Feature-Feature Matching For Object Retrieval in Point Clouds

Michal Staniaszek
Supervisor: John Folkesson

Royal Institute of Technology

June 11, 2015

Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

What is object retrieval?

Finding object locations in point clouds

Can use knowledge about objects to

- Determine the type of room one is in
- Find object in order to perform a task with it
- Predict or track object position over time

Many of these useful in household/office applications

Two main approaches

- Feature matching: match low level representations of objects
- Model matching: match entire object or high level representation

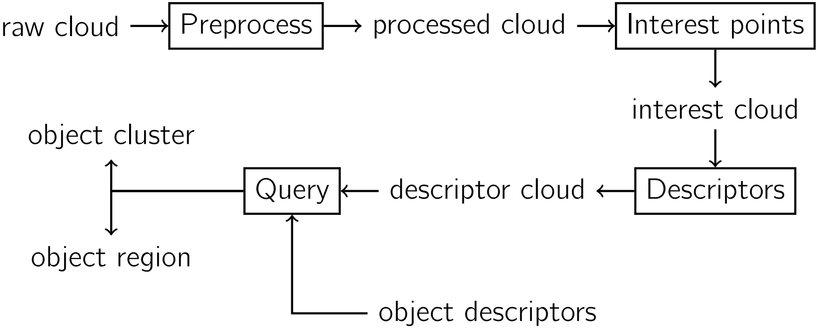
This Project

Focus on feature matching, implement system for object retrieval. Working with 3D dataset.

Questions that the project aims to investigate:

- Can naive representations be effective?
- Which representations are best?

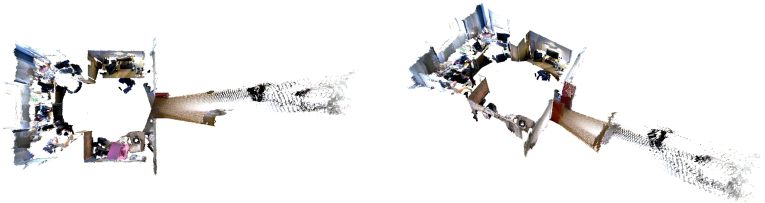
System Structure



Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

Data

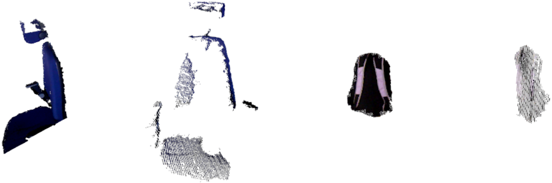


Data set consists of:

- Full scans of a room over 1 month
- Scans constructed from intermediate frames
- Labels for some persistent objects in the rooms

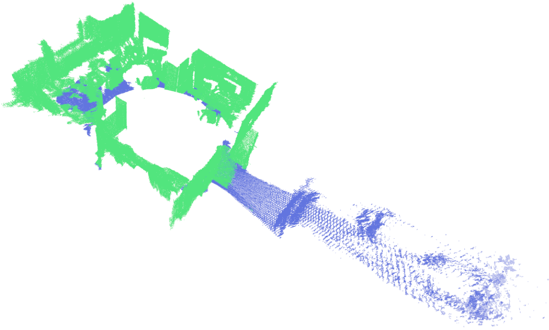
Can exploit the fact that dataset has known structure, but also want to generalise.

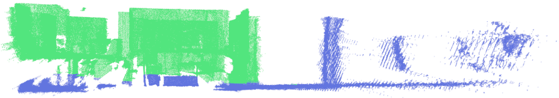
Downsampling, transformation and trimming



Clouds have approx. 4mil points.

- Need to reduce this for more efficient processing
- Can help reduce noise and slicing effects





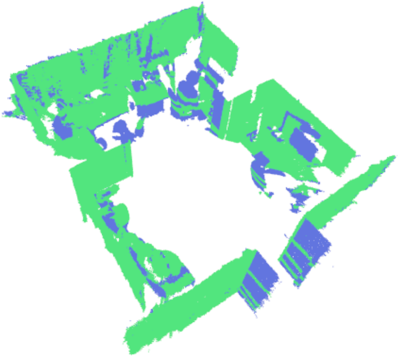
Plane Extraction

Apply RANSAC to trimmed clouds multiple times

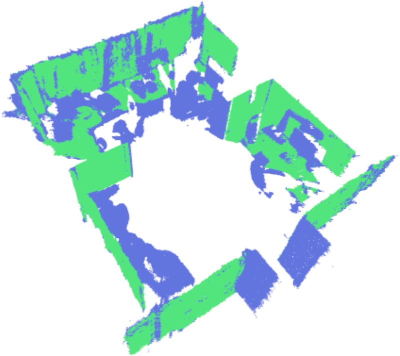
- Remove walls, desks and other flat surfaces
- These are not likely to be part of objects

