

**Jauna mākslas vēstures  
pētniecības metode Latvijas  
17.–18. gadsimta tēlniecības  
autorības noteikšanā**

**Sanita Bitko**

2026



# Kāpēc ir nepieciešama jauna metode?

01

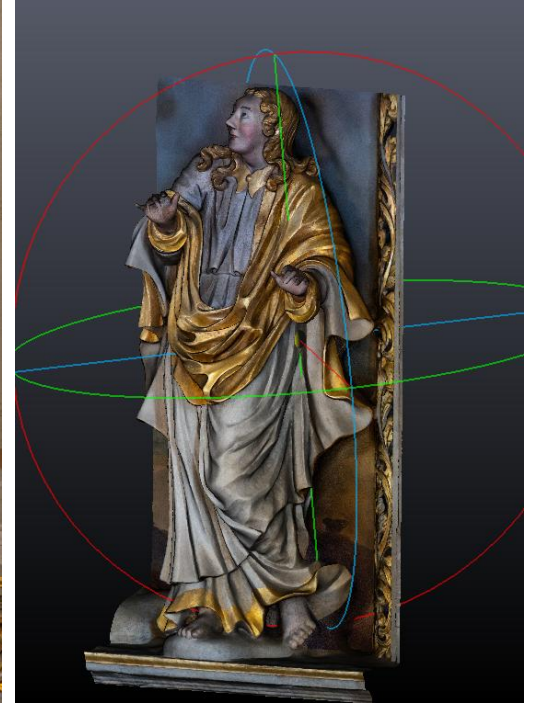
Ja nav dokumentāro avotu, lai cik līdzīgas būtu skulptūras, tās paliek tikai varbūtības līmenī

02

Ar esošo formālo, stilistisko, ikonogrāfisko analīzi iespējams noteikt tikai stilistiskās līdzības un ietekmes

03

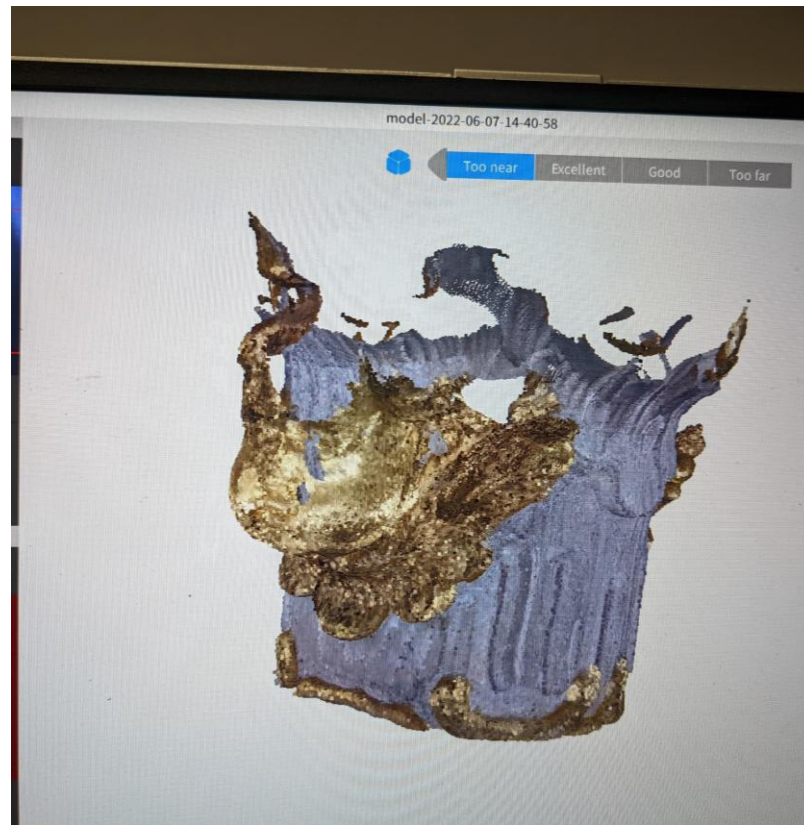
Pat uzrakstītais ir jāvērtē uzmanīgi – kurš un kad rakstījis!



---

## 04 Tehnoloģisko rīku attīstība

Lestenes baznīcas interjera attēlu piemēri



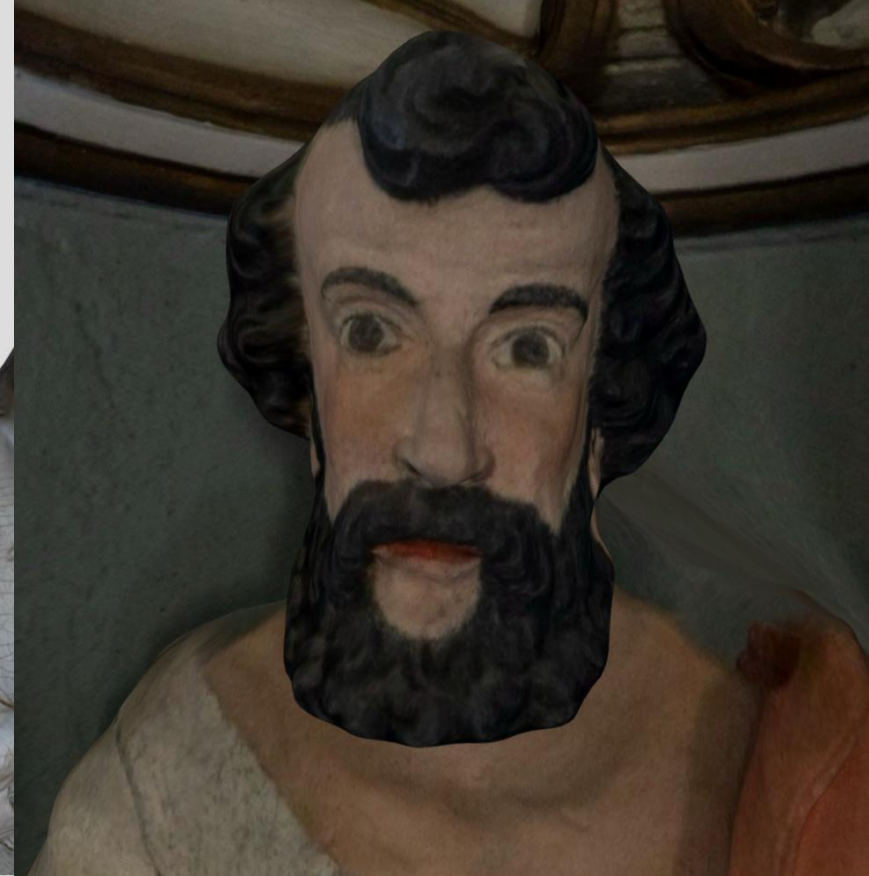
Idejas pirmsākumi – 2022. gads  
«uztrenēju aci»



