



Master Thesis - Final Presentation

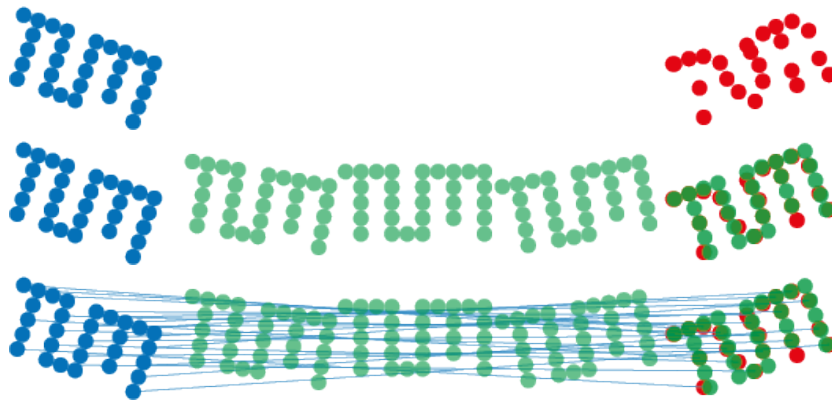
Natural marker extraction and learning for
binocular images and 6D pose estimation

Mahdi Saleh
10.03.2017

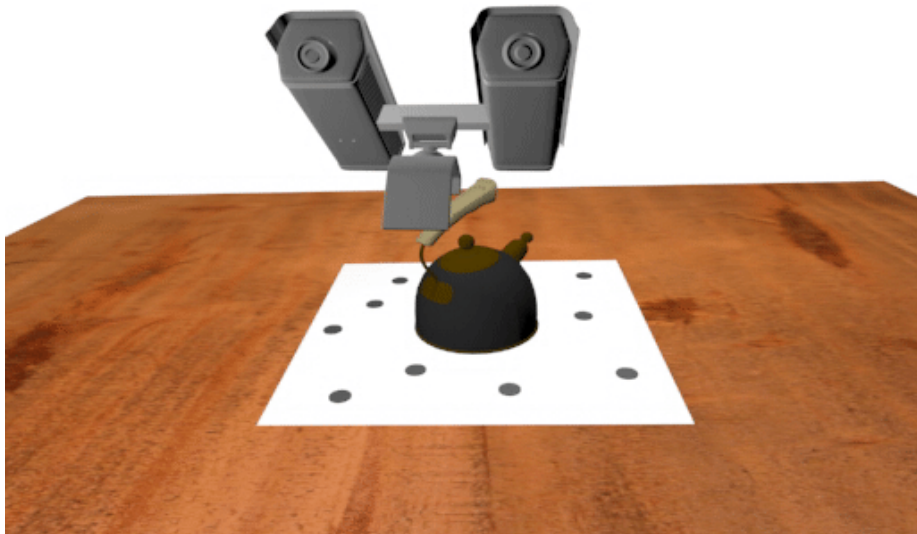
Supervisors
Benjamin Busam
Federico Tombari



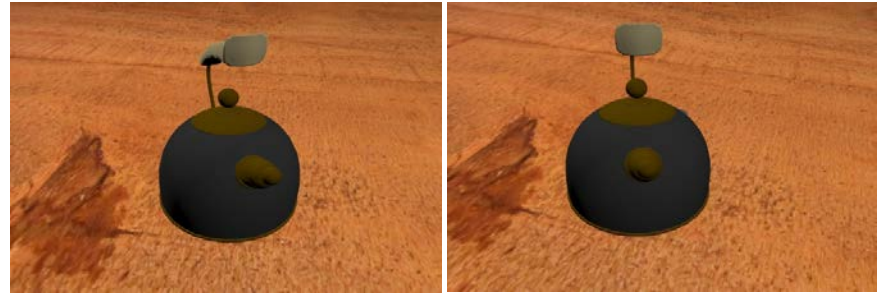
1. Framos OTS detects markers on an object and outputs 3D Pose accurately and fast.
 - Busam et al. **A Stereo Vision Approach for Cooperative Robotic Movement Therapy**, ICCVW 2015
2. Extract and learn natural markers instead
3. Video input with known 6DOF pose



