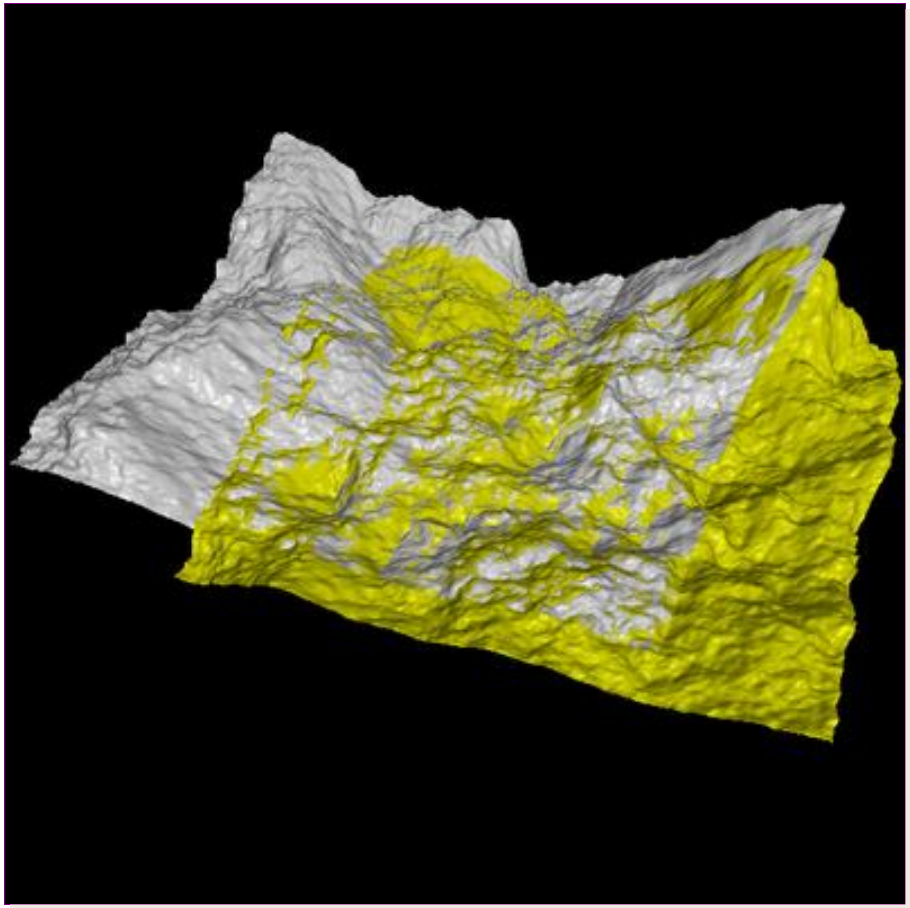
GAMES
Registration

Qixing Huang
July 16th 2021

Motivation

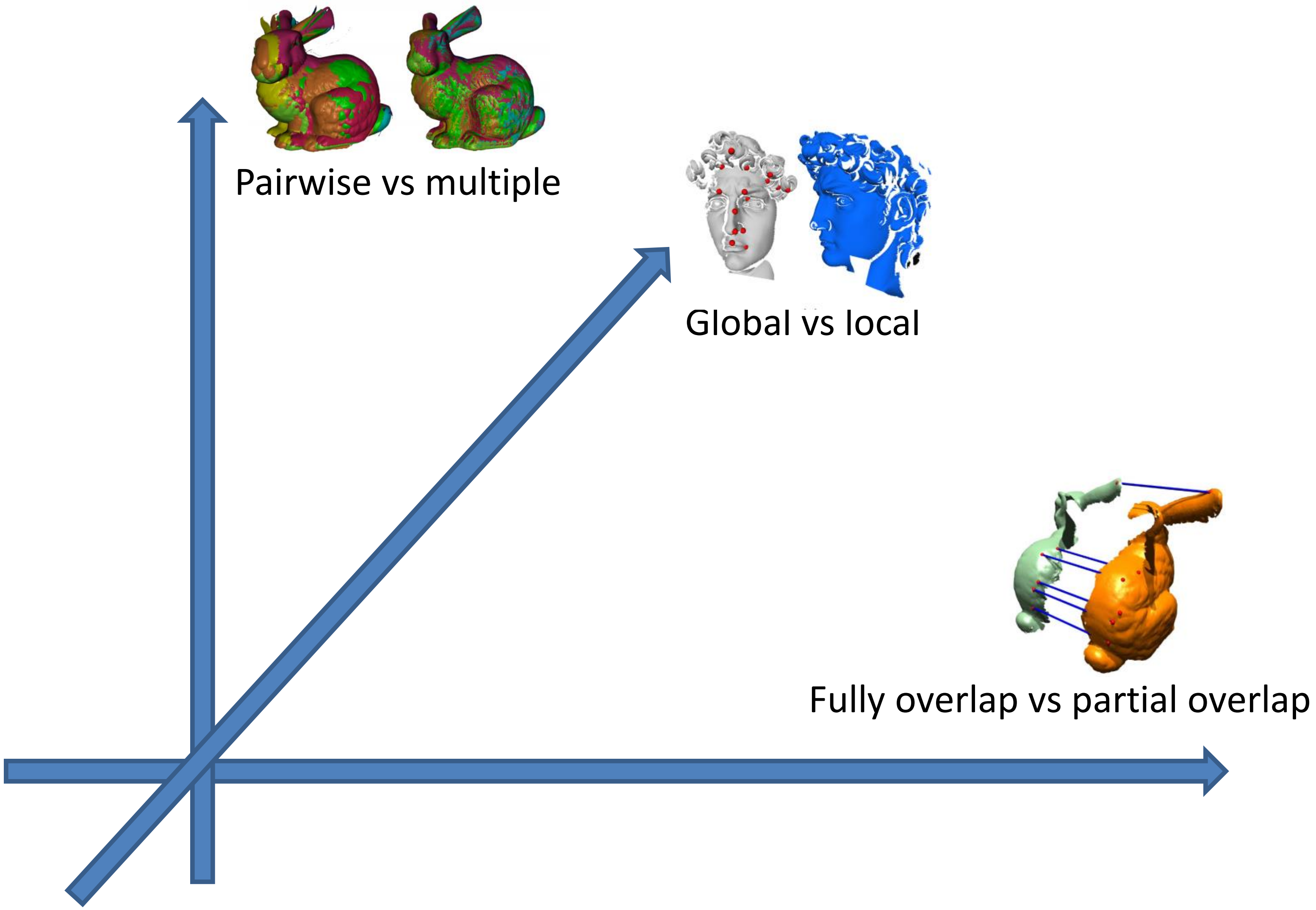
- Align two shapes/scans given initial guess for relative transform



Slide credit: Niloy Mitra and Szymon Rusinkiewicz

Task classification

Three axis



Pairwise vs multiple

Global vs local

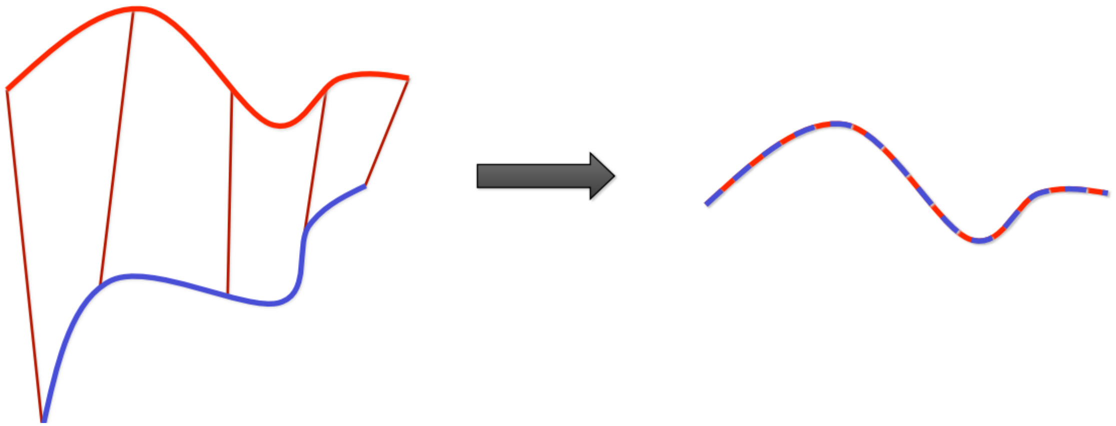
Fully overlap vs partial overlap

Outline

- Pairwise registration
  - Full overlap
  - Partial overlap
  - Global methods
  - Learning-based
- Multiple registration
  - Joint pairwise registration
  - Simultaneous registration and reconstruction

ICP for pairwise registration

- If correct correspondences are known, can find correct relative rotation/translation



Construct error function:

$$E := \sum_i (R\mathbf{p}_i + t - \mathbf{q}_i)^2$$

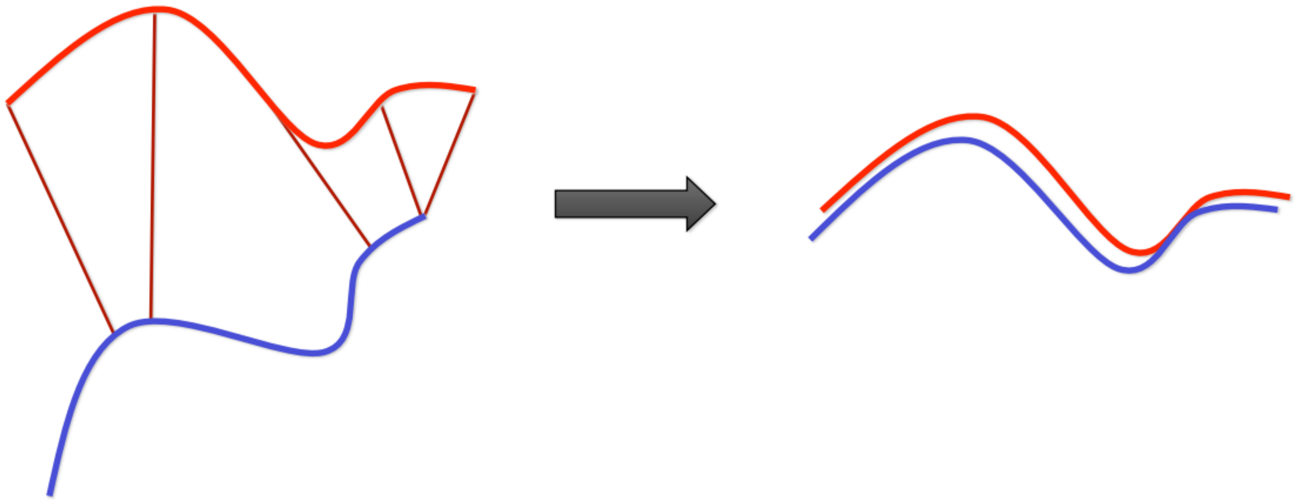
Minimize (closed form solution in [Horn 87])

Slide credit: Niloy Mitra and Szymon Rusinkiewicz

ICP for pairwise registration

- Assume: Closest points as corresponding

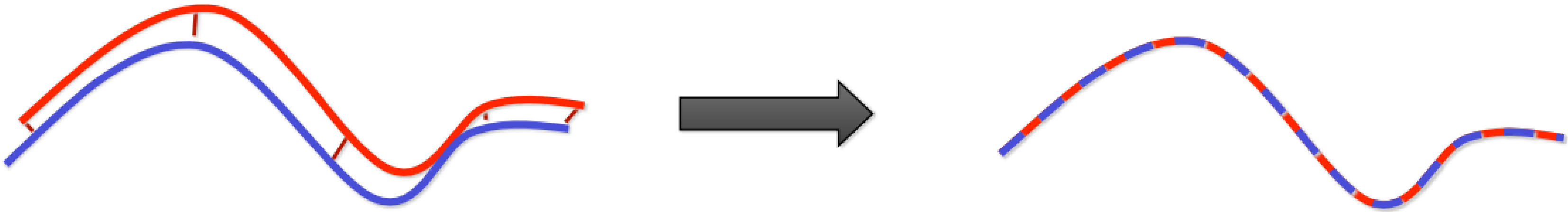
$$\mathbf{p}_i \rightarrow \mathcal{C}(\mathbf{p}_i)$$



Slide credit: Niloy Mitra and Szymon Rusinkiewicz

ICP for pairwise registration

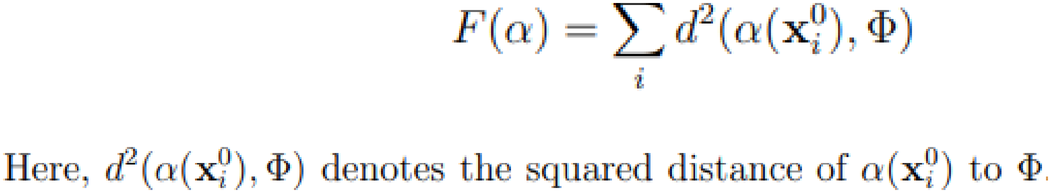
- ... and iterate to find alignment
- Iterative Closest Points (ICP) [Besl and McKay 92]
- Converges if starting poses are close enough



Slide credit: Niloy Mitra and Szymon Rusinkiewicz

From the optimization perspective

- The registration problem shall be formulated in a least squares sense as follows. Compute the rigid body transformation α ∗ , which minimizes



$$F(\alpha) = \sum_i d^2(\alpha(\mathbf{x}_i^0), \Phi)$$

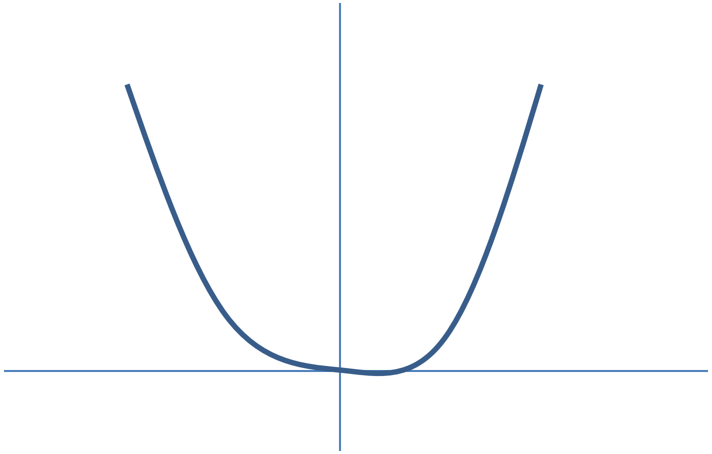
Here, $d^2(\alpha(\mathbf{x}_i^0), \Phi)$ denotes the squared distance of $\alpha(\mathbf{x}_i^0)$ to Φ.

- ICP is alternating minimization
  - Always reduces the objective function
  - Linear convergence

Reducing objective value does not guarantee convergence

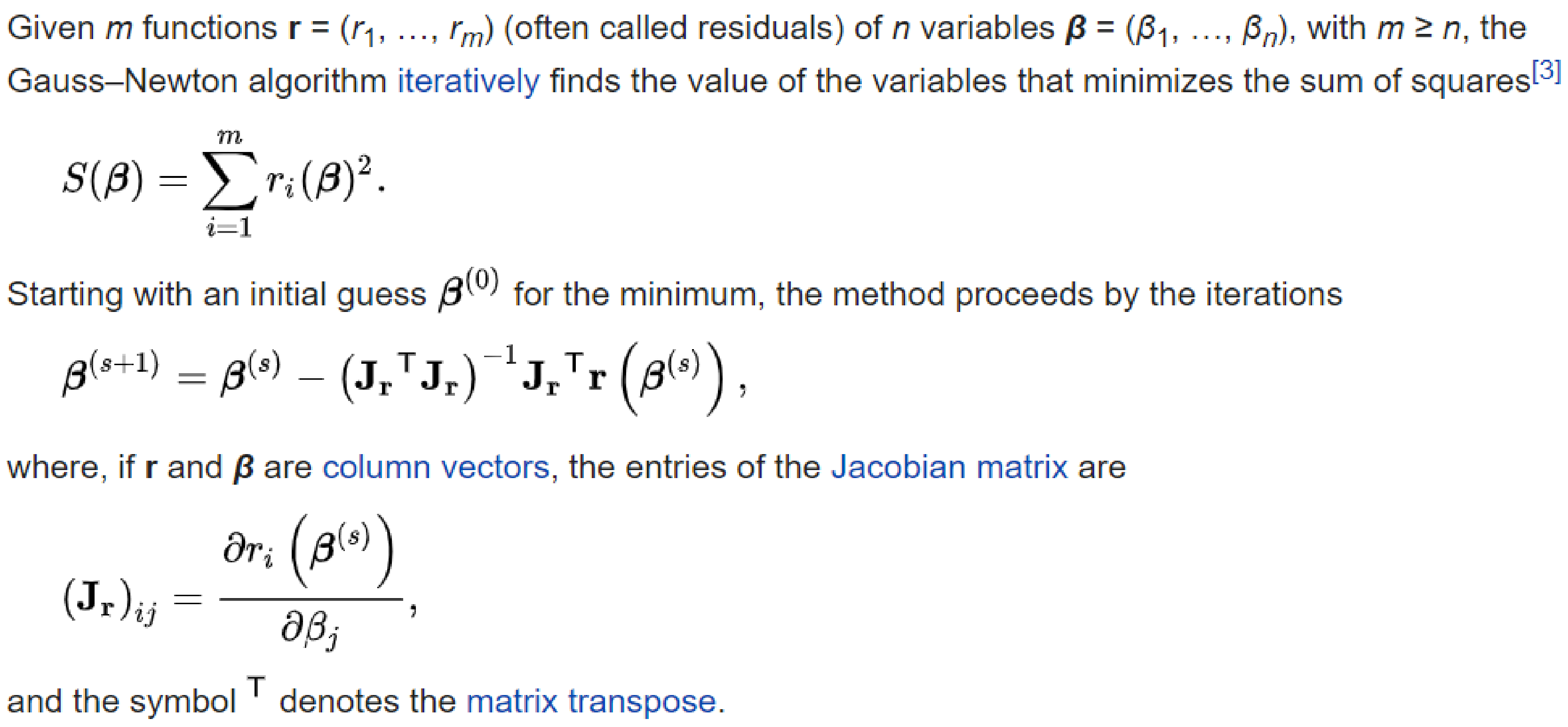
- $f(x) = x*x$

- $x_i = 3 + 1/i$



Gauss-Newton optimization

- Review of Gauss-Newton method



Given m functions $\mathbf{r} = (r_1, \ldots, r_m)$ (often called residuals) of n variables $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$, with $m \geq n$, the Gauss–Newton algorithm iteratively finds the value of the variables that minimizes the sum of squares[3]

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{m} r_i(\boldsymbol{\beta})^2.$$

Starting with an initial guess $\boldsymbol{\beta}^{(0)}$ for the minimum, the method proceeds by the iterations

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - \left(\mathbf{J_r}^\mathsf{T} \mathbf{J_r}\right)^{-1} \mathbf{J_r}^\mathsf{T} \mathbf{r}\left(\boldsymbol{\beta}^{(s)}\right),$$

where, if $\mathbf{r}$ and $\boldsymbol{\beta}$ are column vectors, the entries of the Jacobian matrix are

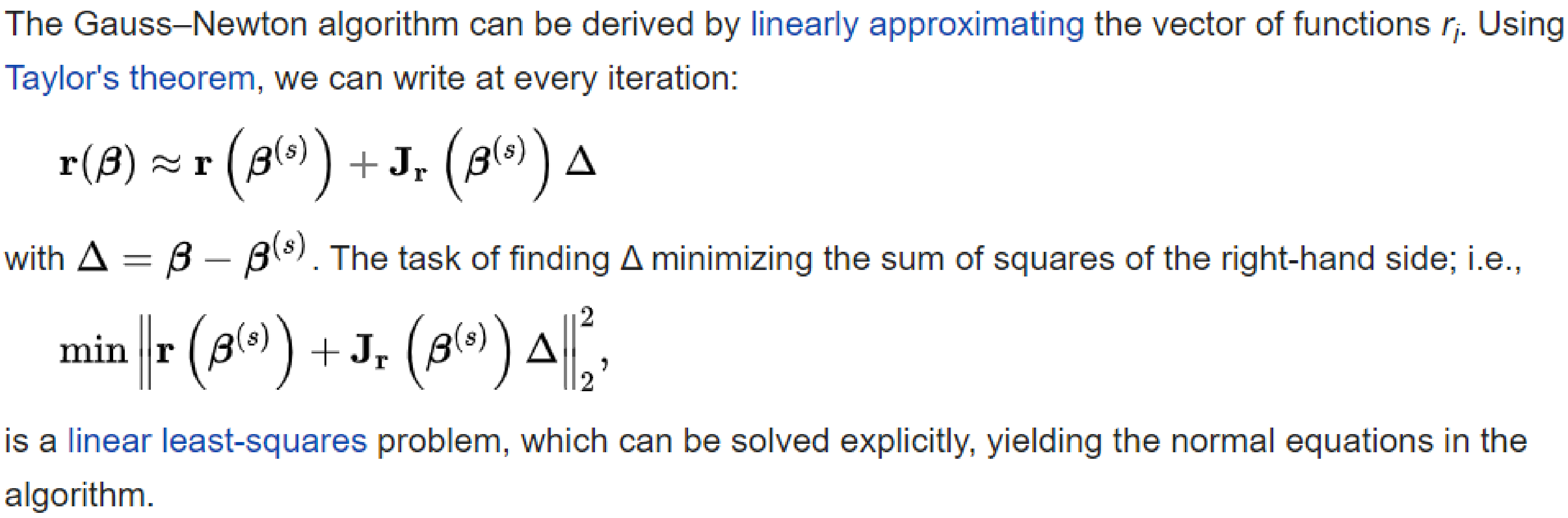
$$(\mathbf{J_r})_{ij} = \frac{\partial r_i\left(\boldsymbol{\beta}^{(s)}\right)}{\partial \beta_j},$$

and the symbol $^\mathsf{T}$ denotes the matrix transpose.

https://en.wikipedia.org/wiki/Gauss-Newton_algorithm

Gauss-Newton optimization

- Review of Gauss-Newton method



The Gauss–Newton algorithm can be derived by linearly approximating the vector of functions r_i. Using Taylor's theorem, we can write at every iteration:

$$\mathbf{r}(\boldsymbol\beta) \approx \mathbf{r}\left(\boldsymbol\beta^{(s)}\right) + \mathbf{J_r}\left(\boldsymbol\beta^{(s)}\right)\Delta$$

with $\Delta = \boldsymbol\beta - \boldsymbol\beta^{(s)}$. The task of finding Δ minimizing the sum of squares of the right-hand side; i.e.,

$$\min \left\|\mathbf{r}\left(\boldsymbol\beta^{(s)}\right) + \mathbf{J_r}\left(\boldsymbol\beta^{(s)}\right)\Delta\right\|_2^2,$$

is a linear least-squares problem, which can be solved explicitly, yielding the normal equations in the algorithm.

https://en.wikipedia.org/wiki/Gauss-Newton_algorithm

Convergence rate of Gauss-Newton method

- Quasi-qu*a*dratic convergence

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\| = O\Big(F(\boldsymbol{x}^\star)\|\boldsymbol{x}_k - \boldsymbol{x}^\star\| + \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|^2\Big)$$

Error of at the next iteration

Error of the current iteration

Residual of the optimal solution

Point-2-plane distance

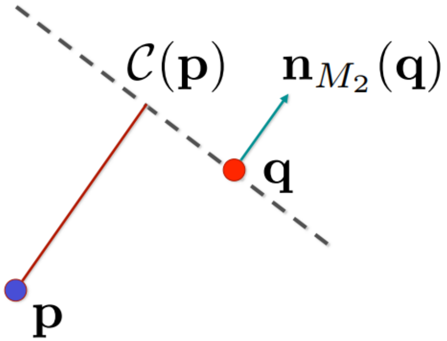
- Gauss-Newton leads to the following optimization problem

$$\min \sum_i [\mathbf{n}_i \cdot (\overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i) + d_i]^2$$

