Iterative Closest Point Algorithm for Accurate Registration of Coarsely Registered Point Clouds with CityGML Models

Steffen Goebbels, Regina Pohle-Fröhlich and Philipp Pricken

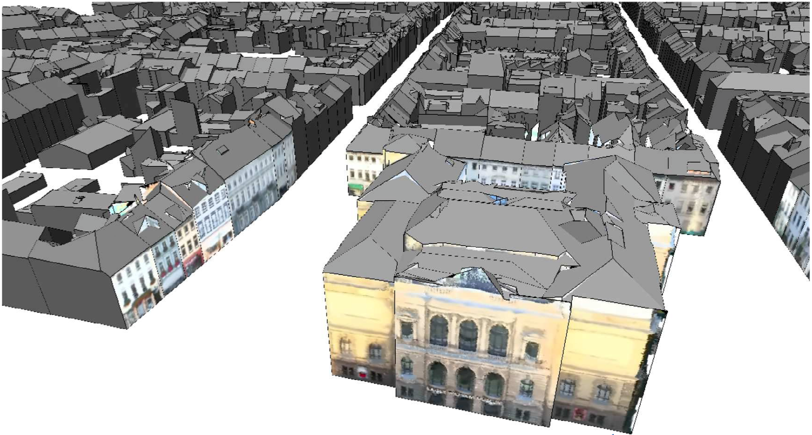
Niederrhein University of Applied Sciences - Institute for Pattern Recognition,
Faculty of Electrical Engineering and Computer Science

ISPRS GSW 2019

Hochschule Niederrhein
University of Applied Sciences

iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

CityGML model, textured with an aligned photogrammetric point cloud



Our aim: Extend LoD2 CityGML models to LoD3.[1]

[1] S. Hensel, S. Goebbels, M. Kada: Facade Reconstruction for Textured LoD2 CityGML Models based on Deep Learning and Mixed Integer Linear Programming, ISPRS GSW 2019

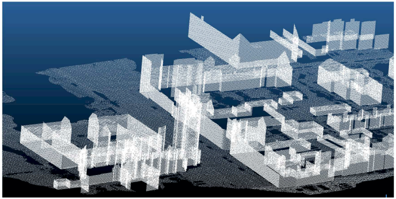
Iterative Closest Point Algorithm (ICP)

Align source point cloud S to target point cloud D.

procedure ICP(point cloud S, point cloud D, convergence criteria)
  while convergence criteria are not fulfilled **do**
    clear correspondences
    for $p \in S$ **do**
      find nearest neighbor $q \in D$ of p
      if neighbor is within threshold distance **then**
        store correspondence (p, q)
    compute affine transform T that maps p to q as close as possible, simultaneously for all pairs (p, q) (least quares minimization)
    apply T to all points in S.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Tests with Point Cloud Library and sampled city models

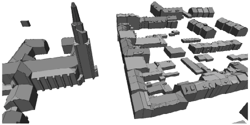


ICP Singular Value Decomposition	ICP Levenberg-Marquard	ICP Point-to-Plane	NDT	Feature-based alignment

fails | **speed** →

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Point-to-Model ICP

Align S:  to D: .

procedure ICP(point cloud S, city model D, convergence criteria)

- **while** convergence criteria are not fulfilled **do**
  - clear correspondences
  - **for** $p \in S$ **do**
    - project p to all wall and roof polygons of D
    - select nearest projection q of p
    - **if** projection is within threshold distance **then**
      - store correspondence (p, q)
  - compute affine transform T that maps p to q as close as possible, simultaneously for all pairs (p, q)
  - apply T to all points in S.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Residuum of affine transform (homogenous coordinates)

Transformation matrix $T = T_{\vec{a},\sigma}$ depends on a scaling factor σ and parameters $\vec{a} := (\alpha, \beta, \gamma, \Delta x, \Delta y, \Delta z)$.

Residuum of affine transform (homogenous coordinates)

Transformation matrix $T = T_{\vec{a},\sigma}$ depends on a scaling factor σ and parameters $\vec{a} := (\alpha, \beta, \gamma, \Delta x, \Delta y, \Delta z)$.

For each pair $(\vec{s}, \vec{d})$ we discuss local residuum

$$\vec{r}(\vec{a}, \sigma) = (r_1, r_2, r_3, 0)^\top := T_{\vec{a},\sigma} \cdot \vec{s} - \vec{d} \in \mathbb{R}^4.$$

Hochschule Niederrhein
University of Applied Sciences

iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Residuum of affine transform (homogenous coordinates)

Transformation matrix $T = T_{\vec{a},\sigma}$ depends on a scaling factor σ and parameters $\vec{a} := (\alpha, \beta, \gamma, \Delta x, \Delta y, \Delta z)$.