Different plane models; with or without normals

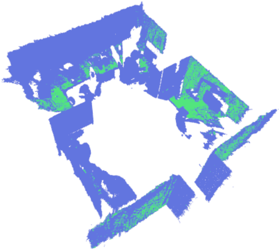
- With normals is better, but requires adjustment of the normal radius



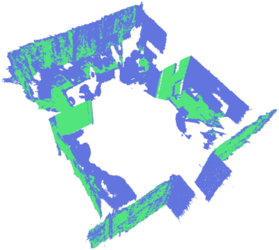
Simple plane



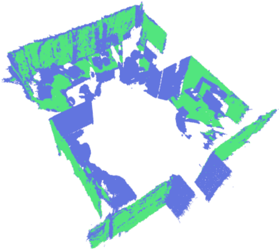
Plane with normal



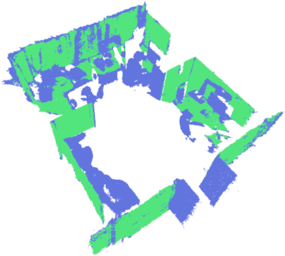
0.02



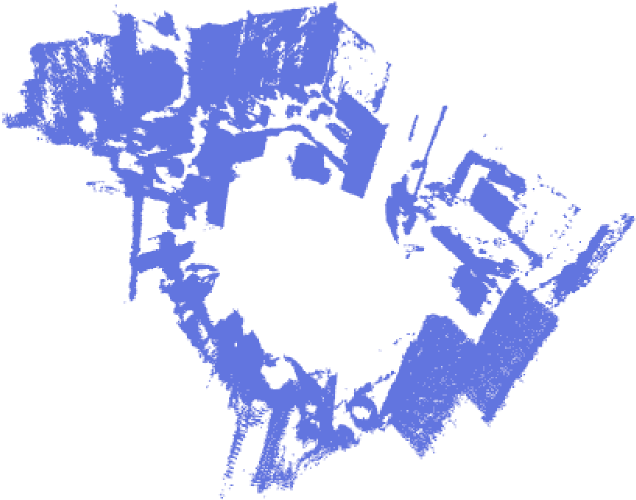
0.04



0.06



0.08



Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

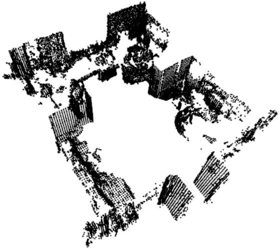
Q: Where does one compute features? **A:** “Interesting” points

Interesting points usually defined by:

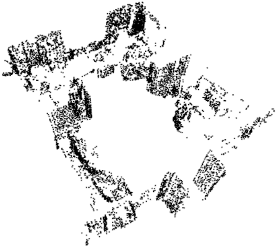
- Minima/maxima of certain properties
- e.g. curvature, intensity, point covariances

Increasing complexity:

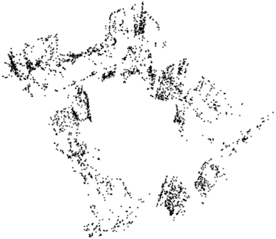
Uniform:	Voxel grid
ISS:	Scatter matrix
SUSAN:	Size of similar regions
SIFT:	Difference of Gaussians over scales



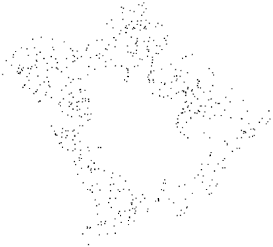
Uniform



ISS



SUSAN



SIFT

Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

What is a descriptor?

Used to represent regions in image or point cloud

- Compresses information about the region
- Simplifies comparison between regions

Desirable qualities:

- Similar regions produce similar descriptors
- Noise tolerance
- Unique reference frame (scale/rotation invariance in 2D)

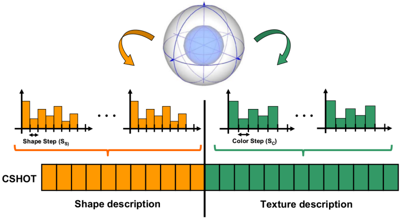
Create compact representations of query and target clouds and compare them.

SHOT descriptors

Computation:

- Select central point p
- Find cosine of angle between normal p and neighbours within radius R
- Add to local histogram bins
- Combine local histograms

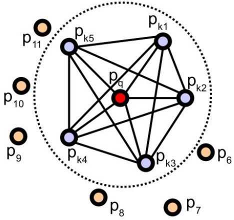
SHOTCOLOR extends the above by adding the L_1 norm of the colour values to the shape description histogram



PFH descriptors

Computation:

- Look at all pairs of points in radius R
- Define coordinate frame using normal
- Compute 4 features based on angles and distances between the two points
- Increment histogram based on index from feature combination



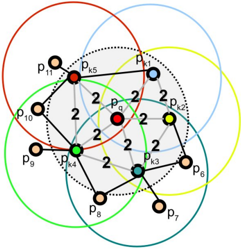
PFH extensions

FPFH:

- Look at point pairs within radius of each point instead of all pairs
- Concatenate separate feature histograms instead of indexing (reduce redundancy)

PFHRGB:

- Adds 3 features for ratio of colour values between points



Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

Query objects

Query objects are:

- Instances of objects taken from annotation clouds
- Assumed to contain only points on the object

Could use individual frames for input, but

- Would require segmentation to extract object
- Would likely be less accurate, since frame can contain multiple objects and also non-objects

Matching

Find closest K descriptors to each point in the query cloud.

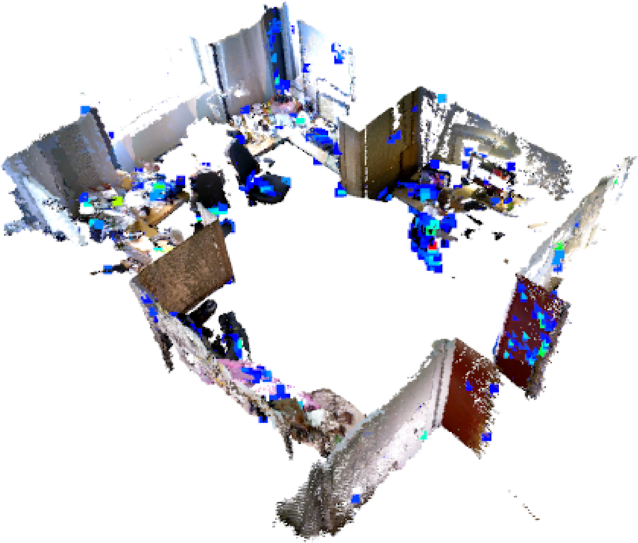
- Distances found using simplified L_2 norm
- Comparing query to target means all query points have a nearest neighbour in target.
- No point finding neighbours for each target point — for most target points, point in query likely to be far away

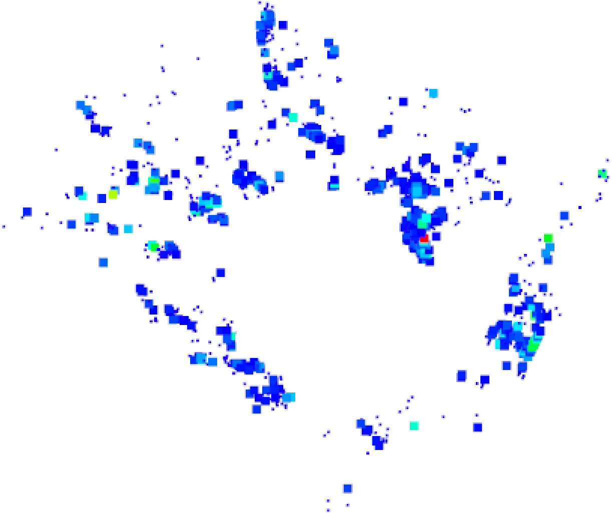
Output of this step is a list of nearest neighbours for each point

Voting

Construct 3D grid and populate cells with values

- Each descriptor has an associated point in 3D space
- Increment cell containing point for each descriptor in nearest neighbour list
- Each cell represented by its centre point