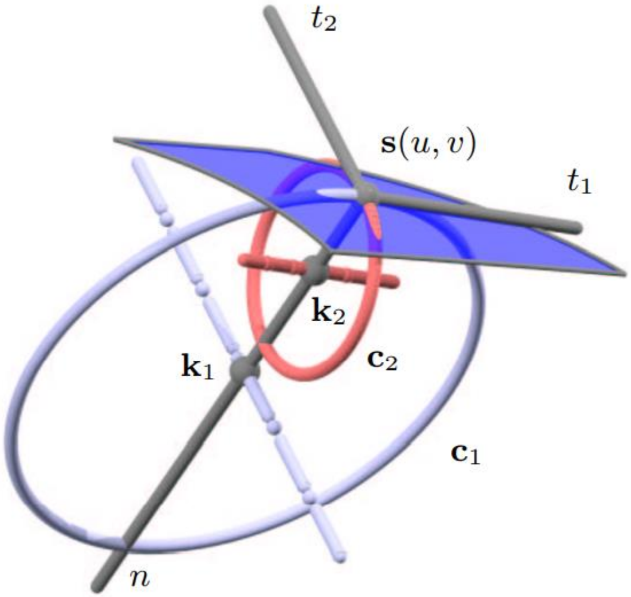
where c gives a linear parameterization of SO(3)

Using point-to-plane distance instead of point-to-point allows flat regions slide along each other

[Chen and Medioni 91]



Squared distance function

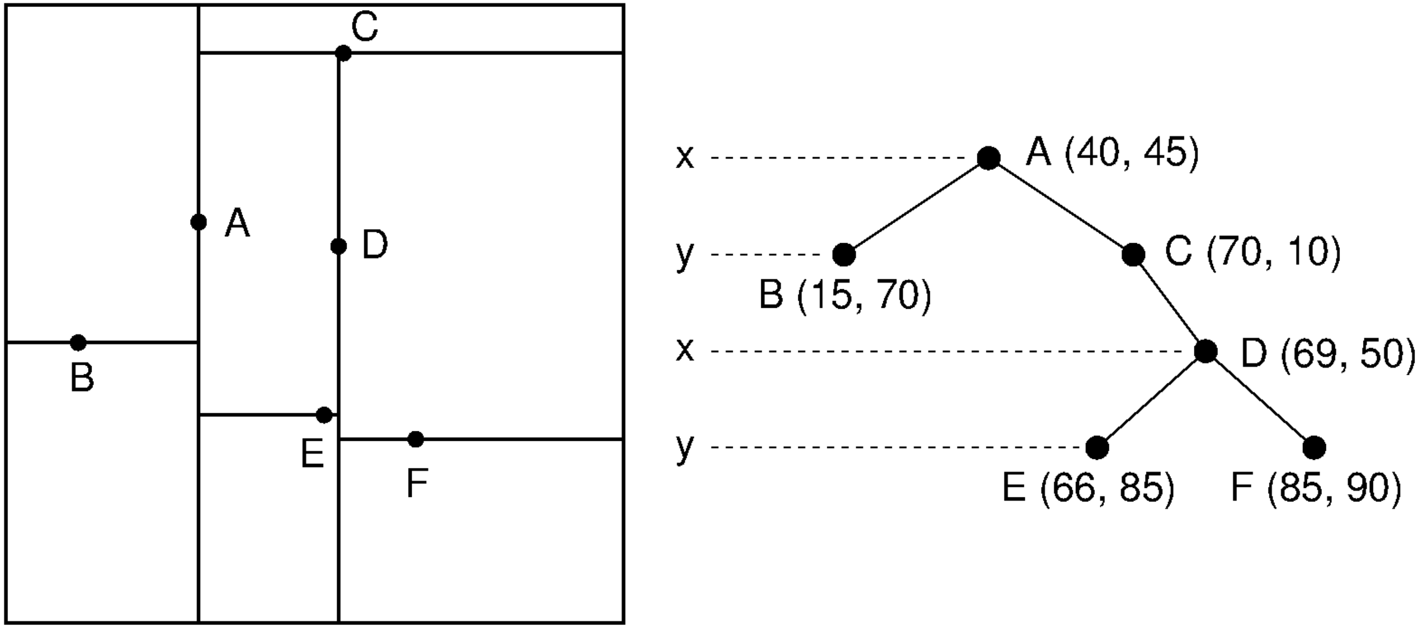


$$F_d(x_1, x_2, x_3) = \frac{d}{d - \varrho_1} x_1^2 + \frac{d}{d - \varrho_2} x_2^2 + x_3^2.$$

Geometry of the Squared Distance Function to Curves and Surfaces. Helmut Pottmann and Michael Hofer.

Practical considerations

- Nearest neighbor computation (Kdtree)

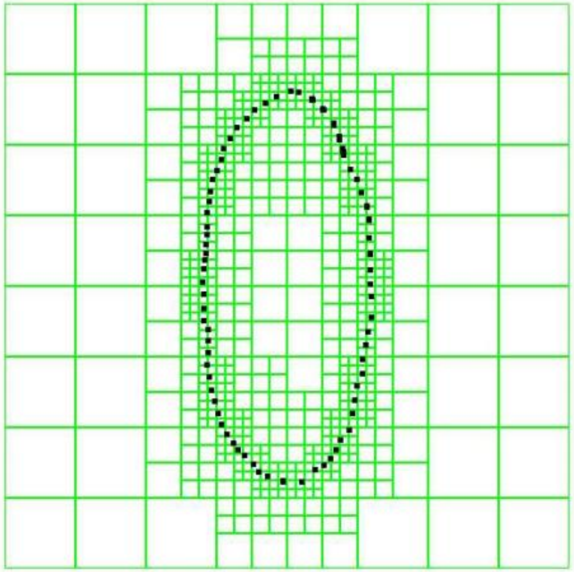


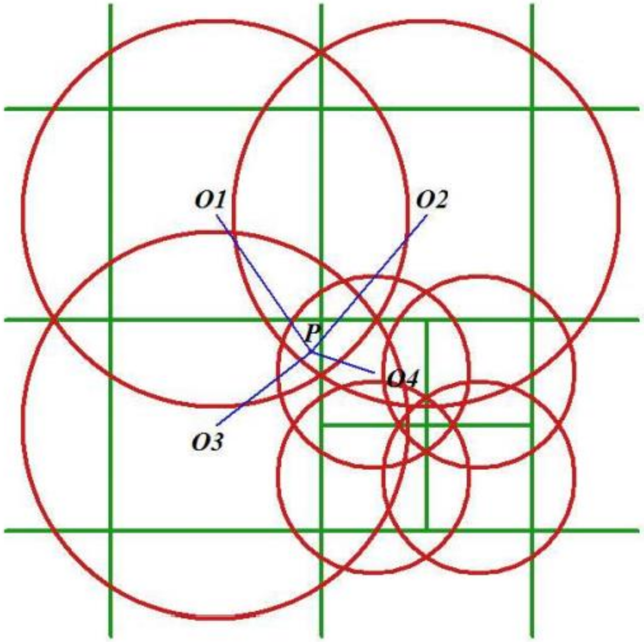
(a)

(b)

Squared distance field

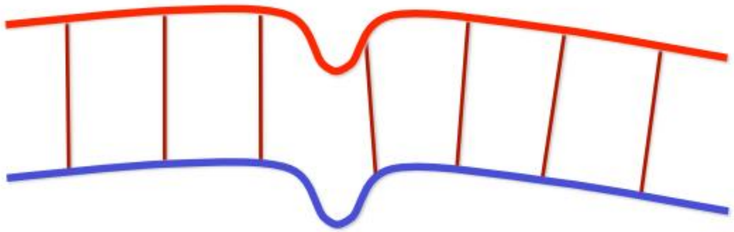
[Pottmann et al. 06]



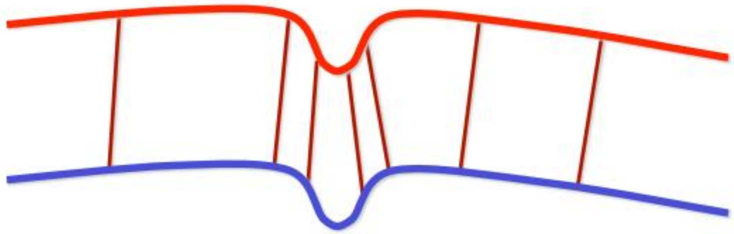


Stable Sampling [Gelfand et al. 2003]

- Select samples that constrain all degrees of freedom of the rigid-body transformation

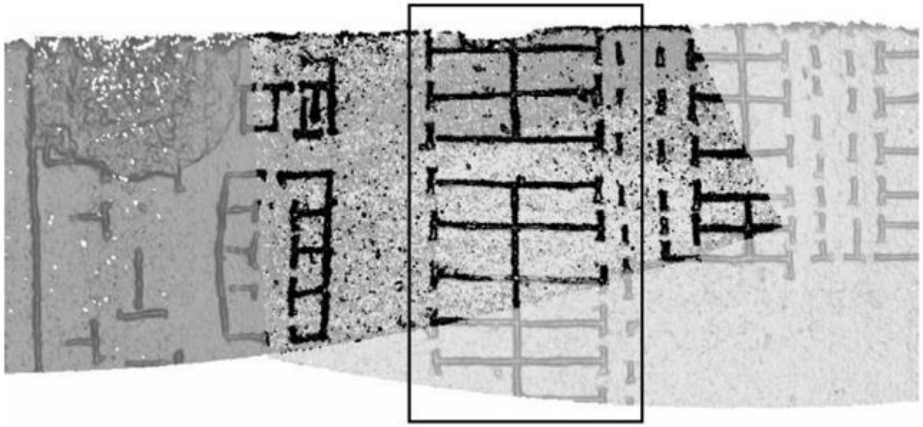


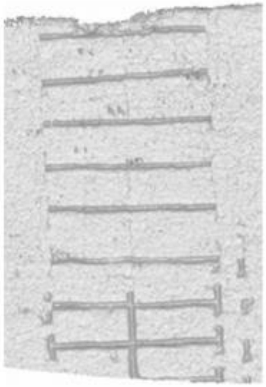
Uniform Sampling



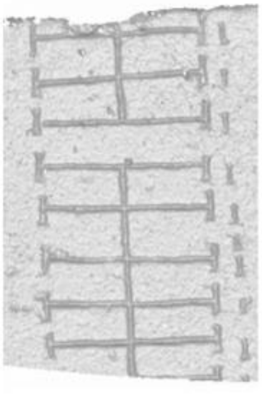
Stable Sampling

Stable Sampling [Gelfand et al. 2003]





Uniform sampling

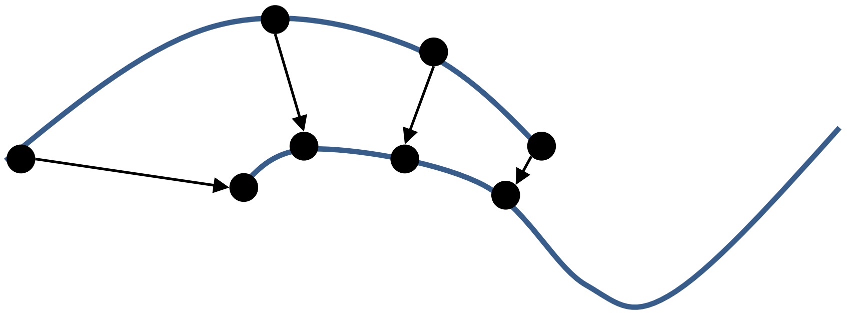


Stable sampling

Partial Overlaps

Registration under robust functions

- Use a robust norm under the point-2-plane distance metric



$$\min_{R,\boldsymbol{t}} \sum_{i=1}^{N} \rho(|(R\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{f}_i)^T \boldsymbol{n}_i|)$$

$$\rho_{GM}(t) = \frac{t^2}{\sigma^2 + t^2} \qquad \rho_1(t) = |t| \qquad \rho_2(t) = t^2$$

Optimization

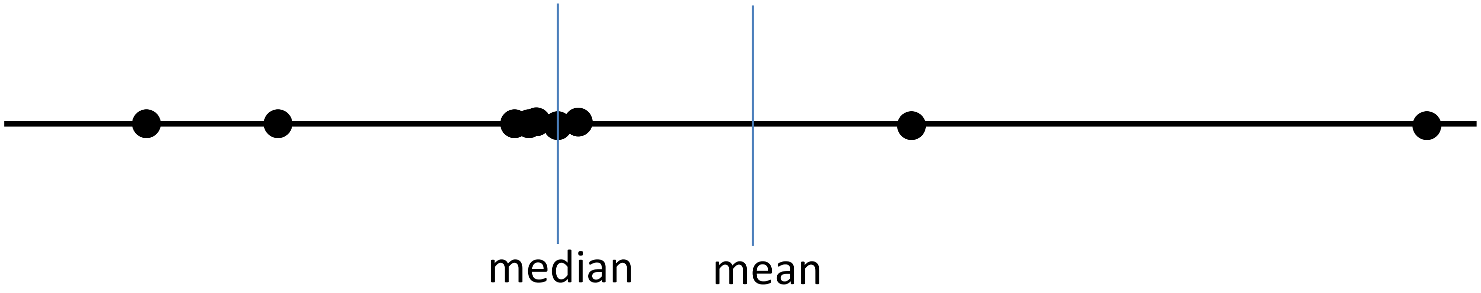
- Still alternate between optimizing the correspondences and optimizing the transformation
- Optimization strategy I: Gauss-Newton optimization
- Optimization strategy II: reweighted least squares

$$\min_{R,\boldsymbol{t}} \sum_{i=1}^{N} w_i |(R\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{f}_i)^T \boldsymbol{n}_i|^2$$

where

$$w_i = \frac{\rho(|(R\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{f}_i)^T \boldsymbol{n}_i|)}{|(R\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{f}_i)^T \boldsymbol{n}_i|^2}$$

Mean versus Median



- Mean

$$\min_x \sum_{i=1}^{N} (x - x_i)^2$$

- Median

$$\min_x \sum_{i=1}^{N} |x - x_i|$$

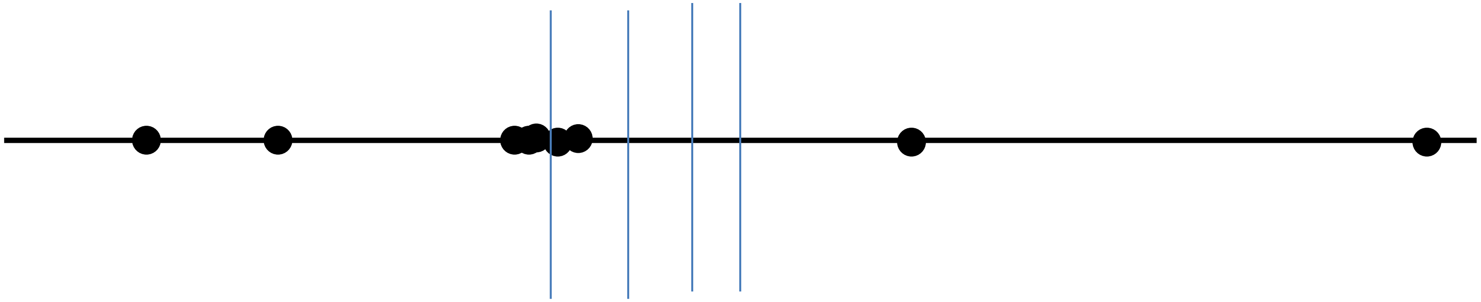
Median computation

- Weighted average

$$\min_x \sum_{i=1}^{N} w_i |x - x_i|^2 \qquad x^\star = \sum_{i=1}^{N} w_i x_i / \sum_{i=1}^{N} w_i$$

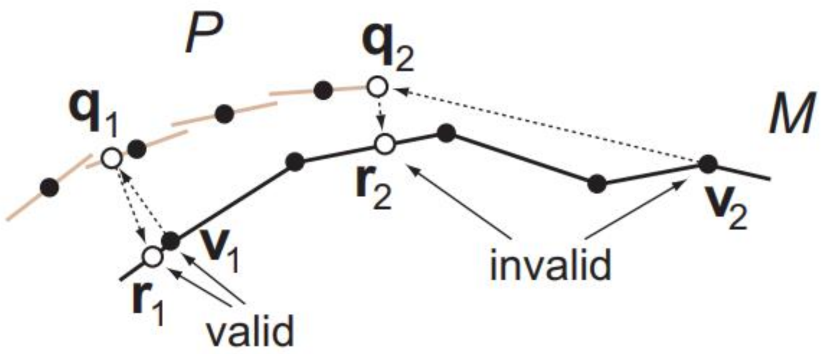
- Weighting

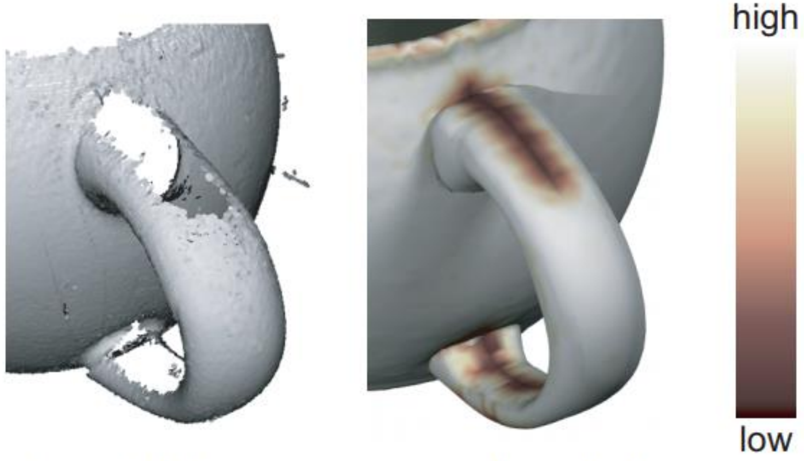
$$w_i = 1/|x^\star - x_i|$$



Bi-directional pruning

[Mitra et al. 05]



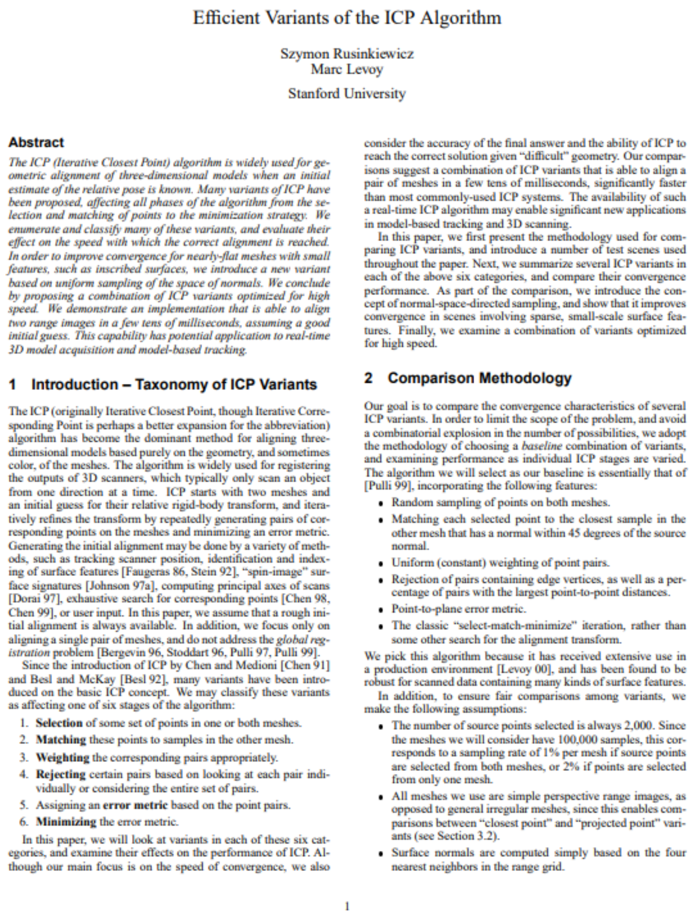


input data

warped model

Efficient variants of ICP Registration

- Selection of points
- Matching points
- Weighting of pairs
- Rejecting pairs
- Error Metric and Minimization



Efficient Variants of the ICP Algorithm

Szymon Rusinkiewicz
Marc Levoy

Stanford University

Abstract

The ICP (Iterative Closest Point) algorithm is widely used for geometric alignment of three-dimensional models when an initial estimate of the relative pose is known. Many variants of ICP have been proposed, affecting all phases of the algorithm from the selection and matching of points to the minimization strategy. We enumerate and classify many of these variants, and evaluate their effect on the speed with which the correct alignment is reached. In order to improve convergence for nearly-flat meshes with small features, such as inscribed surfaces, we introduce a new variant based on uniform sampling of the space of normals. We conclude by proposing a combination of ICP variants optimized for high speed. We demonstrate an implementation that is able to align two range images in a few tens of milliseconds, assuming a good initial guess. This capability has potential application to real-time 3D model acquisition and model-based tracking.

1 Introduction – Taxonomy of ICP Variants

The ICP (originally Iterative Closest Point, though Iterative Corresponding Point is perhaps a better expansion for the abbreviation) algorithm has become the dominant method for aligning three-dimensional models based purely on the geometry, and sometimes color, of the meshes. The algorithm is widely used for registering the outputs of 3D scanners, which typically only scan an object from one direction at a time. ICP starts with two meshes and an initial guess for their relative rigid-body transform, and iteratively refines the transform by repeatedly generating pairs of corresponding points on the meshes and minimizing an error metric. Generating the initial alignment may be done by a variety of methods, such as tracking scanner position, identification and indexing of surface features [Faugeras 86, Stein 92], "spin-image" surface signatures [Johnson 97a], computing principal axes of scans [Dorai 97], exhaustive search for corresponding points [Chen 98, Chen 99], or user input. In this paper, we assume that a rough initial alignment is always available. In addition, we focus only on aligning a single pair of meshes, and do not address the *global registration* problem [Bergevin 96, Stoddart 96, Pulli 97, Pulli 99].

Since the introduction of ICP by Chen and Medioni [Chen 91] and Besl and McKay [Besl 92], many variants have been introduced on the basic ICP concept. We may classify these variants as affecting one of six stages of the algorithm:

1. **Selection** of some set of points in one or both meshes.
2. **Matching** these points to samples in the other mesh.
3. **Weighting** the corresponding pairs appropriately.
4. **Rejecting** certain pairs based on looking at each pair individually or considering the entire set of pairs.
5. Assigning an **error metric** based on the point pairs.
6. **Minimizing** the error metric.

In this paper, we will look at variants in each of these six categories, and examine their effects on the performance of ICP. Although our main focus is on the speed of convergence, we also consider the accuracy of the final answer and the ability of ICP to reach the correct solution given "difficult" geometry. Our comparisons suggest a combination of ICP variants that is able to align a pair of meshes in a few tens of milliseconds, significantly faster than most commonly-used ICP systems. The availability of such a real-time ICP algorithm may enable significant new applications in model-based tracking and 3D scanning.

In this paper, we first present the methodology used for comparing ICP variants, and introduce a number of test scenes used throughout the paper. Next, we summarize several ICP variants in each of the above six categories, and compare their convergence performance. As part of the comparison, we introduce the concept of normal-space-directed sampling, and show that it improves convergence in scenes involving sparse, small-scale surface features. Finally, we examine a combination of variants optimized for high speed.

2 Comparison Methodology

Our goal is to compare the convergence characteristics of several ICP variants. In order to limit the scope of the problem, and avoid a combinatorial explosion in the number of possibilities, we adopt the methodology of choosing a *baseline* combination of variants, and examining performance as individual ICP stages are varied. The algorithm we will select as our baseline is essentially that of [Pulli 99], incorporating the following features:

- Random sampling of points on both meshes.
- Matching each selected point to the closest sample in the other mesh that has a normal within 45 degrees of the source normal.
- Uniform (constant) weighting of point pairs.
- Rejection of pairs containing edge vertices, as well as a percentage of pairs with the largest point-to-point distances.
- Point-to-plane error metric.
- The classic "select-match-minimize" iteration, rather than some other search for the alignment transform.

We pick this algorithm because it has received extensive use in a production environment [Levoy 00], and has been found to be robust for scanned data containing many kinds of surface features.