For each pair $(\vec{s}, \vec{d})$ we discuss local residuum

$$\vec{r}(\vec{a}, \sigma) = (r_1, r_2, r_3, 0)^\top := T_{\vec{a},\sigma} \cdot \vec{s} - \vec{d} \in \mathbb{R}^4.$$

We combine all local residua to one vector in $\mathbb{R}^{3n}$:

$$\vec{R}(\vec{a}, \sigma) := (\vec{r}_{1,1}, \vec{r}_{1,2}, \vec{r}_{1,3}, \vec{r}_{2,1}, \ldots, \vec{r}_{n,3})^\top$$

where $\vec{r}_{k,j}$ is the j-th component, $j \in \{1, 2, 3\}$, of the local residuum of pair $(\vec{s}_k, \vec{d}_k)$.

Objective function for computing $\vec{a} = (\alpha, \beta, \gamma, \Delta x, \Delta y, \Delta z)$

Subject to a scaling factor $\sigma = 1$, we have to minimize an objective function (mean of the squared residua)

$$e(\vec{a}, 1) := \left\| \frac{1}{\sqrt{n}} \vec{R}(\vec{a}, 1) \right\|_2^2 = \frac{1}{n} \sum_{k=1}^{3n} (\vec{R}_k(\vec{a}, 1))^2$$

to find optimal rotation and translation parameters.
Scaling parameter σ will be determined in a second step.

Gauss-Newton method

Let $D(\vec{a})$ be the Jacobian of $\frac{1}{\sqrt{n}}\vec{R}(\vec{a},1)$. This matrix of first derivatives can be composed from Jacobians of local residua. Gauss-Newton iteration are defined via $\vec{a}_0 := (0,0,0,0,0,0)$ and

$$\vec{a}_{l+1} := \vec{a}_l - (D(\vec{a}_l)^\top \cdot D(\vec{a}_l))^{-1} \cdot D(\vec{a}_l)^\top \frac{1}{\sqrt{n}}\vec{R}(\vec{a}_l,1).$$

Levenberg-Marquardt method

Let I be the identity matrix and $\lambda_l \geq 0$ parameters controlling the search radius r (step size) for local minima. Initial value: λ_0.

$$\vec{a}_{l+1} := \vec{a}_l - \left(D(\vec{a}_l)^\top \cdot D(\vec{a}_l) + \lambda_l I\right)^{-1} \cdot D(\vec{a}_l)^\top \frac{1}{\sqrt{n}} \vec{R}(\vec{a}_l, 1).$$

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Levenberg-Marquardt method

Let I be the identity matrix and $\lambda_l \geq 0$ parameters controlling the search radius r (step size) for local minima. Initial value: λ_0.

$$\vec{a}_{l+1} := \vec{a}_l - (D(\vec{a}_l)^\top \cdot D(\vec{a}_l) + \lambda_l I)^{-1} \cdot D(\vec{a}_l)^\top \frac{1}{\sqrt{n}} \vec{R}(\vec{a}_l, 1).$$

Only downhill steps have to be considered:

- If the objective function does not decrease: parameter λ_l is doubled.
- If the objective function directly decreases: $\lambda_{l+1} := \lambda_l / 2$.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Levenberg-Marquardt method

Let I be the identity matrix and $\lambda_l \geq 0$ parameters controlling the search radius r (step size) for local minima. Initial value: λ_0.

$$\vec{a}_{l+1} := \vec{a}_l - \left(D(\vec{a}_l)^\top \cdot D(\vec{a}_l) + \lambda_l I\right)^{-1} \cdot D(\vec{a}_l)^\top \frac{1}{\sqrt{n}} \vec{R}(\vec{a}_l, 1).$$

Only downhill steps have to be considered:

- If the objective function does not decrease: parameter λ_l is doubled.
- If the objective function directly decreases: $\lambda_{l+1} := \lambda_l / 2$.

Remark:

- $\lambda_l = 0$: Gauss-Newton step
- λ_l large: small step of gradient descent.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Determining a scaling factor σ_0

For previously computed parameters $\vec{a}_l$, minimize

$$e(\sigma) := \sum_{k=1}^{3n} (\vec{R}_k(\vec{a}_l, \sigma))^2.$$

Determining a scaling factor σ_0

For previously computed parameters $\vec{a}_l$, minimize

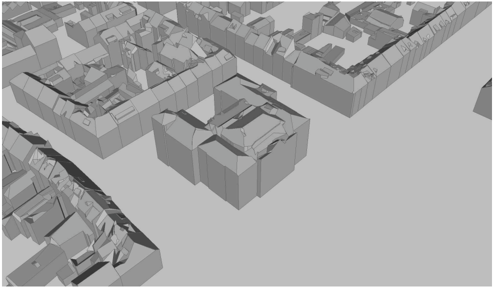
$$e(\sigma) := \sum_{k=1}^{3n} (\vec{R}_k(\vec{a}_l, \sigma))^2.$$

