3D Object Reconstruction using Point Pair Features

Bachelor's thesis final

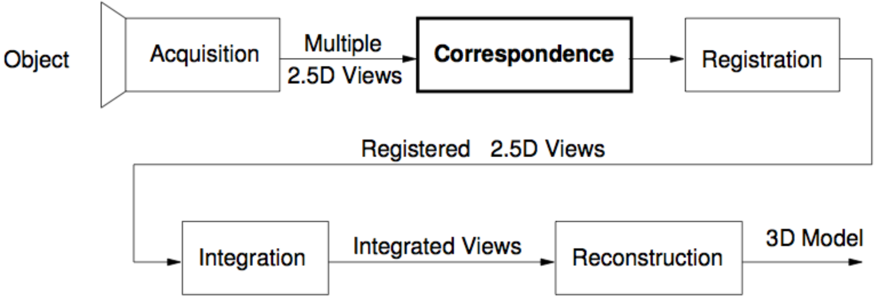
Adrian Haarbach

CAMP

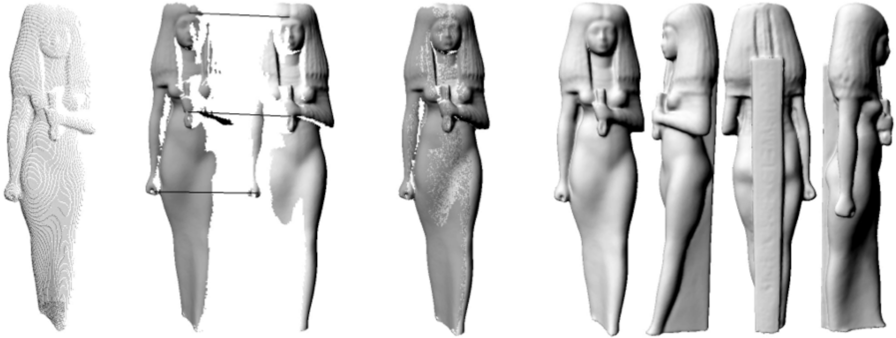
March 20th, 2015

Outline

Components of the 3D modeling process



Range image integration

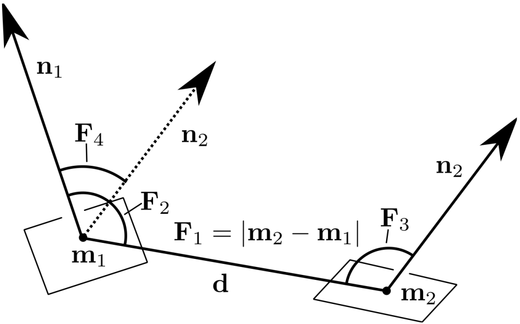


(a) Point cloud of a single view (b) Correspondences identified between two views (c) The two views registered (d) The 3D model after integration and reconstruction of 33 views

Outline

Outline

PPF: Point Pair Feature

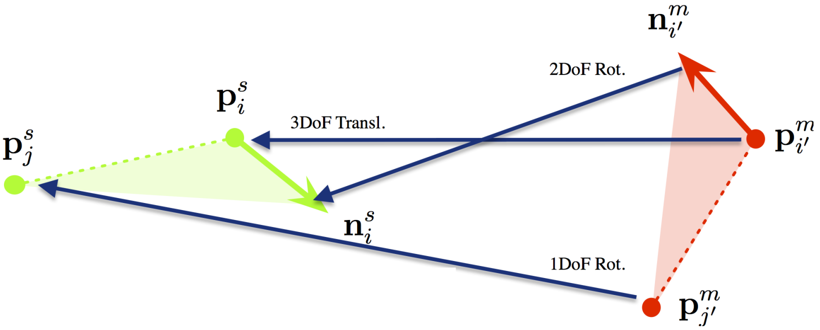


$$F = (F_1, F_2, F_3, F_4) = (||d||_2, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)) \quad (1)$$

PPF

Drost (2010)

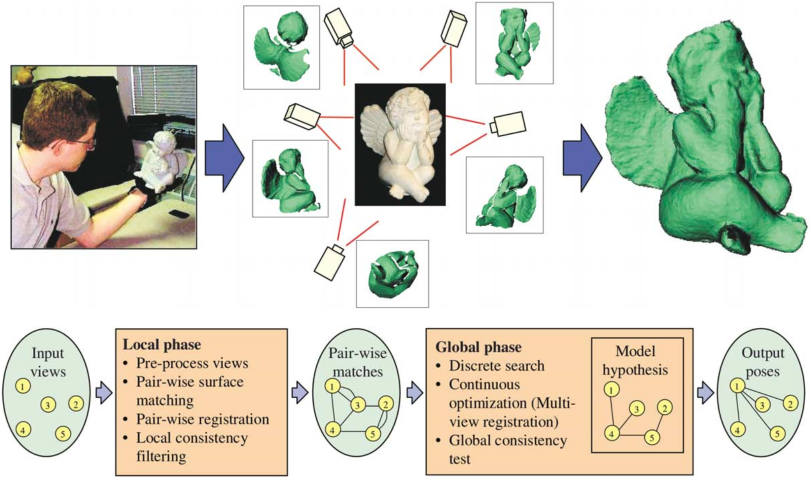
PPF: 6DoF pose estimate by aligning 2 PPF's



Drost (2010)

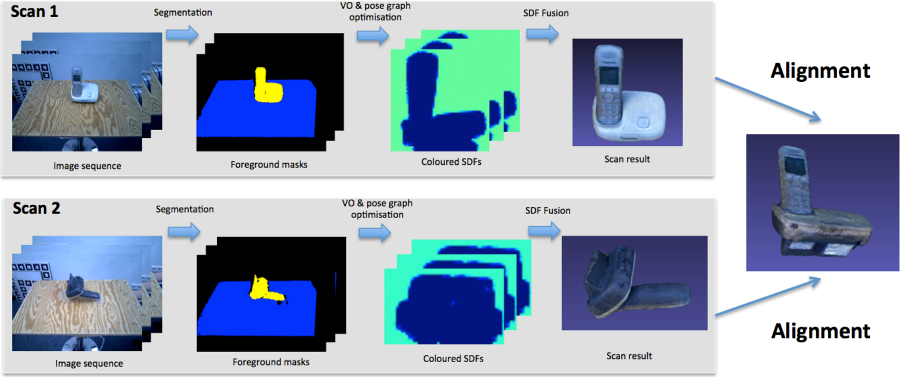
Outline

3D object reconstruction: in-hand scanning



Huber (2003)

3D object reconstruction: tabletop scanning

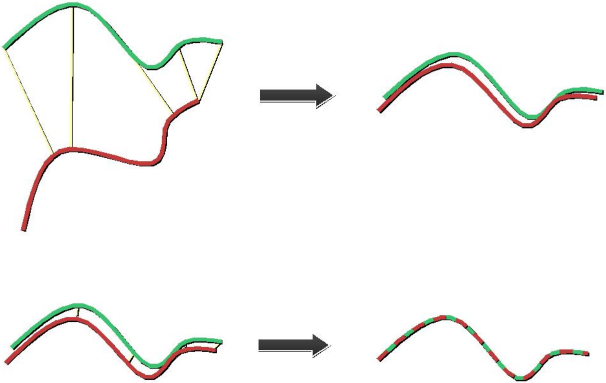


Kehl (2014)

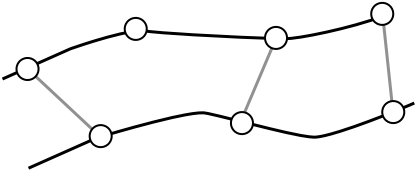
ICP: Iterative Closest Points

iterate:

1. compute correspondences $\{p_i \rightarrow q_i\}_1^N$
2. update current transformation $g = (R, t) \in SO(3)$



ICP: point to point



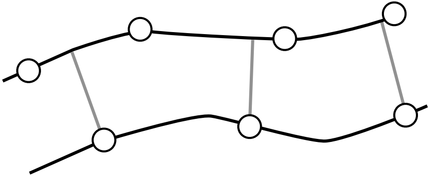
point to point distance

$$E = \sum_{i=1}^{N} ||Rp_i + t - q_i||^2 \qquad (2)$$

point to point

Besl and McKay (1992), Zhang (1994)

ICP: point to plane



point to plane distance

$$E = \sum_{i=1}^{N} ||(Rp_i + t - q_i) \cdot n_{q_i}||^2 \tag{3}$$

point to plane

Chen and Medioni (1991)

ICP: Levenberg-Marquardt ICP

Solve

$$E(g) = \sum_{i=1}^{N} ||e_i||^2$$

iteratively

$$g_{k+1} = g_k + x$$

via nonlinear least-squares techniques

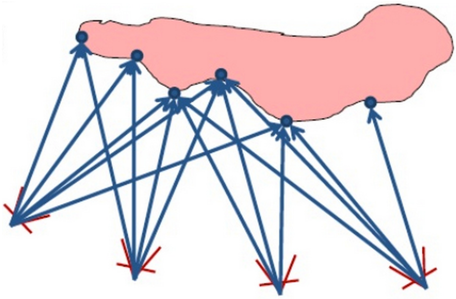
$$x = -(J^T J + \lambda I)^{-1} J^T e \tag{4}$$