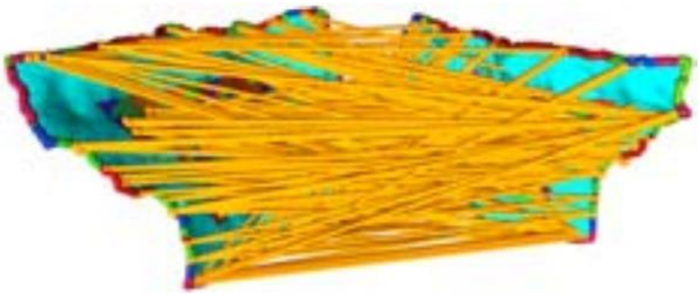
In addition, to ensure fair comparisons among variants, we make the following assumptions:

- The number of source points selected is always 2,000. Since the meshes we will consider have 100,000 samples, this corresponds to a sampling rate of 1% per mesh if source points are selected from both meshes, or 2% if points are selected from only one mesh.
- All meshes we use are simple perspective range images, as opposed to general irregular meshes, since this enables comparisons between "closest point" and "projected point" variants (see Section 3.2).
- Surface normals are computed simply based on the four nearest neighbors in the range grid.

Global Matching

Global matching

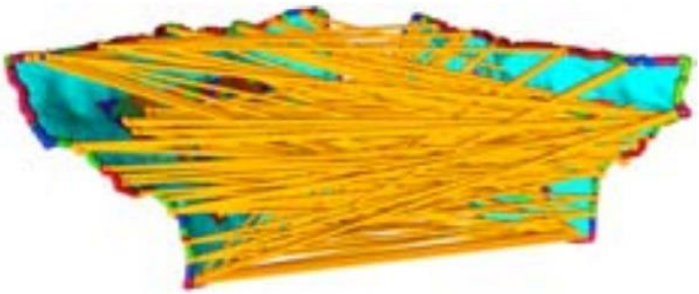
Reassembling fractured surfaces
[Huang et al. 06]



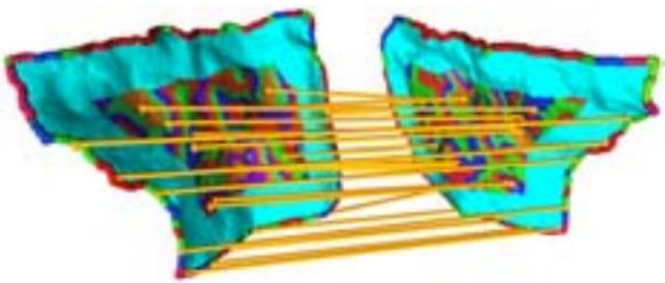
Feature extraction + Feature matching

Global matching

Reassembling fractured surfaces
[Huang et al. 06]

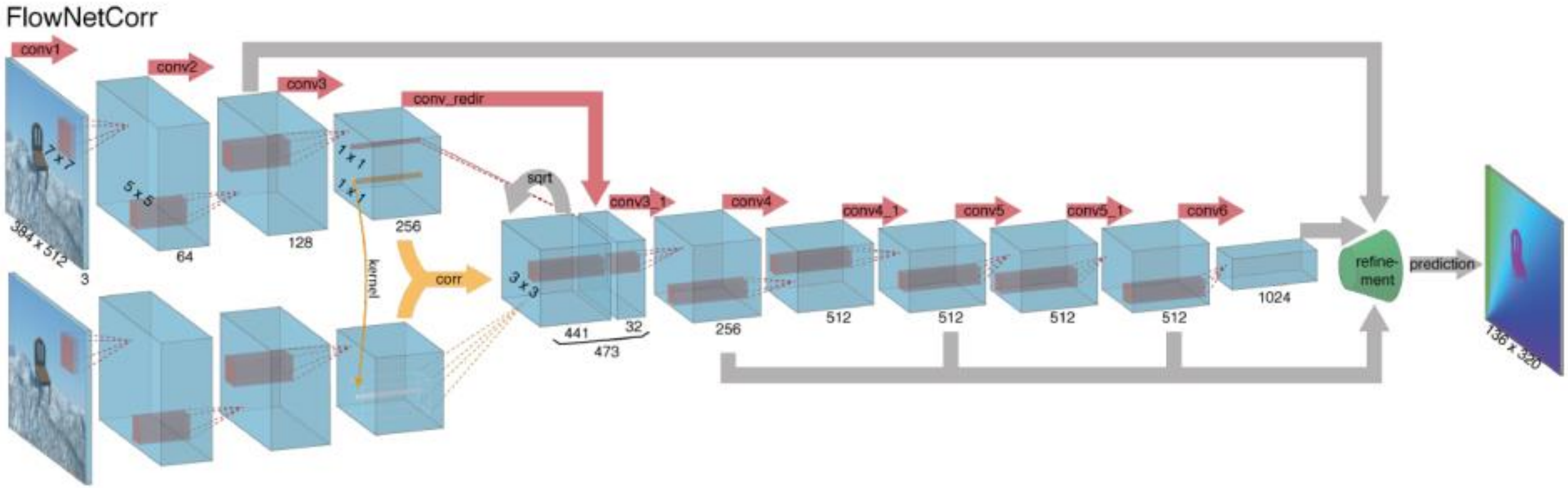


Feature extraction + Feature matching



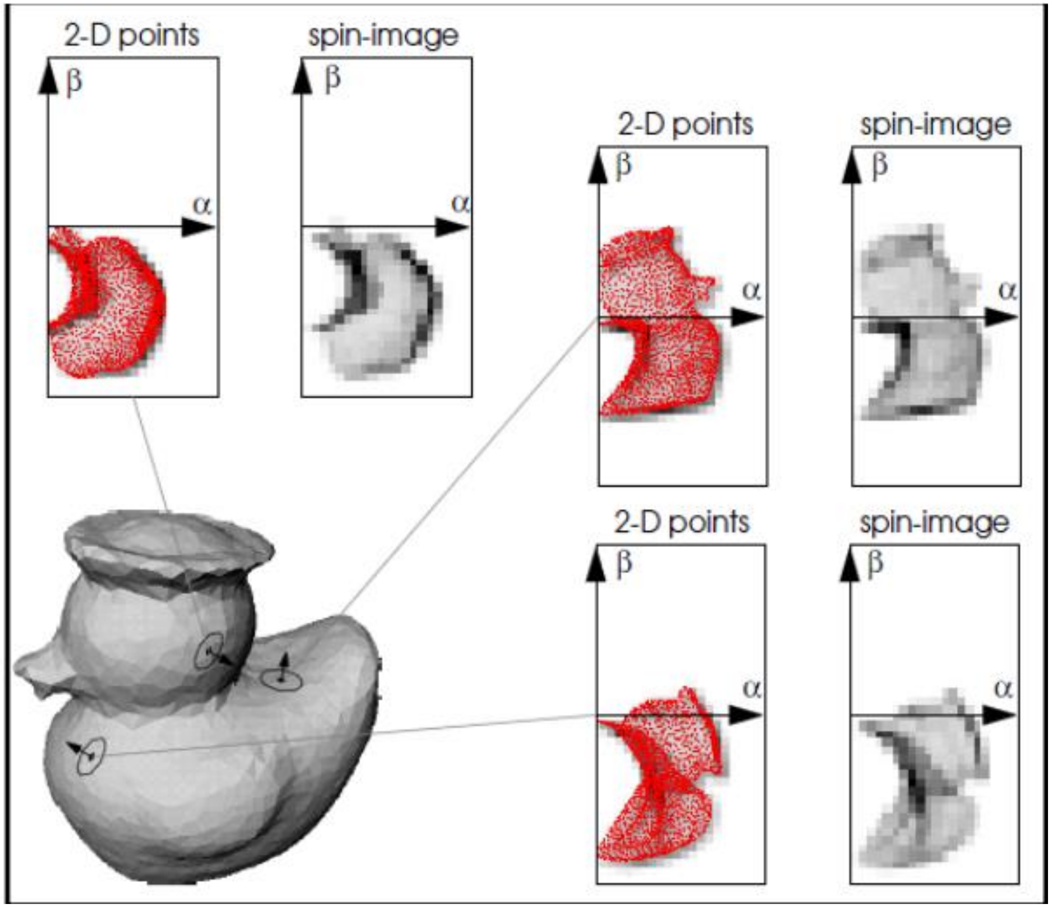
Relative pose extraction

Relative poses/pair-wise matching in the neural network era



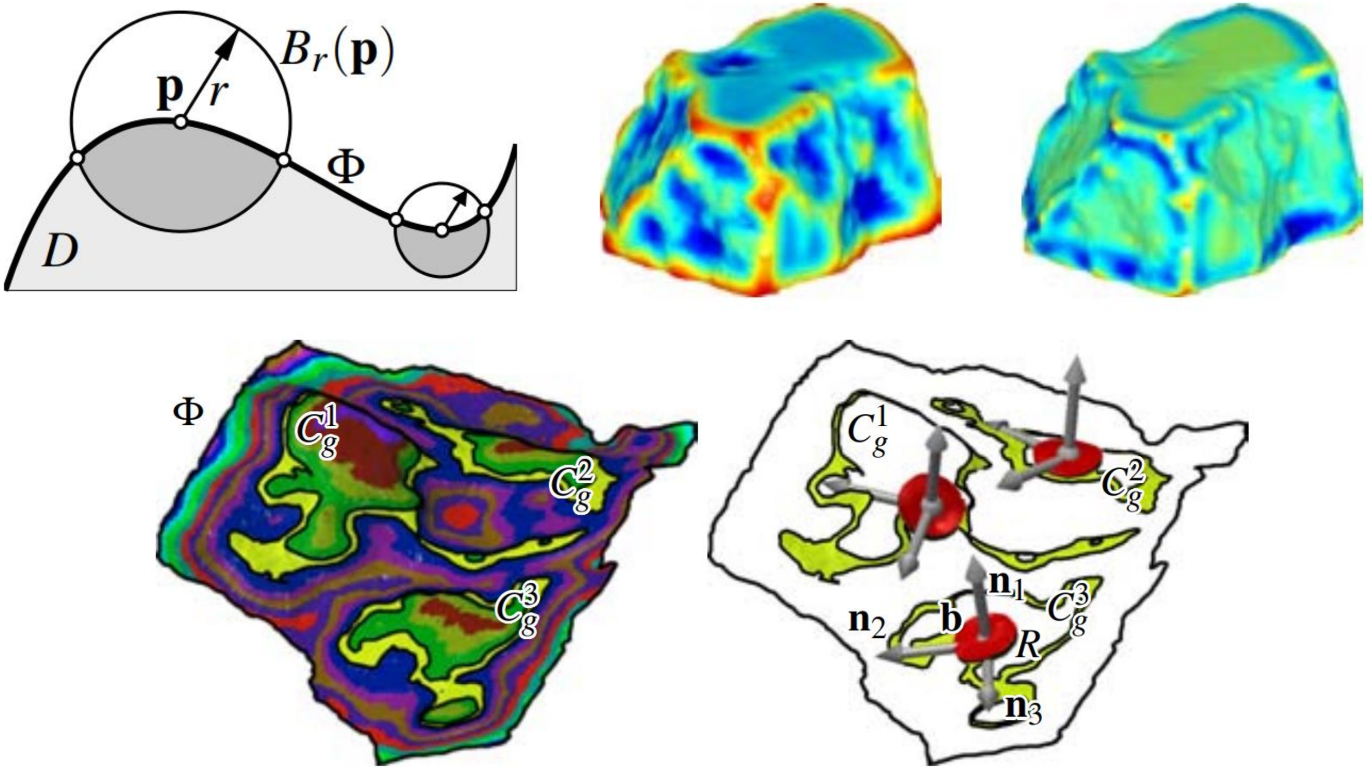
FlowNet: Learning Optical Flow with Convolutional Networks
[Fischer et al. 15]

Feature descriptors – Spin images



https://www.ri.cmu.edu/publications/spin-images-a-representation-for-3-d-surface-matching/

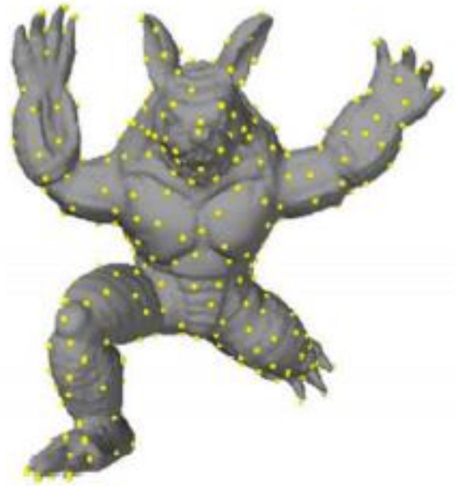
Feature descriptors – integral invariants



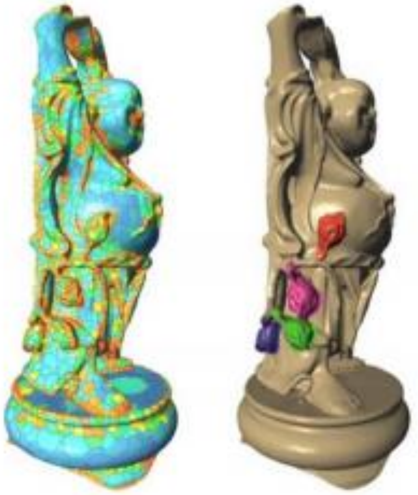
Related to mean curvature and robust

Integral Invariants for Robust Geometry Processing. Pottmann et al., 09.

Other features



3D SIFT

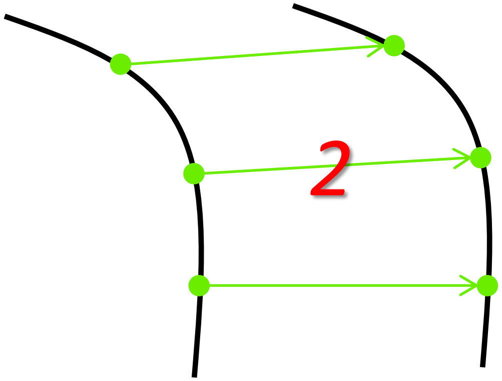


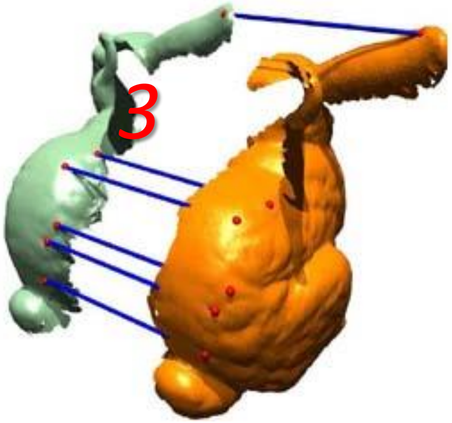
Patch features

A 3-Dimensional Sift Descriptor and Its Application to Action Recognition. Scovanner et al., 07. ACM MM
Salient Geometric Features for Partial Shape Matching and Similarity. Gal and Cohen-Or’ 06. ACM TOG

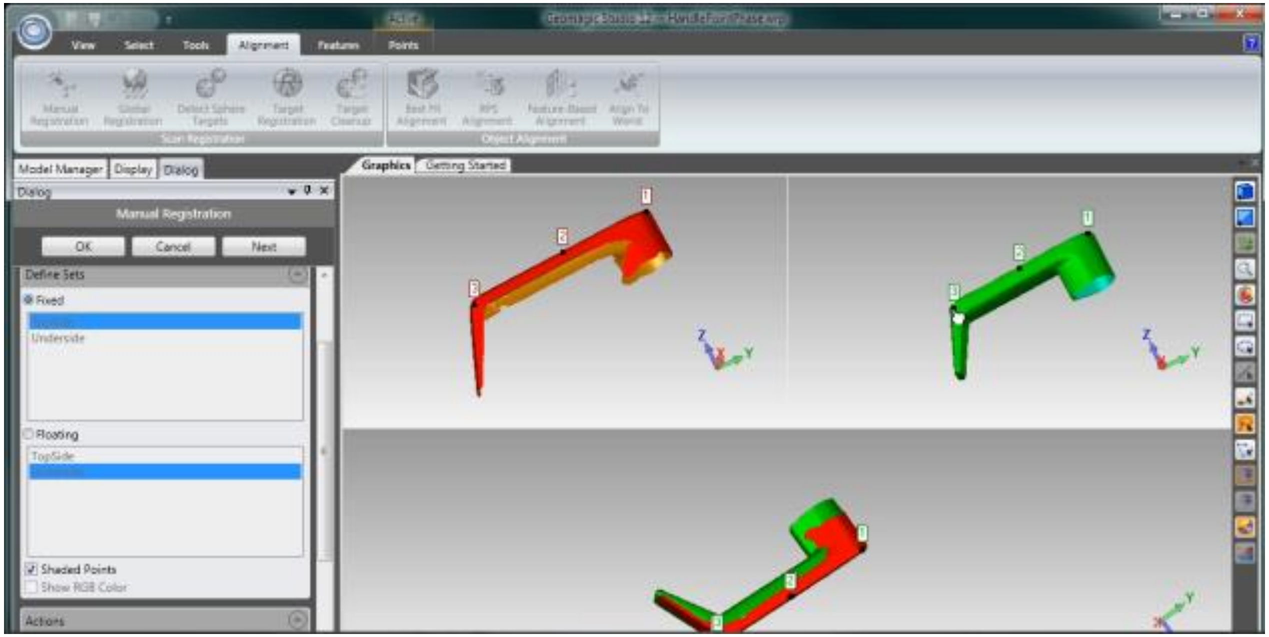
Global matching --- RANSAC

- How many point-pairs specify a rigid transform?
  - In R^2?
  - In R^3?
- Additional constraints?
  - Distance preserving
  - Stability?





Software



Dialog

Manual Registration

● Fixed

Underside

○ Floating

TopSide

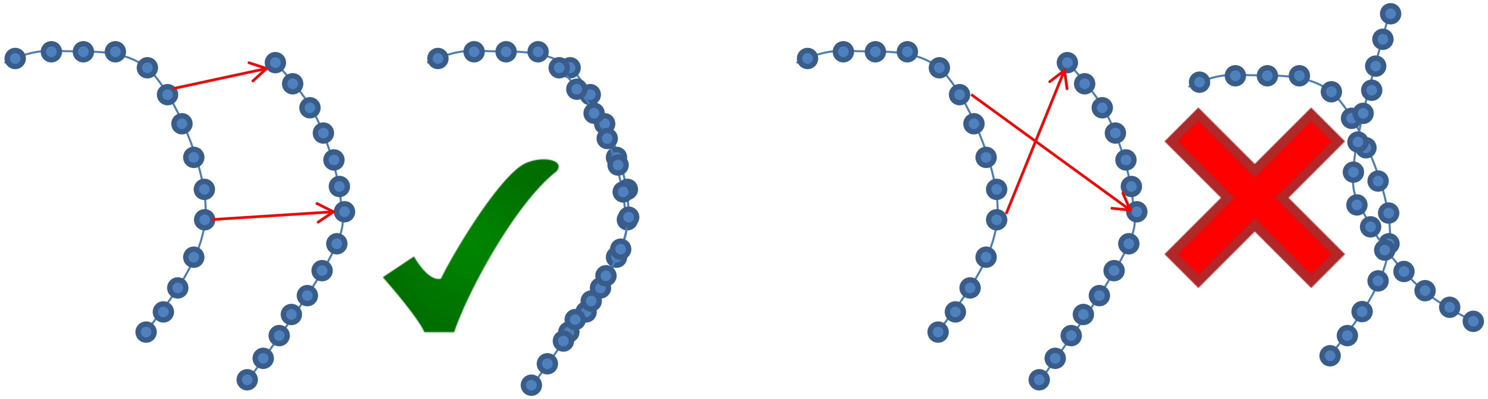
☑ Shaded Points

□ Show RGB Color

Geomagic

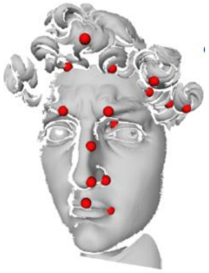
RANSAC

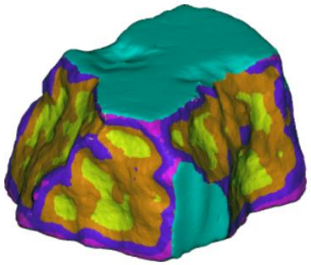
- Preprocessing: sample each object
- Recursion:
  - Step I: Sample three (two) pairs, check distance constraints
  - Step II: Fit a rigid transform
  - Step III: Check how many point pairs agree. If above threshold, terminates; otherwise goes to Step I



RANSAC --- facts

- Sampling
  - Feature point detection

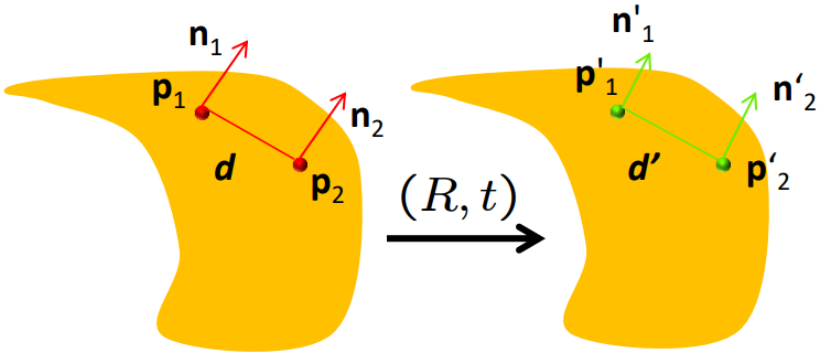




- Correspondences
  - Use feature descriptors
  - The candidate correspondences $m << O(n^2)$
  - Denote the success rate p ¼ $\frac{n}{m}$
- *Basic* analysis
  - The probability of having a valid triplet p^3
  - The probability of having a valid triplet in N trials is $1-(1-p^3)^N$

RANSAC+

- How many surfel (position + normal) correspondences specify a rigid transform?



Constraints:

1. $\|\mathbf{p}_1 - \mathbf{p}_2\| \approx \|\mathbf{p}'_1 - \mathbf{p}'_2\|$
2. $\angle(\mathbf{n}_1, \mathbf{d}) = \angle(\mathbf{n}'_1, \mathbf{d}')$
3. $\angle(\mathbf{n}_2, \mathbf{d}) = \angle(\mathbf{n}'_2, \mathbf{d}')$
4. $\angle(\mathbf{n}_1, \mathbf{n}_2) = \angle(\mathbf{n}'_1, \mathbf{n}'_2)$

$$\mathbf{t} = \frac{\mathbf{p}'_1 + \mathbf{p}'_2}{2} - \frac{\mathbf{p}_1 + \mathbf{p}_2}{2}$$

$$[\mathbf{n}_1, \mathbf{n}_2, d] \xrightarrow{R} [\mathbf{n}'_1, \mathbf{n}'_2, \mathbf{d}']$$

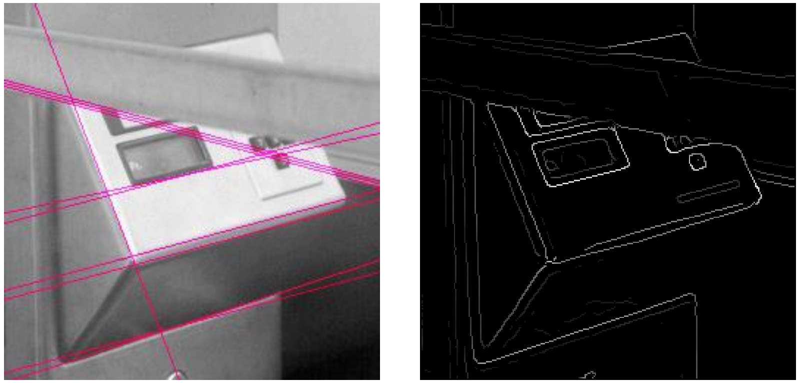
Reduce the number of trials from O(m³) to O(m²)

Success rate:

$$1 - (1 - p^2)^N$$

Hough transform for line fitting

- Line detection in an image
  - what is the line?
  - How many lines?
  - Point-line associations?



- **Hough Transform** is a voting technique that can be used to answer all of these questions
  - Record vote for each possible line on which each edge point lies
  - Look for lines that get many votes.

Voting

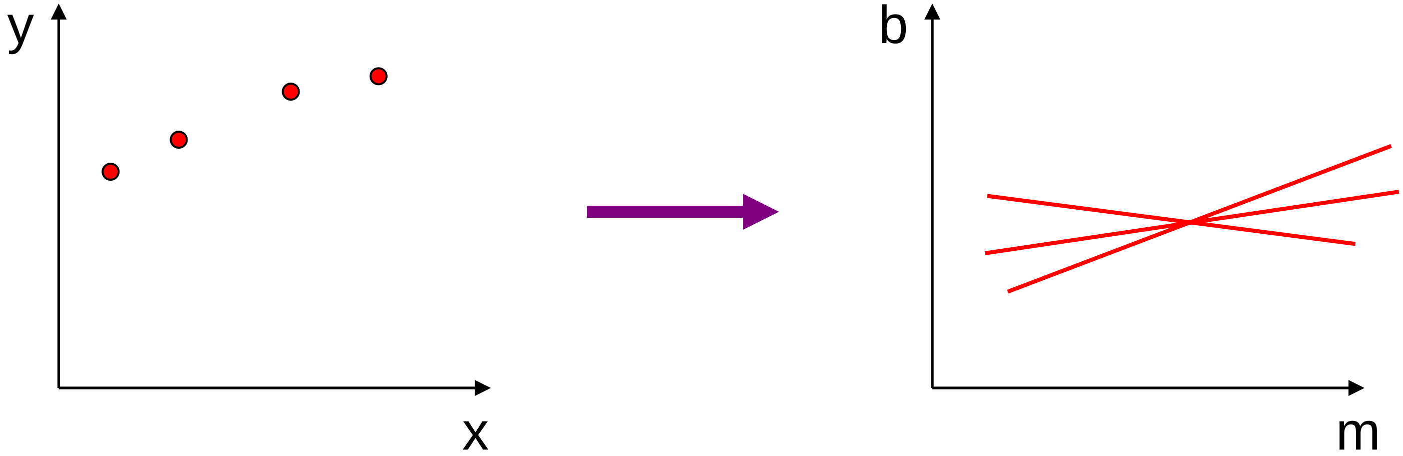
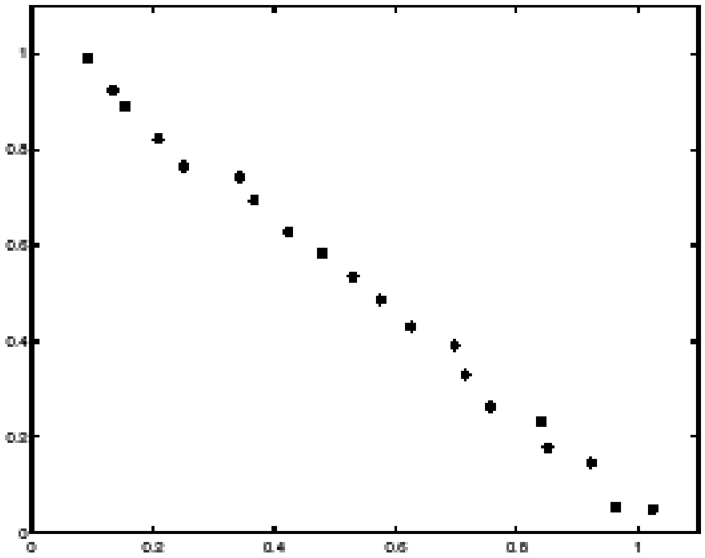
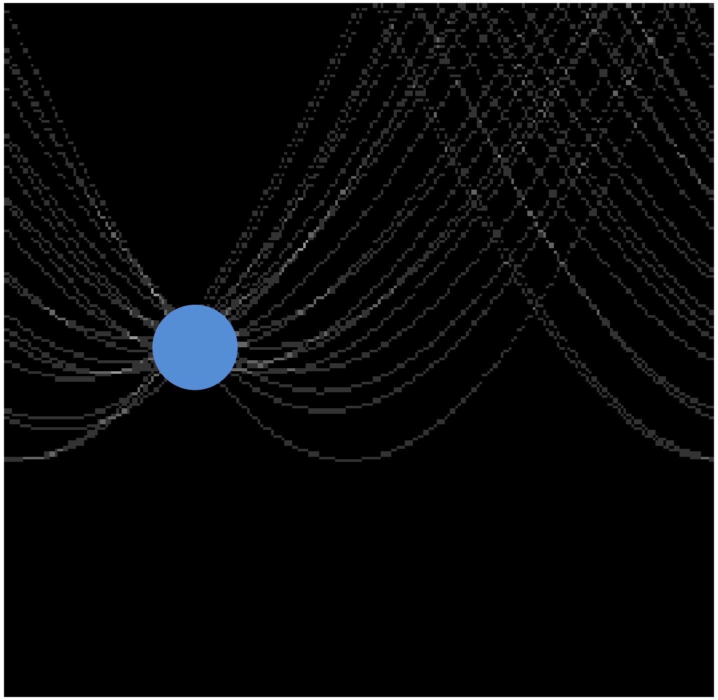


image space

Hough (parameter) space

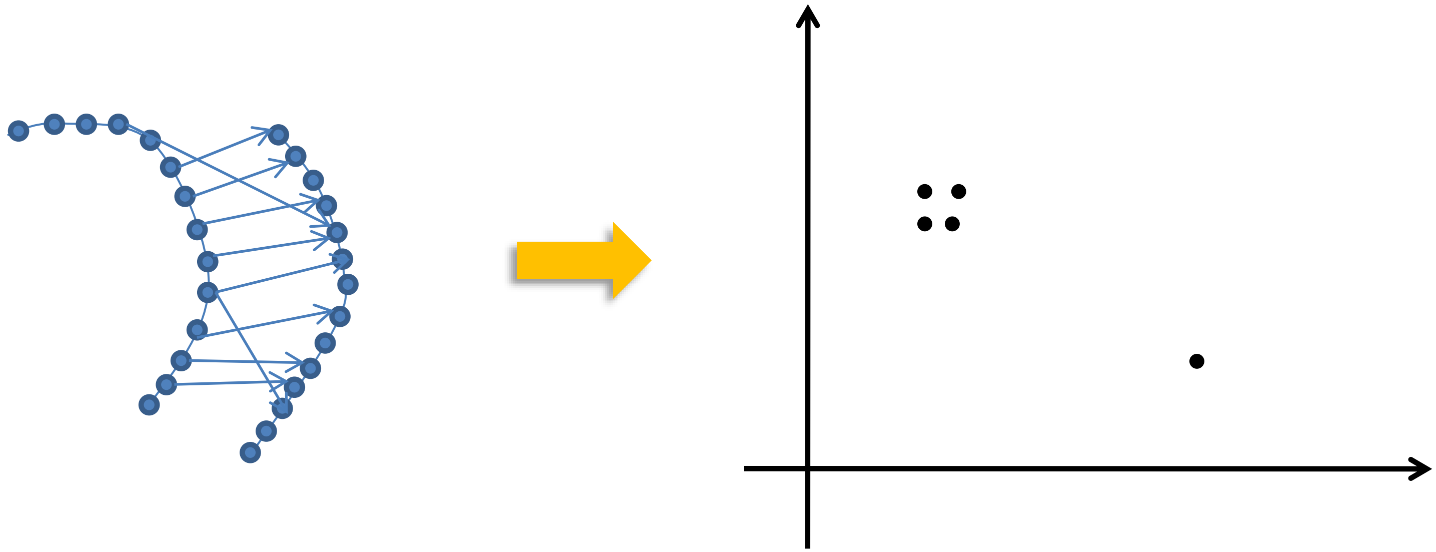
Clustering



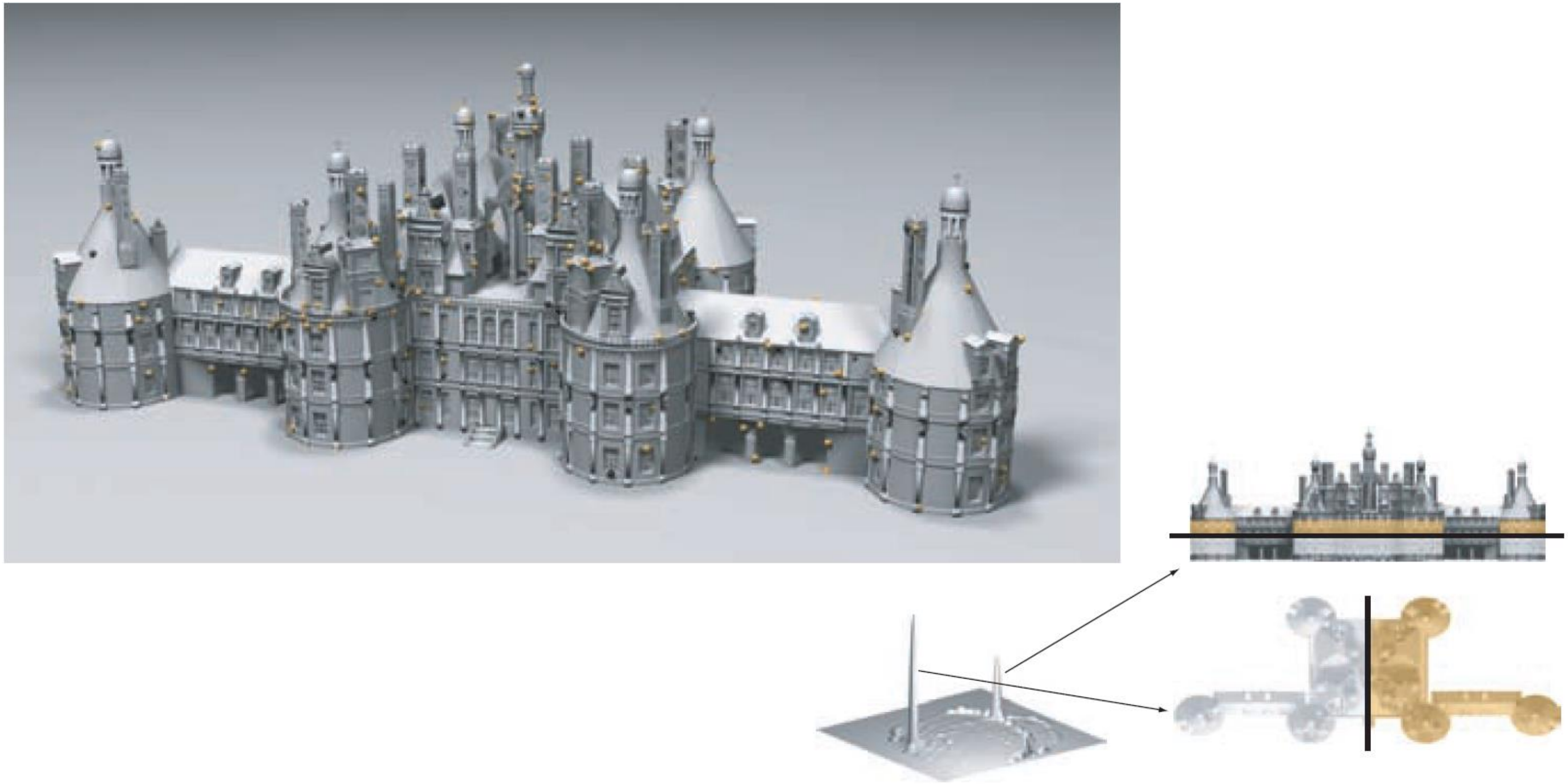


Rigid matching

- Rigid transform detection from feature correspondences



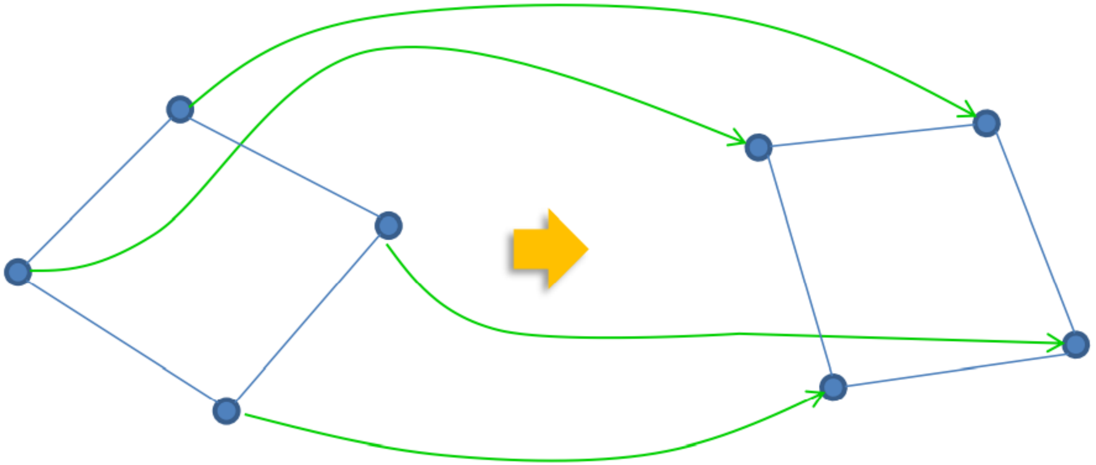
Symmetry detection



Partial and Approximate Symmetry Detection for 3D Geometry, N. Mitra, L. Guibas, and M. Pauly, SIGGRAPH’ 06

Spectral Approach

Distance preservation = Rigidity?

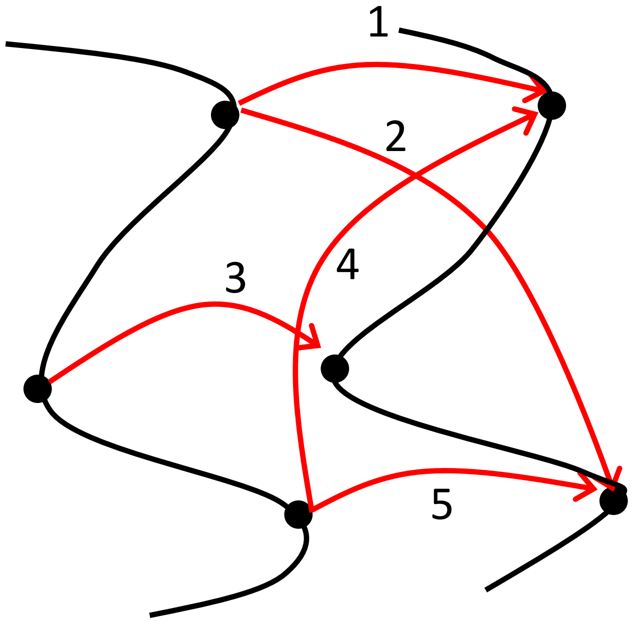


$P = \{\mathbf{p}_i\}$ $Q = \{\mathbf{q}_i\}$

$$\|\mathbf{p}_i - \mathbf{p}_j\| = \|\phi(\mathbf{p}_i) - \phi(\mathbf{p}_j)\| \quad \Longleftrightarrow \quad \phi(\mathbf{p}_i) = R \cdot \mathbf{p}_i + \mathbf{t}$$

$$\det(R) = -1$$

Spectral approach



Correspondences

0: Inconsistent, 1: Consistent

Correspondences

Correspondences

	1	2	3	4	5
1	1	0	1	0	1
2	0	1	0	1	0
3	1	0	1	0	1
4	0	1	0	1	0
5	1	0	1	0	1

Consistency matrix

A Spectral Technique for Correspondence Problems using Pairwise Constraints, M. Leordeanu and M. Hebert, ICCV 2005

Clique extraction

	1	2	3	4	5
1	1	0	1	0	1
2	0	1	0	1	0
3	1	0	1	0	1
4	0	1	0	1	0
5	1	0	1	0	1

Consistency matrix

→

permute

	1	3	5	2	4
1	1	1	1	0	0
3	1	1	1	0	0
5	1	1	1	0	0
2	0	0	0	1	1
4	0	0	0	1	1

Consistency matrix

Algorithm

- Step 1: Compute the maximum eigenvector ***v*** of ***C***

- Step 2: Sort the vertices based on magnitude of v and initialize the cluster

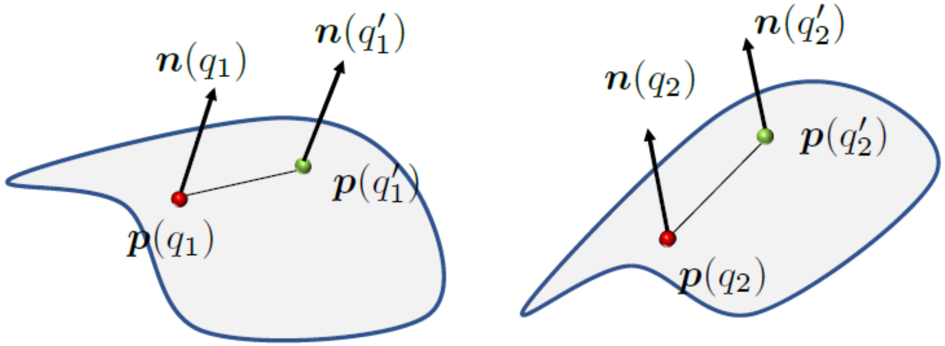
- Step 3: Incrementally insert vertices while checking the clique constraint

- Step 4: Stop if the size of the cluster is small, otherwise accept the cluster and go to Step 1

Geometric consistency

Also used for initializing correspondences

$$\Delta_1^2(c,c') := \|\boldsymbol{f}(q_1) - \boldsymbol{f}(q_2)\|^2 + \|\boldsymbol{f}(q_1') - \boldsymbol{d}(q_2')\|^2$$



$$\begin{aligned}
\Delta_2(c,c') &:= \|\boldsymbol{p}(q_1) - \boldsymbol{p}(q_1')\| - \|\boldsymbol{p}(q_2) - \boldsymbol{p}(q_2')\| \\
\Delta_3(c,c') &:= \angle(\boldsymbol{n}(q_1), \boldsymbol{n}(q_1')) - \angle(\boldsymbol{n}(q_2), \boldsymbol{n}(q_2')) \\
\Delta_4(c,c') &:= \angle(\boldsymbol{n}(q_1), \boldsymbol{p}(q_1)\boldsymbol{p}(q_1')) - \angle(\boldsymbol{n}(q_2), \boldsymbol{p}(q_2)\boldsymbol{p}(q_2')) \\
\Delta_5(c,c') &:= \angle(\boldsymbol{n}(q_1'), \boldsymbol{p}(q_1)\boldsymbol{p}(q_1')) - \angle(\boldsymbol{n}(q_2'), \boldsymbol{p}(q_2)\boldsymbol{p}(q_2'))
\end{aligned}$$

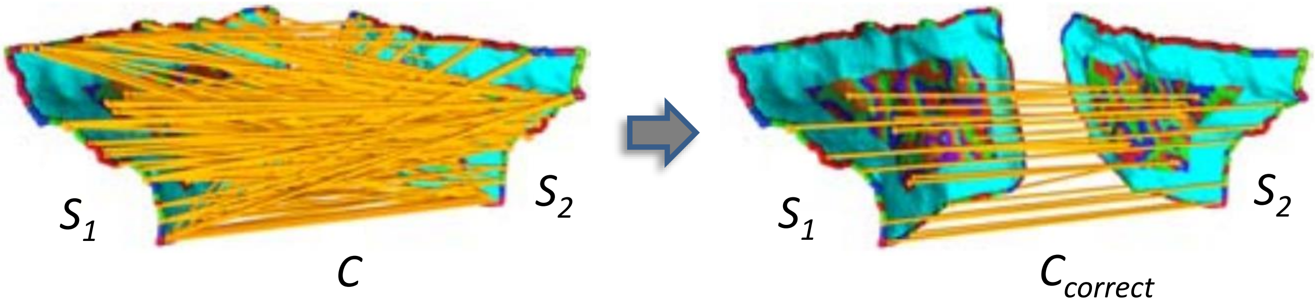
$$w_\gamma(c,c') = \exp\Big(-\frac{1}{2}\sum_{i=1}^{5}\big(\frac{\Delta_i(c,c')}{\gamma_i}\big)^2\Big)$$

Hybrid Method

Robust geometric matching

[Yang et al. 19]

Input:



Reweighted non-linear least squares

Loss term:

$$r_{(R,\boldsymbol{t})}(c) = \left(\|R\boldsymbol{p}(q_1) + \boldsymbol{t} - \boldsymbol{p}(q_2)\|^2 + \|R\boldsymbol{n}(q_1) - \boldsymbol{n}(q_2)\|^2\right)^{\frac{1}{2}}$$

Total objective term:

$$\min_{R,\boldsymbol{t}} \sum_{c \in C} r_{(R,\boldsymbol{t})}(c)$$

Can only tolerate 50% of incorrect correspondences

Spectral matching

$$\max_{\{x_c\}} \sum_{c,c'} w(c,c') x_c x_{c'}$$

$$\text{subject to } \sum_c x_c^2 = 1$$

$\uparrow$ Indicators associated with initial corres.

