3D Sensing and Sensor Fusion

http://cg.elte.hu/~sensing

Dmitry Chetverikov, Iván Eichhardt
csetverikov@inf.elte.hu,
ivan.eichhardt@sztaki.hu

Eötvös Loránd University
Faculty of Informatics

2019 fall (2019-2020-1)

3D Point Cloud Registration

1. Iterative Closest Point
   - Major components of ICP
   - Analysis of ICP
   - Further development of ICP

3D Point Cloud Registration

3D Point Cloud Registration

1. Iterative Closest Point
    - Major components of ICP
    - Analysis of ICP
    - Further development of ICP
2. Trimmed Iterative Closest Point
    - Implementation details and analysis
3. Sample results of 3D data registration
    - Laser scanner data registration
    - Using Genetic Algorithm for pre-alignment
    - Lidar data registration

Euclidean registration of two 3D point sets

- Given:
  - two **roughly pre-registered 3D point sets**

  $\rightarrow$ $\mathcal{P}$ (data) and $\mathcal{M}$ (model)
- Goal:
  - find **optimal motion**

  $\rightarrow$ shift and rotation that bring $\mathcal{P}$ into **best possible alignment** with $\mathcal{M}$
- We consider **Euclidean** alignment
  - other alignments, e.g., affine one, are also used

Applications of 3D point cloud registration

- 3D model acquisition
  - reverse engineering
  - scene reconstruction
- Registration of 3D sensor data
  - laser scanner, Lidar, etc.
- Motion analysis
  - model-based tracking

Problems of 3D point cloud registration

- Partially overlapping point sets
  $\rightarrow$ incomplete measurements
- Noisy measurements
- Erroneous measurements
  $\rightarrow$ outliers
- Shape defects, distortions
- Very large point sets (clouds)
- Point clouds with varying resolution and density

Outline of Iterative Closest Point (ICP)

- Besl and McKay (1992)
- Generic solution to alignment problem

Algorithm: Iterative Closest Point

1. **Pair each point** of $\mathcal{P}$ to closest point in $\mathcal{M}$.
2. **Compute motion** that minimises mean square error (MSE) between paired points.
3. **Apply motion** to $\mathcal{P}$ and update MSE.
4. **Iterate** until convergence.

Assumptions of ICP

- Given 2 sets of 3D points: **data** set $\mathcal{P} = \{\mathbf{p}_i\}_1^{N_p}$ and **model** set $\mathcal{M} = \{\mathbf{m}_i\}_1^{N_m}$.
- $\mathcal{P}$ is **subset** of $\mathcal{M}$: $\mathcal{P} \subset \mathcal{M}$.
- Both $\mathcal{P}$ and $\mathcal{M}$ are **outlier-free**, or have very few outliers.
- **Rough pre-registration**: max initial relative rotation 20°.
- $\mathcal{P}$ is **characteristic enough** to allow for unambiguous matching[1]:
  - no high symmetry
  - no 'featureless' data

[1] Typical for most registration algorithms

ICP: Notation and problem statement

For rotation $\mathbf{R}$ and translation $\mathbf{t}$, transformed points of $\mathcal{P}$ are

$$\mathbf{p}_i(\mathbf{R},\mathbf{t}) = \mathbf{R}\mathbf{p}_i + \mathbf{t}, \quad \mathcal{P}(\mathbf{R},\mathbf{t}) = \{\mathbf{p}_i(\mathbf{R},\mathbf{t})\}_1^{N_p}$$

Individual distance from data point $\mathbf{p}_i(\mathbf{R},\mathbf{t})$ to $\mathcal{M}$:

$$\begin{aligned} \mathbf{m}_{cl}(i,\mathbf{R},\mathbf{t}) &\doteq \arg\min_{\mathbf{m}\in\mathcal{M}} \|\mathbf{m} - \mathbf{p}_i(\mathbf{R},\mathbf{t})\| \\ d_i(\mathbf{R},\mathbf{t}) &\doteq \|\mathbf{m}_{cl}(i,\mathbf{R},\mathbf{t}) - \mathbf{p}_i(\mathbf{R},\mathbf{t})\| \end{aligned}$$

Problem statement: Find rigid motion $(\mathbf{R},\mathbf{t})$ that minimises sum of N_p square distances $d_i^2(\mathbf{R},\mathbf{t})$.