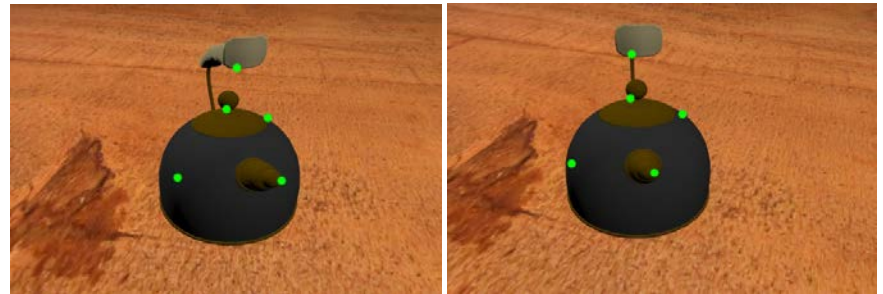
Training video



Input stereo for evaluation



What we want



- Robust and distinctive keypoints
- Symmetric objects
- Highly textured to textureless objects



Image taken from hema.nl, 10.10.2016



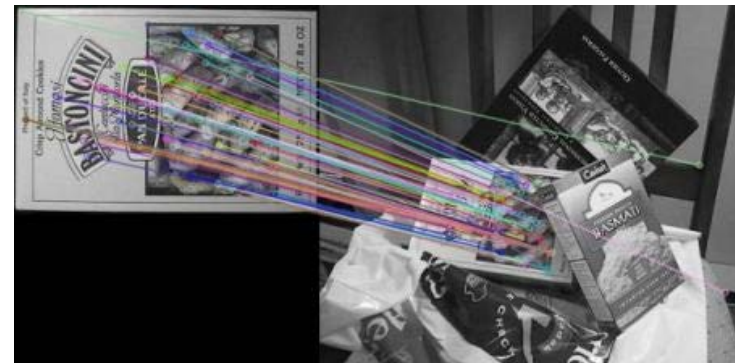
Image taken from living4me.de, 10.10.2016



Image taken from pixabay.com, 10.10.2016

Handcrafted features used for pose estimation

- Harris Corner Detector
 - Harris et al. **A combined corner and edge detector**. Proceedings of the 4th Alvey Vision Conference, 1988.
- SIFT
 - Lowe et al. **Distinctive image features from scale-invariant keypoints**. ICCV 2004.
- HOG
 - Dalal et al. **Histograms of Oriented Gradients for Human Detection**. CVPR 2005.
- SURF
 - Bay et al. **Speeded Up Robust Features**, ECCV 2006
- BRISK
 - Leutenegger et al. **BRISK: Binary Robust Invariant Scalable Keypoints**. ICCV 2011.
- LINEMOD
 - Hinterstoisser et al. **Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes**. ACCV 2012.



Matching using SIFT descriptor,
Image taken from OpenCV Online Documentation, 09.03.2017