Can tolerate more incorrect correspondences
Not a clean separation between inliers/outliers

Spectral matching + reweighted least squares

$$\underset{\{x_c\},R,\boldsymbol{t}}{\text{maximize}} \sum_{c,c'\in\mathcal{C}} w_\gamma(c,c')x_c x_{c'}\big(\delta - r_{(R,\boldsymbol{t})}(c) - r_{(R,\boldsymbol{t})}(c')\big)$$

$$\text{subject to} \sum_{c\in\mathcal{C}} x_c^2 = 1$$

Correspondence pair score

Regression error

Optimization:

1. When R, ***t*** are fixed:

$$\max_{x_c} \sum_{c,c'} a_{cc'} x_c x_{c'} \quad \text{subject to} \sum_{c} x_c^2 = 1$$ → Leading eigenvector computation

$$a_{cc'} := w_\gamma(c,c')\big(\delta - r_{(R,\boldsymbol{t})}(c) - r_{(R,\boldsymbol{t})}(c')\big)$$

2. When {x_c} are fixed:

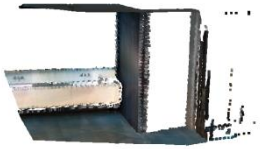
$$\min_{R,\boldsymbol{t}} \sum_{c\in\mathcal{C}} a_c r_{(R,\boldsymbol{t})}(c), \quad a_c := x_c \sum_{c'\in\mathcal{C}} w_\gamma(c,c')x_{c'}$$

Reduces to the standard setting of reweighted non-linear least squares

Training details in the paper

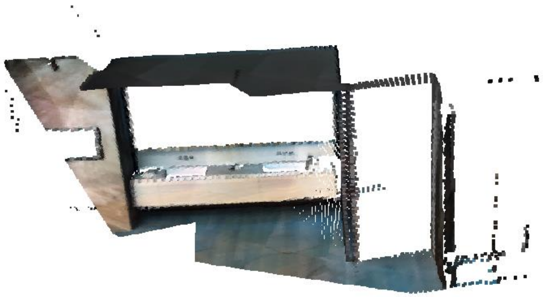
Side-by-side comparison

source



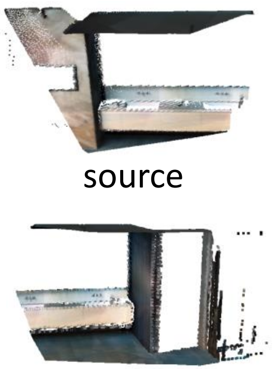
target

Input



Ground truth

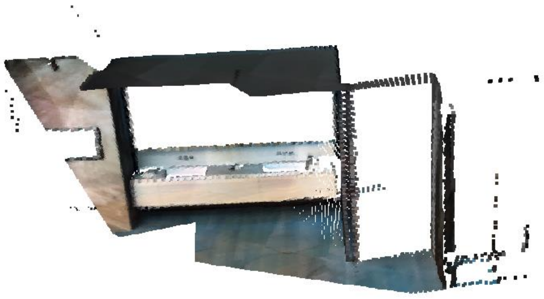
Side-by-side comparison



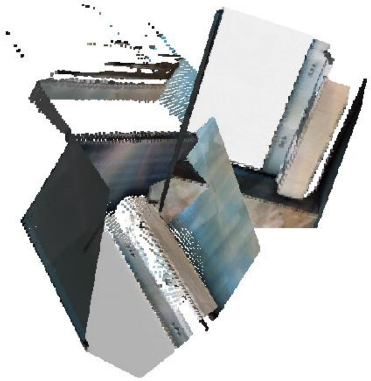
source

target

Input

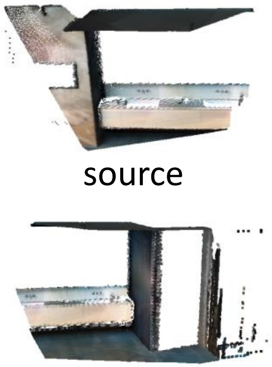


Ground truth



Spectral matching

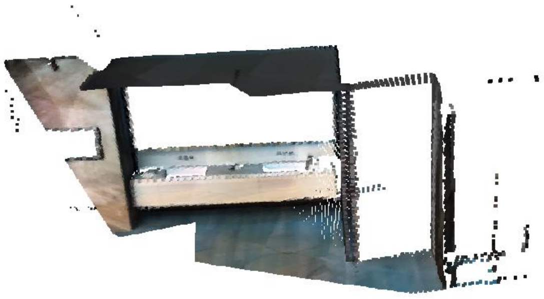
Side-by-side comparison



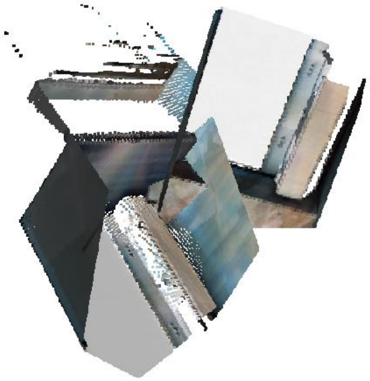
source

target

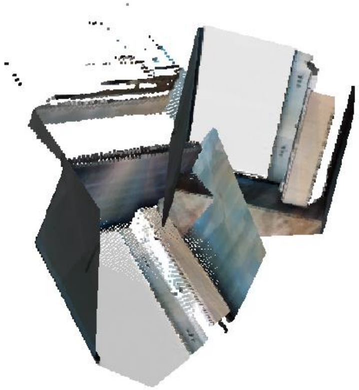
Input



Ground truth

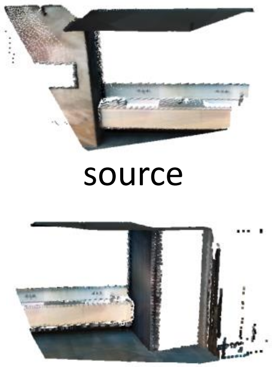


Spectral matching



Reweighted non-linear LS

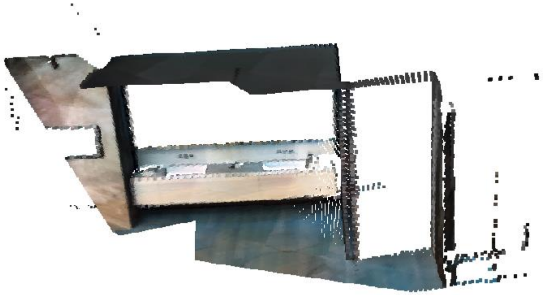
Side-by-side comparison



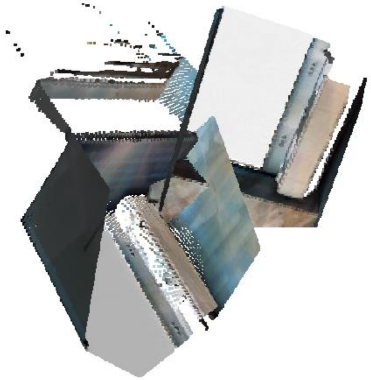
source

target

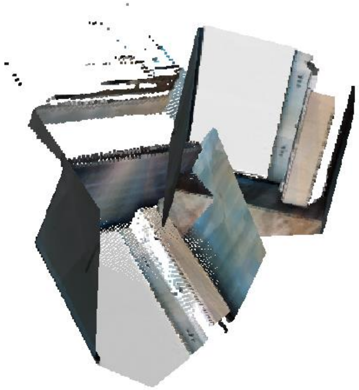
Input



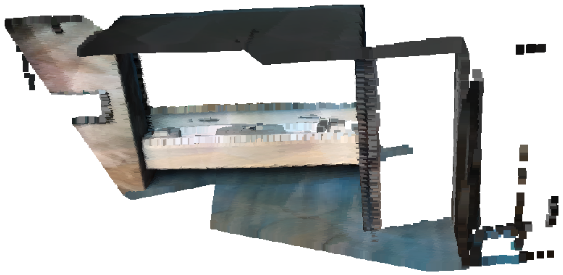
Ground truth



Spectral matching



Reweighted non-linear LS

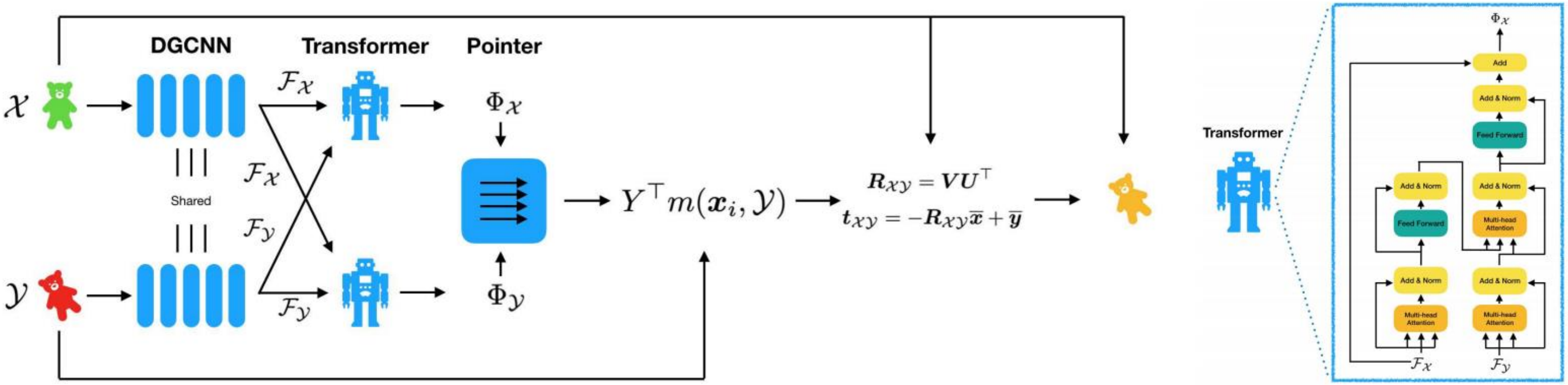


Spectral matching
+ Reweighted non-linear LS

Learning-based methods

Learning registration

[Wang and Solomon 19]



DGCNN
Transformer
Pointer
$\mathcal{X}$
$\mathcal{Y}$
Shared
$\mathcal{F}_\mathcal{X}$
$\mathcal{F}_\mathcal{Y}$
$\Phi_\mathcal{X}$
$\Phi_\mathcal{Y}$
$Y^\top m(\boldsymbol{x}_i, \mathcal{Y})$
$\boldsymbol{R}_{\mathcal{XY}} = \boldsymbol{V}\boldsymbol{U}^\top$
$\boldsymbol{t}_{\mathcal{XY}} = -\boldsymbol{R}_{\mathcal{XY}}\overline{\boldsymbol{x}} + \overline{\boldsymbol{y}}$
Transformer
Add
Add & Norm
Feed Forward
Multi-head Attention

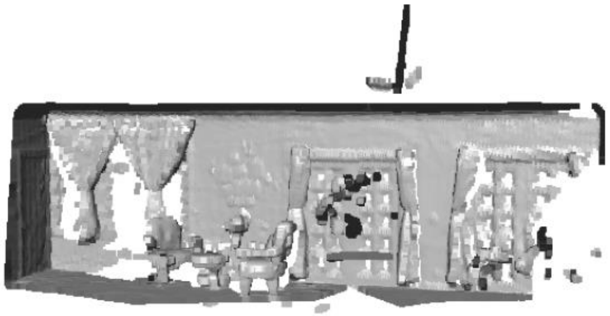
(a) Network architecture

(b) Transformer module

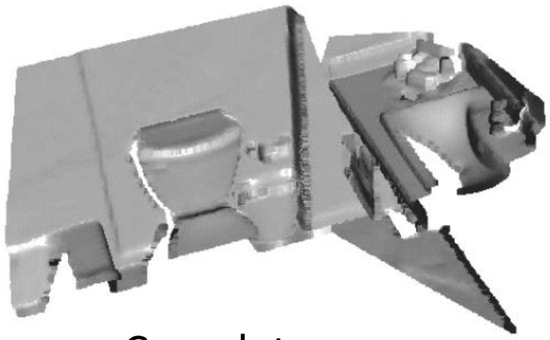
Use transformers to build correspondences

Solve for the rigid transformation

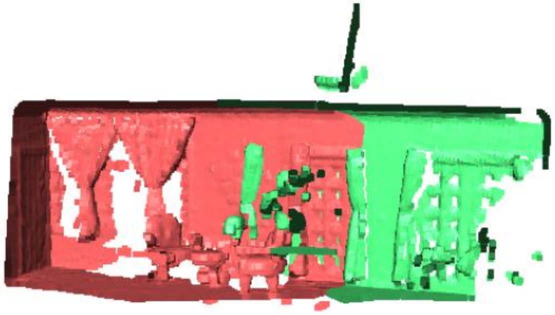
From overlapping scans to non-overlapping scans



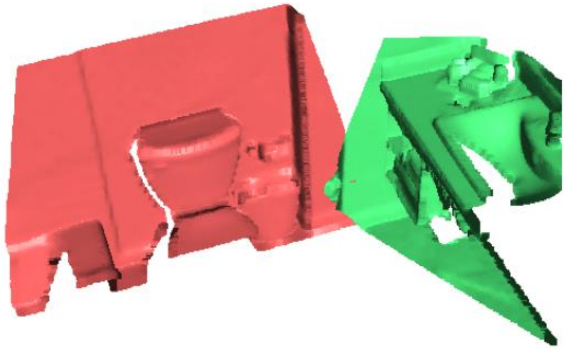
Complete scene



Complete scene

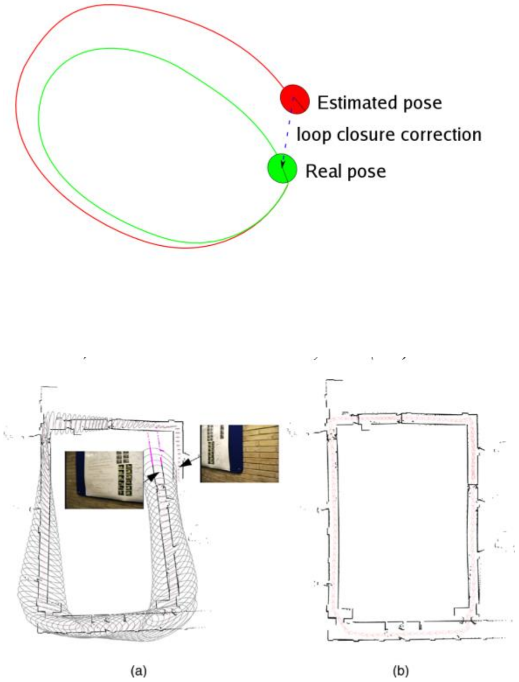


Overlapping scans

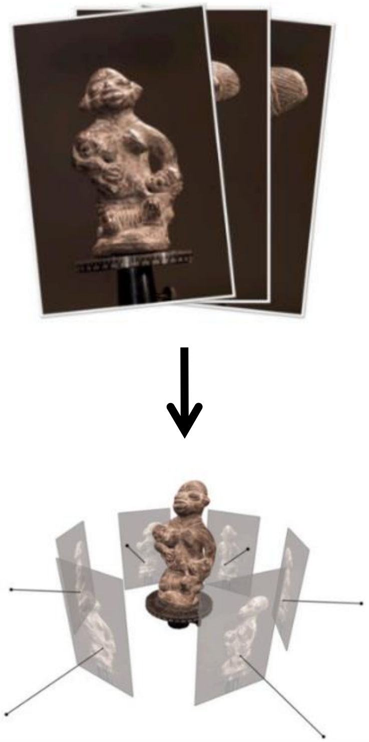


Small/no overlapping scans

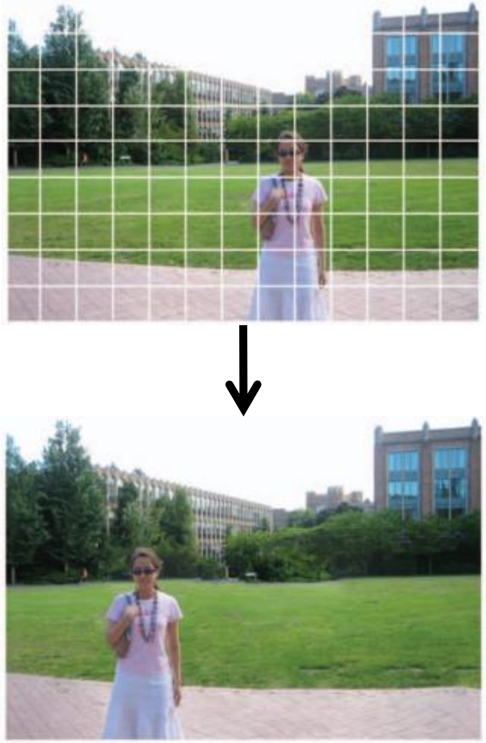
Diverse applications



Early detection of loop closure

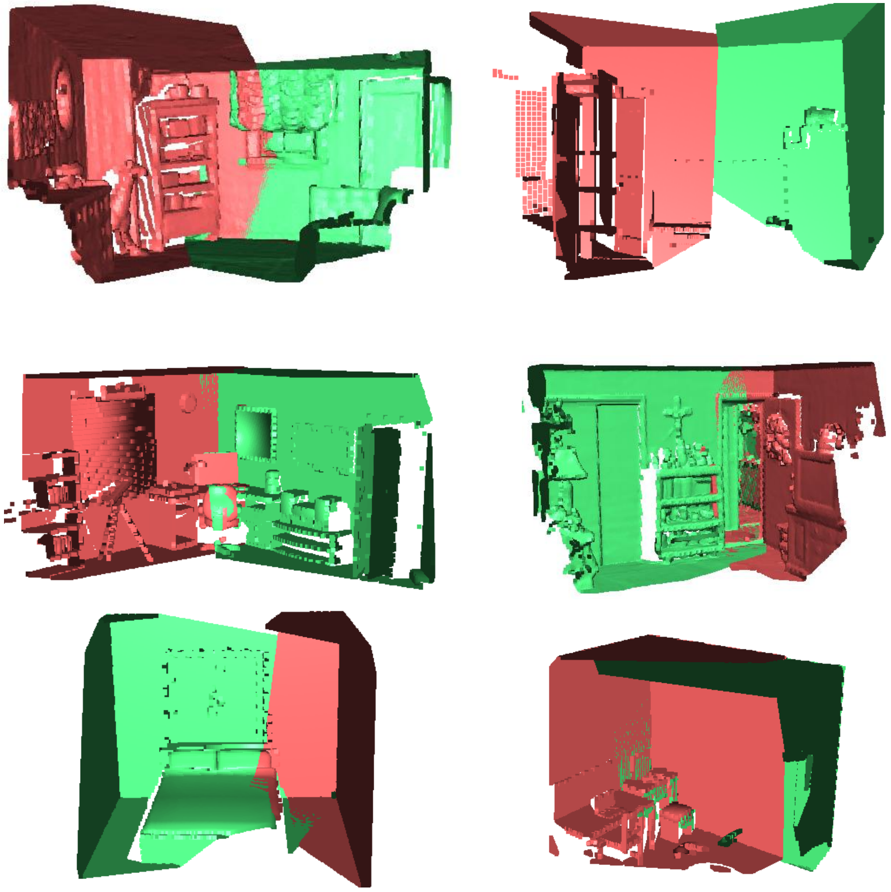


Reconstruction from a few snapshots [Furukawa and Hernandez 15]



Solving jigsaw puzzle [Cho et al. 10]

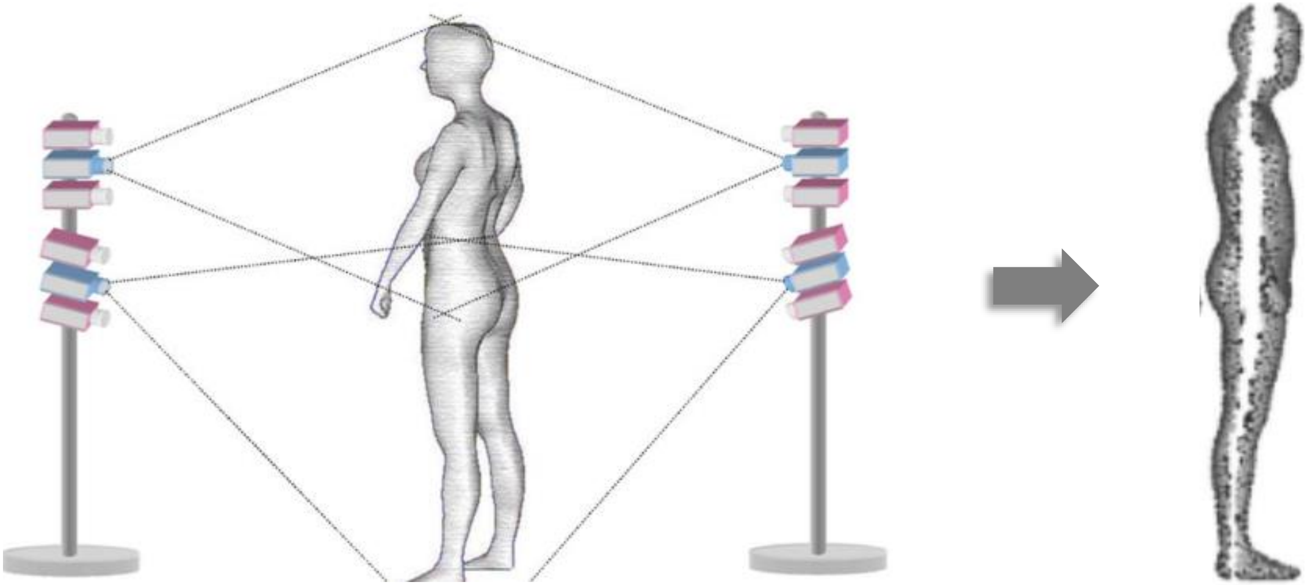
Challenges



Small or No-overlaps

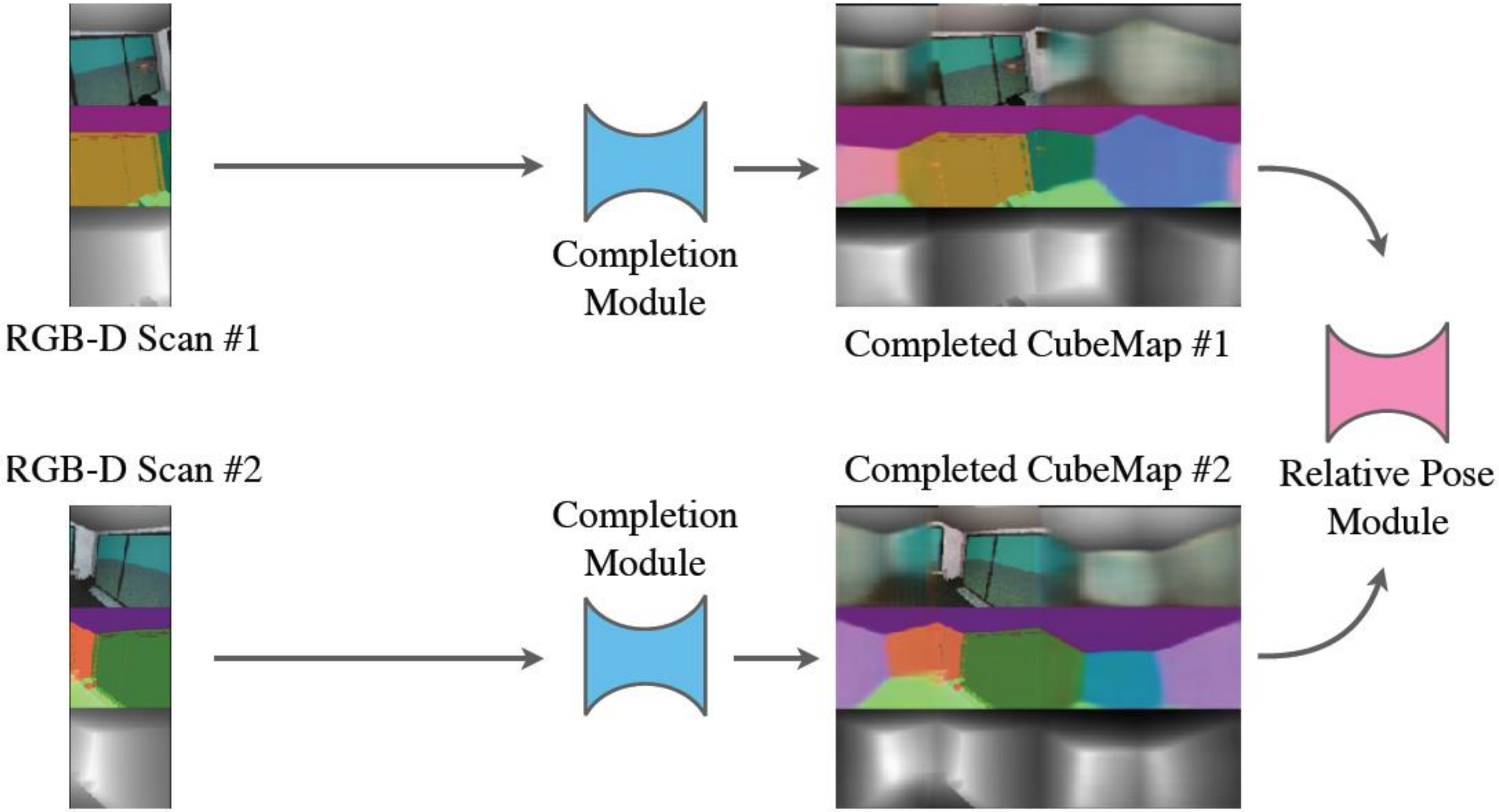
- No or few features to match
- Black-box deep networks do not work
- Overlapping ratios vary

Human perception



Human body reconstruction from a pair of front and back scans

Key Idea: Completion + Relative pose estimation



Scene Completion

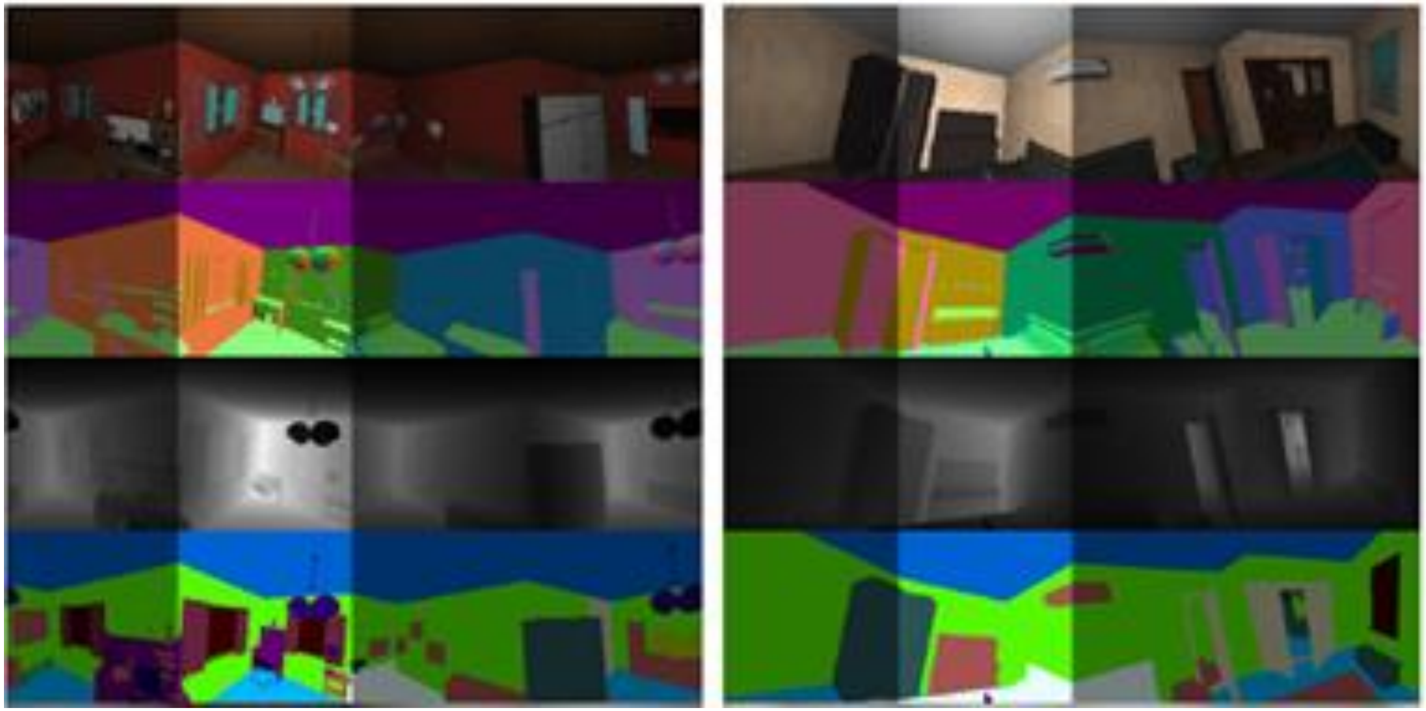
Combine depth/normal/color/learned semantic class descriptors