Registration algorithm by Chen and Medioni (1992)

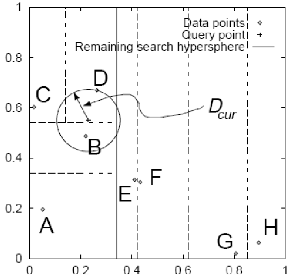
- Iterative scheme similar to Besl and McKay
- Distances from points of $\mathcal{P}$ to surface of $\mathcal{M}$
  $\rightarrow$ **point-to-surface**
- Faster convergence when two sets are close
- Needs triangulation of $\mathcal{M}$ to obtain surface
  - non-trivial operation

Point-to-point vs point-to-surface

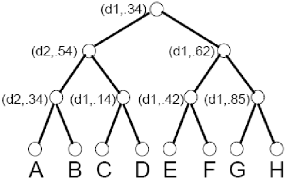
- We use formulation of **Besl and McKay**
- Distances from points of $\mathcal{P}$ to points of $\mathcal{M}$
  - $\rightarrow$ **point-to-point**
  - $\rightarrow$ does not need triangulation
- Easier to use
  - $\rightarrow$ more frequently used in practice

Outline

Searching for closest point: k-d tree 1/2



(a)

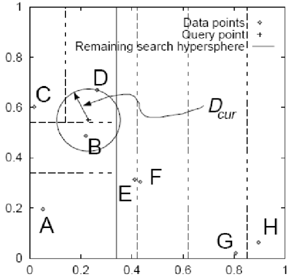


(b)

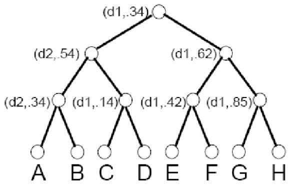
(a) Dashed lines: axis-aligned cutting planes[2]. Small diamonds: data points A–H. (b) K-d tree. Interior nodes: cutting planes with dimensions and cutting values. Leaf nodes: data points.

[2] Source: Szeliski, Computer Vision

Searching for closest point: k-d tree 2/2



(a)



(b)

- Recursively split plane minimizing maximum depth of tree
- Classic search:
  - locate query point (**+**) in its bin (**D**)
  - search nearby leaves (**C**,**B**...) until closest point is found
- Best Bin First search (usually, more efficient):
  - search bins in order of their spatial proximity to query point

Closed-form solutions for optimal rigid motion

- Unit Quaternions: used by original ICP
- Singular Value Decomposition
- Orthogonal Matrices
- Dual Quaternions

Properties of methods for optimal motion

Method	Accuracy	2D Stability[3]	Speed, small N_p	Speed, large N_p
UQ	good	good	fair	fair
SVD	good	good	fair	fair
OM	fair	poor	good	poor
DQ	fair	fair	poor	good

- Comparative analysis[4]
  - **U**nit **Q**uaternions
  - **S**ingular **V**alue **D**ecomposition
  - **O**rthogonal **M**atrices
  - **D**ual **Q**uaternions
  - N_p: number of points

[3] In presence of degenerate (2D) data
[4] Eggert (1997)

Outline

Properties of ICP 1/2

- **Pre-registration** required. Options:
  - manual
  - known sensor motion between two measurements
  - **Genetic Algorithm** $\longrightarrow$ illustrated later in lecture
  - surface feature-based pre-registration
  - . . .
- Point pairing
  - computationally demanding
  - special data structures used to speed up

  $\rightarrow$ k-d trees, spatial bins

Properties of ICP 2/2

- Optimal motion: closed-form solutions available
- ICP has been proven to **converge** to local minimum
- Applicable to different kinds of measurements
  - surface measurements
  - volumetric measurements

Drawbacks of ICP

- Assumes **outlier-free** data
- Assumes that $\mathcal{P}$ is subset of $\mathcal{M}$: $\mathcal{P} \subset \mathcal{M}$

$\rightarrow$ Not robust

$\rightarrow$ Not applicable to parially overlapping measurements

- Relatively slow convergence

Outline

1. Iterative Closest Point
   - Major components of ICP
   - Analysis of ICP
   - Further development of ICP
2. Trimmed Iterative Closest Point
   - Implementation details and analysis
3. Sample results of 3D data registration
   - Laser scanner data registration
   - Using Genetic Algorithm for pre-alignment
   - Lidar data registration

Categorisation of ICP variants

Goal: Improve robustness and convergence (speed)[5].