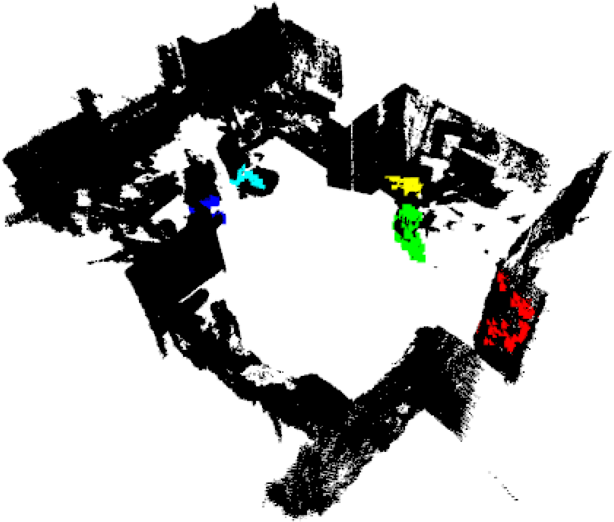


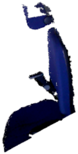


Clustering

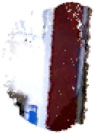
Use cells with top N number of votes to cluster

- Assume that the top votes are likely to be on matching objects
- Regions with many points in the top votes are likely to be on the object
- Use Euclidean clustering on the top points
- Clusters sorted based on summed score of points they contain
- Centroid of cluster used to extract region from target cloud

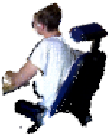




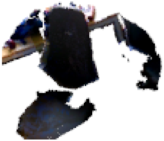
Original



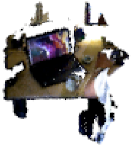
526 votes



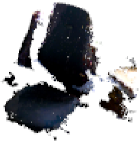
458 votes



295 votes



118 votes



116 votes

Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

Experimental Setup

- Run preprocessing on all clouds using several settings
- Compute descriptor and interest point combinations on these settings
- Run query on 7 objects using settings of the above, and vary query parameter settings

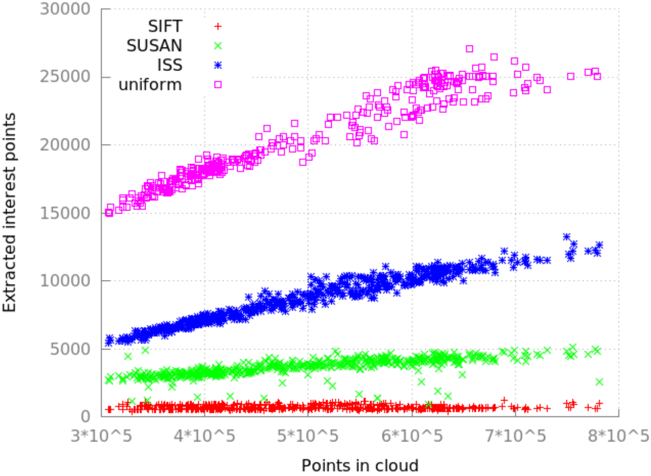
Cloud size reduction from preprocessing

Setting	Downsample	Trim	Trim Orig	Plane	Plane Orig
DEF	0.23±0.01	0.79±0.01	0.18±0.00	0.50±0.06	0.09±0.01
DS15	0.11±0.00	0.76±0.02	0.08±0.00	0.67±0.12	0.06±0.01
DS2	0.06±0.00	0.74±0.02	0.05±0.00	0.89±0.07	0.04±0.00

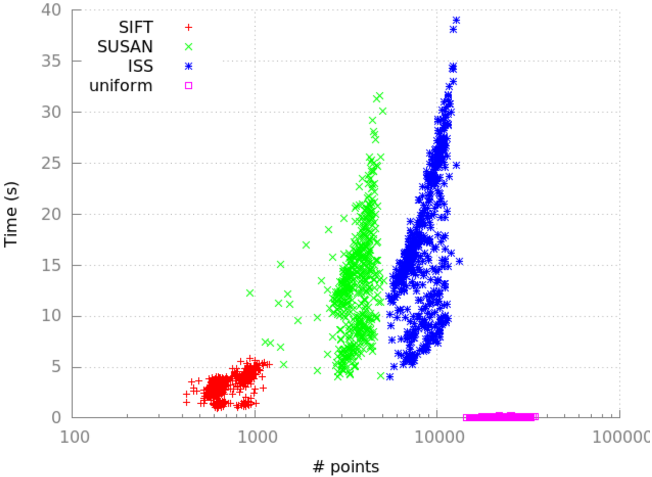
Time taken for preprocessing

Setting	Normals	Planes	PerPlane	Total
DEF	16.33±1.11	116.12±13.63	15.10±1.43	133.32±14.13
DT	12.29±1.30	114.30±18.81	13.69±1.65	127.35±19.44
RI500	13.20±1.23	262.81±60.10	34.19±9.00	278.90±60.37