Levenberg step

Fitzgibbon (2003)

Outline

Multiview refinement

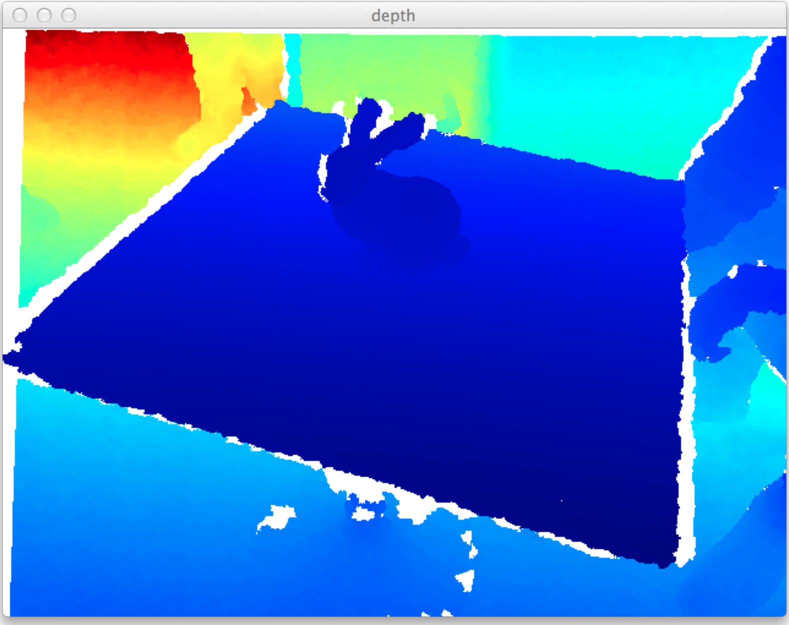


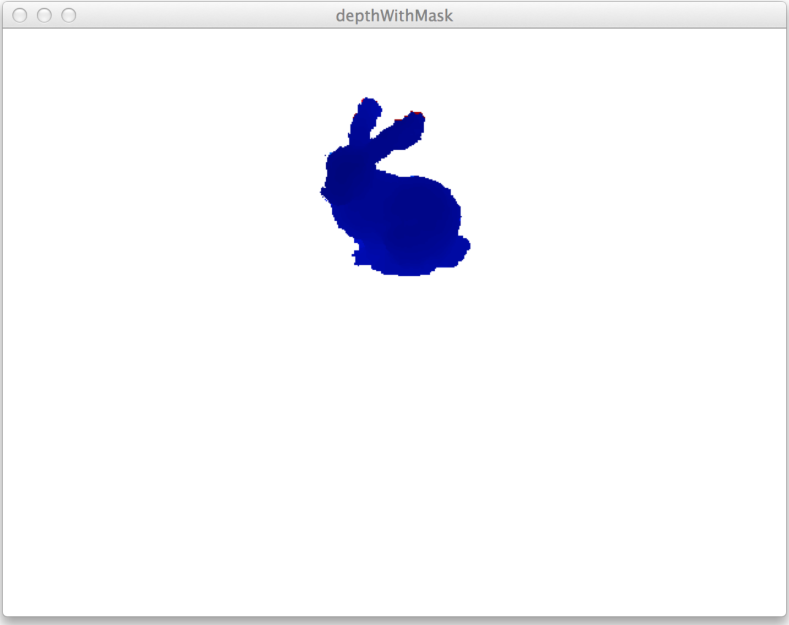
Multiview Levenberg-Marquardt ICP *(Fantoni et al. (2012))*

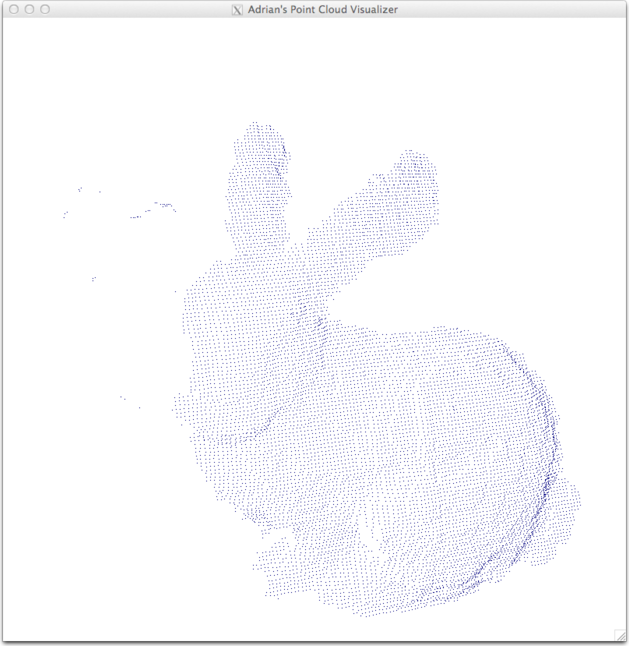
- modeled as pose graph optimization (SLAM context)
- sparsity needs to be exploited for efficiency
- solved with the g2o framework *(Kümmerle et al. (2011))*