- **Selecting** subsets of $\mathcal{P}$ and $\mathcal{M}$
  - random sampling for Monte-Carlo technique
- **Matching** (pairing) selected points
  - closest point
  - point-to-surface: faster when normals are precise
- **Weighting** and **rejecting** pairs
  - distribution of distances between paired points
  - geometric constraints (e.g., compatibility of normal vectors)
- Assigning **error metric** and **minimising** it
  - iterative: original ICP
  - direct: Levenberg-Marquardt algorithm

[5] Rusinkiewicz (2001)

Robustifying ICP algorithm

- ICP assumption:
  - each point of $\mathcal{P}$ has valid correspondence in $\mathcal{M}$
- Not applicable to:
  - partially overlapping sets
  - sets containing outliers
- How to robustify ICP?
  - **reject wrong point pairs**

  $\rightarrow$ based on different criteria

Rejection criteria

Statistical criteria: Monte-Carlo with robust statistics

- Least **median of squares** (LMedS)
- Least **trimmed squares** (LTS)

Geometric criteria: Iterative Closest *Reciprocal* Point[6]

- Similar to backward consistency check in stereo matching
- Use ϵ-reciprocal correspondence:
  - **if** point $\mathbf{p} \in \mathcal{P}$ has closest point $\mathbf{m} \in \mathcal{M}$, **then**
  - back-project $\mathbf{m}$ onto $\mathcal{P}$ by finding closest point $\mathbf{p}' \in \mathcal{P}$
  - reject pair $(\mathbf{p}, \mathbf{m})$ if $\|\mathbf{p} - \mathbf{p}'\| > \epsilon$

[6] Pajdla (1995)

Convergence of above methods

- Different heuristics combined

$\rightarrow$ Heterogeneous algorithms

$\rightarrow$ **Convergence cannot be proved**

- E.g., Iterative Closest Reciprocal Point does not converge

$\rightarrow$ Stopping conditions difficult to use

Robust statistics

- **Sort** distances between paired points
- **Minimize**:
- For Least **median of squares**,
  - value in the middle of sorted sequence
  - **incompatible** with computation of optimal motion
- For Least **trimmed squares**,
  - sum of certain number of least values (e.g., least 50%)
  - **compatible** with computation of optimal motion
  - better convergence rate, smoother objective function

Randomised robust regression

Use LMedS or LTS to

- Repeatedly draw random samples
- Estimate optimal motion parameters
- Detect and reject outliers
- Find least squares solution for inliers

Robust to outliers, but breakdown point is 50%

$\rightarrow$ **Minimum overlap** is 50%

Assumptions of Trimmed ICP

- 2 sets of 3D points: **data** set $\mathcal{P} = \{\mathbf{p}_i\}_1^{N_p}$ and **model** set $\mathcal{M} = \{\mathbf{m}_i\}_1^{N_m}$. ($N_p \neq N_m$.)
- Minimum guaranteed rate of data points that can be paired is known: **minimum overlap** ξ. Number of data points that can be paired $N_{po} = \xi N_p$.
- Rough pre-registration: max initial relative rotation 20°.
- Overlapping part is characteristic enough to allow for unambiguous matching:
  - no high symmetry
  - no ‘featureless’ data

Problem statement: TrICP

- Informal statement
  - Find Euclidean transformation that brings an N_{po}-point subset of $\mathcal{P}$ into best possible alignment with $\mathcal{M}$.
- Formal statement
  - Find rigid motion $(\mathbf{R}, \mathbf{t})$ that minimises sum of least N_{po} square distances $d_i^2(\mathbf{R}, \mathbf{t})$.
- Conventional ICP
  - $\xi = 1$ and $N_{po} = N_p$.
- Trimmed ICP
  - Smooth transition to ICP as $\xi \to 1$.

Outline of of TrICP

Basic idea: Consistent use of LTS in **deterministic** way

Algorithm: Trimmed Iterative Closest Point [7]

1. **Closest point**: For each point $\mathbf{p}_i \in \mathcal{P}$, find closest point in $\mathcal{M}$ and compute d_i^2.
2. **Trimmed Squares**: Sort d_i^2, select N_{po} least values and calculate their sum S_{TS}.
3. **Convergence test**: If any of stopping conditions is satisfied, exit; otherwise, set $S'_{TS} = S_{TS}$ and continue.
4. **Motion calculation**: For N_{po} selected pairs, compute optimal motion $(\mathbf{R}, \mathbf{t})$ that minimises S_{TS}.
5. **Data set motion**: Transform $\mathcal{P}$ by $(\mathbf{R}, \mathbf{t})$ and go to 1.

[7] Chetverikov et al. (2002)

Stopping conditions

- Maximum allowed number of iterations N_{iter} has been reached, **or**
- Trimmed MSE is sufficiently small, **or**
- Change of Trimmed MSE is sufficiently small.

Trimmed MSE e: For sorted distances
$d_{s1} \leq d_{s2} \leq \ldots \leq d_{sN_{po}} \leq \ldots \leq d_{sN_p}$,

$$S_{TS} \doteq \sum_{si=s1}^{sN_{po}} d_{si}^2 \qquad\qquad e \doteq \frac{S_{TS}}{N_{po}}$$

Change of Trimmed MSE: $|S_{TS} - S'_{TS}|$

Outline

1. Iterative Closest Point
   - Major components of ICP
   - Analysis of ICP
   - Further development of ICP
2. Trimmed Iterative Closest Point
   - Implementation details and analysis
3. Sample results of 3D data registration
   - Laser scanner data registration
   - Using Genetic Algorithm for pre-alignment
   - Lidar data registration

Implementation details

- Finding closest point
  - use k-d tree
- Sorting individual distances and calculating LTS
  - use heap sort
- Computing optimal motion
  - use Unit Quaternions
  - robust to noise
  - stable in presence of degenerate data (‘flat’ point sets)
  - relatively fast

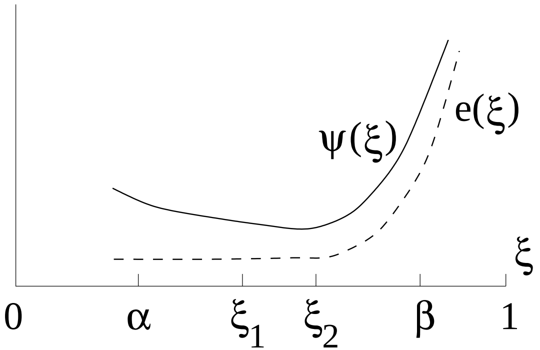
Automatic setting of overlap parameter ξ 1/2

When ξ is unknown, it is set automatically by minimising objective function

$$\psi(\xi) = \frac{e(\xi)}{\xi^{1+\lambda}}, \qquad \lambda = 2$$

- In given range $\alpha \leq \xi \leq \beta$, $\psi(\xi)$ minimises trimmed MSE $e(\xi)$ and tries to use as many points as possible.
- Larger λ: avoid undesirable alignments of symmetric and/or ‘featureless’ parts.
- $\psi(\xi)$ is minimised using modified **Golden Section Search Algorithm.**

Automatic setting of overlap parameter ξ 2/2



Typical shapes of objective functions $e(\xi)$ and $\psi(\xi)$.

Convergence of Trimmed ICP

Theorem: *TrICP always converges monotonically to local minimum with respect to trimmed MSE objective function.*

Convergence to global minimum depends on initial guess.

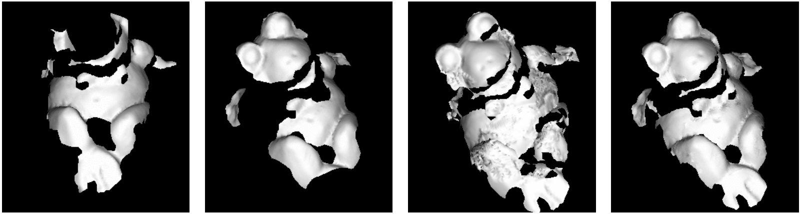
Sketch of proof of convergence

- **Optimal motion** does not increase MSE: if it did, it would be inferior to identity transformation, as the latter does not change MSE.
- **Updating the closest points** does not increase MSE: no individual distance increases.
- **Updating the list of N_{po} least distances** does not increase MSE: to enter the list, any new pair has to substitute a pair with larger distance.
- Sequence of MSE values is nonincreasing and bounded below (by zero), hence it converges to a local minimum.

Outline

1. Iterative Closest Point
   - Major components of ICP
   - Analysis of ICP
   - Further development of ICP
2. Trimmed Iterative Closest Point
   - Implementation details and analysis
3. Sample results of 3D data registration
   - Laser scanner data registration
   - Using Genetic Algorithm for pre-alignment
   - Lidar data registration

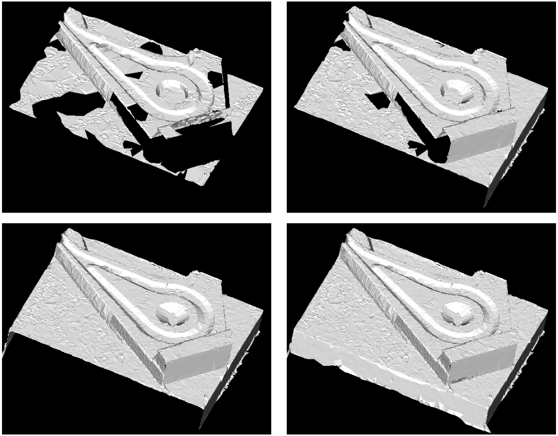
Results for Frog



set $\mathcal{P}$ | set $\mathcal{M}$ | result of ICP | result of TrICP

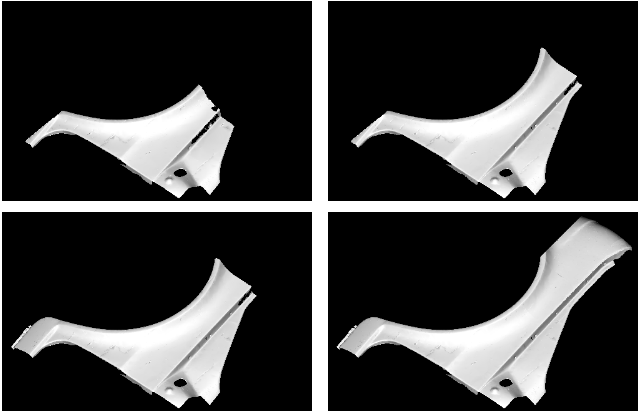
Aligning two partial measurements of Frog.
($\approx$3000 points)

Results for Skoda part



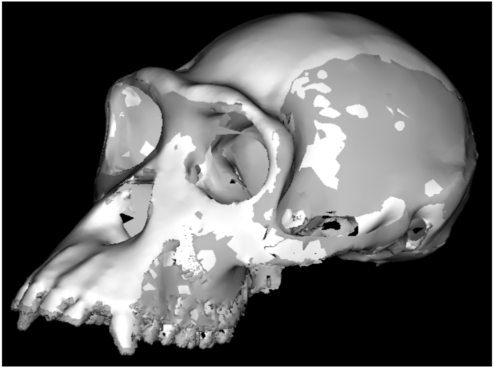
TrICP: Aligning partial measurements of **Skoda part.**
(≈6000 points)

Results for Fiat part



TrICP: Aligning partial measurements of Fiat part.
(overlap $\approx$ 20%.)

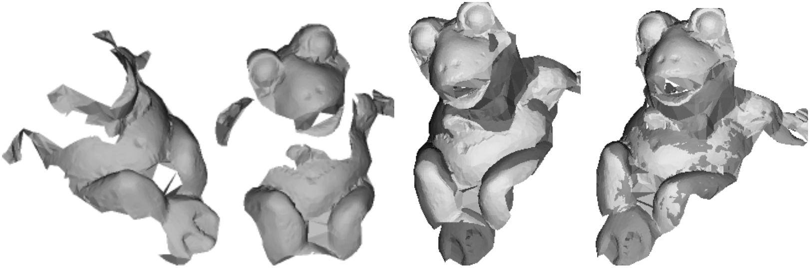
Results for Skull



TrICP: Aligning two measurements of chimpanzee skull.
(≈100000 points)

Outline

Results for Frog dataset



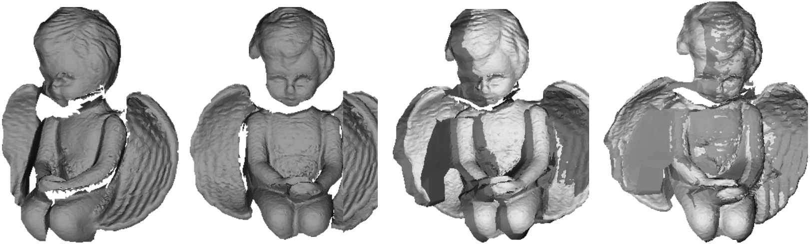
Frog data GA GA+TrICP

Frog dataset, GA alignment[8] and final alignment

- Significant relative rotation in data compensated
- TrICP refines result of GA

[8] Lomonosov, Chetverikov, Ekárt (2006)