Salgales baznīcas skulptūra



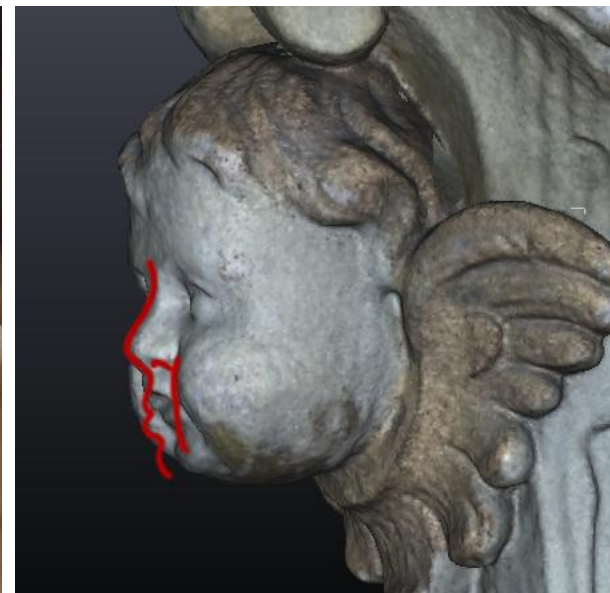
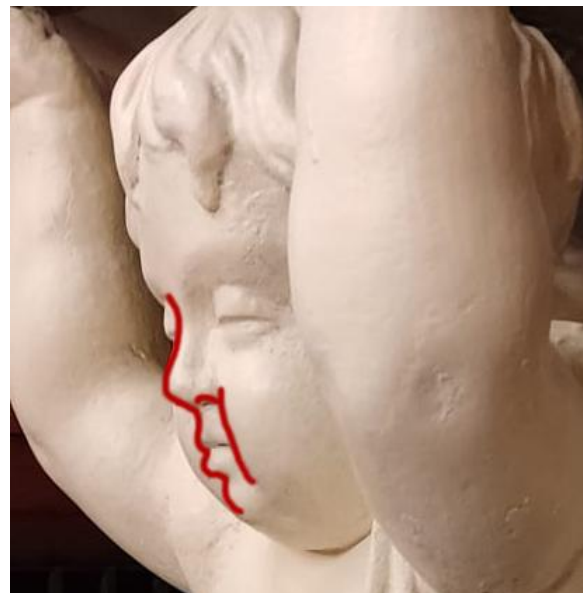
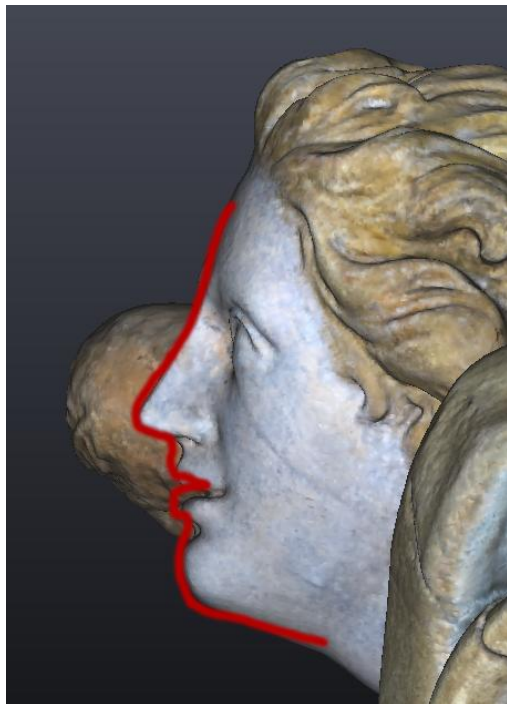
Dzirciema baznīcas skulptūra



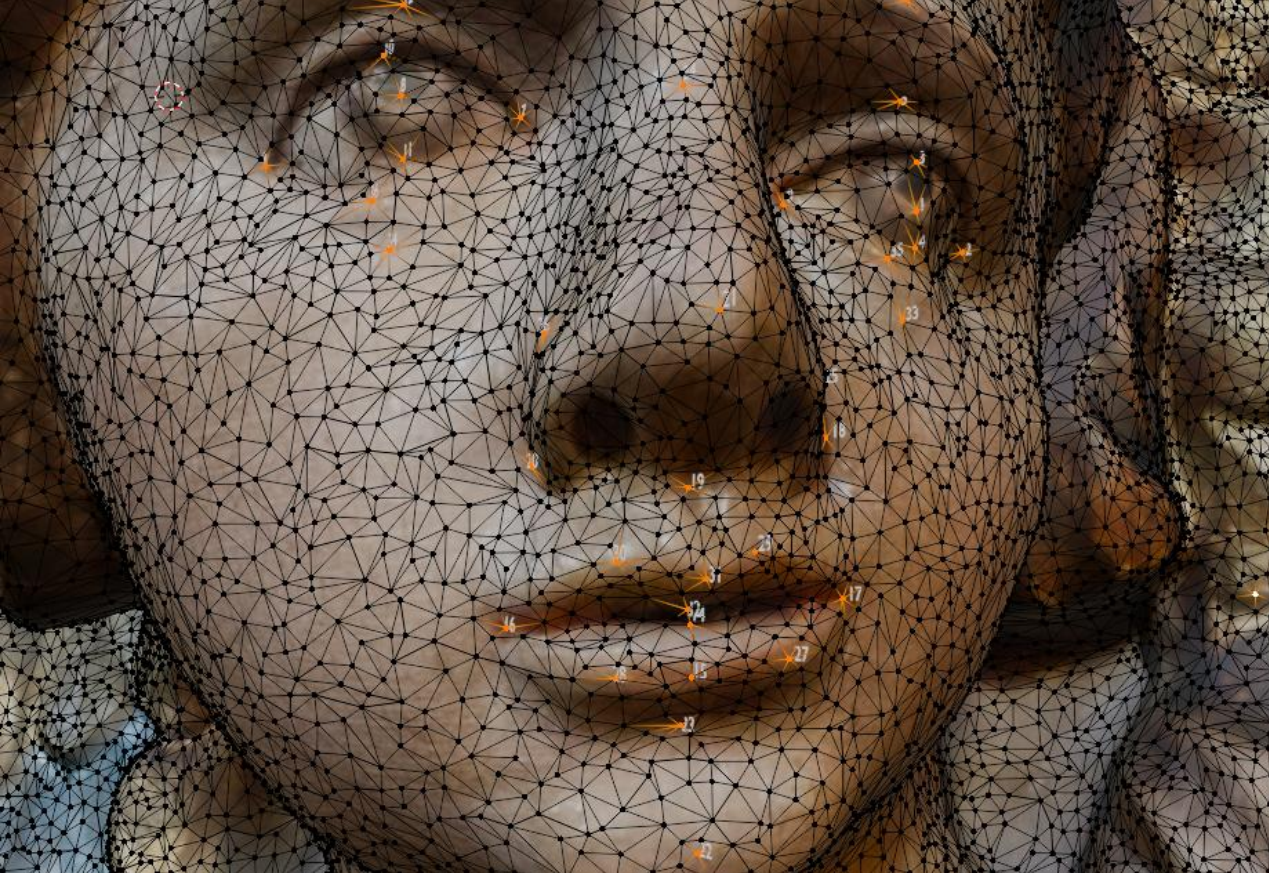
Alsungas baznīcas skulptūra

---

05 Subjektīvs vērtējums - vai ir viens autors vai nav?



Katrs redz un uztver savādāk



Risinājums – jauna mākslas vēstures  
pētniecības metode:

1. balstīta sejas formas analīzē (3D modeļi)
2. eksaktajās zinātnēs

```
all_to_all_simetrisks_v1.py - D:\LMA\Magistrs\Metode_Ai\Pitons\dati4\all_to_all_simetrisks_v1.py (3.13.5)
File Edit Format Run Options Window Help

"""
from pathlib import Path
import pandas as pd
import numpy as np
import sys
import re
from typing import List, Dict, Tuple

# ==== Ceļi ====
rezultati_path = Path(r"D:\LMA\Magistrs\Metode_Ai\Pitons\dati4\rezultati.csv")

# ==== Parametri ====
point_mode = "youden"
classify_threshold = None
min_group_size = 2
validate_prefix = None
EPS = 1e-12 # tikai skaitliskai stabilitātei

# ==== Palīgfunkcijas ====
def load_results(path: Path) -> pd.DataFrame:
    if not path.exists():
        print(f"ERROR: file not found: {path}", file=sys.stderr); sys.exit(1)
    df = pd.read_csv(path)
    df.columns = [str(c) for c in df.columns]
    name_col = df.columns[0]
    df[name_col] = df[name_col].astype(str)
    df.set_index(name_col, inplace=True)
    return df

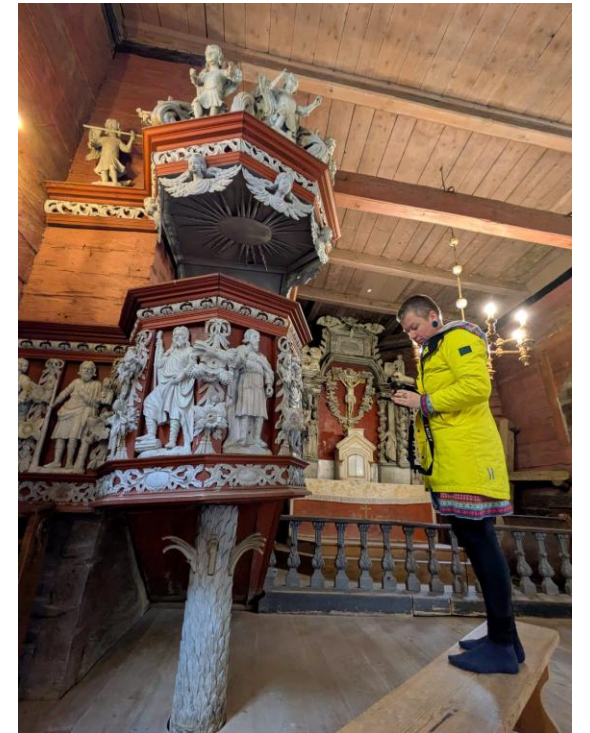
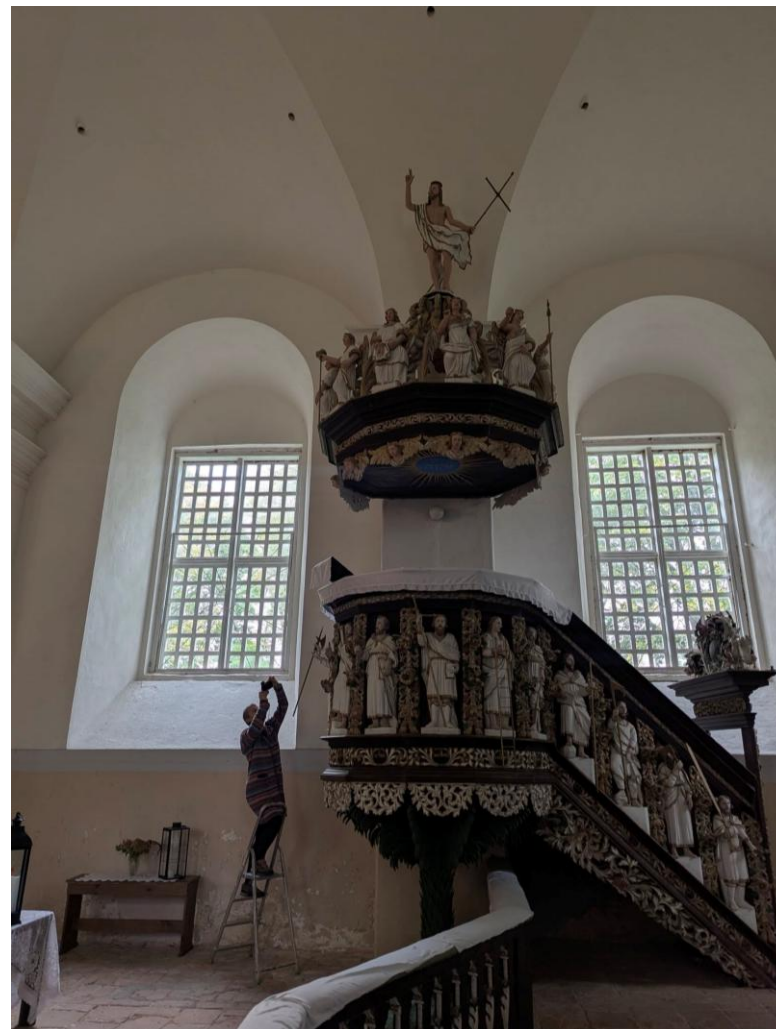
def detect_prefix(id_str: str) -> str:
    m = re.match(r"^[A-Za-z0-9]+", str(id_str))
    return m.group(1) if m else ""

def build_groups(ids) -> Dict[str, list]:
    buckets: Dict[str, list] = {}
    for i in ids:
        pref = detect_prefix(i)
        if not pref:
            pref = ""
        buckets.setdefault(pref, []).append(i)
    return buckets

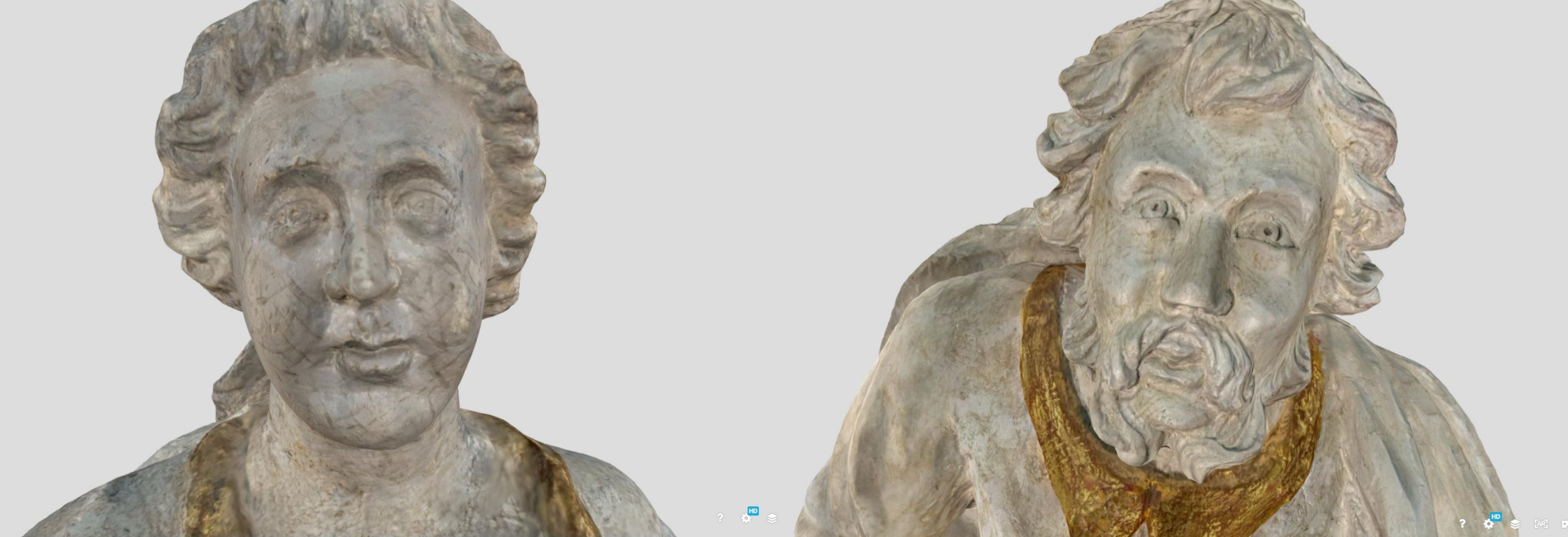
def _normalize_prefixes(x) -> List[str]:
    if x is None:
        return []
    if isinstance(x, (list, tuple, set)):
        prefs = [str(p).strip() for p in x]
    else:
        tmp = str(x).replace("|", ",")
        prefs = [p.strip() for p in tmp.split(",")]
    return [p for p in prefs if p]

# ==== Skaitliski stabila logistiskā funkcija ====
def stable_sigmoid(z: float) -> float:
    # izmanto tanh formu, lai izvairītos no overflow exp()
    return float(0.5 * (1.0 + np.tanh(0.5 * z)))

# ==== Īsākā loga J (savienātais) ====
```



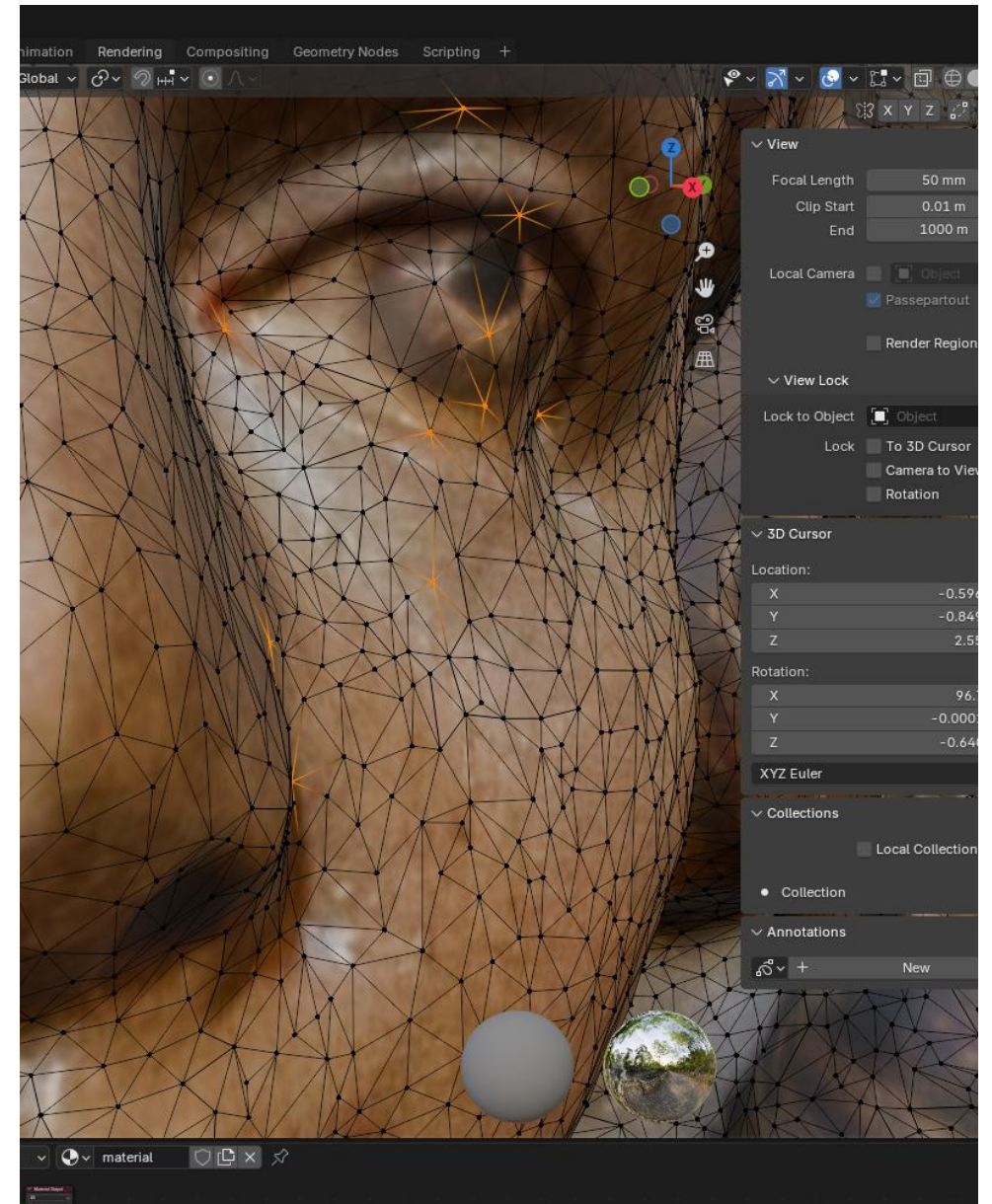
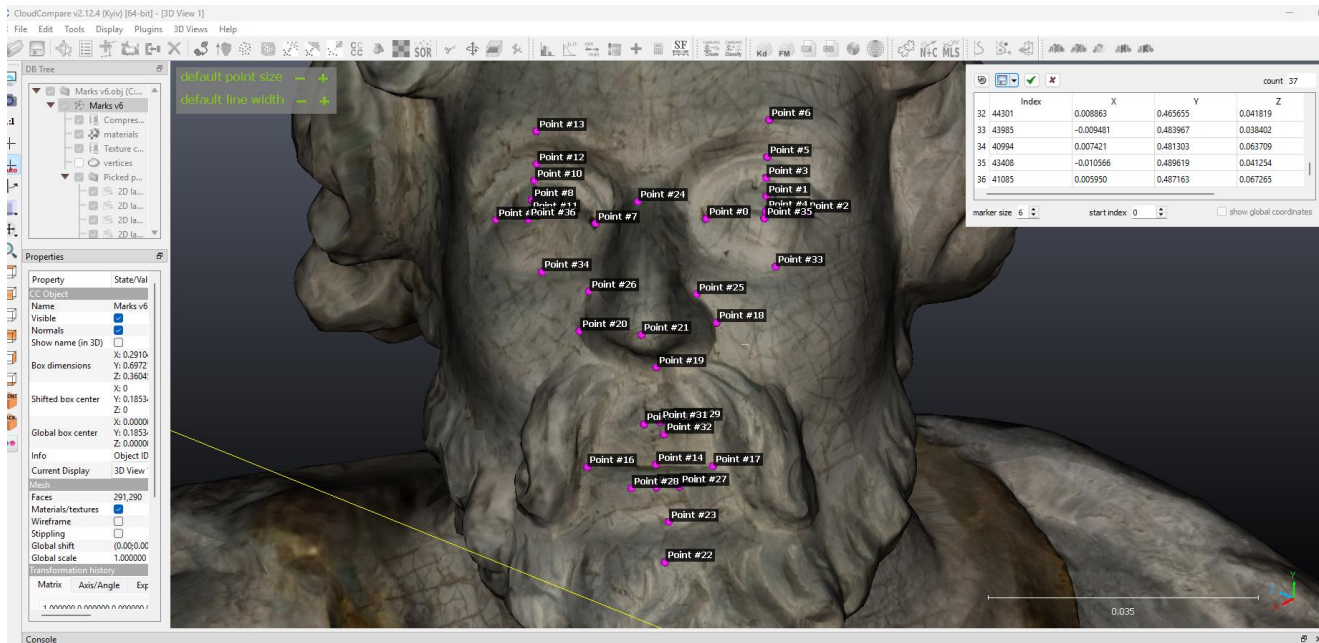
3D modeļu izveidošana  
ar fotogrammetriju