Color

Depth

Normal

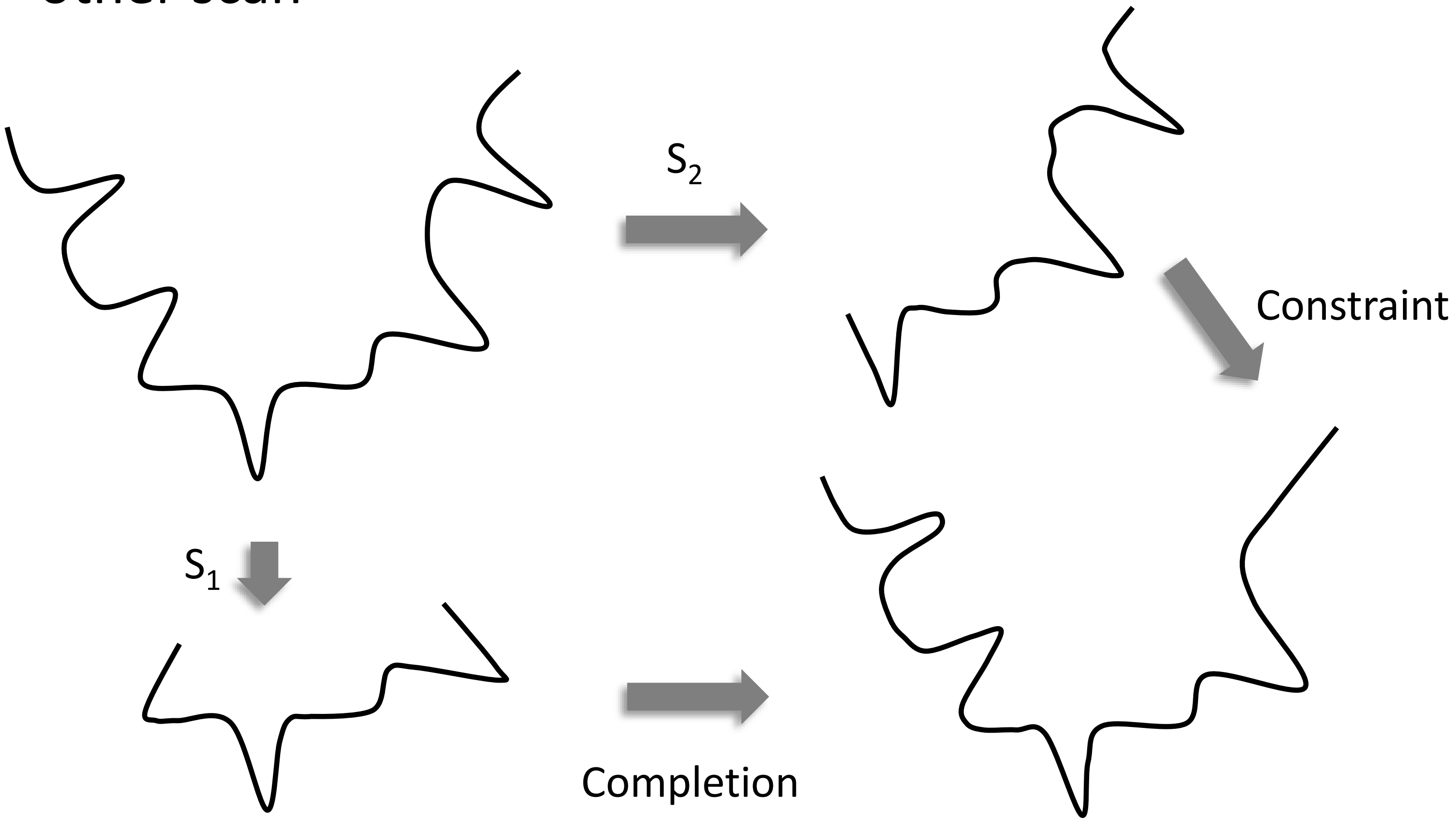
Semantic descriptor



Scene completion from two inputs

A generic constraint

- The completed scene from each scan should contain the other scan



S_2

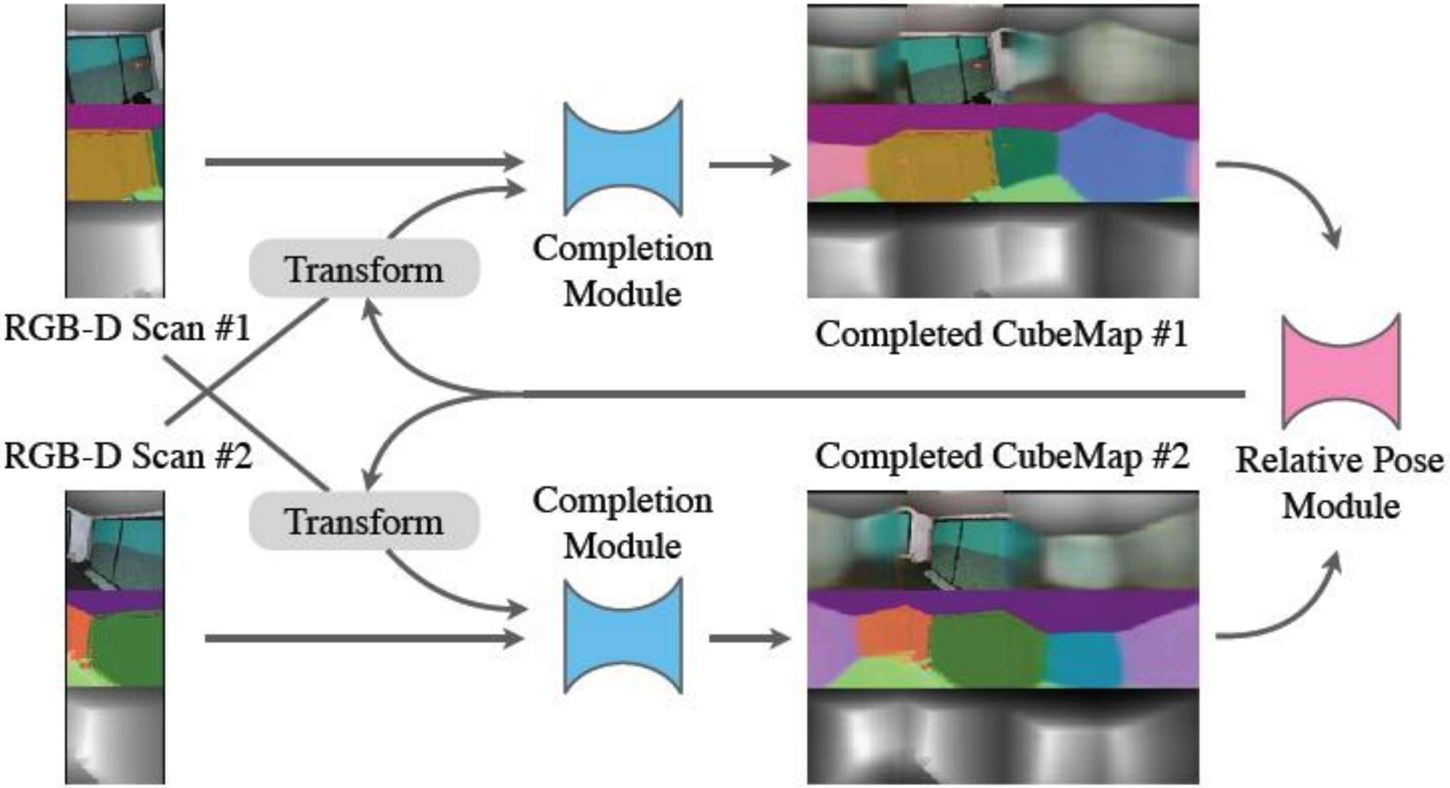
Constraint

S_1

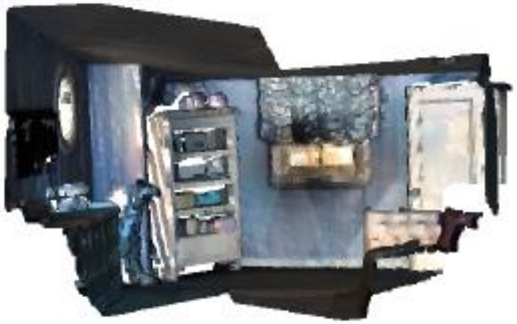
Completion

Update the completed scenes using both input scans

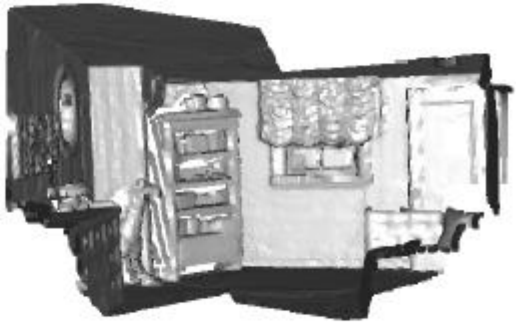
– Allow the geometry of the second scan to move when performing completion