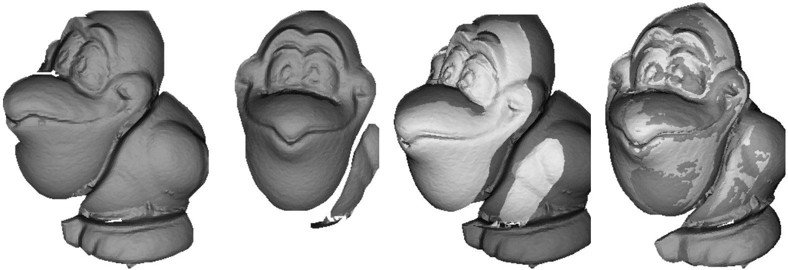
Results for Angel dataset



Angel data GA GA+TrICP

Angel dataset, GA alignment and final alignment.

Results for Bird dataset

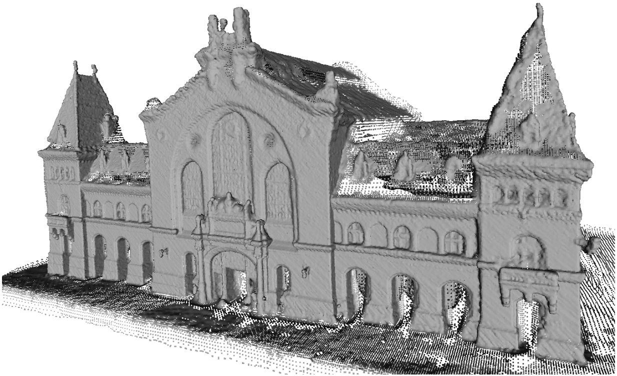


Bird data GA GA+TrICP

Bird dataset, GA alignment and final alignment.

Outline

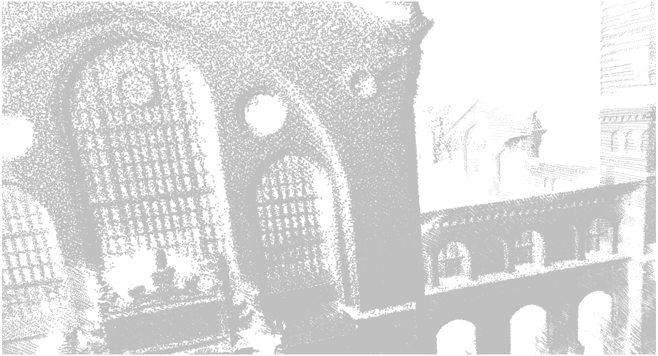
Central Market Hall



- 160 point clouds acquired by car-mounted Lidar[9]
- Registered by TrICP without knowledge of motion
- Surface obtained by triangulation where possible

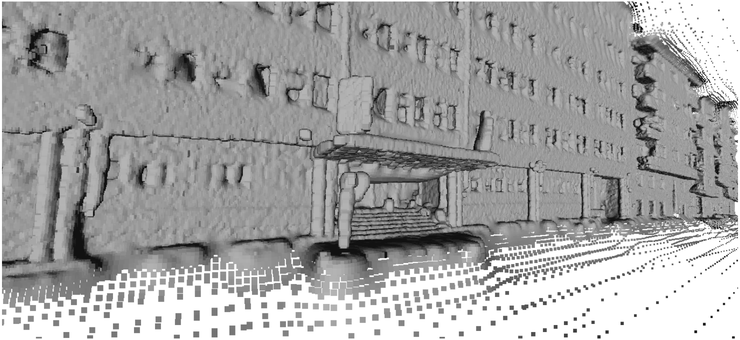
[9] SZTAKI project i4D

Detail of point cloud



- Varying spatial resolution
  - point cloud dense at bottom
  - decreasing with height

Part of Kende street



- 360 point clouds acquired by car-mounted Lidar
- Registered by TrICP without knowledge of motion
- Part shown: entrance of SZTAKI

Issues to be addressed

- Point clouds with varying resolution and density
  - → parts with different impacts on registration
  - → problems when flat, featureless parts dominate
- Long sequences of point clouds
  - registration errors tend to accumulate
  - → growing shape distortion
- Closed loops of point clouds for complete 3D models
  - accumulated registration errors disturb fitting in the end
  - → need for global cost function
  - → adjust all partial registrations, smooth errors
- Very large point clouds
  - large computational load
  - → slow operation
  - → need for hardware support (e.g., GPU)