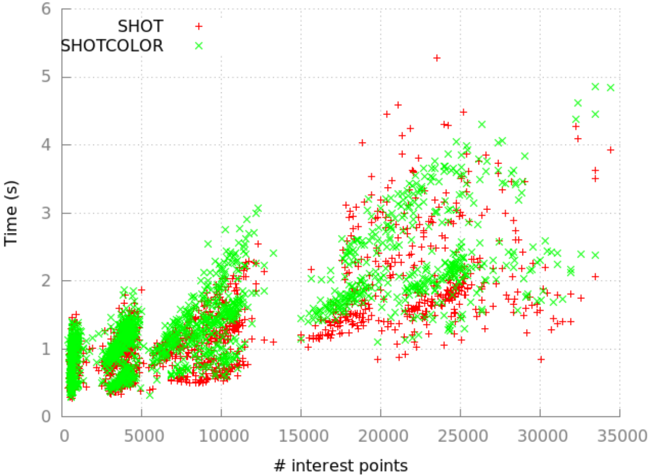
Interest points extracted vs. method



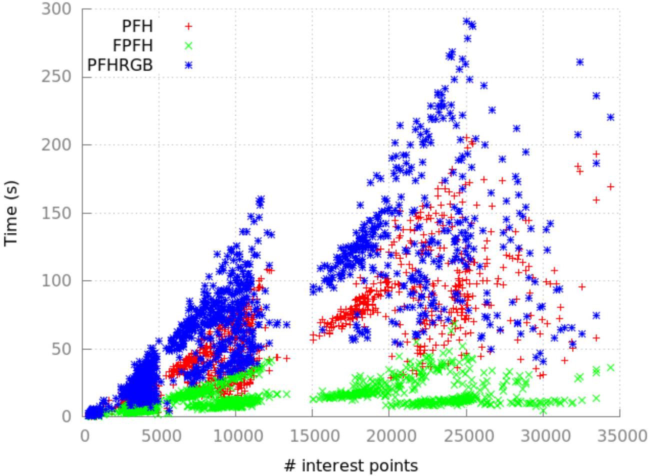
Interest points computation time vs. method



Descriptors computation time (SHOT)



Descriptors computation time (PFH)



Query Objects

Object Query retrieval rates

		All		SHOTC,PFHRGB,Uniform,ISS	
Query	Setting	Backpack	Chair	Backpack	Chair
DEF	DEF	14.2±22.7	29.2±24.4	31.1±28.6	39.9±31.2
	DT	18.0±25.6	33.0±24.4	40.5±28.3	50.8±29.2
K10R2	DEF	18.1±22.2	40.4±33.1	30.5±28.0	43.4±34.8
	DT	**21.7**±24.6	49.9±29.1	**39.7**±28.5	56.3±31.5
K10R25	DEF	12.2±16.6	40.2±33.4	20.6±22.1	44.5±34.9
	DT	19.3±22.3	**53.3**±30.1	35.8±26.1	**57.2**±31.4
Actual counts		79	85	79	85

Object query retrieval rates (DT/DEF), descriptor

Object	Descriptor	Top	Total	Unique	Actual
Backpack	FPFH	2.7±1.5	3.3±2.5	3.3±2.5	80
	PFH	1.0±1.0	2.3±3.2	2.3±3.2	– " –
	PFHRGB	**49.3**±7.2	**56.0**±13.2	**56.0**±13.2	– " –
	SHOT	2.7±3.8	3.3±4.9	3.3±4.9	– " –
	SHOTCOLOR	9.3±8.6	25.0±33.3	25.0±33.3	– " –
Chair	FPFH	15.0±6.2	21.7±10.1	21.7±10.1	85
	PFH	10.0±9.3	17.8±12.3	17.2±11.6	– " –
	PFHRGB	**36.7**±18.9	**60.7**±39.6	**54.0**±33.8	– " –
	SHOT	8.3±9.1	29.7±19.7	29.7±19.7	– " –
	SHOTCOLOR	25.7±21.1	55.0±37.3	47.7±30.9	– " –