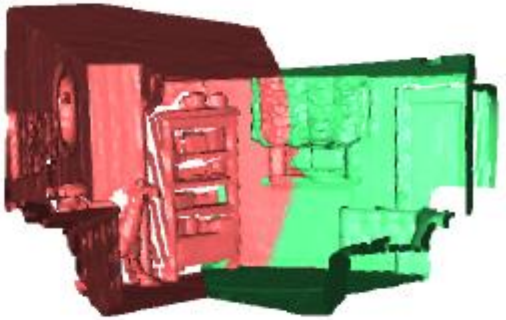
Qualitative Results --- Small overlap



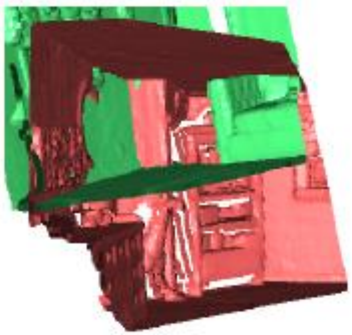
G.T. Color



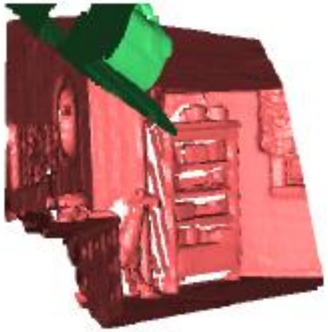
G.T. Scene



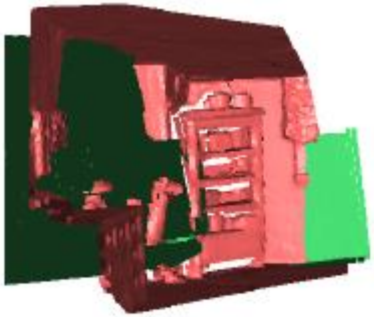
Ours



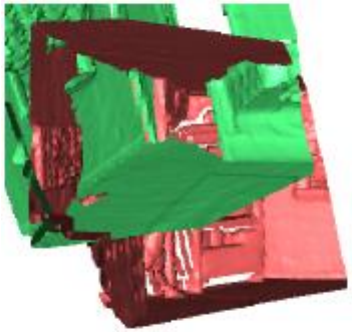
4PCS



DL

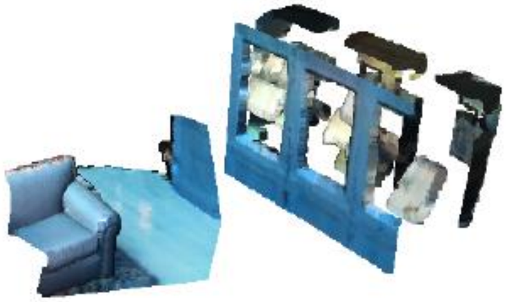


GReg

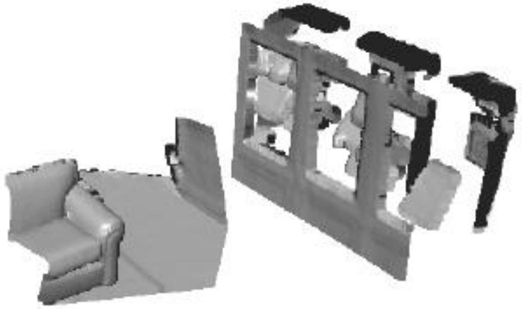


CGReg

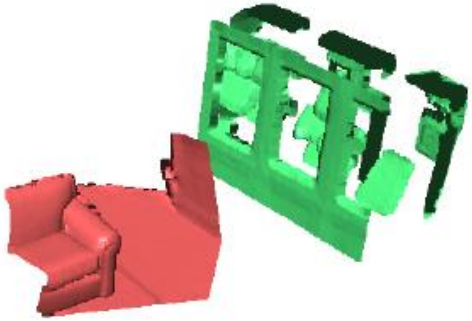
Qualitative Results --- No overlap



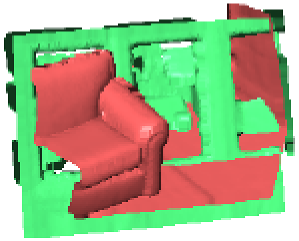
G.T. Color



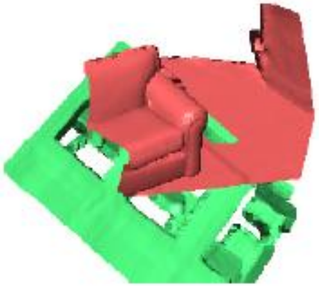
G.T. Scene



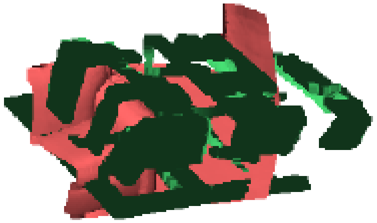
Ours



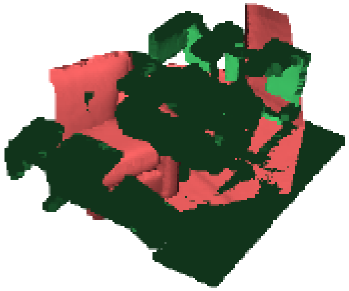
4PCS



DL

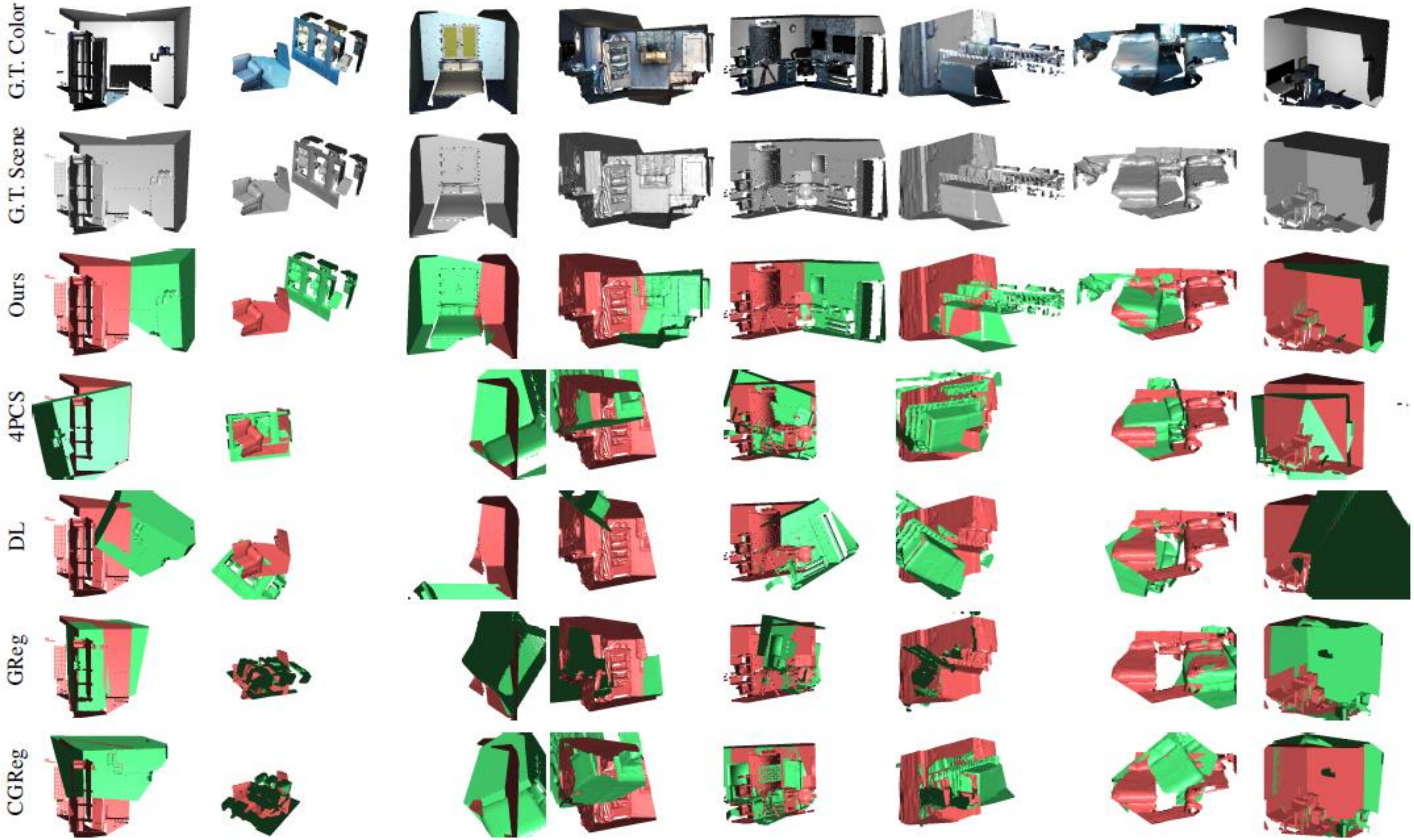


GReg

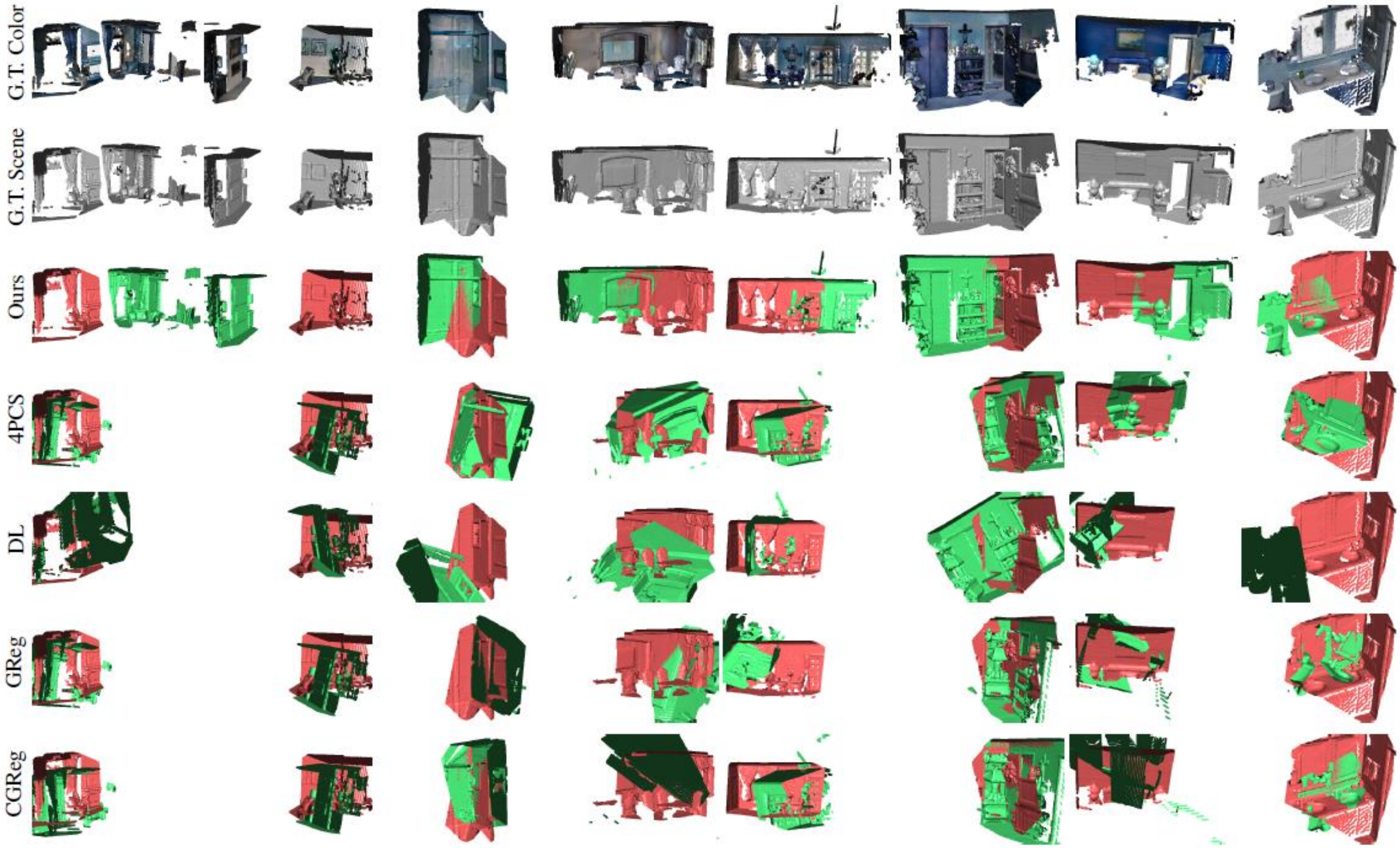


CGReg

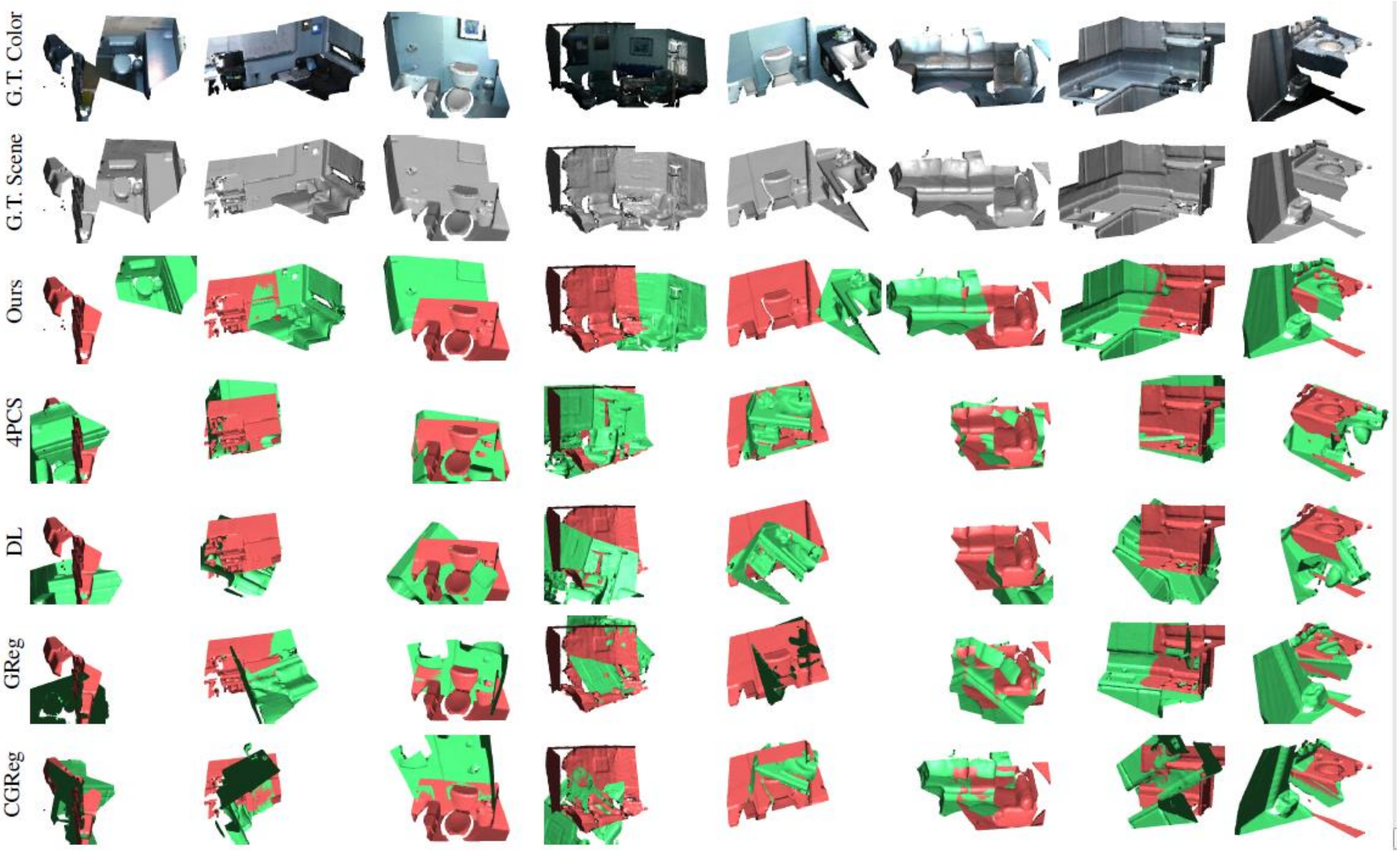
Qualitative Results --- SUNCG



Qualitative Results --- Matterport



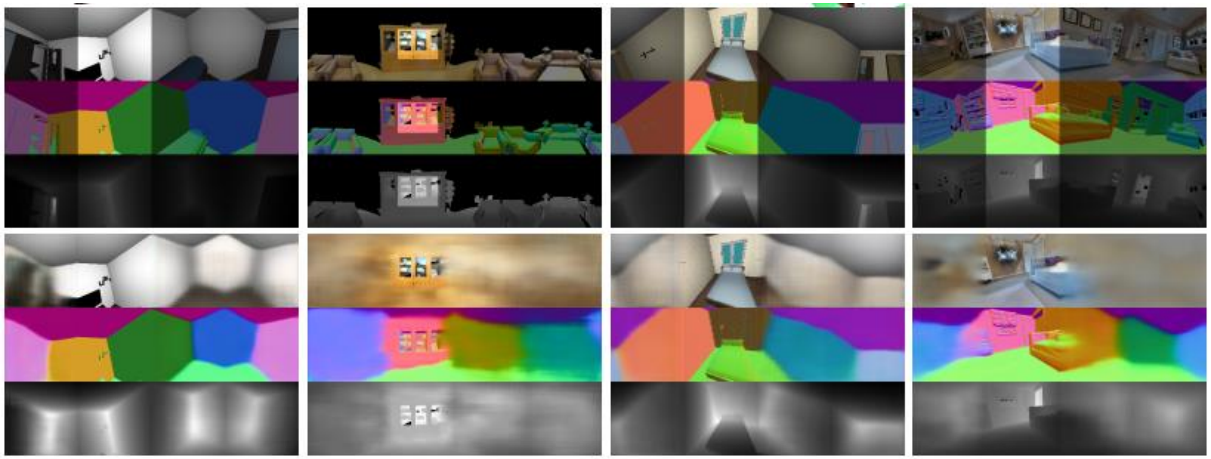
Qualitative Results --- ScanNet

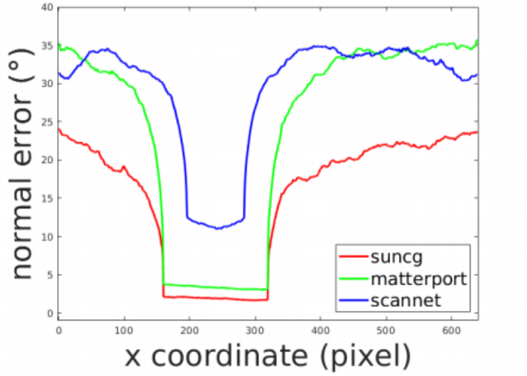


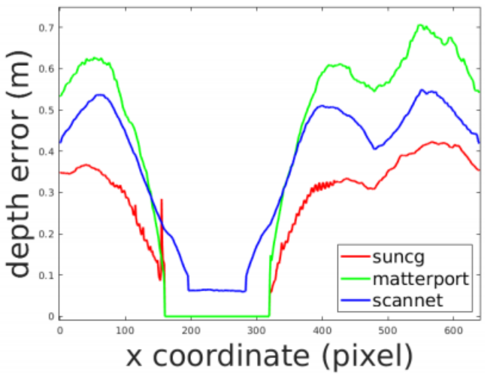
Understand the quality of the completions

Ground Truth:

Predicted:

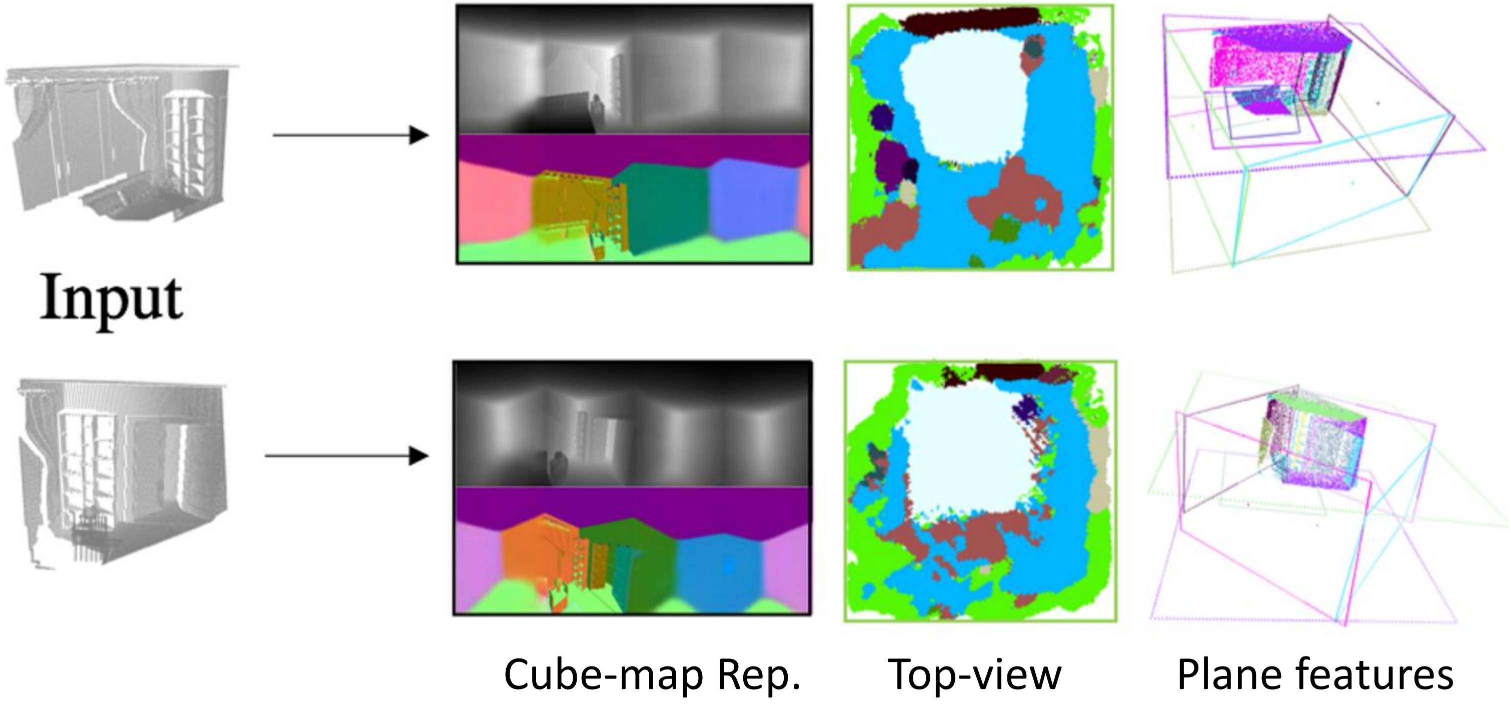






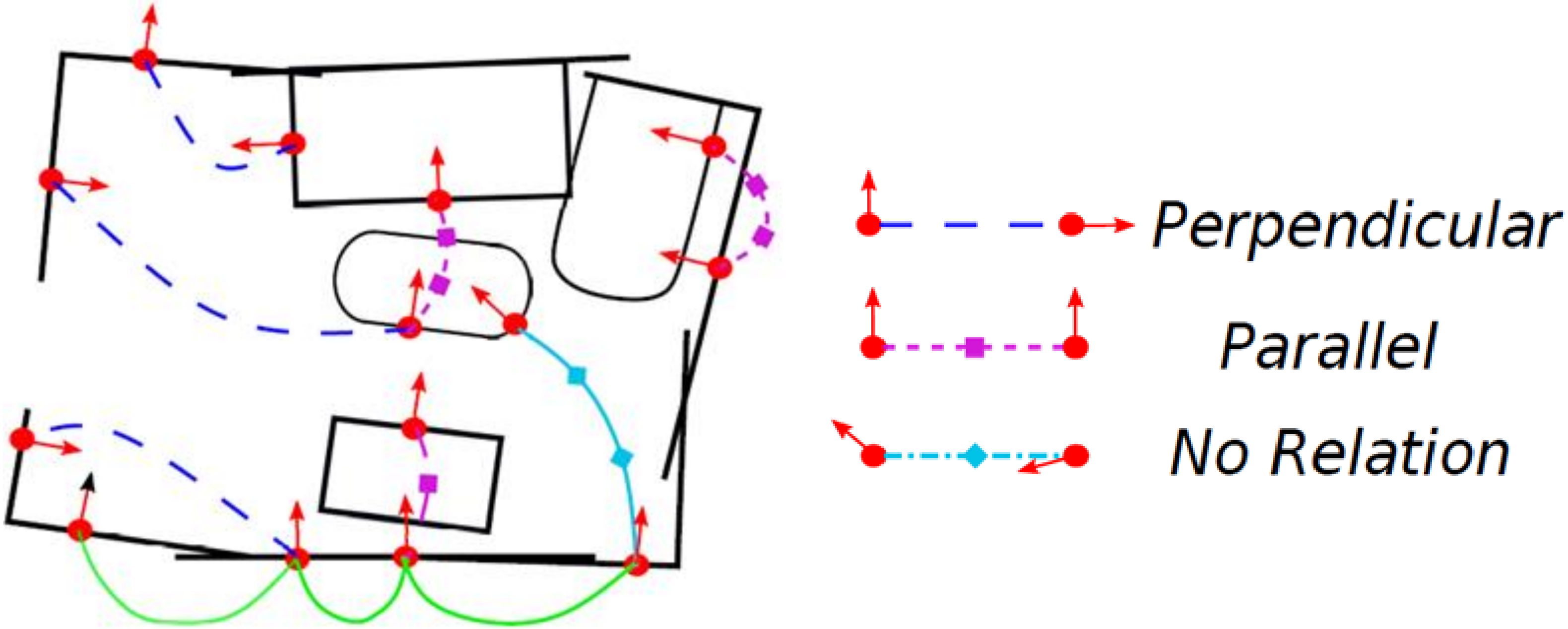
Relative pose estimation in the presence of large outlier ratios

Representations for Completion

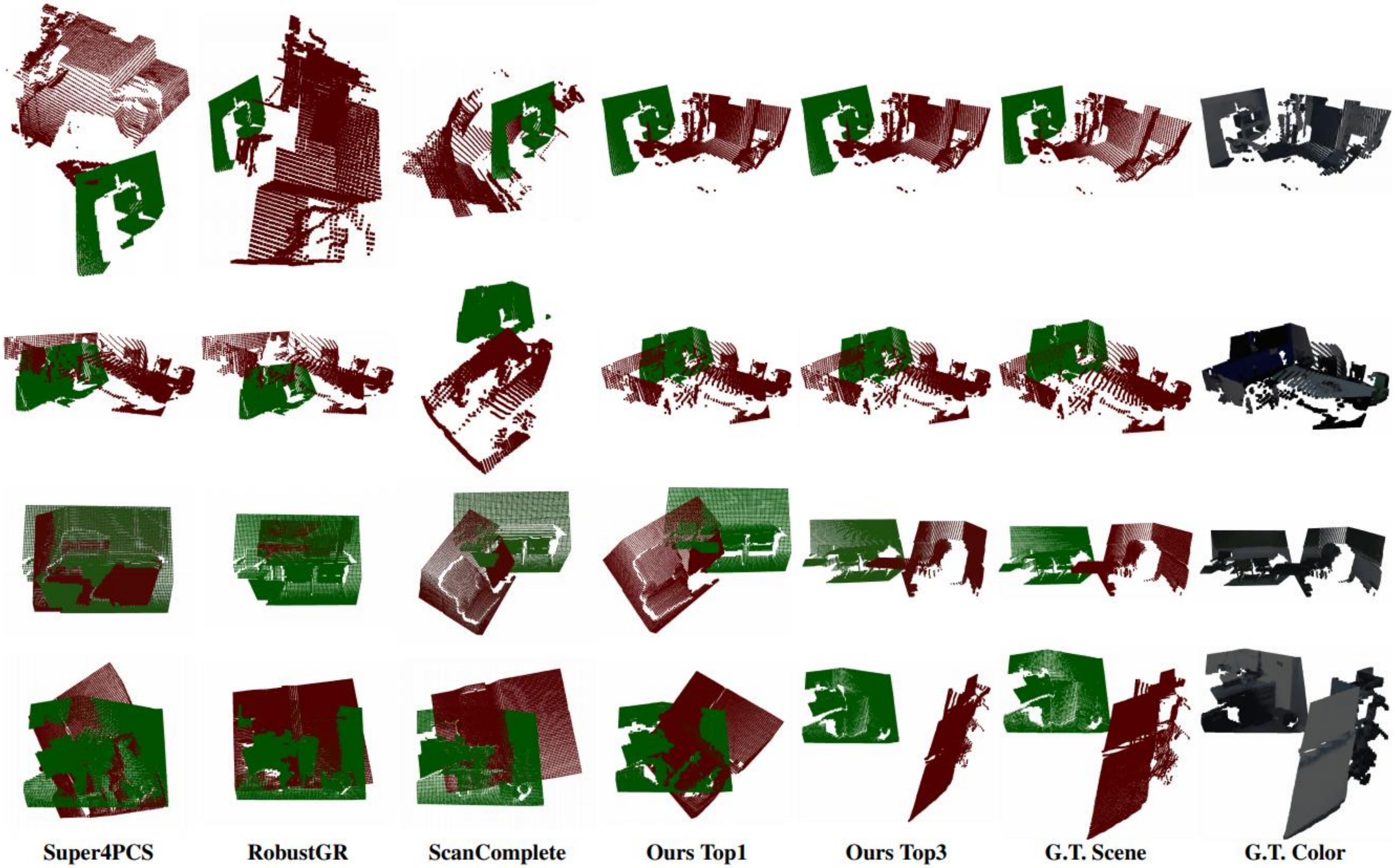


Top-view and plane features generalize better in far-away invisible regions

Representations for local registration



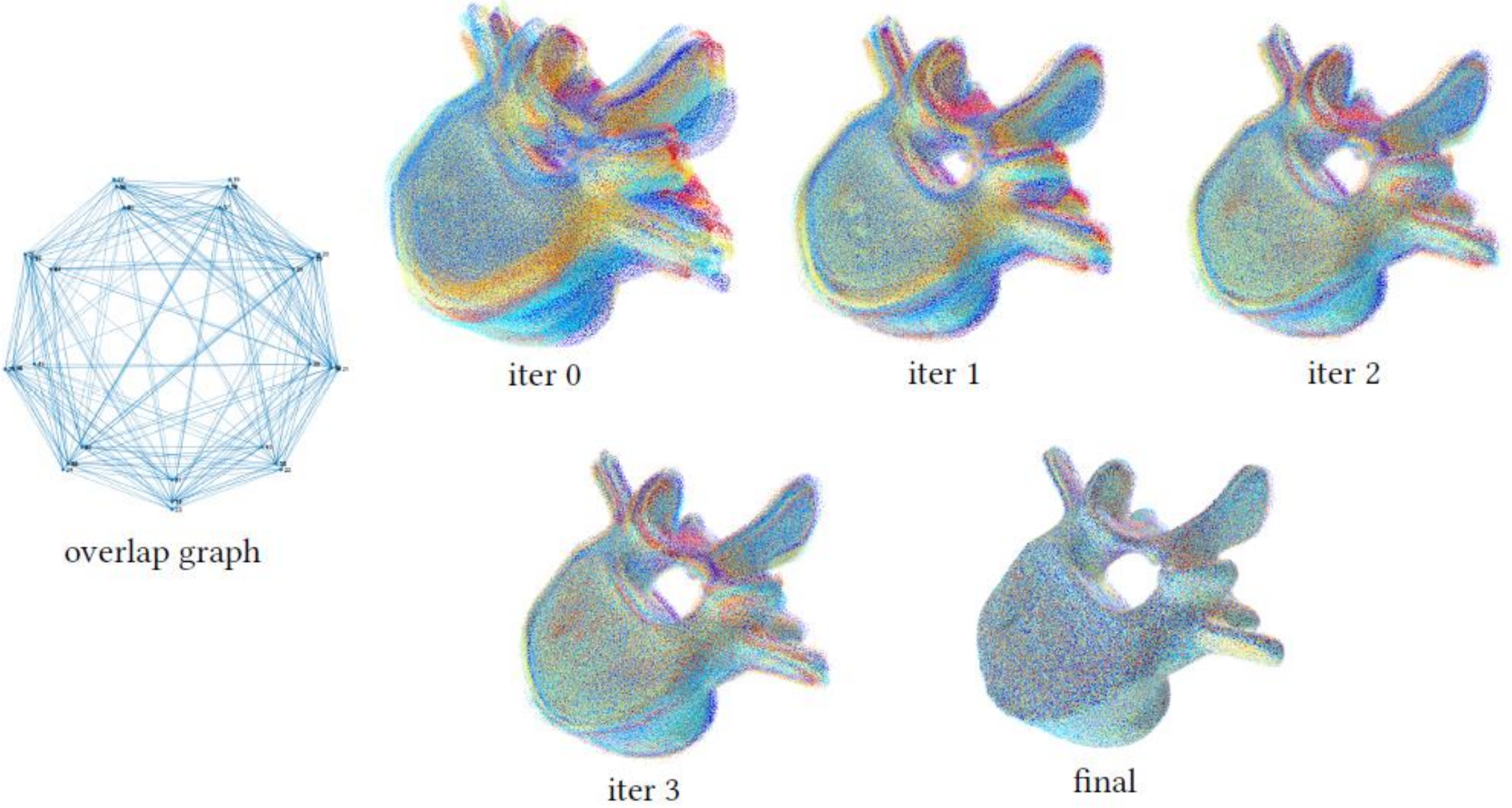
Qualitative results



Super4PCS RobustGR ScanComplete Ours Top1 Ours Top3 G.T. Scene G.T. Color

Multiple methods

Joint pairwise registration



overlap graph

iter 0

iter 1

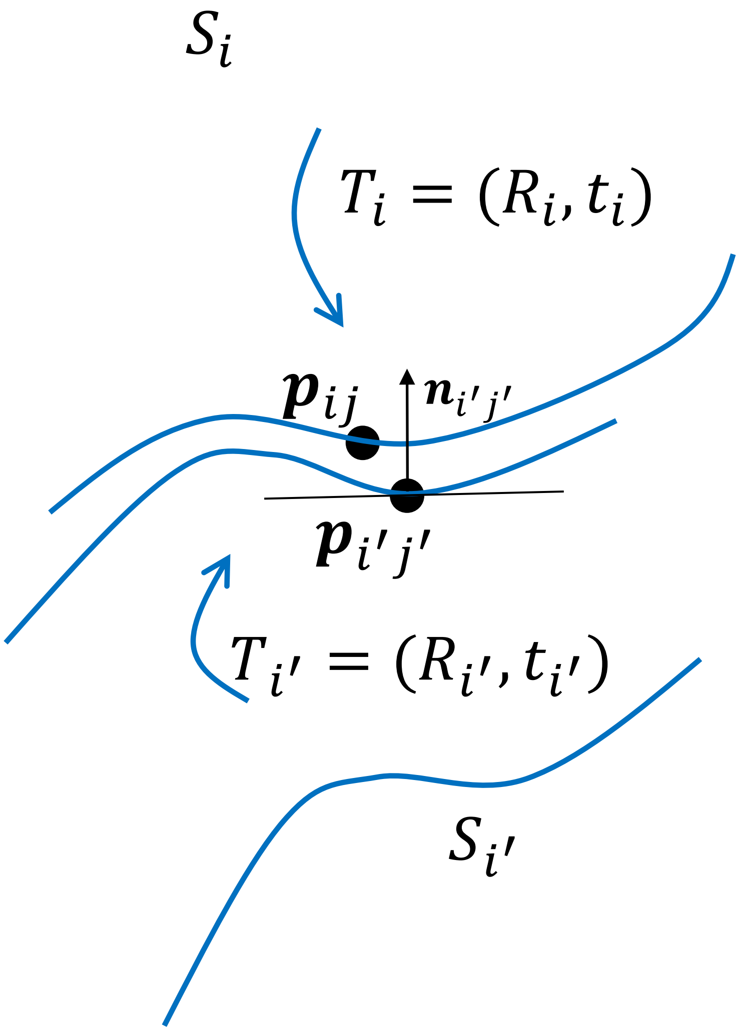
iter 2

iter 3

final

Tam, Gary KL et al. 2012

Joint pairwise registration



$$\underset{\{T_i\}}{\text{minimize}} \quad \sum_{(i,i')\in\mathcal{E}} d^2(S_i, T_i, S_{i'}, T_{i'})$$

$$\text{subject to} \quad R_1 = I_3, \boldsymbol{t}_1 = \boldsymbol{0}.$$

$$d^2(S_i, T_i, S_{i'}, T_{i'}) :=$$

$$\sum_{(p_{ij}, p_{i'j'})\in C_{ii'}} \left((R_i\boldsymbol{p}_{ij} + \boldsymbol{t}_i - R_{i'}\boldsymbol{p}_{i'j'} - \boldsymbol{t}_{i'})^T (R_{i'}\boldsymbol{n}_{i'j'})\right)^2$$