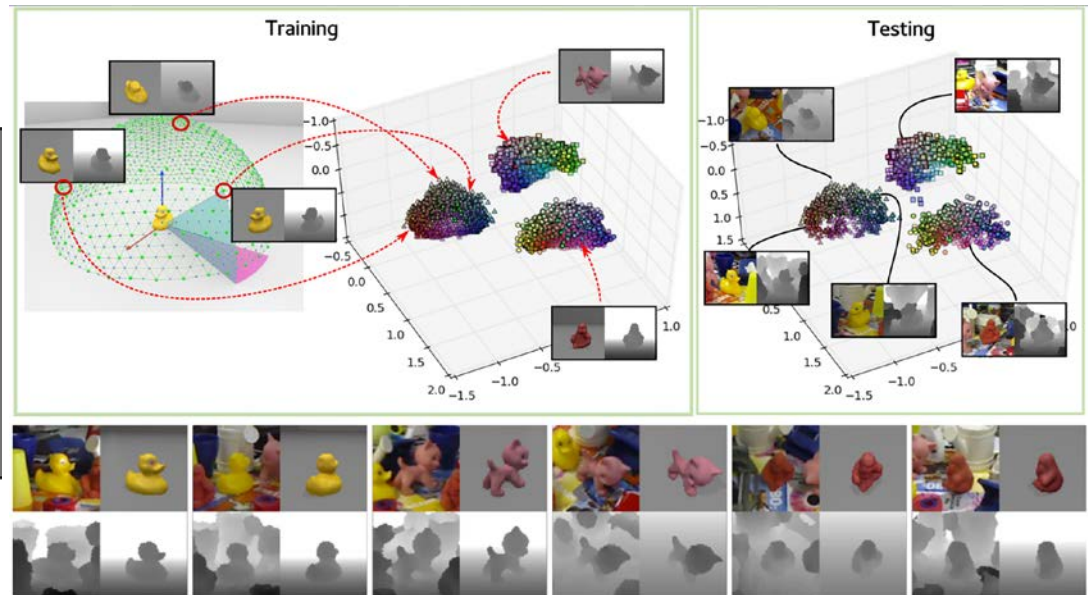
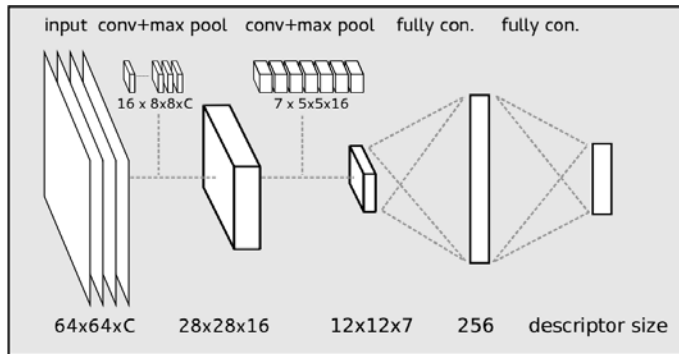
Training keypoints by SVM or boosting

- Learning HOG models for viewport classification
 - Gu et al. **Discriminative Mixture-of-Templates for Viewpoint Classification**. ECCV 2010.
- Without pose estimation problem
 - Malisiewicz et al. **Ensemble of Exemplar-SVMs for Object Detection and Beyond**. ICCV 2011.
- Exemplars were recently used for 3D object detection and pose estimation, but still rely on a handcrafted representation.
 - Aubry et al. **Seeing 3D Chairs: Exemplar Part-Based 2D-3D alignment Using a Large Dataset of CAD Models**. CVPR 2014.



Wohlhart et al. "Learning descriptors for object recognition and 3d pose estimation." *CVPR* 2015.

- Trains RGB-D objects from different poses on a CNN network
 - Finds object class and its 3D pose
 - K-nearest neighbor search in descriptor space



Verdie et al. "TILDE: A Temporally Invariant Learned DEtector." CVPR 2015

- Finding keypoints under severe illumination changes
 - Extracts keypoints on all image
 - If the batch of images has frequent keypoints in a point, the point is a positive location
 - Uses regression to train patches
 - Tries different regressions



(a) Original images

(b) SIFT

(c) SURF

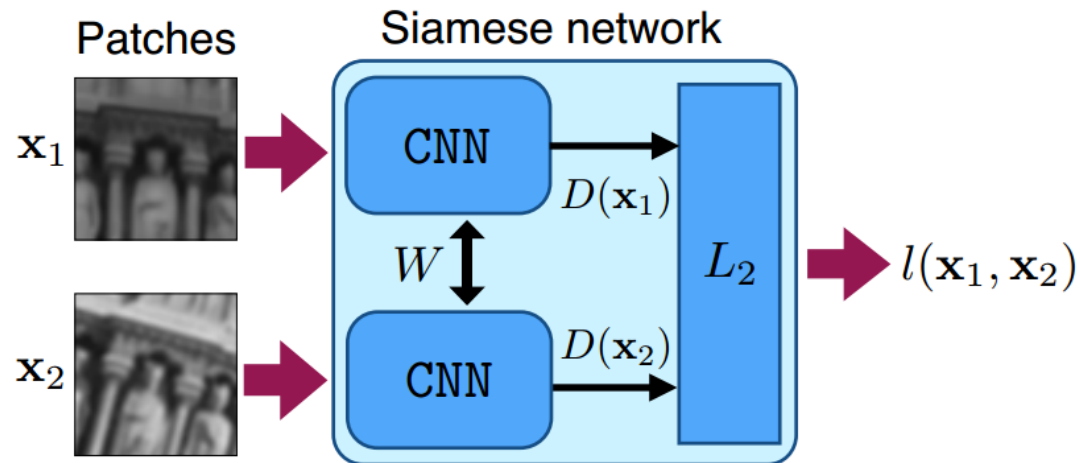
(d) FAST-9

(e) Our keypoints

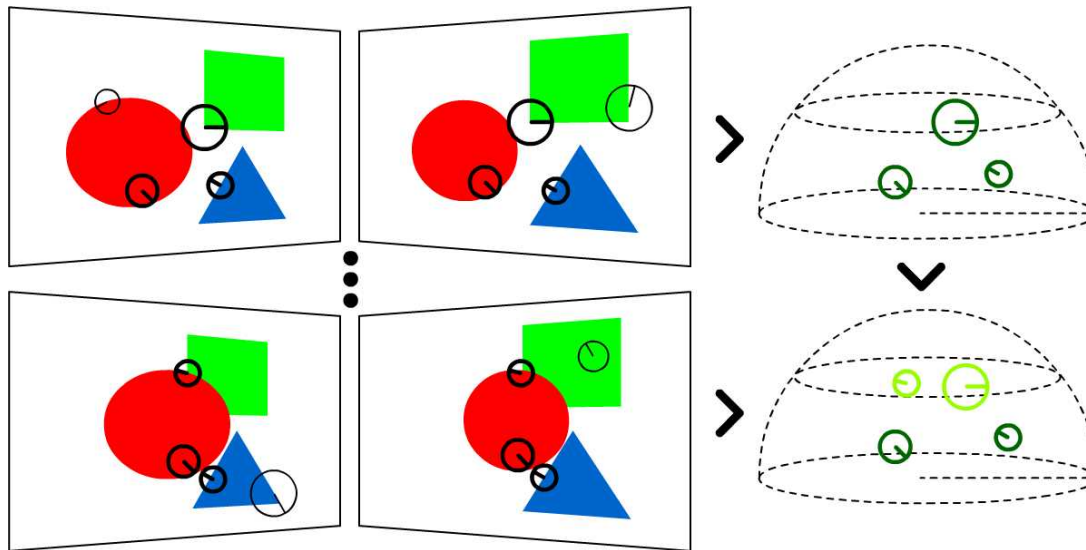
#keypoints	Webcam		Oxford		EF	
	(2%)	Stand.	(2%)	Stand.	(2%)	Stand.
TILDE-GB	33.3	54.5	32.8	43.1	16.2	
TILDE-CNN	36.8	51.8	49.3	43.2	27.6	
TILDE-P24	40.7	58.7	59.1	46.3	33.0	
TILDE-P	48.3	58.1	55.9	45.1	31.6	
FAST-9	26.4	53.8	47.9	39.0	28.0	
SFOP	22.9	51.3	39.3	42.2	21.2	
SIFER	25.7	45.1	40.1	27.4	17.6	
SIFT	20.7	46.5	43.6	32.2	23.0	
SURF	29.9	56.9	57.6	43.6	28.7	
TaSK	14.5	25.7	15.7	22.8	10.0	
WADE	27.5	44.3	51.0	25.6	28.6	
MSER	22.3	51.5	35.9	38.9	23.9	
LCF	30.9	55.0	40.1	41.6	23.1	
EdgeFoci	30.0	54.9	47.5	46.2	31.0	