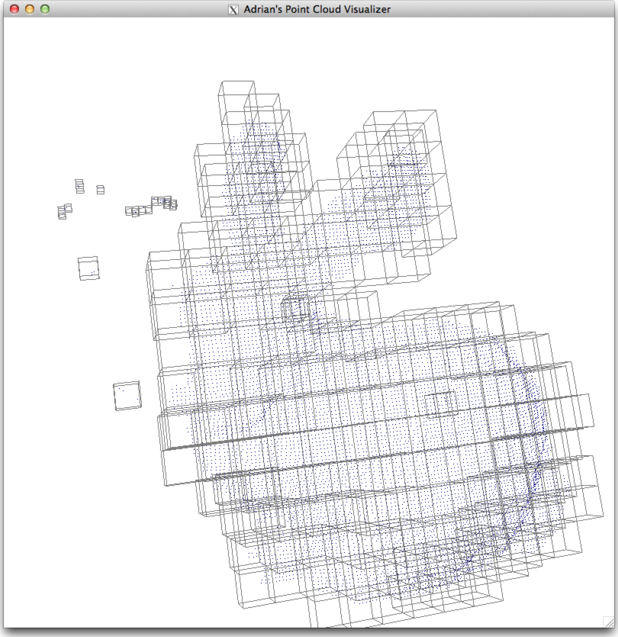
Outline

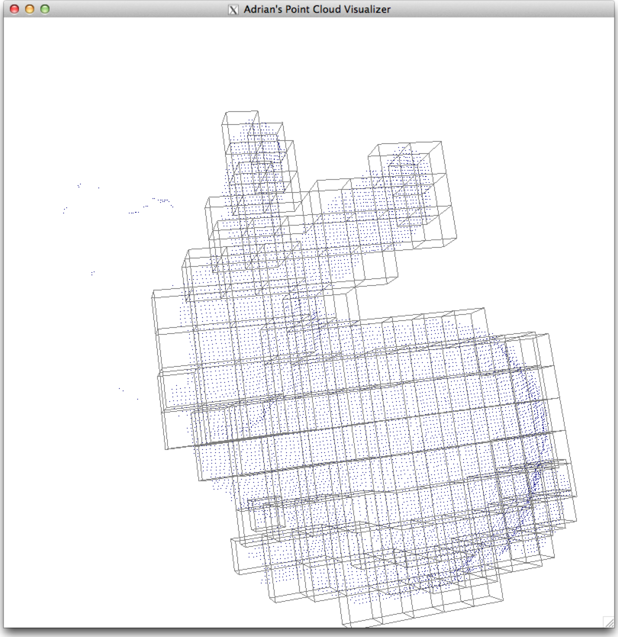
Outline

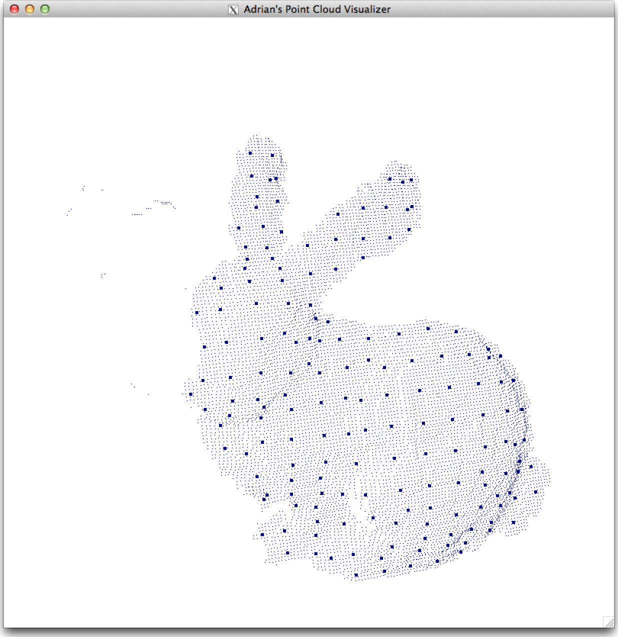

depth

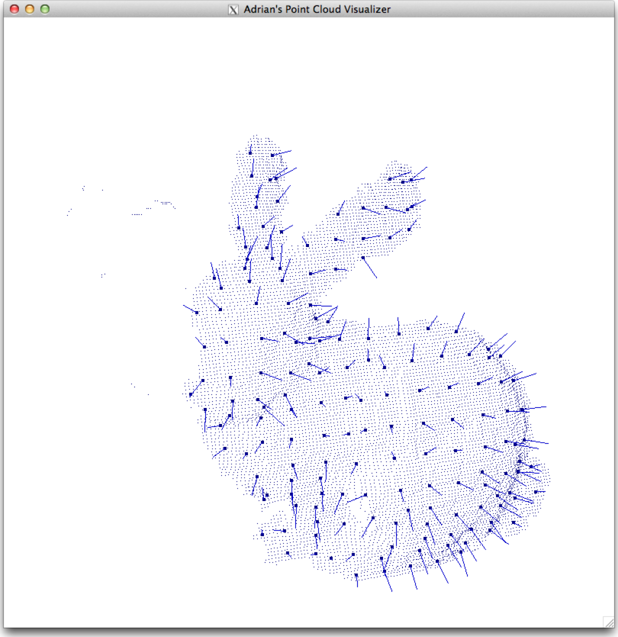

depthWithMask


Adrian's Point Cloud Visualizer


Adrian's Point Cloud Visualizer


Adrian's Point Cloud Visualizer


Adrian's Point Cloud Visualizer


Adrian's Point Cloud Visualizer

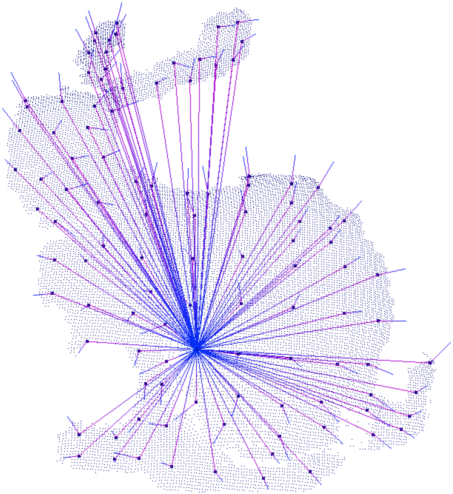
Outline

Pairwise coarse alignment

Consists of three substeps

1. Learning
2. Matching
3. Pose clustering and averaging

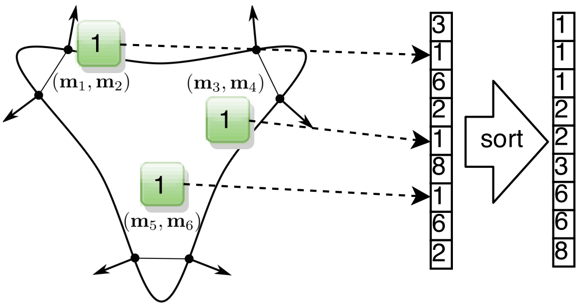
Learning: compute $O(N^2)$ PPF's



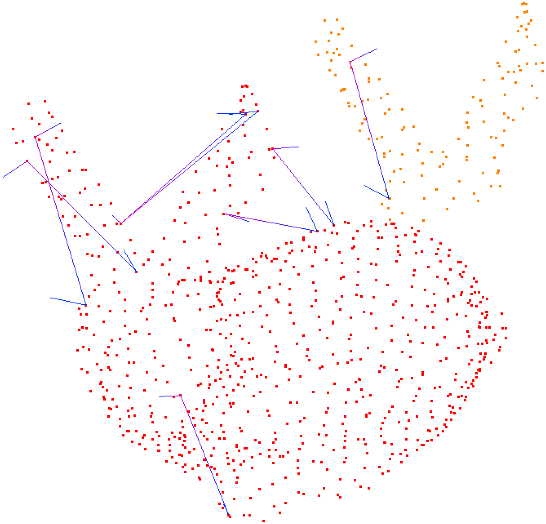
$$N \times (N-1) \text{ PPF's } F \in \mathbb{R}^4$$

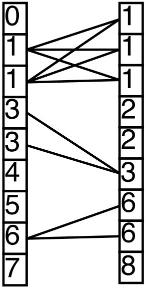
Learning: quantization and storage

$$quantizePPF(F,\delta,\theta) = (\lfloor\frac{||d||_2}{\delta}\rfloor, \lfloor\frac{\angle(n_1,d)}{\theta}\rfloor, \lfloor\frac{\angle(n_2,d)}{\theta}\rfloor, \lfloor\frac{\angle(n_1,n_2)}{\theta}\rfloor)$$

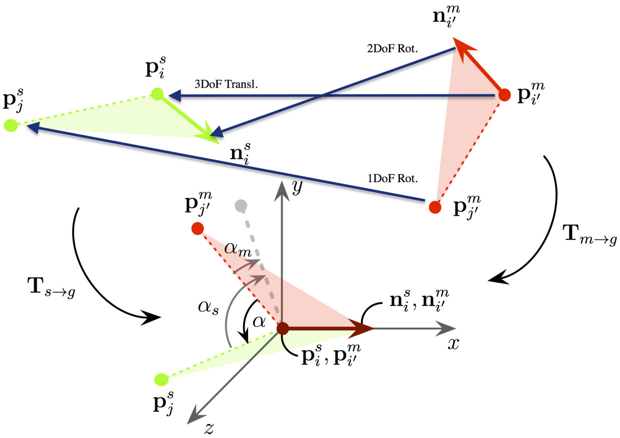


Matching: intersect 2 sorted lists



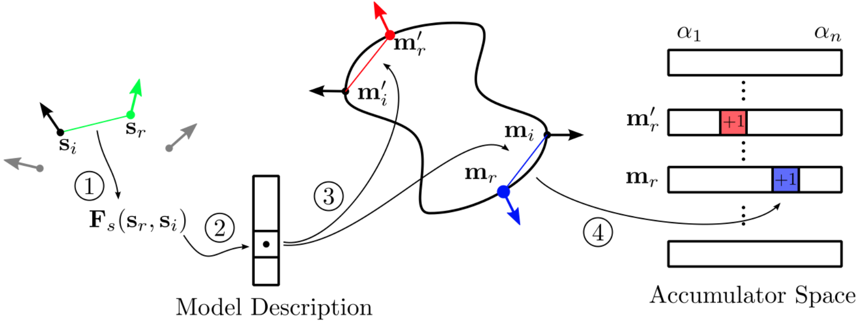


Matching: Intermediate Coordinate System



$$T_{m\to s} = T^{-1}_{s\to g} R_x(\alpha) T_{m\to g}$$

Matching: Voting Scheme

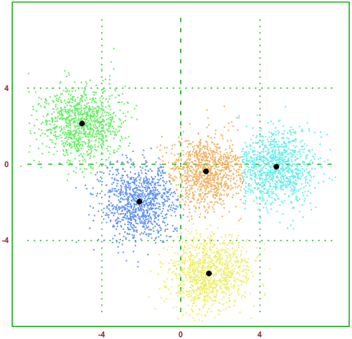


- Accumulator space $\mathcal{A}$: list of $|N^s|$ matrices $M^{|N^m| \times |\frac{360^\circ}{\theta}|}$
- $v \leftarrow max(\mathcal{A}(i^s))$
- $i^m, i^\alpha \leftarrow argmax(\mathcal{A}(i^s))$
- $\mathcal{P}_{i^s} = T^{-1}_{i^s \rightarrow g} R_x((i^\alpha + 0.5) * \theta) T_{i^m \rightarrow g}$

Pose clustering and averaging

clustering:

- sort poses by their votes
- agglomerative clustering (bottom up)
- complete linkage criterium : $d(\mathcal{K}(i), \mathcal{K}(j)) = max\{d(\mathcal{P}_i, \mathcal{P}_j), \mathcal{P}_i \in \mathcal{K}(i), \mathcal{P}_j \in \mathcal{K}(j)\}$



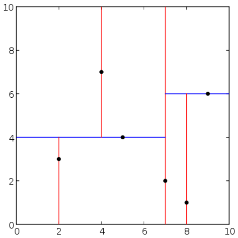
averaging:

- sum up votes
- translation: Euclidean mean
- rotation: quaternion mean
- return cluster average with highest number of votes