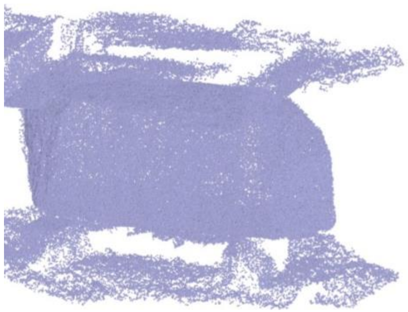
Gauss-Newton method

Newton’s method

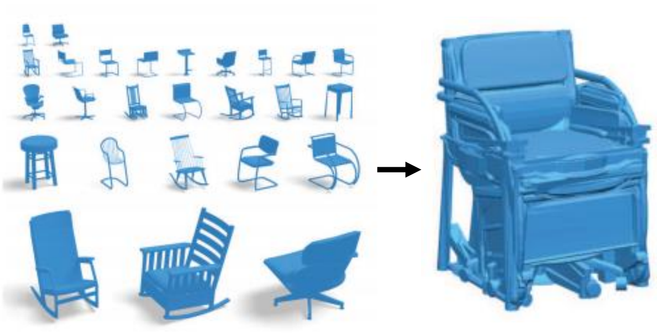
Joint pairwise registration - applications

- Surface reconstruction

- Graph/Depth-based SLAM



- Organizing shape collections



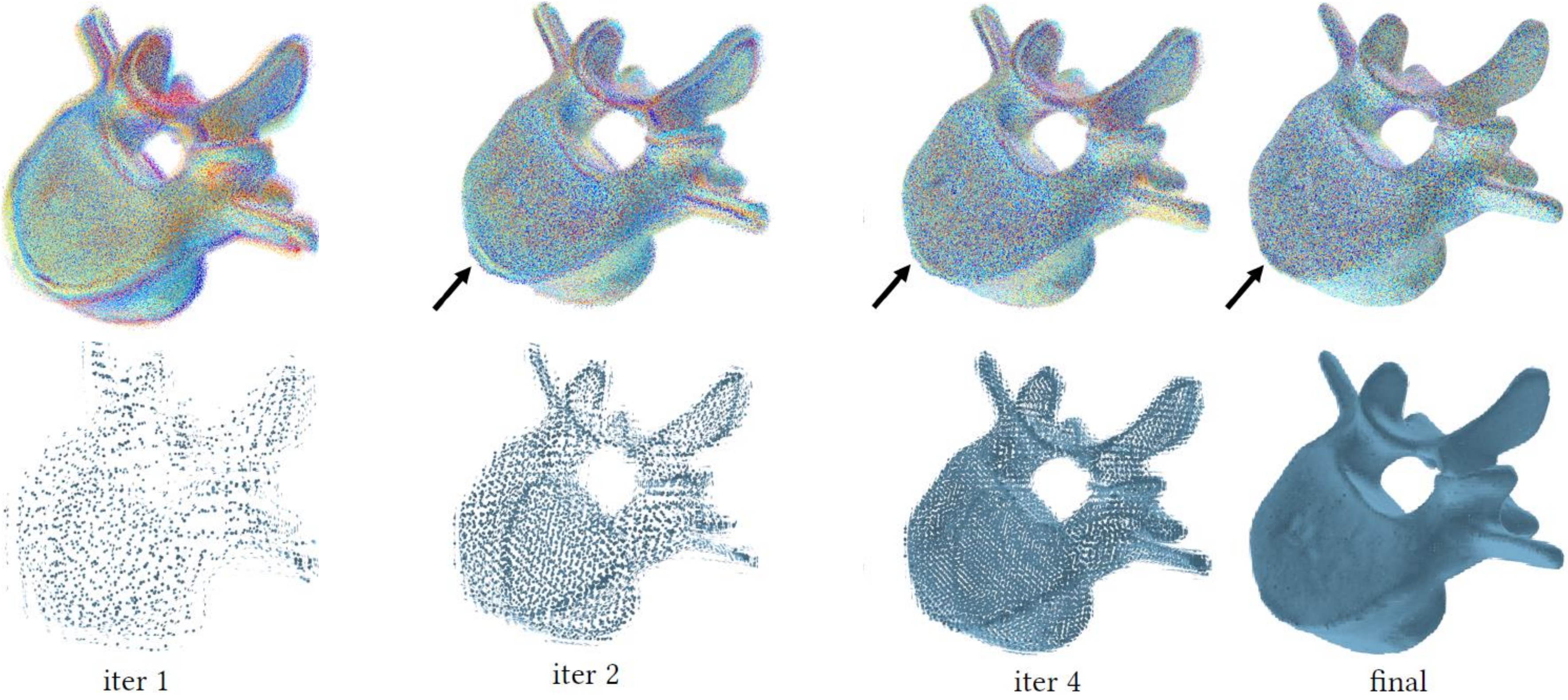
Joint pairwise registration - limitations

- Need to determine overlapping scans and overlapping regions
  - Potentially a quadratic number of pairs

- Slow convergence when there are a lot of scans

Simultaneous registration and reconstruction

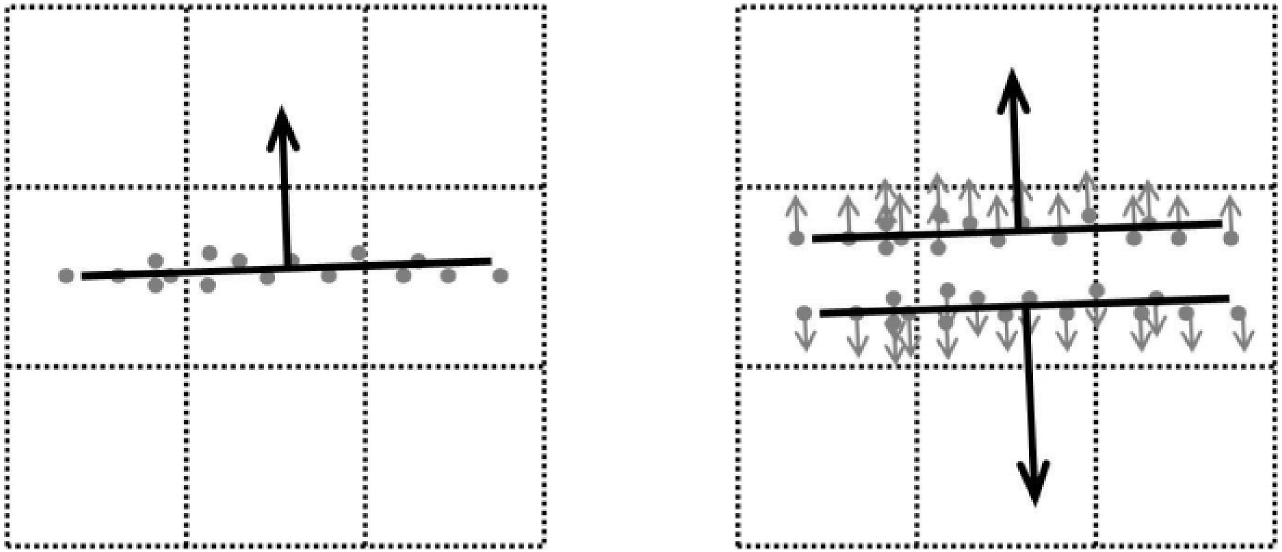
Simultaneous registration and reconstruction



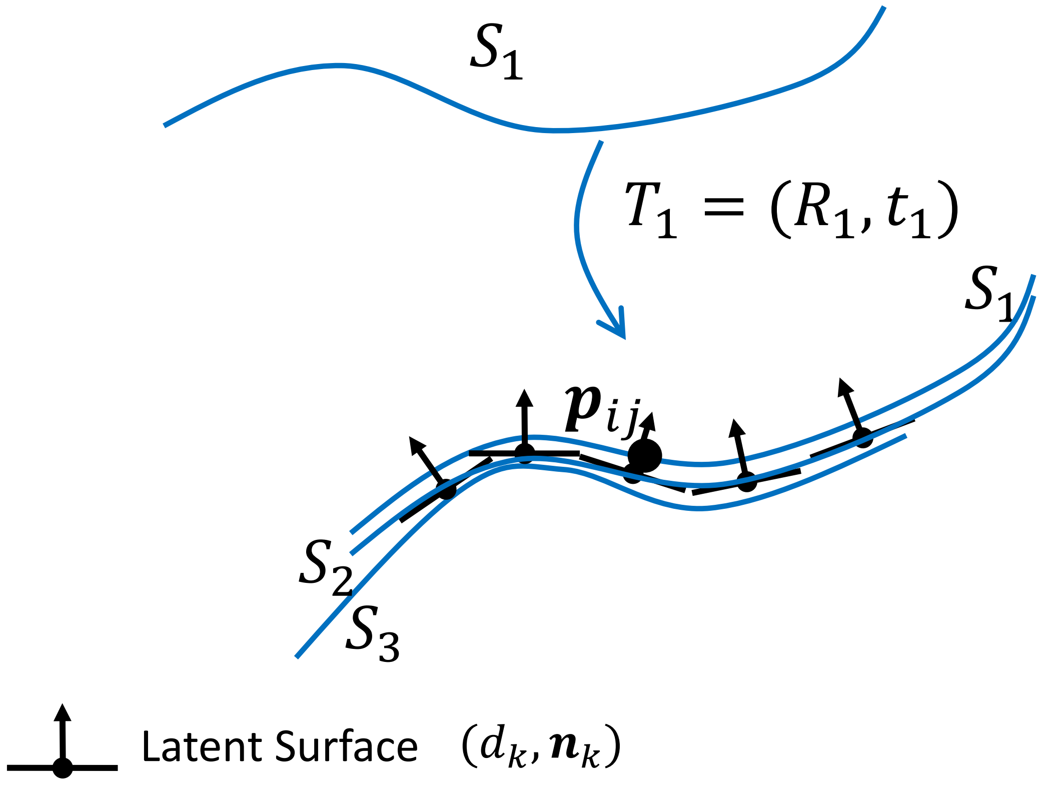
Huang et al. 2007, Huang and Anguelov 2010

Step I – Latent surface creation

- Fit planes to points associated with each cell



Step II – Scan-surface alignment



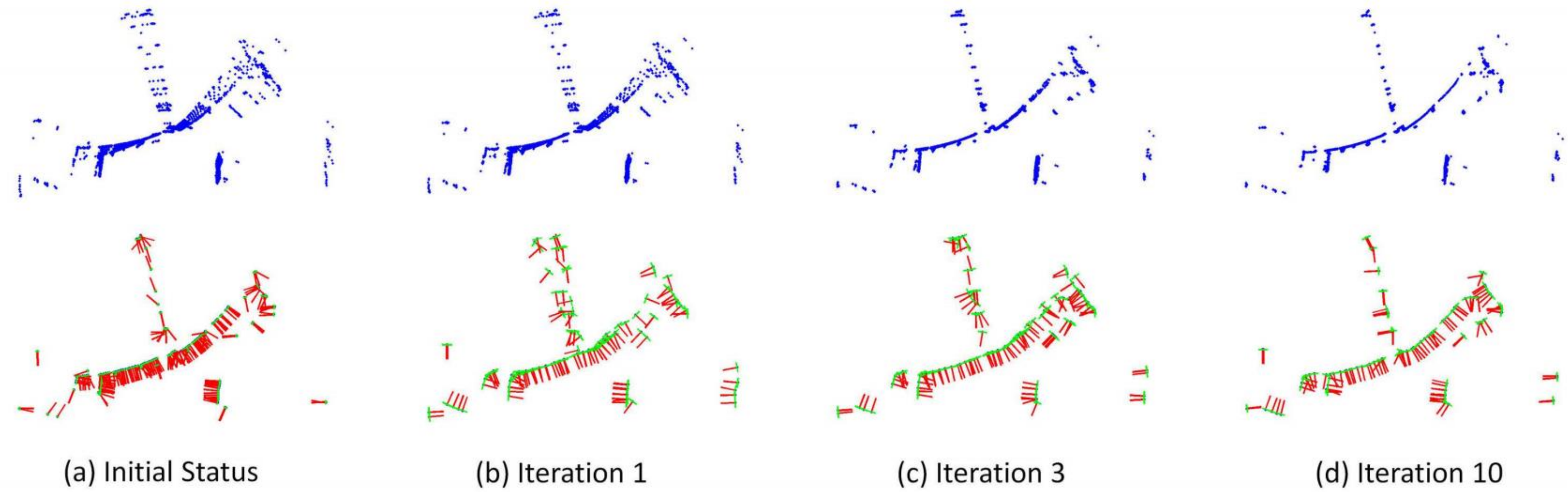
Simultaneous registration and reconstruction

$$\begin{aligned} \operatorname*{argmin}_{\{R_i,\boldsymbol{t}_i\},\{(d_k,\boldsymbol{n}_k)\}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N_i}\left((R_i\boldsymbol{p}_{ij}+\boldsymbol{t}_i)^T\boldsymbol{n}_{k_{ij}}-d_{k_{ij}}\right)^2 \\ \text{subject to} \quad & R_1 = I_3, \boldsymbol{t}_1 = \mathbf{0}. \end{aligned}$$

- Fix the scans to optimize the latent surface

- Fix the latent surface to optimize the scan poses

Simultaneous registration and reconstruction



(a) Initial Status

(b) Iteration 1

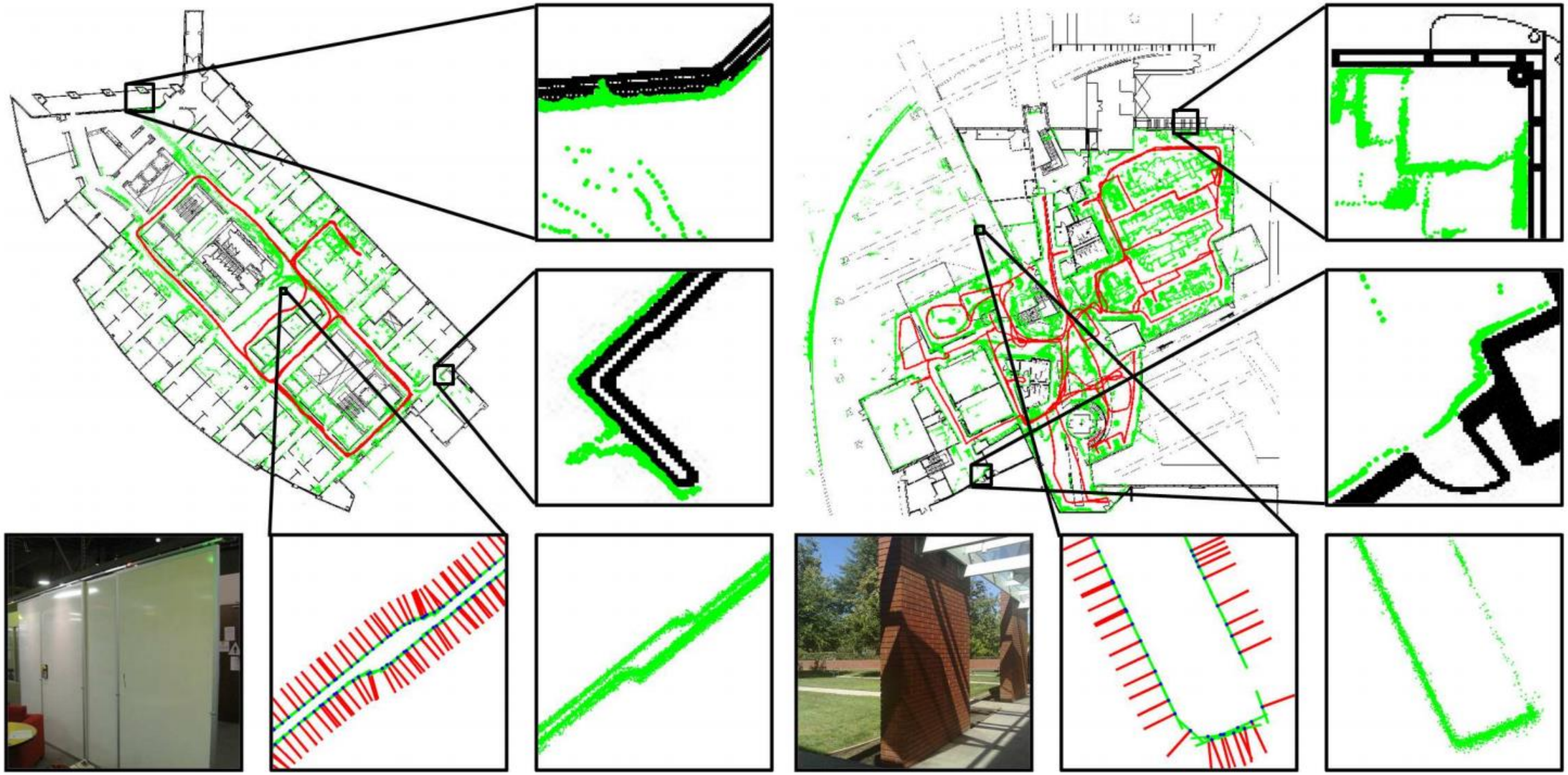
(c) Iteration 3

(d) Iteration 10

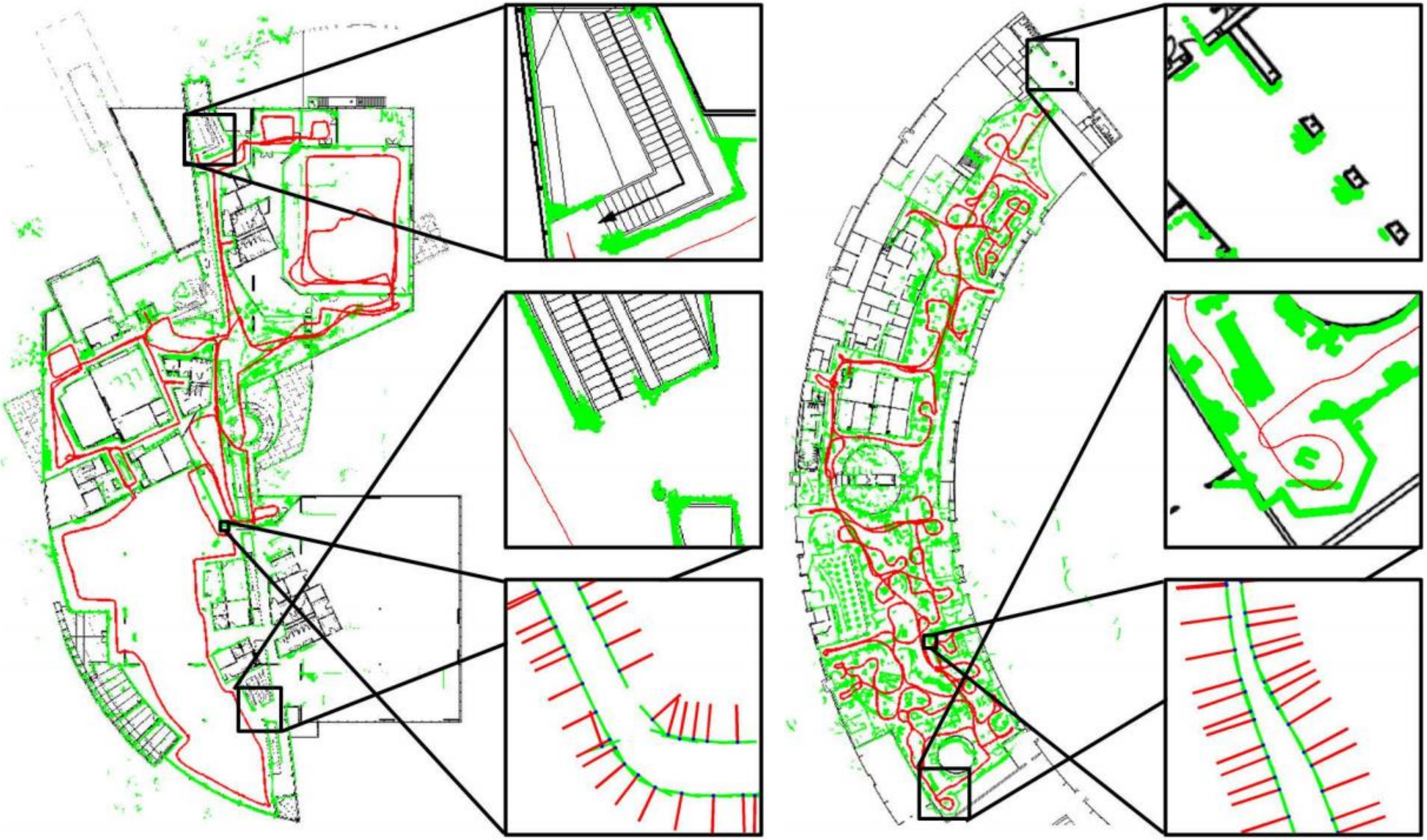
Works for un-organized point sets

Efficient – no range query

Large-scale registration



Large-scale registration



Topics that are not covered

- Non-rigid registration
  - Will have a guest lecture on this
- Methods that are based on probabilistic modeling
- Other learning-based methods
  - Will talk about this later

IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 19, NO. 7, JULY 2013 1

Registration of 3D Point Clouds and Meshes: A Survey From Rigid to Non-Rigid

Gary K.L. Tam[1], Zhi-Quan Cheng[2], Yu-Kun Lai[1], Frank C. Langbein[1], Yonghuai Liu[3], David Marshall[1], Ralph R. Martin[1], Xian-Fang Sun[1] and Paul L. Rosin[1]

Abstract—3D surface registration transforms multiple 3D datasets into the same coordinate system so as to align overlapping components of these sets. Recent surveys have covered different aspects of either rigid or non-rigid registration, but seldom discuss them as a whole. Our study serves two purposes: (i) to give a comprehensive survey of both types of registration, focusing on 3D point clouds and meshes, and (ii) to provide a better understanding of registration from the perspective of data fitting. Registration is closely related to data fitting in that it comprises three core interwoven components: *model selection*, *correspondences & constraints* and *optimization*. Study of these components (i) provides a basis for comparison of the *novelties* of different techniques, (ii) reveals the similarity of rigid and non-rigid registration in terms of problem representations, and (iii) shows how over-fitting arises in non-rigid registration and the reasons for increasing interest in intrinsic techniques. We further summarise some practical issues of registration which include initializations and evaluations, and discuss some of our own observations, insights and foreseeable research trends.

Index Terms—Deformation modeling, digital geometry processing, surface registration, point clouds, meshes, 3D scanning

1 Introduction

Surface registration transforms multiple 3D datasets into the same coordinate system so as to align overlapping components of these sets. The datasets comprise measured points representing surfaces of 3D objects or scenes. Due to limitations of 3D scanning technology, typically multiple datasets must be captured from different viewpoints, each is associated with a different coordinate system. To allow them to be recombined to reconstruct the surfaces that represent the original objects or scenes [1], these data must be registered. *Surface registration* is thus an essential component of the 3D acquisition pipeline and is fundamental to computer vision, computer graphics and reverse engineering. Registering templates to a set of deforming surfaces provides cross-parametrization, and facilitates texture and skeleton transfer, shape interpolation, and statistical shape analysis. Numerous applications also benefit from the continual research on correspondences and registration (e.g. features and saliency), including symmetry detection and articulated object matching, finding object correspondences, fractured object reassembly, sub-part identification, and skeleton and pose construction.

Surface registration may consider *rigid* or *non-rigid* shapes. The former assumes that two (or more) surfaces are related by a rigid transformation. The latter allows deformation (e.g. morphing, articulation) between them. Rigid registration is a challenging problem. Firstly, the data itself poses many difficulties, which may include noise, outliers, and limited amounts of overlap. Noise may take the form of perturbations of points, or unwanted points close to a 3D surface. Outliers are unwanted points far from the surface, which can seriously affect results if not discarded. Limited overlap arises due to different parts of the object being in view in each scan; typically the number of scans is kept low for efficiency, with few points in common between successive scans. Further problems may arise due to self-occlusion when the object is scanned from certain viewing angles. While such problems can be mitigated by careful scanning, they are hard to avoid completely. Secondly, variations in initial positions and orientations (and what is known about them), as well as resolutions of data, can also affect algorithm performance, and must be taken into account when comparing rates of convergence, methods of correspondence determination, and approaches to optimization.

Non-rigid registration is even more difficult, as it not only faces the above challenges but also needs to account for *deformation*, so the solution space is much larger. Unlike the rigid case, where a few correspondences are sufficient to define one candidate rigid transformation for hypothesis testing, both deformation and alignment in the non-rigid case, without strong prior assumptions, often require a lot more reliable correspondences to define. Establishing meaningful and natural correspondences, however, is a

- *Gary K.L. Tam, E-mail: kwok-leung.tam@cs.cardiff.ac.uk*