Simo-Serra et al. "Discriminative learning of deep convolutional feature point descriptors." *ICCV* 2015.

- Trains the network on sets of matching and unmatching patches
 - The CNN outputs are 128-D descriptor
 - Some tied weights between two networks
 - Easy matching using Euclidean distance



1. Extract a pool of handcrafted features
2. Triangulate and build up a point cloud
3. Score and filter keypoints
4. Backprojection of the best keypoints on the images
5. Train a CNN classifier on the patches with/without markers



Synthetic dataset

- 6 models from the website Turbosquid* and self-created
- The objects have different level of textures and geometries
- 1,000 frames rendered from a list of different poses in 1024x768 pixel resolution
- Maxscript used for reading ground truth, transformation and automatic rendering
- Random office and home space as background



Dwarf



Camera



Earth



Pillow

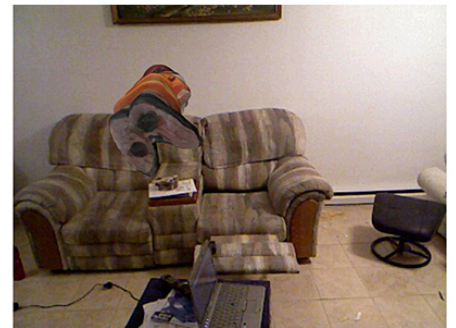


Beets



Box

*www.turbosquid.com, access on 21.11.2016



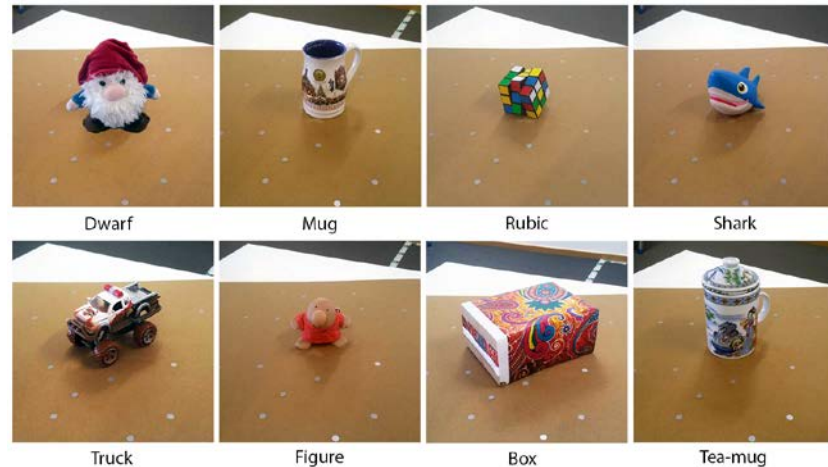
Sample images of Dwarf model

Real dataset

- Reflective markers are attached to a plane
- The plane is moved and rotated freely
- 5 different streams are captured Simultaneously at 5Hz
- Ground truth pose is calculated via marker registration of plane
 - Busam et al. **A Stereo Vision Approach for Cooperative Robotic Movement Therapy**, ICCVW 2015
- Kinect depth images are captured for RGB-D based algorithms in dataset