Object query retrieval rates (DT/K10R2), descriptor

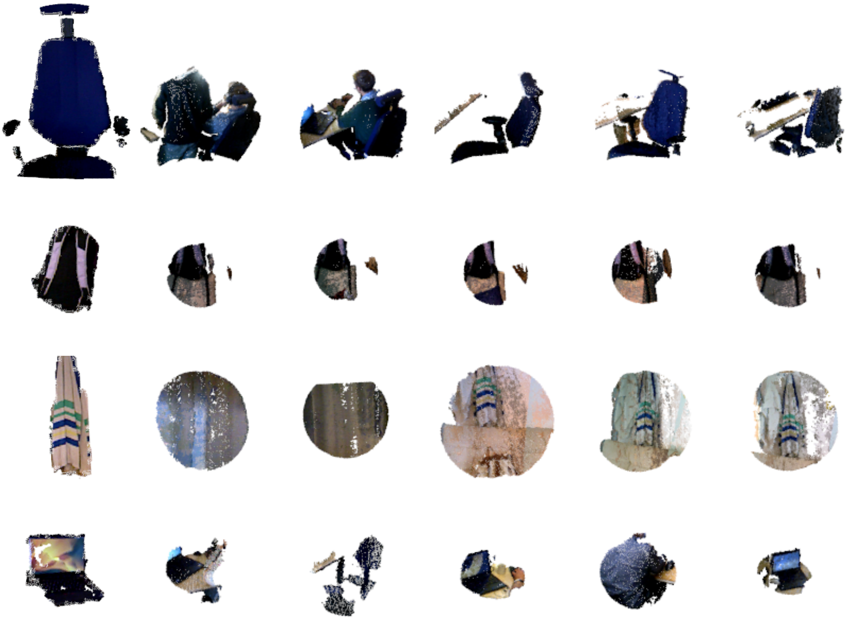
Object	Descriptor	Top	Total	Unique	Actual
Backpack	FPFH	2.7±1.5	13.0±14.8	13.0±14.8	80
	PFH	2.0±2.6	12.3±16.4	12.3±16.4	– " –
	PFHRGB	**42.3**±11.0	**53.3**±19.7	**53.3**±19.7	– " –
	SHOT	1.0±1.7	4.0±6.9	4.0±6.9	– " –
	SHOTCOLOR	6.7±5.5	26.0±32.9	26.0±32.9	– " –
Chair	FPFH	16.0±7.8	43.7±28.4	42.0±27.1	85
	PFH	10.7±10.0	45.0±30.3	43.0±28.6	– " –
	PFHRGB	**38.0**±15.7	**70.0**±41.6	**59.0**±32.0	– " –
	SHOT	6.3±6.5	55.7±43.7	51.7±39.8	– " –
	SHOTCOLOR	25.3±21.7	62.3±45.5	53.7±37.9	– " –

Object query retrieval rates (DT/DEF), interest

Object	Interest	Top	Total	Unique	Actual
Backpack	ISS	**16.8**±21.6	**30.0**±31.5	**30.0**±31.5	80
	SIFT	1	1	1	— " —
	SUSAN	10.8±17.5	10.8±17.5	10.8±17.5	— " —
	UNIFORM	14.0±26.0	16.2±29.9	16.2±29.9	— " —
Chair	ISS	15.0±18.0	50.0±27.6	46.2±22.6	85
	SIFT	5	5	5	— " —
	SUSAN	10.8±2.9	10.8±2.9	10.8±2.9	— " —
	UNIFORM	**32.6**±16.0	**52.6**±26.1	**47.6**±20.6	— " —

Object query retrieval rates (DT/K10R2), interest

Object	Interest	Top	Total	Unique	Actual
Backpack	ISS	**15.4**±21.3	**41.0**±24.1	**41.0**±24.1	80
	SUSAN	8.2±13.1	8.2±13.1	8.2±13.1	— " —
	UNIFORM	9.2±18.9	16.0±25.3	16.0±25.3	— " —
Chair	ISS	14.2±16.6	77.2±15.5	69.4±8.3	85
	SUSAN	11.4±5.7	11.8±6.0	11.8±6.0	— " —
	UNIFORM	**32.2**±17.7	**77.0**±16.3	**68.4**±11.6	— " —



Outline

1. Introduction
2. Preprocessing
3. Interest Points
4. Descriptors
5. Object Query
6. Experimental Results
7. Conclusion

Conclusion

Successfully implemented a system for object retrieval

- Best results (50% retrieved objects) were with uniform interest point selection and PFHRGB descriptor.
- Some of the objects were not retrievable or had very low retrieval rates (10%). Perhaps due to size, shape or preprocessing issues?
- Might be possible to perform better with tweaks, but wouldn't work as a standalone system

Questions?