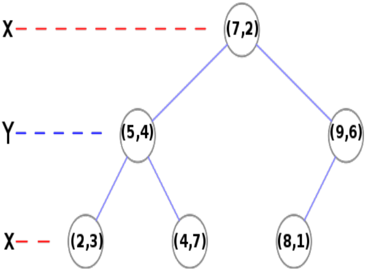
Outline

Pairwise refinement

- precomputed kd-tree



median splits

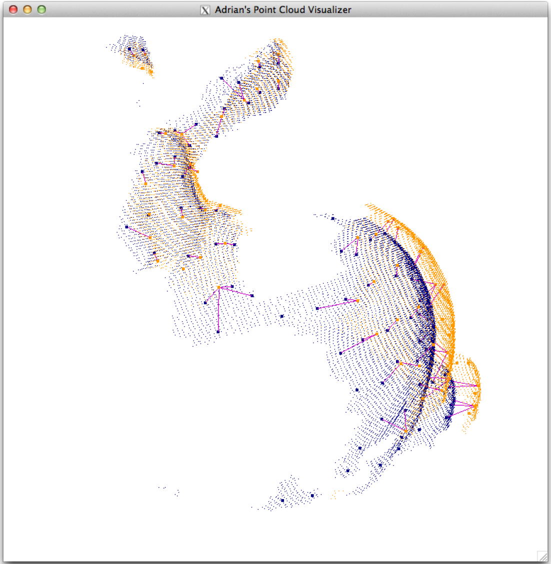


the resulting tree

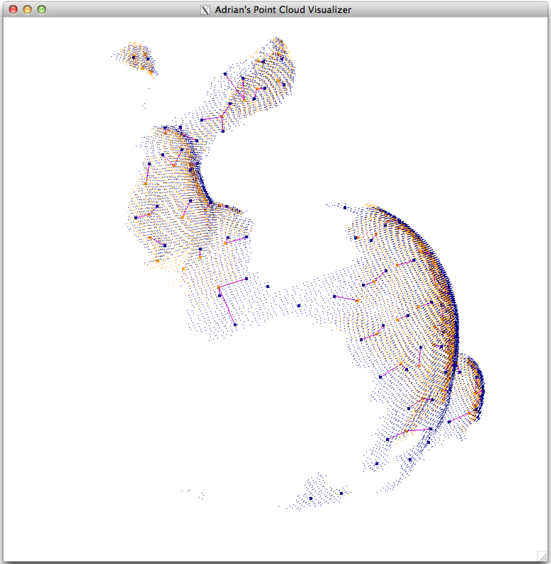
A k-d tree of $(2,3),(5,4),(9,6),(4,7),(8,1),(7,2)$