The necessary condition $\frac{d}{d\sigma} e(\sigma) = 0$ results in

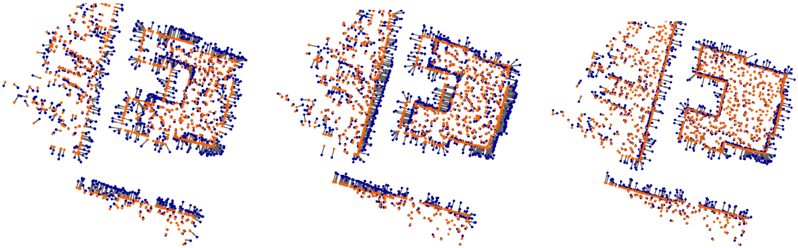
$$\sigma_0 := \frac{\sum_{k=1}^{n} \sum_{j=1}^{3} (T_{\vec{a}_l,1} \vec{s}_k)_j \cdot (\vec{d}_k)_j}{\sum_{k=1}^{n} \sum_{j=1}^{3} (T_{\vec{a}_l,1} \vec{s}_k)_j^2}.$$

Scaling has to be restricted to factors near 1 to avoid contraction of the cloud to one point.

Example: Kaiser Wilhelm museum point cloud



Correspondences at start, after 10, and after 25 iterations



(One Gauss-Newton step, projection to bounding rectangles)

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Levenberg Marquardt optimization (LM) compared with one Gauss-Newton (GN) step

Kaiser Wilhelm museum point cloud: All runs reduce an error of 6.10032 to the same final error 1.976.			
method	number of outer iterations	number of inner iterations	running time
GN	61	limited to 1	326 s
LM $\lambda_0 = 1$	91	limited to 1	488 s
LM $\lambda_0 = 1$	61	average: 4.5	816 s
LM $\lambda_0 = 1/16$	61	average: 2	483 s
LM $\lambda_0 = 1/64$	61	average: 2	484 s

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Projection to roof and wall polygons with gradient descent

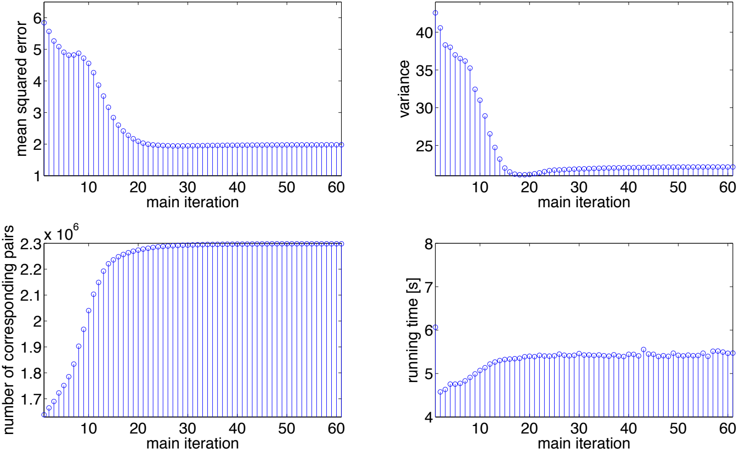
- Project point to bounding rectangle.
- If projected point is outside the polygon then move point to polygon's border using gradient descent on a distance transform.

black: section of a polygon

grey value indicates distance

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

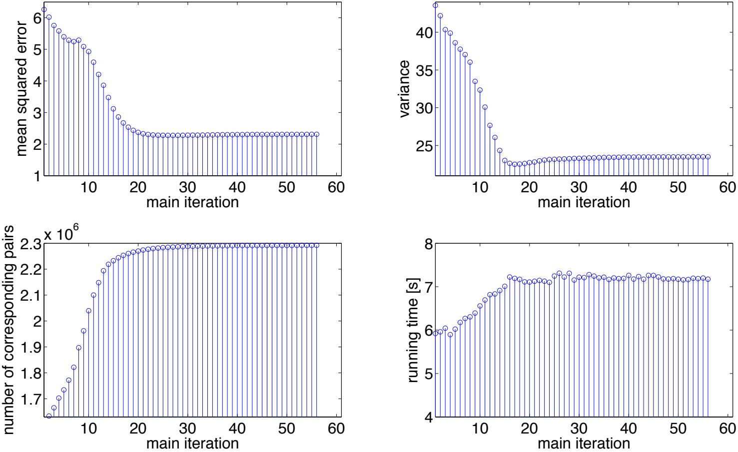
Computational results for projection to rectangles



projection to bounding rectangles (using one Gauss-Newton step, 61 iterations, running time: 326 s)

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Computational results for projection to polygons



projection to wall and roof polygons (using one Gauss-Newton step, 56 iterations, running time: 392 s)

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition