The image depicts how our robot Irma3D sees itself in a mirror. The laser looking into itself creates distortions as well as changes in intensity that give the robot a single eye, complete with iris and pupil. Thus, the image is called "Self Portrait with Duckling".
The ICP Algorithm (1)

**Scan registration** Put two independent scans into one frame of reference

**Iterative Closest Point** algorithm [Besl/McKay 1992]

For prior point set \( M \) (“model set”) and data set \( D \)

1. Select point correspondences \( w_{i,j} \) in \( \{0,1\} \)
2. Minimize for rotation \( R \), translation \( t \)

\[
E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \| m_i - (R d_j + t) \|^2
\]

3. Iterate 1. and 2.

SVD-based calculation of rotation
• works in 3 translation plus 3 rotation dimensions
  \( \Rightarrow \) 6D SLAM with closed loop detection and global relaxation.
The ICP Algorithm (2)

Closed form (one-step) solution for minimizing of the error function

1. Cancel the double sum:

\[
E(R, t) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{i,j} \| m_i - (R d_j + t) \|^2
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \| m_i - (R d_i + t) \|^2,
\]

2. Compute centroids of the matching points

\[
c_m = \frac{1}{N} \sum_{i=1}^{N} m_i,
\]

\[
c_d = \frac{1}{N} \sum_{i=1}^{N} d_j
\]

\[
M' = \{ m'_i = m_i - c_m \}_{1,\ldots,N},
\]

\[
D' = \{ d'_i = d_i - c_d \}_{1,\ldots,N}.
\]

3. Rewrite the error function

\[
E(R, t) = \frac{1}{N} \sum_{i=1}^{N} \| m'_i - R d'_i - (t - c_m + R c_d) \|^2
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \| m'_i - R d'_i - \bar{t} \|^2,
\]
3. Rewrite the error function

\[ E(R, t) = \frac{1}{N} \sum_{i=1}^{N} \left\| m'_i - Rd'_i - \left( t - c_m + Rc_d \right) \right\|^2 \]

\[ = \frac{1}{N} \sum_{i=1}^{N} \left\| m'_i - Rd'_i \right\|^2 - \frac{2}{N} \tilde{t} \cdot \sum_{i=1}^{N} (m'_i - Rd'_i) + \frac{1}{N} \sum_{i=1}^{N} \left\| \tilde{t} \right\|^2. \]

Minimize only the first term! (The second is zero and the third has a minimum for \( \tilde{t} = 0 \)).

\[ E(R, t) = \sum_{i=1}^{N} \left\| m'_i - Rd'_i \right\|^2. \]

Arun, Huang und Blostein suggest a solution based on the singular value decomposition.

The ICP Algorithm (4)

Theorem: Given a $3 \times 3$ correlation matrix

$$H = \sum_{i=1}^{N} m_i^T d_i' = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}$$

with $S_{xx} = \sum_{i=1}^{N} m_i' d_i' x$, $S_{xy} = \sum_{i=1}^{N} m_i' d_i' y$, \ldots, then the optimal solution for $E(R, t) = \sum_{i=1}^{N} \|m_i' - Rd_i'\|^2$ is $R = UV^T$ with $H = U\Lambda V^T$ from the SVD.

Proof:

$$E(R, t) = \sum_{i=1}^{N} \|m_i' - Rd_i'\|^2.$$ 

Rewrite

$$E(R, t) = \sum_{i=1}^{N} \|m_i'\|^2 - 2 \sum_{i=1}^{N} m_i' \cdot Rd_i' + \sum_{i=1}^{N} \|d_i'\|^2.$$ 

Rotation is length preserving, i.e., maximize the term

$$\sum_{i=1}^{N} m_i' \cdot Rd_i' = \sum_{i=1}^{N} m_i'^T Rd_i'$$
The ICP Algorithm (5)

**Theorem:** Given a 3 x 3 correlation matrix

\[
H = \sum_{i=1}^{N} m_i^T d'_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}
\]

with \(S_{xx} = \sum_{i=1}^{N} m_{ix}^' d_{ix}^', \ S_{xy} = \sum_{i=1}^{N} m_{ix}^' d_{iy}^', \ \ldots\), then the optimal solution for \(E(R, t) = \sum_{i=1}^{N} ||m_i^' - Rd_i^'||^2\) is \(R = VU^T\) with \(H = U\Lambda V^T\) from the SVD.

**Proof:**

\[
\sum_{i=1}^{N} m_i^' \cdot Rd_i' = \sum_{i=1}^{N} m_i'^T R d_i'
\]

Rewrite using the trace of a matrix

\[
\text{Trace} \left( \sum_{i=1}^{N} R d_i' m_i'^T \right) = \text{Trace} \left( RH \right)
\]

**Lemma:** For all positive definite matrices \(A^TA\) and all orthonormal matrices \(B\) the following equation holds:

\[
\text{Trace} \left( A^TA \right) \geq \text{Trace} \left( BAA^T \right)
\]
The ICP Algorithm (6)

Theorem: Given a 3 x 3 correlation matrix

\[ H = \sum_{i=1}^{N} m_i'^T d_i' = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix} \]

with \( S_{xx} = \sum_{i=1}^{N} m_i'^x d_i'^x, \ S_{xy} = \sum_{i=1}^{N} m_i'^x d_i'^y, \ldots \), then the optimal solution for \( E(R, t) = \sum_{i=1}^{N} \| m_i' - Rd_i' \|^2 \) is \( R = VU^T \) with \( H = U\Lambda V^T \) from the SVD.

Proof: Suppose the singular value decomposition of \( H \) is \( H = U\Lambda V^T \). \( U \) and \( V \) are orthonormal 3 x 3 and \( \Lambda \) a diagonal matrix without negative entries.

\[ R = VU^T. \]

\( R \) is orthonormal and

\[ RH = VU^TU\Lambda V^T = V\Lambda V^T. \]

And using the lemma it is \( \text{Trace}(RH) \geq \text{Trace}(BRH) \).

Therefore \( R \) maximizes \( \sum_{i=1}^{N} m_i'^T Rd_i' \).
The ICP Algorithm (7)

- Estimating the transformation can be accomplished very fast $O(n)$

- Closest point search
  - Naïve $O(n^2)$, i.e., brute force
  - K-d trees for searching in logarithmic time

Recommendation: Start with
ANN: A Library for Approximate Nearest Neighbor Searching by David M. Mount and Sunil Arya (University of Maryland)

- Easy to use
- Many different methods are available
- Quite fast

http://www.cs.umd.edu/~mount/ANN/
One has to search all buckets according to the ball-within-bounds-test. ⇒ Backtracking
NNS Search – the Critical Issue

Properties for all tested NNS libraries.

<table>
<thead>
<tr>
<th>Library</th>
<th>revision</th>
<th>Data structure</th>
<th>$k$-NN search</th>
<th>fixed radius</th>
<th>ranged search</th>
<th>optimized for</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DTK [2]</td>
<td>rev. 470</td>
<td>k-d tree</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>shape registration</td>
</tr>
<tr>
<td>3DTK</td>
<td>rev. 470</td>
<td>octree</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>shape registration &amp; efficient storage</td>
</tr>
<tr>
<td>ANN [3]</td>
<td>Ver. 1.1.1</td>
<td>k-d tree</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>high dimensions</td>
</tr>
<tr>
<td>CGAL [4]</td>
<td>Ver. 3.5.1-1</td>
<td>k-d tree</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>multithreading</td>
</tr>
<tr>
<td>FLANN [5]</td>
<td>bcf3a56e5f6ed2d4dc3a340725fa341fa36ef79a4</td>
<td>k-d tree</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>libnabo [6]</td>
<td>Ver. 1.0.0</td>
<td>k-d tree</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SpatialIndex [7]</td>
<td>Ver. 1.4.0-1.1</td>
<td>R-tree</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>STANN [8]</td>
<td>Ver. 0.71 beta</td>
<td>SFC</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

Matching Time in s

- actuated SICK LMS200
- Microsoft Kinect
- Riegl VZ-400

3D Point Cloud Processing
Dr. Andreas Nüchter
July 23, 2014
NNS Search – the Critical Issue
The ICP Algorithm (8)

• Point reduction – another key for fast ICP algorithms
  – Start with cube surrounding the 3D point cloud
The ICP Algorithm (9)

- Point reduction – another key for fast ICP algorithms
  - Start with cube surrounding the 3D point cloud
  - Divide

- Another key issue: maximal point-to-point distance.
Registering Surfaces (1)

• Given

The main idea:
– Pairwise matching technique
– We want to minimize the distance between the two parts
– We set up a variational problem
– Minimize distance “energy” by rigid motion of one part
Registering Surfaces (2)

Problem:
- How to compute the distance
- This is simple if we know the corresponding points.
- Of course, we have in general no idea of what corresponds...

• ICP-idea: set closest point as corresponding point

• Full algorithm:
  - Compute closest point points
  - Minimize distance to these closest points by a rigid motion
  - Recompute new closest points and iterate
Registering Surfaces (3)

- Distances

Part B
(moves, rotation & translation)

Part A
(stays fixed)

- Closest Point Distances

Part B
(moves, rotation & translation)

Part A
(stays fixed)
Registering Surfaces (4) – ICP iterations

Part A

Part B

final result

Part A

Part B

Part A

Part B
Generalizations (1)

- “point-to-plane” ICP
- First order approximation
  - Match points to tangential planes rather than points
  - Converges much faster
Generalizations (2)

Implementation:

– We need normals for each point (unoriented/oriented)
– Compute closes point along normal direction

or

– Compute closest point as usual, project it to surface defined by query point and normal
– Desirable: reduced points with normals
Comparisons

- In literature it is claimed, that point-to-plane is faster and more accurate.
More Tricks and Tweaks

- **ICP Problems:**
  - Partial matching might lead to distortions / bias
  - Remove outliers
    - M-estimator
    - delete “far away points”, e.g. 20% percentile in point-to-point distance
    - hard point-to-point distance threshold (for environments 20cm)
  - Remove normal outliers (if connection direction deviates from normal direction)

- **Sampling problems**
  - Problem: for example flat surface with engraved letters
  - No convergence in that case
  - Improvement: Sample correspondence points with distribution to cover unit sphere of normal directions as uniformly as possible
More Tricks and Tweaks

- **ICP Problems:**
  - Partial matching might lead to distortions / bias
  - Remove outliers
    - M-estimator
    - delete “far away points”, e.g. 20% percentile in point-to-point distance or
    - hard point-to-point distance threshold (for environments 20cm)
  - Remove normal outliers (if connection direction deviates from normal direction)

- **Sampling problems**
  - Problem: for example flat surface with engraved letters
  - No convergence in that case
  - Improvement: Sample correspondence points with distribution to cover unit sphere of normal directions as uniformly as possible
Things to try...

bin/slam6D dat
bin/show dat

bin/slam6D -r 10 dat
bin/show dat
We see: small matching errors accumulate

```bash
bin/slam6D -s 1 -e 65 -r 10 -i 100 -d 75
   --epsICP=0.00001 ~/dat/hannover/
```
6D SLAM – Global Relaxation (1)

- In SLAM loop closing is the key to build consistent maps
- Notice: Consistent vs. correct or accurate

- GraphSLAM
  - Graph Estimation
  - Graph Optimization

- Graph Estimation
  - Simple strategy: Connect poses with graph edges that are close enough
  - Simple strategy: Connect poses, they have enough point pairs (closest points)
The global algorithm

**Scan registration** Put two independent scans into one frame of reference

**Iterative Closest Point** algorithm [Besl/McKay 1992]¹

For prior point set $M$ ("model set") and data set $D$

1. Select point correspondences $w_{i,j}$ in \{0,1\}
2. Minimize for rotation $R$, translation $t$

$$E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \| m_i - (Rd_j + t) \|^2$$

3. Iterate 1. and 2.

Four closed form solution for the minimization

For globally consistent scan matching use the following error function plus 3 rotation dimensions

$$E = \sum_{j \rightarrow k} \sum_{i} \| R_j m_i + t_j - (R_k d_i + t_k) \|^2$$

Minimize for all rotations $R$ and translations $t$ at the same time
Parametrizations for the Rigid Body Transformations

\[ E = \sum_{j \rightarrow k} \sum_{i} \left| R_j m_i + t_j - (R_k d_i + t_k) \right|^2 \]

- Helix transformation

\[ v(p) = \bar{x} + x \times p \]

\[ E = \sum_{j \rightarrow k} \sum_{i} (m_i - d_i + (\bar{x}_j + x_j \times m_i) - (\bar{x}_k + x_k \times m_i))^2 \]

... solving a system of linear equations
Parametrizations for the Rigid Body Transformations

\[ E = \sum_{j \rightarrow k} \sum_{i} \left| R_{j} m_{i} + t_{j} - (R_{k} d_{i} + t_{k}) \right|^2 \]

- Small angle approximation

\[ \sin \theta \approx \theta - \frac{\theta^3}{3} + \frac{\theta^5}{5} - \ldots \]
\[ \cos \theta \approx 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4} - \ldots \]

\[
R \approx \begin{pmatrix}
1 & -\theta_z & \theta_y \\
\theta_x \theta_y + \theta_z & 1 - \theta_x \theta_y \theta_z & -\theta_x \\
\theta_x \theta_z - \theta_y & \theta_x + \theta_y \theta_z & 1
\end{pmatrix}
\]

\[
R \approx \begin{pmatrix}
1 & -\theta_z & \theta_y \\
\theta_z & 1 & -\theta_x \\
-\theta_y & \theta_x & 1
\end{pmatrix}
\]

... solving a system of linear equations
Parametrizations for the Rigid Body Transformations

\[
E = \sum_{j \to k} \sum_{i} |R_{j} m_{i} + t_{j} - (R_{k} d_{i} + t_{k})|^{2}
\]

- Explicit modeling of uncertainties
- Assumptions: The unknown error is normally distributed

\[
W = \sum_{j \to k} (\bar{E}_{j,k} - E'_{j,k})^T C_{j,k}^{-1} (\bar{E'}_{j,k} - E'_{j,k})
\]

\[
= \sum_{j \to k} (\bar{E}_{j,k} - (X'_{j} - X'_{k})) C_{j,k}^{-1} (\bar{E'}_{j,k} - (X'_{j} - X'_{k})).
\]

\[
E_{j,k} = \sum_{i=1}^{m} \|X_{j} \oplus d_{i} - X_{k} \oplus m_{i}\|^{2} = \sum_{i=1}^{m} \|Z_{i}(X_{j}, X_{k})\|^{2}
\]

... solving a system of linear equations
## Comparisons of the Parametrizations

<table>
<thead>
<tr>
<th>Global ICP</th>
<th>Classical Pose GraphSLAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Gaussian noise in the „3D Point Cloud“ space</td>
<td>• Gaussian noise in the space of poses</td>
</tr>
<tr>
<td>• Locally optimal</td>
<td>• Gradient descent needed</td>
</tr>
<tr>
<td>• ICP-like iterations using new point correspondences</td>
<td>• ICP-like iterations using new point correspondences needed as well</td>
</tr>
</tbody>
</table>

- Riegl Laser Measurement GmbH
  (video) (video) (video)
Closed Loop Detection and Global Relaxation

3D data acquisition

3D scan matching

Compute new pose

$\Delta p > \varepsilon$

Loop detection

$\Delta p < \varepsilon$

no loop

graph calculation

3D scan matching

Linearization

Solve linear Eqs.

$\Delta p > \varepsilon$

$\Delta p < \varepsilon$
Processing Large Data Sets (2)

We see: small matching errors accumulate

```
bin/slam6D -s 1 -e 65 -r 10 -i 100 -d 75
   --epsICP=0.00001 ~/dat/hannover/
```

```
bin/slam6D -s 1 -e 65 -r 10 -i 100 -d 75
   --epsICP=0.00001
   -D 250 -l 50 --cldist=750 -L 0 -G 1 ~/dat_hannover

bin/show -s 1 -e 65 ~/dat/dat_hannover
```