- point to plane ICP

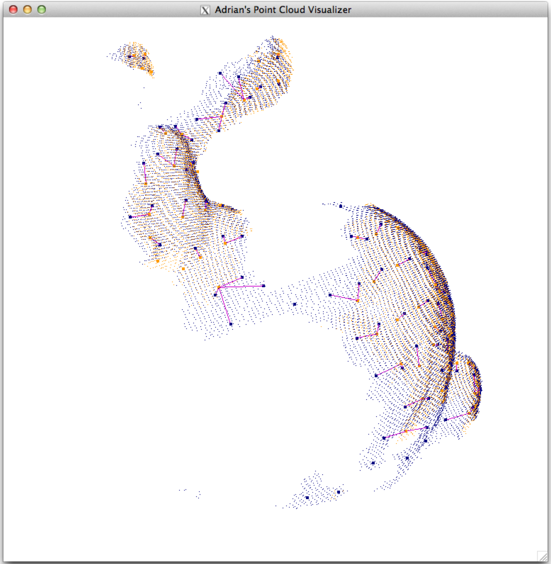
Pairwise refinement: ICP example 1



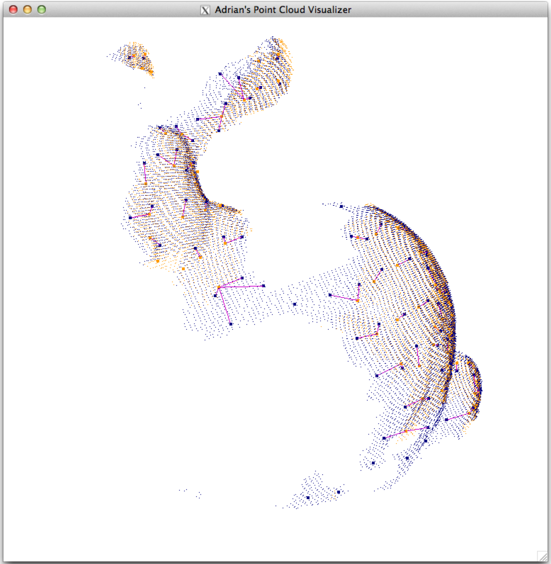
Pairwise refinement: ICP example 2



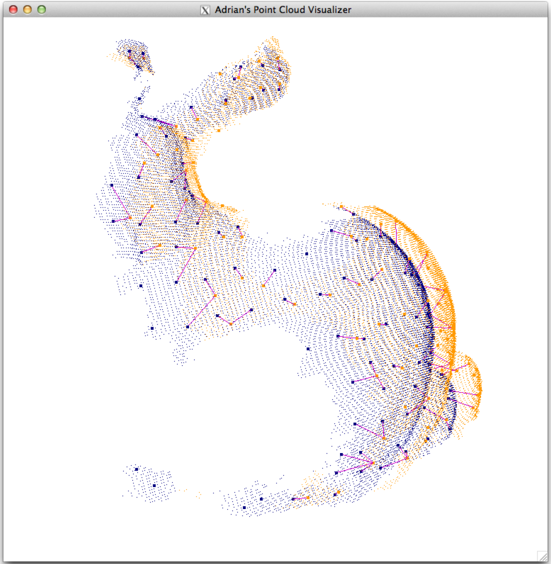
Pairwise refinement: ICP example 3



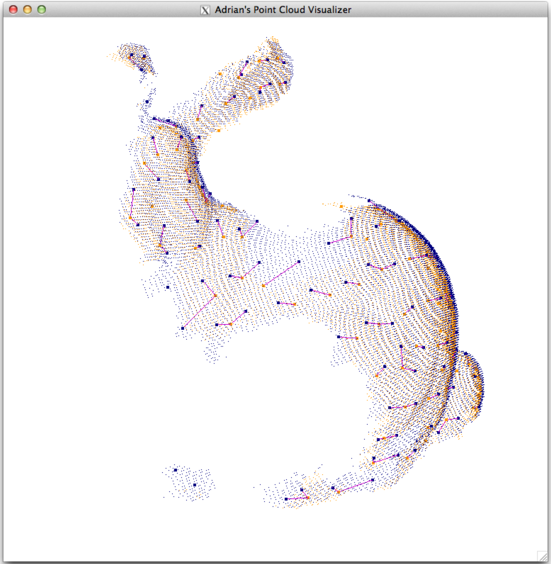
Pairwise refinement: ICP example 4



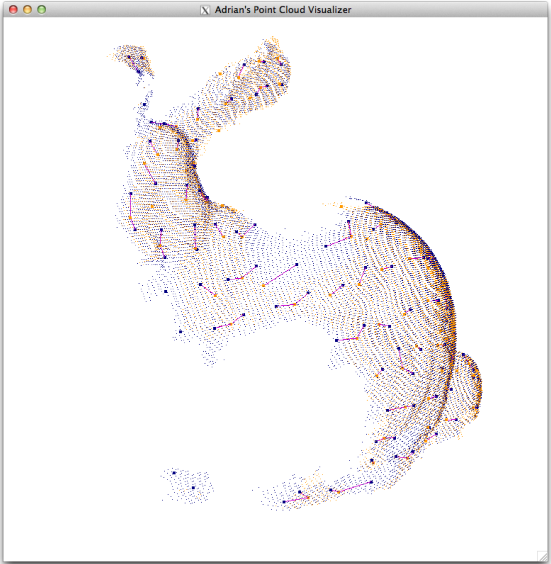
Pairwise refinement: ICP example 5



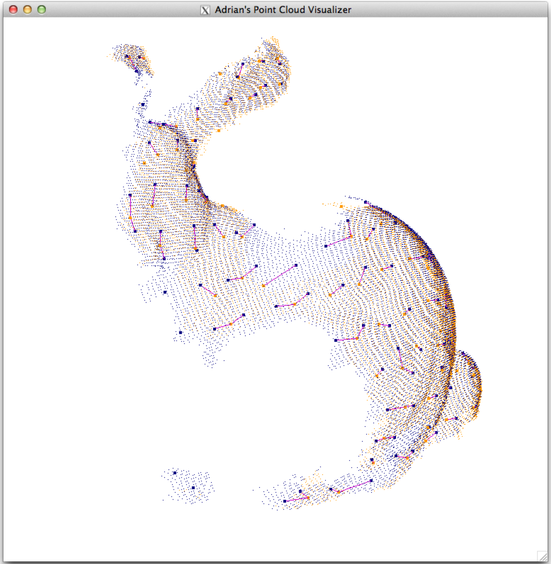
Pairwise refinement: ICP example 6



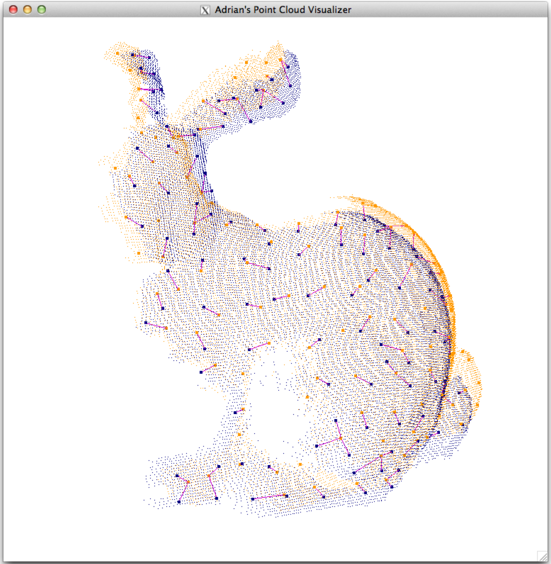
Pairwise refinement: ICP example 7



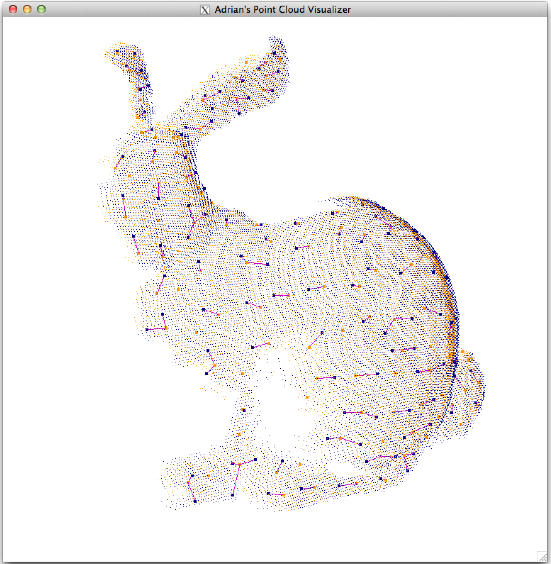
Pairwise refinement: ICP example 8



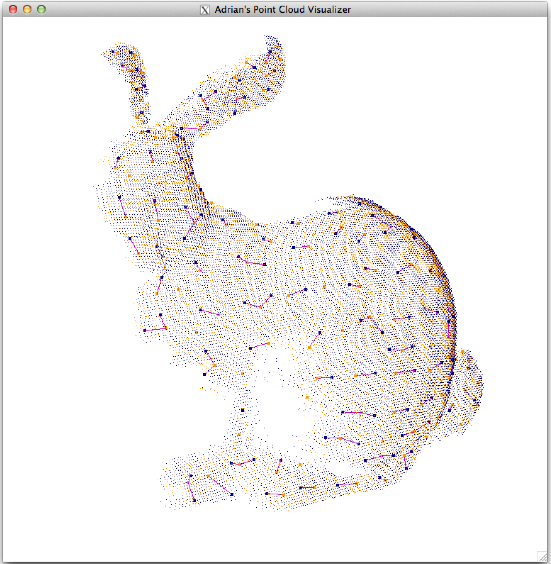
Pairwise refinement: ICP example 9



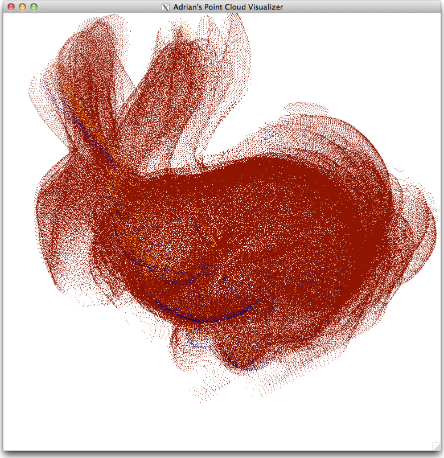
Pairwise refinement: ICP example 10

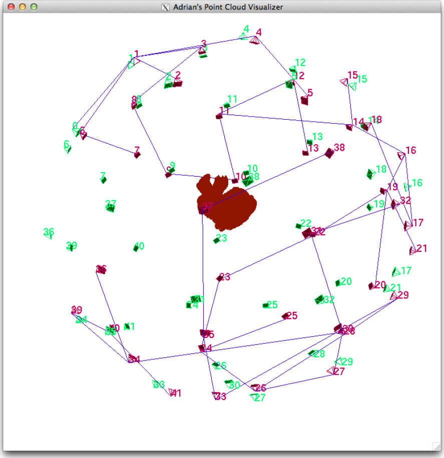


Pairwise refinement: ICP example 11

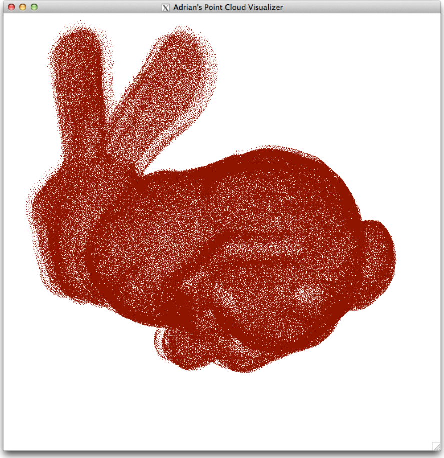


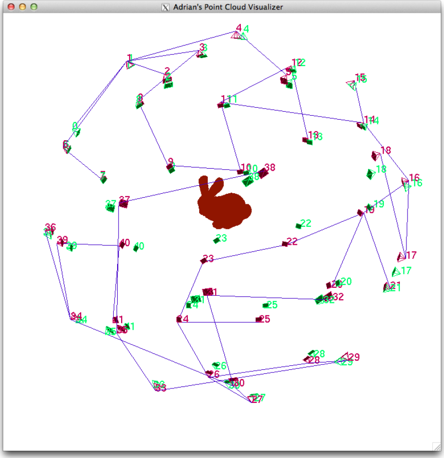
Pairwise refinement: reconstruction 1





Pairwise refinement: reconstruction 2



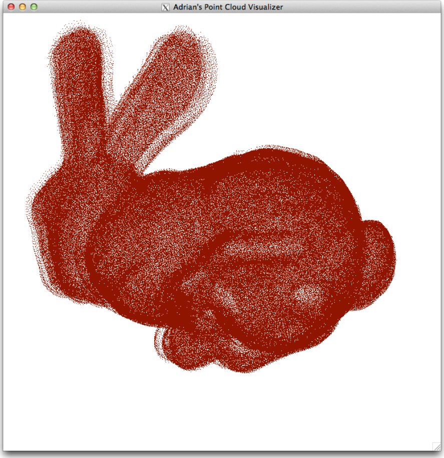


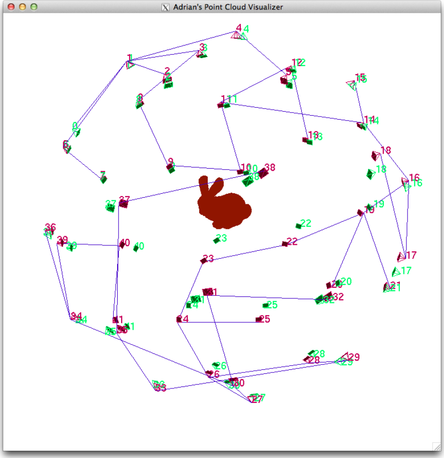
Outline

Multiview refinement

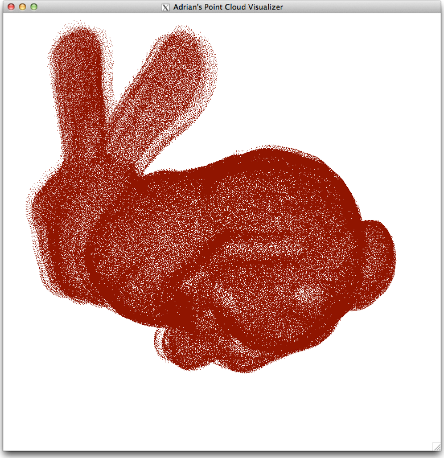
- relative $\rightarrow$ absolute pose graph
- add knn nearest poses
- closes loops

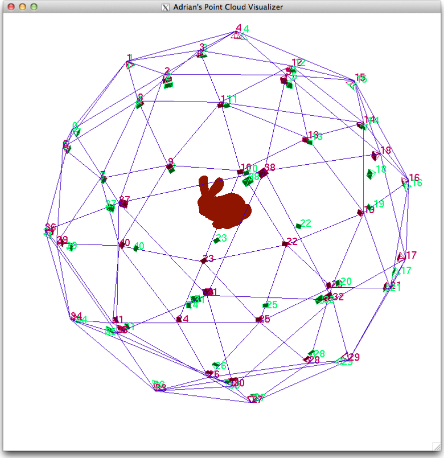
Multiview refinement: reconstruction 2





Multiview refinement: reconstruction 3



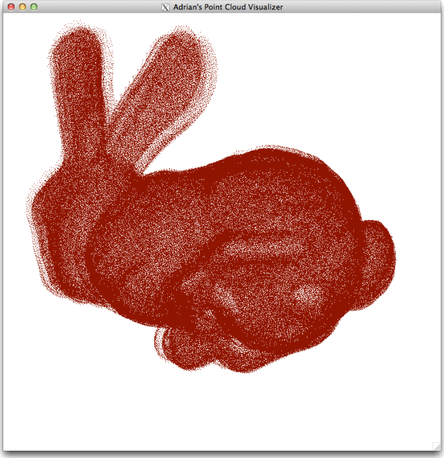


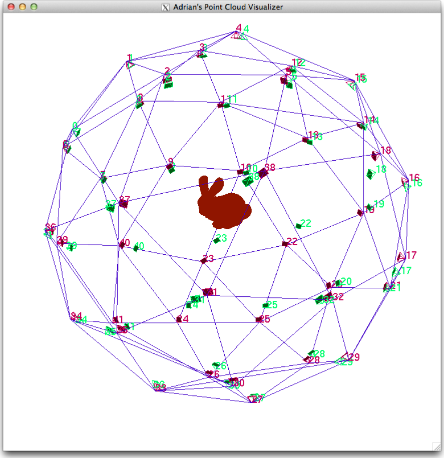
Multiview refinement

$$E(g_1, ...g_M) = \sum_{h=1}^{M} \sum_{k=1}^{M} A(h,k) \sum_{i=1}^{N_h} ||d(g_h(p_i^h), g_k(q_i^h))||^2 \qquad (5)$$