Salgales baznīcas skulptūras – autors Johans Mertenss

---

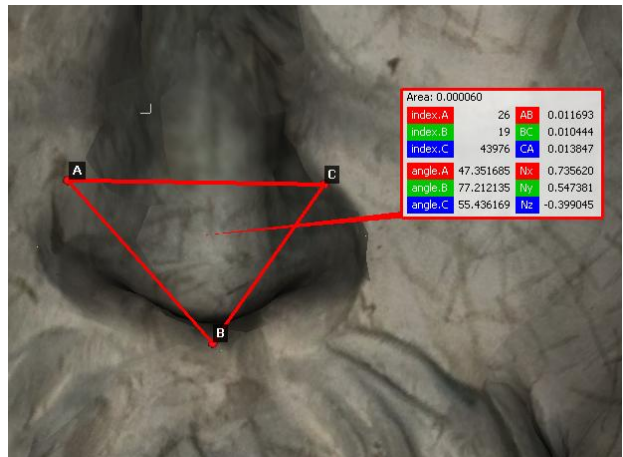
Esošie atpazīšanas algoritmi nav piemērojami kokaēlniecībā



## Metodes darbības principi

- Uz katra 3D modeļa tiek marķēti 37 punkti – raksturlielumi
- Punktu izvēle balstīta empīriskajos novērojumos un veiktajos eksperimentos (matu līnija, ausis)
- Fiksēts punktu izvietojums nodrošina metodes atkārtamību
- Atkārtamība nodrošina rezultātu pārbaudāmību un zinātnisku ticamību.

0,0. 166083440185,0. 573711574078,-0. 288570834875  
 1,0. 173510074615,0. 576229155064,-0. 274862051801  
 2,0. 179344311357,0. 570711791515,-0. 269996523857  
 3,0. 175209656358,0. 576993048191,-0. 27216739645  
 4,0. 171658426523,0. 574142217636,-0. 272129178047  
 5,0. 175976872444,0. 579373061657,-0. 278924018145  
 6,0. 178358510137,0. 583242893219,-0. 281581103808  
 7,0. 150408998132,0. 57121771574,-0. 293353825808  
 8,0. 142933443189,0. 572646677494,-0. 297828406096  
 9,0. 136311680079,0. 565291702747,-0. 303802192211  
 10,0. 144442722201,0. 573115706444,-0. 301074594259  
 11,0. 140683889389,0. 571041464806,-0. 29555118084  
 12,0. 145455390215,0. 575601041317,-0. 303259640932  
 13,0. 149328738451,0. 581399559975,-0. 306645512581  
 14,0. 13666107235,0. 573389232159,-0. 25747987628  
 15,0. 133831530809,0. 572333455086,-0. 254628956318  
 16,0. 127966552973,0. 568265259266,-0. 267574459314  
 17,0. 148061320186,0. 57073199749,-0. 251802682877  
 18,0. 154107168317,0. 575488984585,-0. 262955844402  
 19,0. 141975939274,0. 579196333885,-0. 264769881964  
 20,0. 136662587523,0. 573963313961,-0. 276402324438  
 21,0. 146458879113,0. 594517767429,-0. 270139336586  
 22,0. 127446711063,0. 565071642399,-0. 246557697654  
 23,0. 133365318179,0. 569230675999,-0. 253004504871  
 24,0. 159445002675,0. 585736453333,-0. 286990252405  
 25,0. 155777171254,0. 578002870083,-0. 266340494457  
 26,0. 139971643685,0. 576086256081,-0. 27923566103  
 27,0. 137958496809,0. 57272952795,-0. 251809567213  
 28,0. 13136407733,0. 572871135044,-0. 257000287901  
 29,0. 142217755318,0. 578869044781,-0. 257931679487  
 30,0. 136390835047,0. 578434586525,-0. 262506783009  
 31,0. 138363867998,0. 578829586596,-0. 259591382542  
 32,0. 138034939766,0. 575072050095,-0. 258691132069  
 33,0. 166969612241,0. 573578298092,-0. 266962558031  
 34,0. 135922789574,0. 570175766945,-0. 290053427219  
 35,0. 171029701829,0. 57384043932,-0. 270732998848  
 36,0. 1394405514,0. 57056593895,-0. 294775128365



```

***
from pathlib import Path
import pandas as pd
import numpy as np
import sys
import os
from typing import List, Dict, Tuple

# *** Cell ***
resultati_path = Path("D:\10\Magistra\Irodna_1\Python\data\resultati.csv")

# *** Parametri ***
point_name = "puzinads"
point_mode = "puzinads"
classif_y_threshold = "None"
min_group_size = 2
validata_prefix = "None"
EPS = 1e-10 # tīksma stabilitātes stabilitātes

# *** Paliģināšana ***
def load_result(path):
    df = pd.DataFrame()
    if not path.exists():
        print(f"File not found: {path}, file may not exist")
        return df
    df = pd.read_csv(path)
    df.columns = [col for col in df.columns
                  if not col.startswith('df')]
    df.name_col = df.columns[0]
    df.set_index(name_col, inplace=True)
    return df

def detect_point_id(point):
    m = re.match("^(?P<id>[a-z0-9]+)$", point)
    return m.group(1) if m else ""

def build_group(ids):
    dict = {}
    buckets = dict.fromkeys(ids, [])
    for id in ids:
        pref = detect_point_id(id)
        if not pref:
            buckets.setdefault(pref, []).append(id)
    return buckets

def _normalize_prefix(ids):
    list = []
    return list

if len(ids) > 1:
    list = list(ids)
    if len(list) > 1:
        list = list(ids)
    else:
        list = [ids]
    pref = [id.split('.')[0] for id in list]
    return [id for id in list if id.startswith(pref[0])]

# *** Stabilitātes stabilitātes funkcija ***
def stable_group(ids):
    return list(ids)

# Amatsloka saraksts, kas ievadīti no overflow map()
return list(ids * (1.0 + np.tanh(0.5 * x)))

