :

1. Inicializace pro $\sigma = 0$ ... směr natočení budovy;
2. $\Sigma s_i = 0$ ... obvod budovy;
3. σ ... směrnice první hrany budovy
4. Opakování přes všechny body polygonu:
5. $$s_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \tag{4.3}$$
6. $$\sigma_j = \arctan 2 \frac{y_{j+1-y_j}}{x_{j+1} - x_j} \tag{4.4}$$
7. $$d\sigma_i = \sigma_i - \sigma \tag{4.5}$$
8. $$k_i = round \frac{2 * d\sigma_i}{\pi} \tag{4.6}$$

9.
$$r_i = d\sigma_i - k_i * \frac{\pi}{2} \tag{4.7}$$

10.
$$\sigma+ = r_i * s_i \tag{4.8}$$

11.
$$\Sigma s_i+ = s_i \tag{4.9}$$

12.
$$\sigma = \sigma + \frac{\sigma}{\Sigma s_i} - \textit{váž.průměr} \tag{4.10}$$

13. Vytvoření enclosing rectangle (ohraničující obdélník).

5 Vzhled aplikace

Spuštěním aplikace se otevře okno "Building Simplify". Okno je rozděleno na dvě části. Na levé straně se nachází Canvas, část vyhrazená pro vykreslování geometrie. Vpravo je menu, které obsahuje tlačítka tříd QPushButton a rolovací menu QComboBox.

Tlačítka QPushButton slouží k nahrání dat, generalizaci, pročištění grafického okna (Canvas). Rolovací menu slouží k výběru metody generalizace.