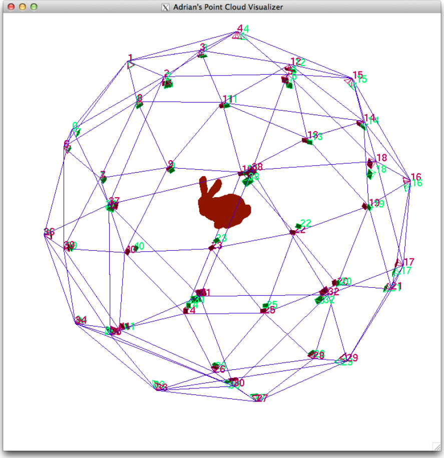
- A : graph adjacency matrix
- Multiview LM-ICP modeled as a graph in g2o
- error minimized using LM / Dogleg

Multiview refinement: reconstruction 3





Multiview refinement: reconstruction 4



Outline

Evaluation

Stages:

0	A	B	C
preproc.	pairwise align.	pairwise refin.	multiview refin.

Metrics:
The relative pose error at time step i with interval width Δ:

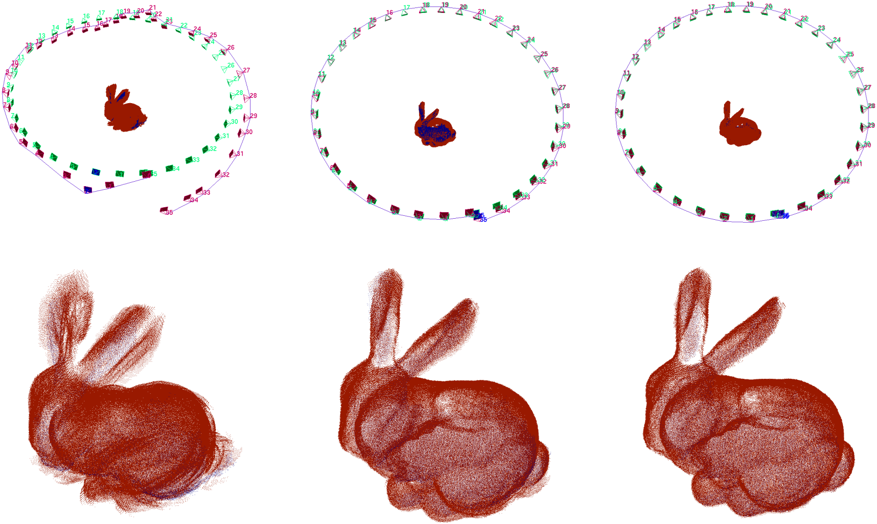
$$E_{i,\Delta} := (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta})^{-1} \qquad (\textbf{RPE})$$

The absolute trajectory error at time step i:

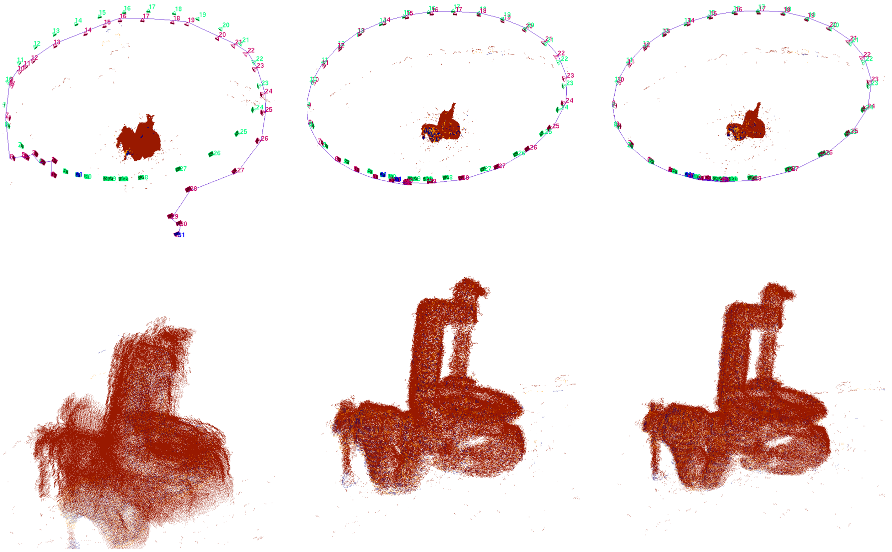
$$F_i := Q_i^{-1} S P_i \qquad (\textbf{ATE})$$

$$\underbrace{\{P_1, \cdots, P_n\}}_{\textit{estimates}}, \underbrace{\{Q_1, \cdots, Q_n\}}_{\textit{groundtruth}} \in SE(3)$$

Outline

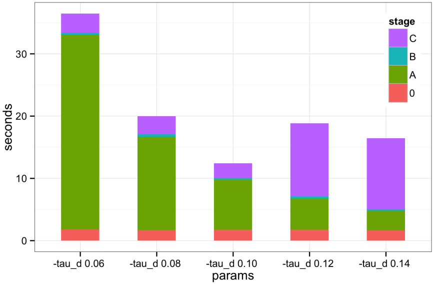


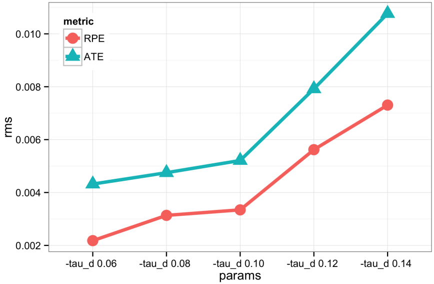
bunny2



bench vise -step 15

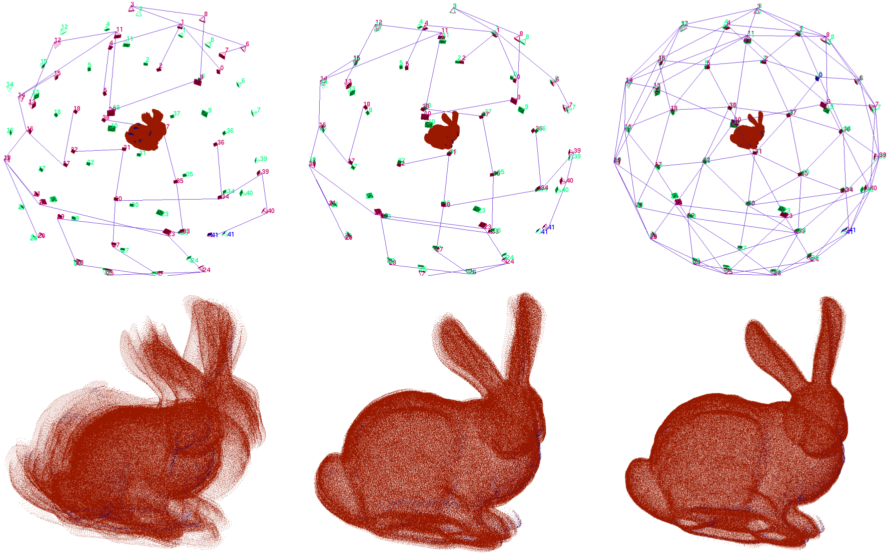
sequence params	S	RPE [mm/%] rms	RPE [mm/%] mean	RPE [mm/%] std	ATE [mm/%] rms	ATE [mm/%] mean	ATE [mm/%] std
bunny2	A	32.311	28.996	14.254	74.097	68.679	27.812
	↓	88	89	85	91	92	89
	B	3.936	3.330	2.098	6.610	5.803	3.165
	↓	15	14	17	21	15	47
	C	3.344	2.859	1.735	5.212	4.932	1.685
bench vise -step 15	A	72.998	67.519	27.747	215.475	184.514	111.283
	↓	83	85	74	76	75	78
	B	12.287	9.979	7.168	52.579	46.602	24.346
	↓	26	23	34	59	58	62
	C	9.031	7.676	4.758	21.737	19.662	9.270