```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	Column4	Column5	Column10	Column11	Column12	Column13	Column14	Column15	Column16	Column17	Column18	Column19	Column20	Column21
	Lagža Leņķis	Maiņvirziena acim	Deguna leņķis	Uzaciņķis	111-Ac maiņvirziena	apaļlāpņu leņķis	note vs acis							
969154	150.13317425799146	152.4544708207332	139.5797170997643	53.72426397042323	150.16220115890115	70.9440409970562	2.52524273							
969155	147.872402108978	150.132405030176	147.50931350307787	54.9204704334344	152.00947716507105	153.2109101202924	2.17812454							
649996	145.459240558749	143.5372584448184	138.6665110450312	52.175245237922	125.777963485303	147.025610515746	2.03879153							
609676	152.677025330026	149.1915004272407	133.10278663627617	52.117890881518	128.67115535393634	151.1408020208135	2.56007250							
607674	154.45627957310113	150.47036901227707	146.1804410473098	51.80264209740205	145.4015274700147	152.959604147711	2.01539172							
213027	144.6784080601004	156.3218002891976	133.3334257789818	50.0505008180098	138.5426915242005	151.4728073214023	2.03847373							
41581	150.2953708179823	148.5629608674205	126.4955219050056	54.7387850973978	146.7634813557483	156.568421203852	2.40139704							
979302	158.7984880228335	133.1286581312037	142.8232364240078	50.764451960889599	124.18574970324185	154.485429738387	2.068972199							
102915	150.7505162062707	138.7796204809712	132.7138094967876	63.6128312204699	142.389516281418	147.62161417023015	2.40001483							
921495	138.4240068081183	148.1872227751798	140.0102689719348	46.2046490869078	138.12248240170114	155.9014702879153	2.112035350							
287555	138.371524512387	150.24384621039715	129.64097974933313	60.34344448879623	148.91488279562	152.30726874866	2.02107044							
697715	164.0573537127556	184.9712018048057	163.0270806129917	55.7683622196231	162.8713517054499	141.3887850921346	2.068363648							
822671	172.506262811108	155.506039180071	154.4091867184707	53.2620189712137	156.832781505118	142.670084119621	2.11254537							
433995	127.3894406422276	133.8079514507957	134.5747342974536	51.8644079427425	156.2480968703744	143.4033105995945	2.43801077							
22877	152.8791628519646	167.7953681110488	144.21895832620786	48.39112531824025	143.08038537223054	151.23126063353324	2.23944931							
61378	154.3085308115045	165.2391558712693	144.4471788462628	68.30610820003	153.53485629874793	152.733224851394	2.362350632							
95332	165.0223674423339	169.8347007920449	134.03096496201	67.4027449370205	163.6781389961189	141.89971831445	2.091117962							
18187	156.190680683894	169.8268820270436	143.74542868868141	64.0430400267837	163.6830013335976	151.4936890597582	1.87702082							
862187	154.6884260830344	143.788920001904	146.54845476263063	56.0419889926991	138.4082812628504	149.9367183610443	2.679873098							
8126	146.0087701091034	155.54584865221753	143.516426272806	53.52279593142933	152.9964557341086	142.2248733846428	2.369882896							
25982	144.220616221747	154.5684408062037	138.664516720749	64.5485997693114	156.5847638644574	150.4020898090597	2.30068987							
63918	162.3625886887465	142.87683120792134	130.9608146560478	54.4224670204224	147.3773501010542	150.7411679599515	2.42372868							
34143	163.4412918832578	135.8825100472373	131.77100832701307	53.9188348091648	148.81261036137712	150.9079211740732	2.502810005							
93168	159.4201748881816	127.14380624154849	135.48533084867182	57.7370873197896	150.10840091829	147.7448181833032	2.76207028							
59129	144.1699370041159	160.5884372482114	137.9482792070093	52.0204140206071	148.86914026110244	150.85149510772114	2.76134091							
917907	149.18242489827	157.989132621291	145.762629291238	59.8509607839584	154.4882236999112	148.4898732296378	2.44891181							
105047	163.5986997859063	163.5212267147454	142.3420780107	64.2420780107	142.40415320780107	151.44712207772	2.536341462							
23976	156.7271859679034	88.6888815224415	132.154271569854	57.0964259959881	148.9874745033706	156.020571075645	2.26543396							
72926	128.9731674540314	153.15768977459	129.2750468386305	51.815338452136	153.0001879616782	128.588462063501	2.178922028							
115919	141.8596744020064	186.19697146632934	141.2127879764893	55.05280752310996	156.8358509111032	132.95248984644	2.178922028							
57085	147.1325052399449	149.5109699707908	138.2307023765262	65.1961597163076	151.87344236976	149.9748462200348	2.70434587							
87242	167.8718484951478	156.504174964239	133.18430514631604	43.223202324657	145.779845210605	154.83127145839902	2.71323253							
102413	161.4418789112096	129.2300714087706	133.5719518312958	53.5569833028671	144.8520687019745	151.914215145238032	2.52965168							
697283	143.1848484868977	147.14845454901795	145.1422712062391	60.8434139738744	145.49612606391999	148.9842196629767	3.107172489							
311698	142.820015205204	145.9876191841474	139.122554480091	64.276202874538	154.6370424042645	158.3629447254	2.19397210							
141868	146.454545082298	160.8913686909725	136.78822863883897	55.3971185240546	152.6144707891707	152.4400040477833	1.041127629							
159687	137.3984644126377	186.1811043623227	139.5097745070772	56.4602058922733	155.9682568381254	148.352158863946	1.991263544							
376047	168.8370313700323	164.402230871242	147.8691878969894	56.9158044004969	163.9147718103075	142.297226287121	1.49090602							
106007	133.7150740844254	155.5676304688403	142.596859990442	60.5382389001225	153.537038966263	138.974215831408	2.142740547							
810882	161.987246886184	133.43049897295349	129.5240297614927	54.5608289172795	138.8100712827007	134.9071033469172	2.09427829							
88814	131.085020849585	169.927714570778	133.57619519474283	57.842779014765076	153.7453614862108	136.3815808322075	2.257133081							
24244	146.4181336831843	166.8417845430307	130.1052444870247	67.8474112928613	163.90303131346424	148.410146297876	2.344437792							

# Matemātiskie aprēķini

- Starp 37 atskaites punktiem tiek aprēķināti leņķi un sejas elementu garuma–platuma attiecības = indeksi
- Normalizēti indeksi nodrošina no mēroga neatkarīgu salīdzināšanu
- Rezultātā šobrīd iegūti 17 kvantitatīvi indeksi
- Aprēķini tiek veikti Python programmēšanas valodā