Obrázek 5.1: Vzhled aplikace

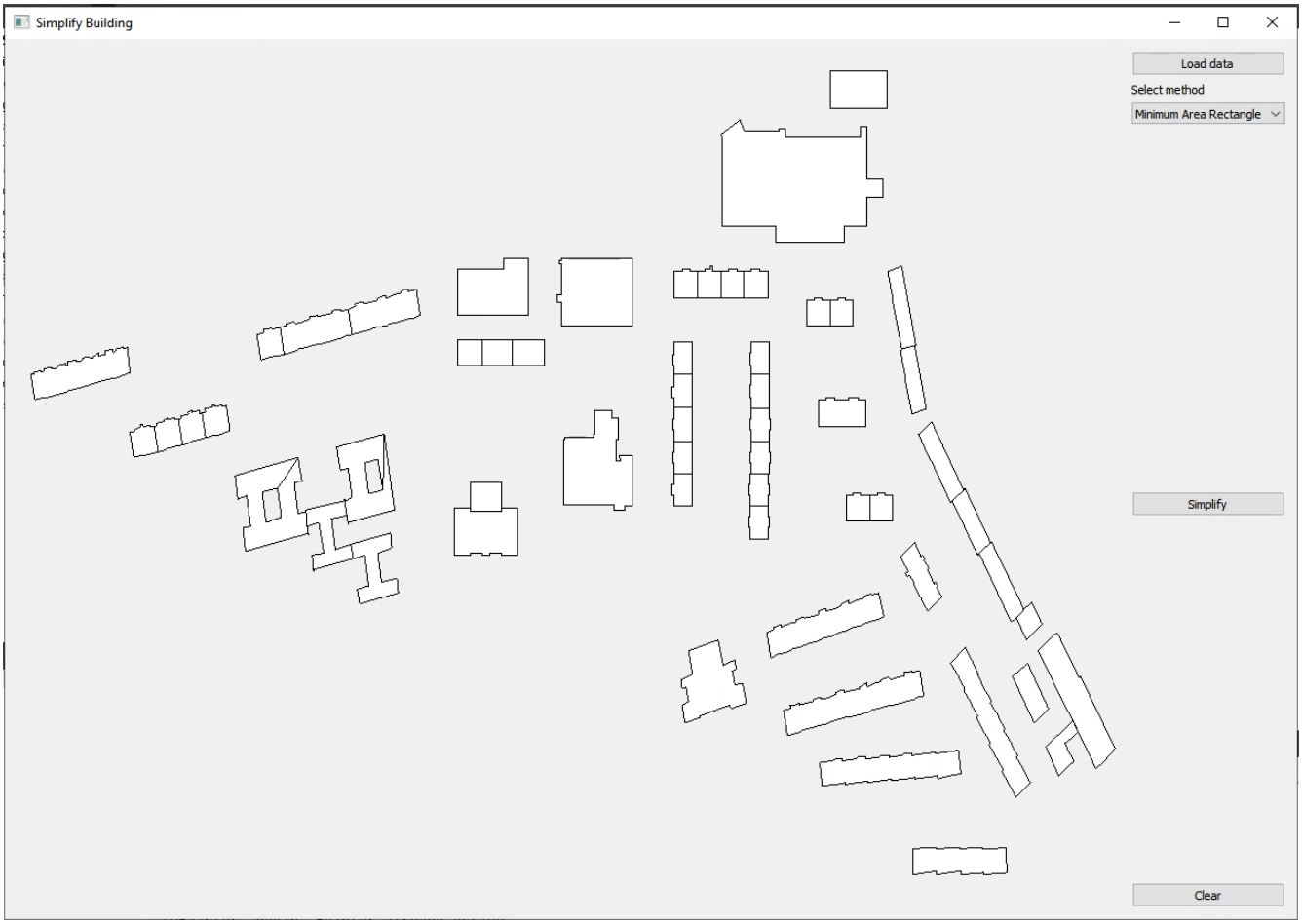
6 Vstupní a výstupní data

Pomocí tlačítka Load data lze do aplikace nahrát soubor se vstupními daty, po kliknutí na něj se objeví dialog pro výběr souboru. Aplikace na vstupu očekává soubor ve formátu csv obsahující geometrie polygonů ve formátu WKT. Vzorová vstupní data byla získána z programu QGIS pomocí exportu dat do WKT, jedná se o polygony stavebních objektů z datové sady ZABAGED. Jako vzorová data jsou přiloženy 3 csv soubory představující 3 druhy městské zástavby - sídliště, centrum města, vilovou čtvrť. Souřadnice polygonů jsou v S-JTSK.

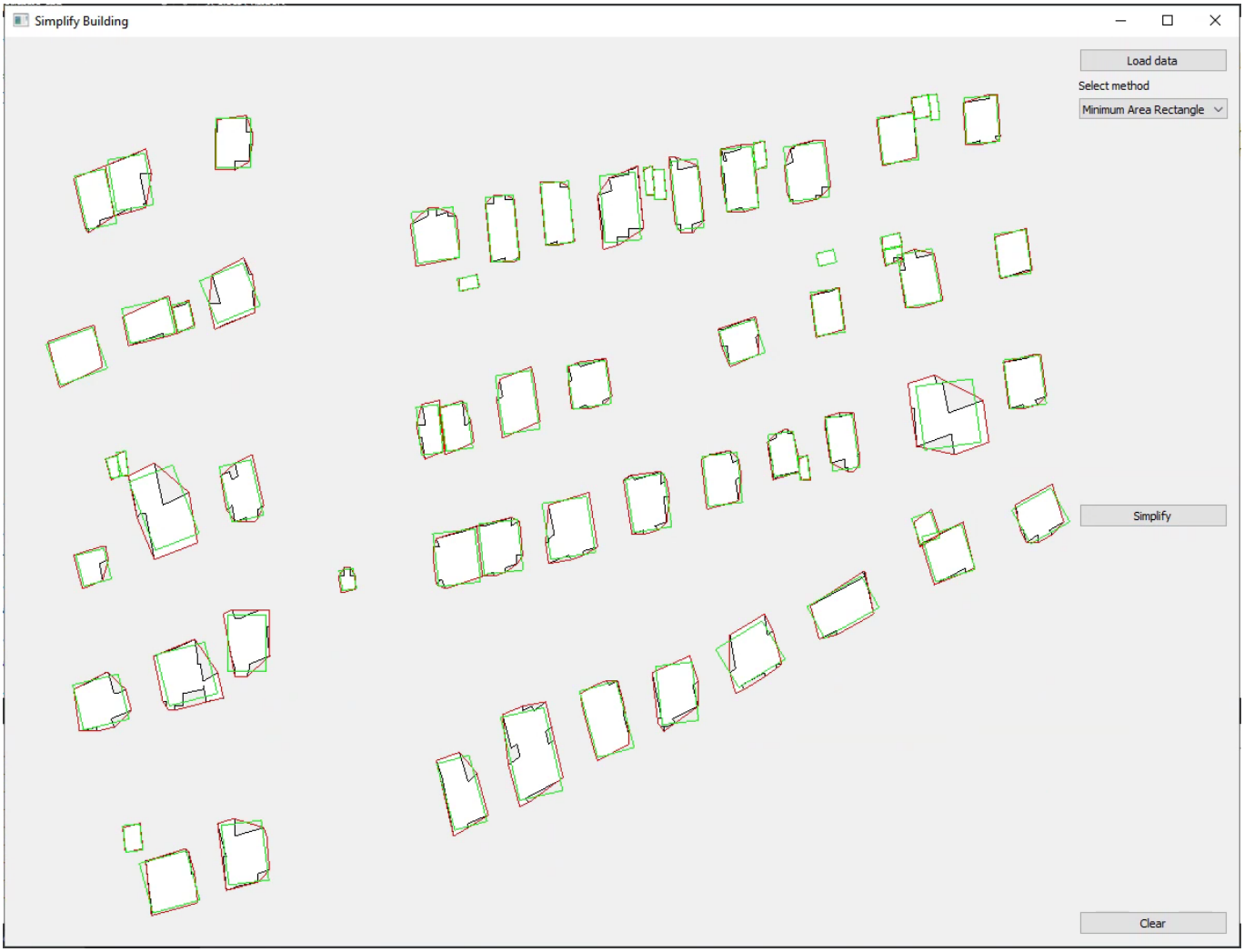
1	MultiPolygon (((744339.66 1051550.48, 744341.28 1051556.27, 744343.09 1051562.74, 744367.14 1051556.01, 744366.34 1051552.8,
2	MultiPolygon (((744310.9 1051397.07, 744328.65 1051423.25, 744338.4 1051437.64, 744338.79 1051437.35, 744355.76 1051462.27, 7
3	MultiPolygon (((744316.05 1051450.93, 744326.6 1051466.69, 744338.46 1051458.83, 744334.87 1051453.47, 744331.72 1051448.77,