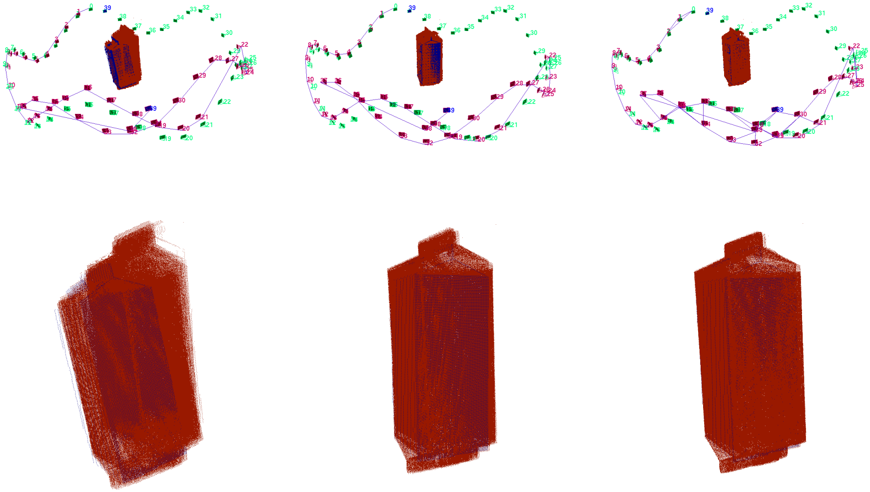


Speed vs. accuracy on **bunny2** : *The distance sampling rate τ_d is modified*

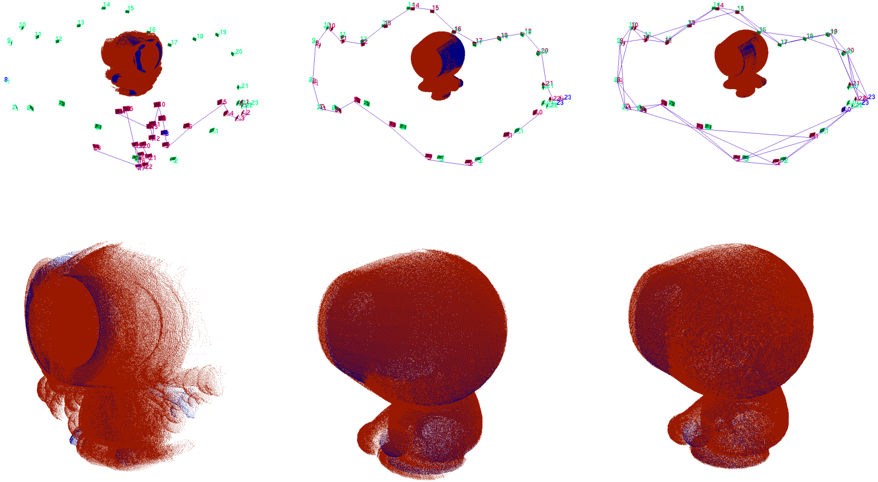
Outline



bunny sphere -knn 5 -nFrames 5

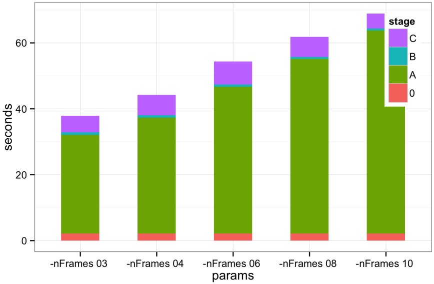


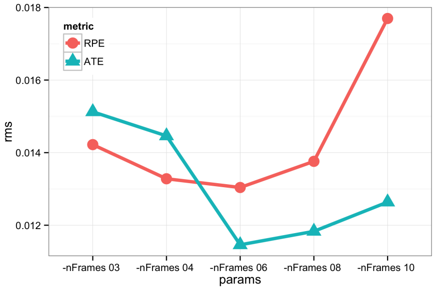
juice box wavy -step 3



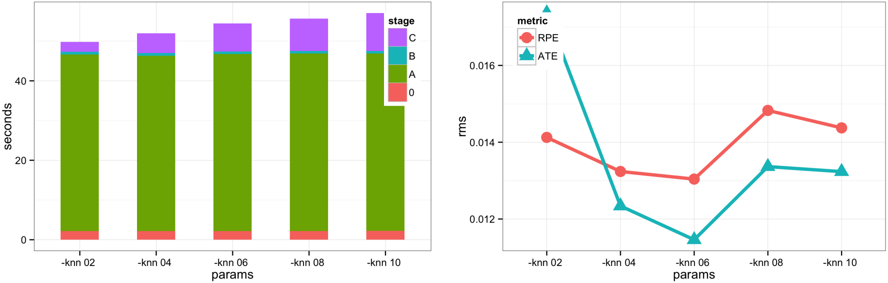
Kenny wavy -step 5 -diamM 0.4 -knn 4

sequence params	S	RPE [mm/%] rms	mean	std	ATE [mm/%] rms	mean	std
bunny sphere -knn 5 -nFrames 5	A	65.799	54.217	37.283	125.697	108.222	63.937
	↓	69	69	70	76	76	78
	B	20.212	16.833	11.189	29.737	26.353	13.776
	↓	34	35	33	58	57	61
	C	13.278	10.929	7.540	12.445	11.256	5.309
juice box wavy -step 3	A	124.981	39.776	118.482	520.406	442.792	273.418
	↓	3	21	1	2	1	4
	B	121.612	31.256	117.526	511.973	438.984	263.457
	↓	-1	-0	-1	0	1	-2
	C	122.356	31.375	118.265	509.811	433.165	268.841
Kenny wavy -step 5 -diamM 0.4 -knn 4	A	146.957	137.339	52.292	478.807	467.906	101.589
	↓	86	89	76	90	92	74
	B	20.004	15.523	12.618	46.516	38.589	25.973
	↓	12	15	7	26	25	28
	C	17.619	13.126	11.753	34.490	29.064	18.571





Speed vs. accuracy on **bunny sphere** -knn 06 : *The number nFrames of frames to match to during the coarse pairwise alignment is modified*



Speed vs. accuracy on **bunny sphere** -nFrames 06 : *The number of knn nearest neighbors to add to the pose graph before multiview refinement is modified*

Outline

Conclusion

- reconstruct small objects
- from a few range images
- in about 1 minute

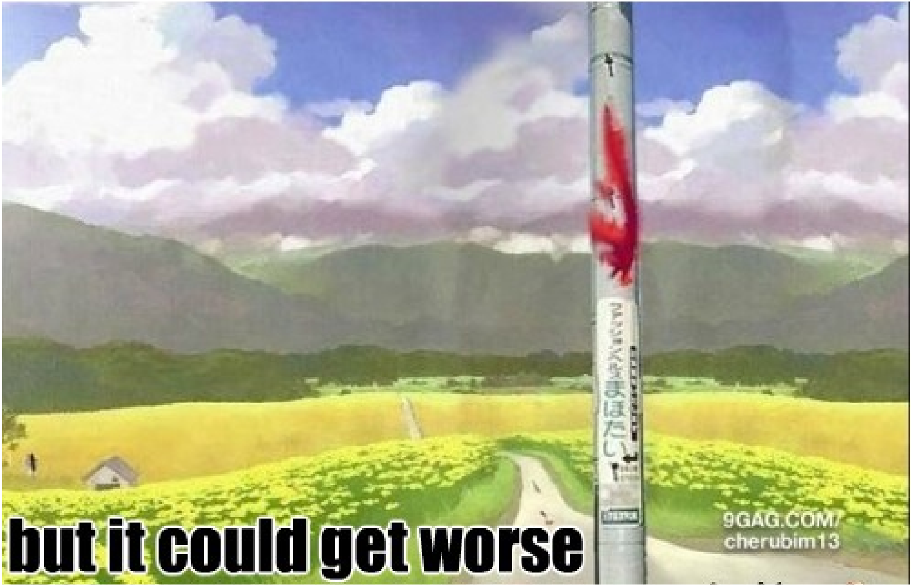
- need variation in geometry

- need to find right set of parameters
- so that accuracy improves in each step

Future work

- reduce number of parameters
- use curvature to speedup PPF matching
- use color for robustness
- Ceres Solver for multiview refinement $\rightarrow$ Master's thesis


it hurts to say GOODBYE


but it could get worse
9GAG.COM/
cherubim13

Backup slides

Point set PCA

$$Cov(P) = \frac{1}{P}\sum_{i=1}^{P}(p_i - \bar{p})(p_i - \bar{p})^T = C, C \in \mathbb{R}^{3\times 3}$$

$$e_1, e_2, e_3 \in \mathbb{R}^3 \tag{6}$$

eigenvectors(C)