# [01] Salīdzināšana

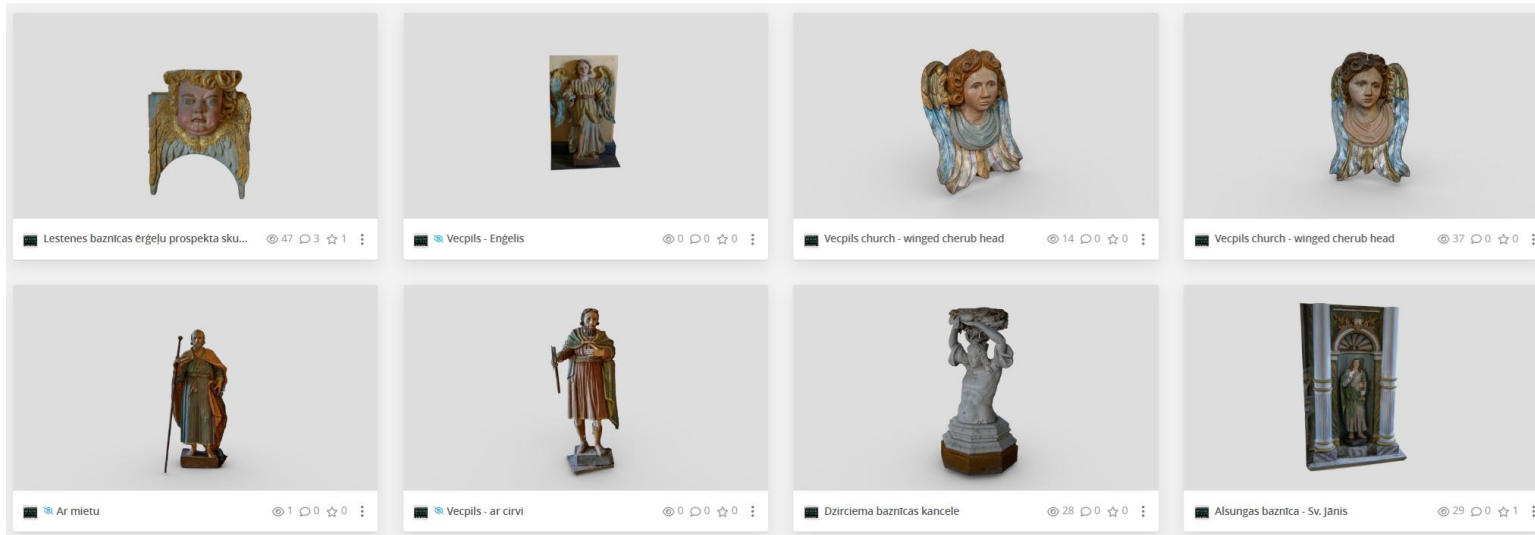
## Bāzes grupa

Salgalas baznīcas skulptūru grupa. Autors Johans Mertenss

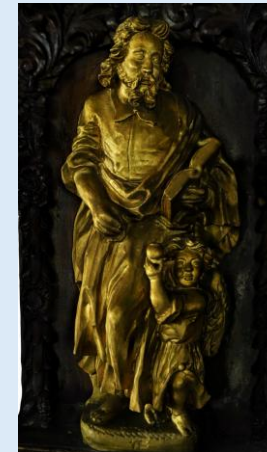


## Svešie - ārpus grupas skulptūras

Citas skulptūras, kas nav bāzes grupā un kandidātu grupā



**Kandidātu grupa - Kandidāts**  
Kandavas baznīcas skulptūra.



- Skulptūras netiek salīdzinātas viena pret vienu
- Kandidāta skulptūra tiek salīdzināta pret bāzes grupu.
- **Vai Kandavas skulptūrai ir tas pats autors, kas Salgalas skulptūrām**

Column1	Column2
1-Matejs-salgale	123.74912633378545
1-Janis-salgale	128.87007712140033
99-Kandava-marks	129.96957312590033
1-Damite-Ljubestiba	131.71941180448346
1-Salgale-ticiba2	134.3251524744269
blender-engela_galvina_1_blender	138.06418528896145
1-Mozus	138.66723399495623
1-juris-salgale	138.95992834735537
99-Kandava-kancele	141.42566273927378
99-Kandava-Lukas	141.80043297880758
99-Kandava-ceriba	143.3670761715032
d-Dzirciems	145.08994514250952
1-chuvaks	146.7362698356514
1-kirstus	149.20433170511865
1-kancele	150.36785565685216
1-Ticiba	150.45447682961293
1-marks	151.39736057956645
99-Kandava-matejs	151.46483850148957
99-Kandava-milestiba	153.13710607185783
V-engelis	154.1135670512075
stel-armietu	154.8702867966772
2-Bikeru_vaze_restored2	154.9569195630661
piltene	155.00197080719
99-Kandava-ticiba	157.38458512390463
V-engelis2	160.22265363845645
99-Kandava-janis	160.94401880834198
sub-Andrejs	161.63653294529578
vecpiebalga	162.16452173817555
2-Bikeru_vaze_restored	164.33195816120065
3-jekaba_kungs1	165.41363547611314
Ergelu_engelis	166.74177214440226
A-janis	167.1881481093431
98-sparnotagalva	167.508535276098
Cloud_sompare_v-galva-1	168.06953154967152
V-ar-cirvi	168.72577602236498
V-engelu_galvina_2	169.856205744173

## [02] Jūdena (Youden's J) statistika

### Cik katrs indekss ir informatīvs?

Katram meistaram ir savs stils, tādēļ katram meistaram būs savi informatīvie indeksi.

Cik bieži kandidāta rādītājs trāpa bāzes grupā?

Cik bieži tas parādās svešinieku grupā?

### Piemēram rādītājs «deguna leņķis»

Bāzes grupā ir 5 skulptūras

-> 4 no tām rādītājs trāpa diapazonā

Svešinieku grupā ir 20 skulptūras

-> tikai 2 no tām iekrīt tajā pašā diapazonā

Secinājums:

Indekss ir raksturīga bāzes grupai,

bet neraksturīga svešajiem.

$$TPR_i = \frac{\#(B_i \cap [a_i, b_i])}{|B_i|}$$

$$FPR_i = \frac{\#(O_i \cap [a_i, b_i])}{|O_i|}$$

$$J_i = TPR_i - FPR_i$$

$$w_i = \max(J_i, 0)$$

## [03] Logaritmiskās ticamības attiecība (Log-Likelihood Ratio LLR) ar Gausa sadalījumu (Gaussian splatting)

**Gausa sadalījums – nosaka cik raksturīga ir šī indeksa vērtība katrā grupā.**

Vai šī vērtība atrodas tuvu bāzes grupas tipiskajām vērtībām ( $\mu$ )?

«Cik raksturīga šī vērtība ir bāzes grupā?»

«Cik raksturīga šī vērtība ir svešinieku grupā?»

Katram rādītājam tiek iedota «atzīme».

**LLR – salīdzina** vai indeksa vērtība izskatās pēc bāzes grupas vai svešinieku grupas?

Ja kandidāta vērtība:

(\*) bieži sastopama bāzes grupā, bet reti sastopama svešinieku grupā

-> stiprs arguments par labu bāzes grupai (LLR > 0)

Ja ir otrādi:

(\*) bieži sastopama svešinieku grupā, bet reti bāzes grupā

-> arguments pret piederību bāzes grupai (LLR < 0)

Salīdzina divas rādītāja vērtības.

$$LLR = \log \frac{P(x | base)}{P(x | out)} = \log \frac{P(x | \mu_{base}, \sigma_{base})}{P(x | \mu_{out}, \sigma_{out})}$$

**LLR nebalso – jā / nē,  
bet pieļauj robežgadījumus  
LLR dod katra rādītāja svaru,  
kas beigās tiek sasummēts**

## [03] Ja vērtība būtu nejauša, vai es to biežāk ieraudzītu svešinieku vai bāzes grupā?

Bāzes grupā acu proporcijas rādītājs:  
vairums darbu: 32 – 34  
dažos gadījumos: 31 vai 35  
ārpus tā gandrīz nekad

Kandidātam:  
acu attālums: 33

**LLR = šajā gadījumā ir pozitīvs ->**  
«par labu bāzes grupai»

Kandidāta deguna izvirzījuma proporcija: 7  
Bāzē: parasti 6 – 7  
Svešiem: ļoti bieži 6 – 8  
Vērtība ir “vispārīga”.  
**LLR = gandrīz neko nesaka.**

Bāzes grupa:

5 skulptūras ar mutes platuma proporciju ap 48 – 50  
1 skulptūra ar 53

Kandidāts:  
mutes platuma proporcija: 53

-> Kandidāts neizkrīt pilnībā,  
-> bet tas līdzinās retākam gadījumam, nevis tipiskajam.  
**LLR = «iespējams, bet piesardzīgi»**

## [03] LLR – kā svāri ...



Vienā kausā — “bāzes grupa”

Otrā — “svešinieki”

Katrs no šiem jautājumiem:

pieliek nelielu svaru vienā vai otrā pusē.