Obrázek 6.1: Formát vstupních dat

Výstupem aplikace je grafické znázornění nahraných objektů (polygonů) a jejich následná generalizace zvolenou metodou.



Obrázek 6.2: Načtená data před generalizací



Obrázek 6.3: Generalizace metodou Minimum Area Enclosing Rectangle

Obrázek 6.4: Generalizace metodou Longest Edge

Obrázek 6.5: Generalizace metodou Wall Average

7 Zhodnocení generalizačních metod

Použité generalizační metody byly hodnoceny na 3 vzorových datových sadách. Charakter zástavby centra města a vilové čtvrti nejlépe zjednodušuje metoda Wall Average, podobné výsledky dává i metoda Minimum Area Enclosing Rectangle. Sídliště je nejlépe generalizováno metodou Minimum Area Enclosing Rectangle, Wall Average poskytuje také dobré výsledky. Naopak metoda Longest Edge není vhodná ani pro jeden typ zástavby, pro některé samostatné budovy (např. ze sady sídliště) ale může poskytnout uspokojivé výsledky.

Generalizační metoda	Procentuální úspěšnost metody na testovaných datech
Minimum Area Enclosing Rectangle	98%
Wall Average	96%
Longest Edge	90%

Tabulka 7.1: Sídliště

Generalizační metoda	Procentuální úspěšnost metody na testovaných datech
Wall Average	97%
Minimum Area Enclosing Rectangle	95%
Longest Edge	85%

Tabulka 7.2: Centrum města

Generalizační metoda	Procentuální úspěšnost metody na testovaných datech
Wall Average	96%
Minimum Area Enclosing Rectangle	92%
Longest Edge	85%