$$\lambda_1 > \lambda_2 > \lambda_3 \in \mathbb{R} \tag{7}$$

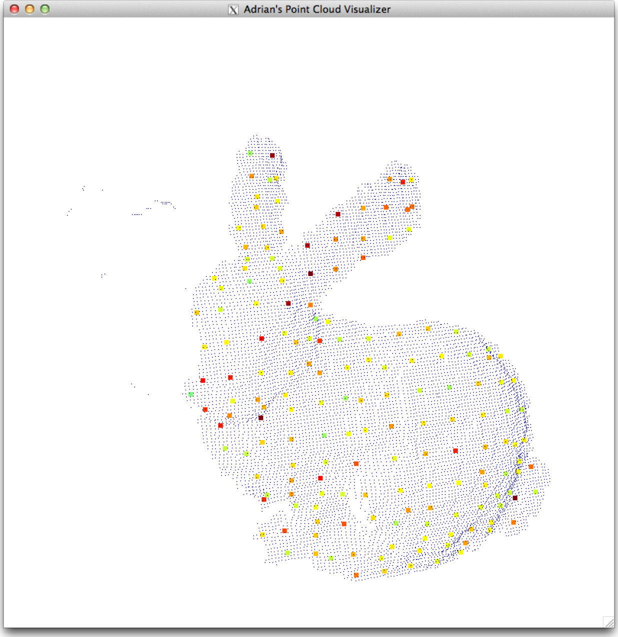
eigenvalues(C)

$$n(\{p_i\}) = -sign(e_{3,z})\frac{e_3}{||e_3||} \tag{8}$$

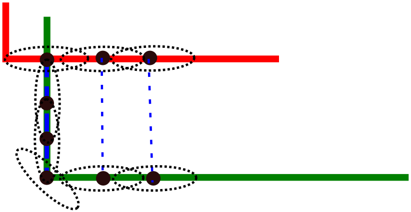
normal

$$\sigma(\{p_i\}) = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \tag{9}$$

curvature


Adrian's Point Cloud Visualizer

ICP: Generalized ICP

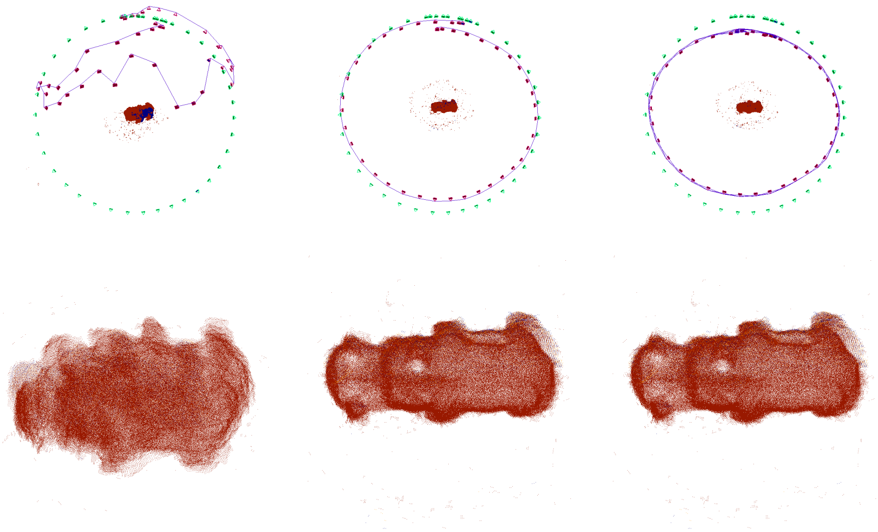


plane to plane distance

$$E = \sum_{i=1}^{N} f(p_i, q_i, n_{p_i}, n_{q_i}) \tag{10}$$

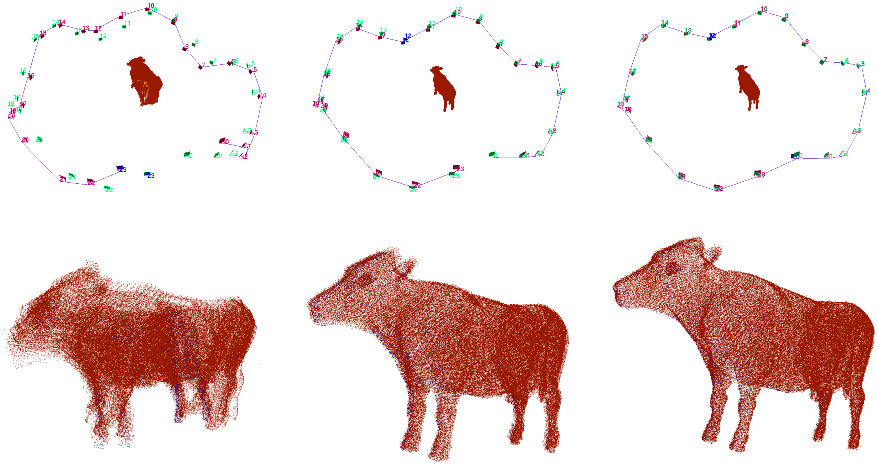
plane to plane

Segal et al (2009)

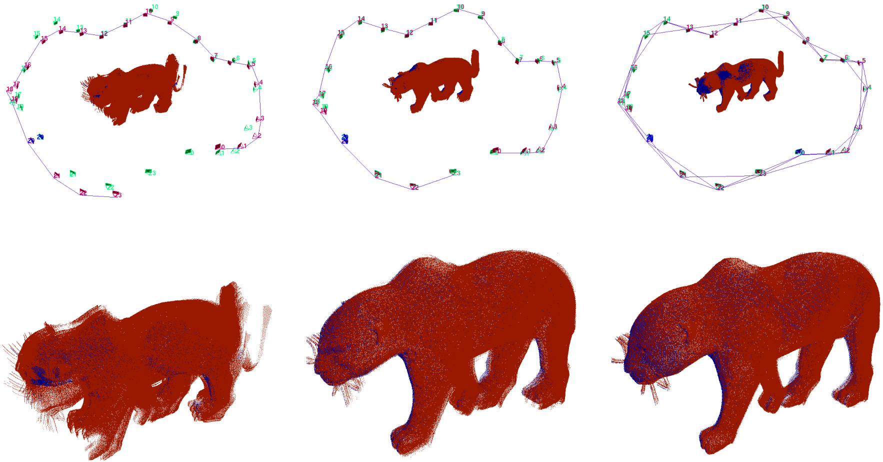


Pumba -diamM 0.07 -step 15 -knn 4 -dmax 0.05

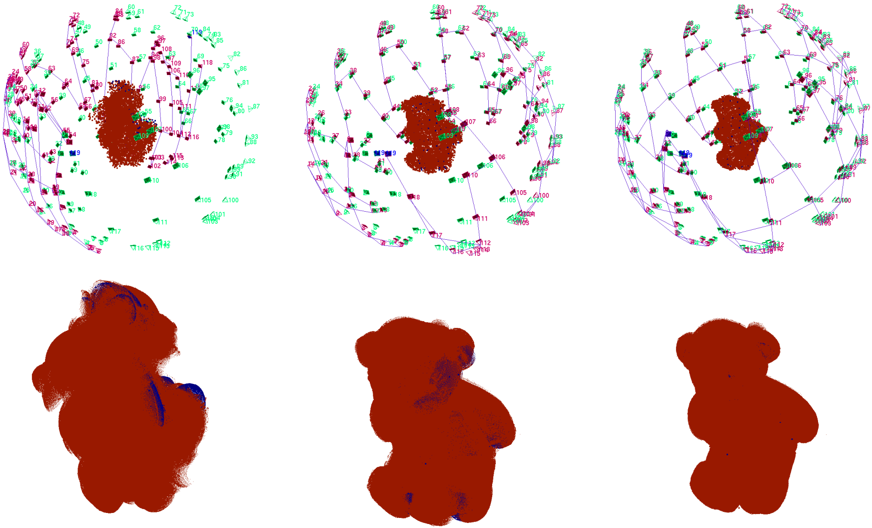
sequence params	S	RPE [mm/%] rms	mean	std	ATE [mm/%] rms	mean	std
Pumba	A	111.926	90.269	66.172	367.296	337.233	145.534
-diamM 0.07	↓	83	84	81	74	72	82
-step 15	B	19.295	14.810	12.367	96.967	93.496	25.710
-knn 4	↓	-12	-8	-17	-16	-17	-14
-dmax 0.05	C	21.519	15.982	14.411	112.917	109.038	29.342



cow wavy -step 5



leopard wavy -step 5 -diamM 0.3 -knn 4



teddy abrupt -nFrames 5 -knn 4 -diamM 0.5

sequence params	S	RPE [mm/%] rms	mean	std	ATE [mm/%] rms	mean	std
cow wavy -step 5	A	38.893	32.589	21.228	78.353	71.332	32.418
	↓	87	88	86	86	86	87
	B	4.915	3.947	2.929	10.674	9.814	4.196
	↓	18	29	2	50	51	46
	C	4.021	2.803	2.883	5.301	4.789	2.273
leopard wavy -step 5 -diamM 0.3 -knn 4	A	36.106	33.840	12.590	70.558	62.355	33.019
	↓	82	83	74	87	87	88
	B	6.569	5.671	3.314	9.335	8.415	4.041
	↓	15	28	-15	43	43	43
	C	5.594	4.109	3.796	5.280	4.756	2.292
teddy abrupt -nFrames 5 -knn 4 -diamM 0.5	A	73.390	59.036	43.599	274.761	255.011	102.287
	↓	82	81	84	86	86	82
	B	13.423	11.397	7.091	39.428	35.027	18.102
	↓	33	30	40	50	48	56
	C	9.028	7.976	4.230	19.715	18.059	7.908