LLR ir kopējais svērumš, nevis atsevišķš spriedums.

## [04] Rādītāju svars un vērtību normalizācija

Tiek noteikts katra rādītāja svars Jūdena indekss \* LLR indekss

**Derīgs rādītājs + Raksturīgs bāzes grupai = lielāks svars**

Sigmoīda normalizācija – pārvērš rezultātus vērtībās no 0 līdz 1;

Vērtība tuvu 1 – ļoti līdzīgs grupai

Vērtība tuvu 0 – ļoti atšķirīgs

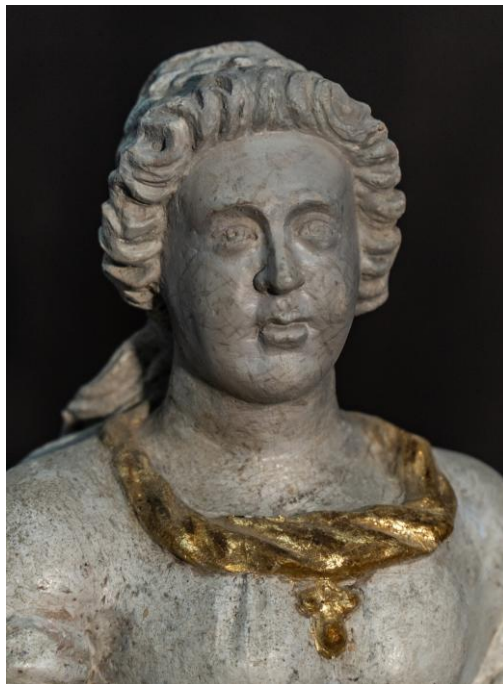
Ap 0.5 – neitrāls / neskaidras gadījums

$$LLR_{total}(c) = \sum_{i=1}^k w_i \cdot LLR_i(x_i)$$

$$score(c) = \frac{1}{1 + e^{-LLR_{total}(c)}}$$

	A	B	C	D	E
1	targ	result	pair	skulptura	score
14	1-	99-	99- vs 1-	99-Kandava-kancele	0.987494682293
15	1-	99-	99- vs 1-	99-Kandava-Lukas	0.987247075103
16	1-	99-	99- vs 1-	99-Kandava-ceriba	0.953124129854
17	1-	99-	99- vs 1-	99-Kandava-marks	0.933296608385
18	1-	99-	99- vs 1-	99-Kandava-janis	0.903428540069
19	1-	99-	99- vs 1-	99-Kandava-milestiba	0.899345856727
20	1-	99-	99- vs 1-	99-Kandava-ticiba	0.866697641849
222					

# [05] Metodes validācija



## [05] Metodes validācija

Salgales un Kandavas baznīcu autors:  
Johans Mertenss /dokumentāri apstiprināts/

target_group	result_group	pair	skulptura	score	raw_p
1-	(no-prefix)	(no-prefix) vs 1-	vecpiebalga	0.003292316174	
1-	(no-prefix)	(no-prefix) vs 1-	piltene	0.001103501745	
1-	(no-prefix)	(no-prefix) vs 1-	Ergelu_engelis	0.000800894398	
1-	(no-prefix)	(no-prefix) vs 1-	kerklini_alegorija	0.000000006251	
1-	(no-prefix)	(no-prefix) vs 1-	Cloud_sompare_v-galva-1	0.000000000000	
1-	2-	2- vs 1-	2-Bikeru_vaze_restored2	0.000772175163	
1-	2-	2- vs 1-	2-Bikeru_vaze_restored	0.000156446031	
1-	2-	2- vs 1-	2-sparnotagalva	0.000099558966	
1-	2-	2- vs 1-	2-Bikeru_vaze_unrestored3	0.000000000051	
1-	3-	3- vs 1-	3-jekaba_dama1	0.000342039095	
1-	3-	3- vs 1-	3-jekaba_kungs1	0.000000006710	
1-	98-	98- vs 1-	98-sparnotagalva	0.000003737472	
1-	99-	99- vs 1-	99-Kandava-kancele	0.987494682293	
1-	99-	99- vs 1-	99-Kandava-Lukas	0.987247075103	
1-	99-	99- vs 1-	99-Kandava-ceriba	0.953124129854	
1-	99-	99- vs 1-	99-Kandava-marks	0.933296608385	
1-	99-	99- vs 1-	99-Kandava-janis	0.903428540069	
1-	99-	99- vs 1-	99-Kandava-milestiba	0.899345856727	
1-	99-	99- vs 1-	99-Kandava-ticiba	0.866697641849	
1-	99-	99- vs 1-	99-Kandava-matejs	0.482295088732	

### Sagaidāmie rezultāti:

Salgale vs Kandava - pozitīvs

Kandava vs Salgale – pozitīvs

Kandava vs citi – negatīvs

Salgale vs citi - negatīvs

target_group	result_group	pair	skulptura	score	raw_p
99-	(no-prefix)	(no-prefix) vs 99-	vecpiebalga	0.039648182103	
99-	(no-prefix)	(no-prefix) vs 99-	piltene	0.000587575836	
99-	(no-prefix)	(no-prefix) vs 99-	kerklini_alegorija	0.000001895341	
99-	(no-prefix)	(no-prefix) vs 99-	Ergelu_engelis	0.000000070051	
99-	(no-prefix)	(no-prefix) vs 99-	Cloud_sompare_v-galva-1	0.000000000001	
99-	1-	1- vs 99-	1-Mozus	0.999622253709	
99-	1-	1- vs 99-	1-juris-salgale	0.987095455296	
99-	1-	1- vs 99-	1-Janis-salgale	0.984174868008	
99-	1-	1- vs 99-	1-chuvaks	0.981051210304	
99-	1-	1- vs 99-	1-kirstus	0.976741293346	
99-	1-	1- vs 99-	1-Matejs-salgale	0.965104221339	
99-	1-	1- vs 99-	1-marks	0.953389204794	
99-	1-	1- vs 99-	1-kancele	0.929388361537	
99-	1-	1- vs 99-	1-Ticiba	0.915048173926	
99-	1-	1- vs 99-	1-Damite-Ljubestiba	0.706549149567	
99-	1-	1- vs 99-	1-Salgale-ticiba2	0.616267057633	
99-	2-	2- vs 99-	2-Bikeru_vaze_restored2	0.061697366637	
99-	2-	2- vs 99-	2-Bikeru_vaze_restored	0.057951351416	
99-	2-	2- vs 99-	2-sparnotagalva	0.001313716234	
99-	2-	2- vs 99-	2-Bikeru_vaze_unrestored3	0.000000929783	



**Paldies par  
uzmanību!**