Tabulka 7.3: Vilová čtvrť

8 Dokumentace

8.1 Třída Algorithms

- $int\ getPointLinePosition(QPointF\ \&p1, QPointF\ \&p2, QPointF\ \&q)$
  - analyzuje vzájemný vztah bodu a přímky
- $double\ getTwoLinesAngle(QPointF\ \&p1, QPointF\ \&p2, QPointF\ \&p3, QPointF\ \&p4)$
  - vypočte úhel mezi dvěma vektory
- $QPolygonFcreateCH(vector < QPointF > \&points)$
  - vypočte konvexní obálku polygonu
- $vector < QPointF > rotate(vector < QPointF > \&points,\ double\ sig)$
  - otočí množinu bodů o úhel
- $double\ getArea(vector < QPointF > \&points)$
  - vypočte plochu polygonu
- $tuple < std :: vector < QPointF > minMaxBox(vector < QPointF > \&points)$
  - vytvoří minimální ohraničující obdélník množiny bodů
- $QPolygonF\ minAreaEnclosingRectangle(vector < QPointF > \&points)$
  - vytvoří ohraničující obdélník množiny bodů s minimální plochou
- $vector < QPointF > resizeRectangle(vector < QPointF > \&points,$ $vector < QPointF > \&er)$
  - změní velikost obdélníku
- $QPolygonF\ wallAverage(ector < QPointF > \&points)$
  - vytvoří ohraničující obdélník množiny bodů metodou Wall Average
- $QPolygonF\ longestEdge(vector < QPointF > \&points)$
  - vytvoří ohraničující obdélník množiny bodů metodou Longest Edge

8.2 Třída CSV

- $vector\ < QPolygonF > read\_CSV(string\ \&filename,\ double\ \&xmin,\ double\ \&xmax,$ $double\ \&ymin,\ double\ \&ymax,\ )$
  - načte vstupní csv soubor

8.3 Třída Draw

- $void\ mousePressEvent(QMouseEvent * event)$
  - vrátí souřadnice kurzoru po kliknutí na Canvas
- $void\ paintEvent(QPaintEvent * event)$
  - vykreslí polygony na Canvas
- $void\ clearAll(){}$
  - vyčistí Canvas
- $void\ setCH(QPolygonF\ \&ch){chs.push\_back(ch\_);}$
  - vrátí konvexní obálku
- $void\ setMAER(QPolygon\ \&er){ers.push\_back(er\_);}$
  - vrátí minimální ohraničující obdélník

- $void\ clearCH(){chs.clear();}$
  - smaže položky vektoru konvexních obálek
- $void\ clearMAER(){ers.clear();}$
  - smaže položky vektoru minimálních ohraničujících obdélníků
- $vector < QPointF >\ getPoints(){return\ points;}$
  - získá souřadnice načtených polygonů
- $vector < QPointF >\ getPolygons(){return\ polygons();}$
  - získá souřadnice načtených polygonů
- $void\ drawPolygons(std :: vector < QPolygonF >\ \&polygons,\ double\ \&xtrans,\ double\ \&ytrans,\ double\ \&xratio,\ double\ \&y_ratio)$
  - vykreslí polygony na Canvas
- $QPolygonF\ transformPolygon(QPolygonF\ \&polygon,\ double\&xtrans,\ double\&ytrans,\ double\ \&xratio,\ double\ \&yratio)$
  - transformuje polygon min-max boxu

8.4 Třída Mainform

- $void\ on\_load\_data\_clicked()$
  - otevře průzkumníka souborů
- $void\ on\_simplify\_clicked()$
  - provede vybranou metodu
- $void\ on\_\_clear\_clicked()$
  - vyčistí Canvas
- $void\ simplifyBuildings(vector < QPointF > \&points)$
  - provede vybranou metodu

8.5 Třída SortPointsByX

- $bool\ operator\ ()\ (QPointF\ \&p,\ QPointF\&q)$
  - seřadí body podle velikosti souřadnice X

8.6 Třída SortPointsByY

- $bool\ operator\ ()\ (QPointF\ \&p,\ QPointF\&q)$
  - seřadí body podle velikosti souřadnice Y

9 Závěr

Byla vytvořena aplikace Building Simplify s grafickým rozhraním. Aplikace byla napsána v programovacím jazyce C++ a umožňuje generalizaci nahraných polygonů vybranou metodou. Ke generalizaci je možné vybrat jednu z metod Minimum Area Enclosing Rectangle, Wall Average a Longest Edge.

Literatura

[1] Environmental Systems Research Institute, Inc., *Generalizing polygon coverage data*, https://desktop.arcgis.com/en/arcmap/10.3/tools/coverage-toolbox/generalizing-polygon-coverage-data.htm, [23.11.2022].

[2] Tomáš Bayer, *Konvexní obálka množiny bodů*, https://agony.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf, [28.10.2022].