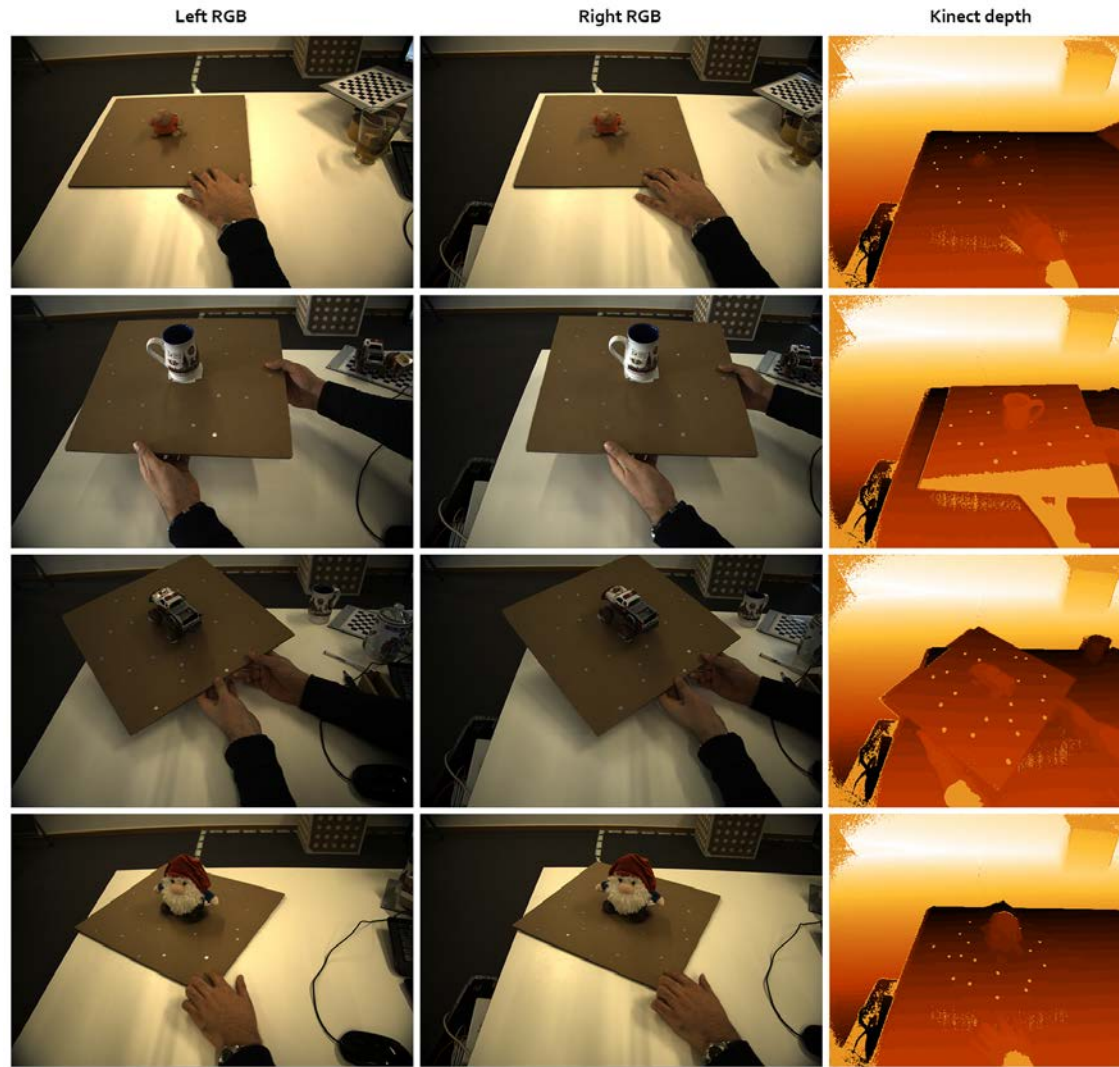
Multi Camera setup



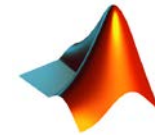
Dataset objects on the trackable plane



Color stereo views for video sequence and monochrome stereo for marker detection and pose estimation



Sample images for 4 objects in a known pose

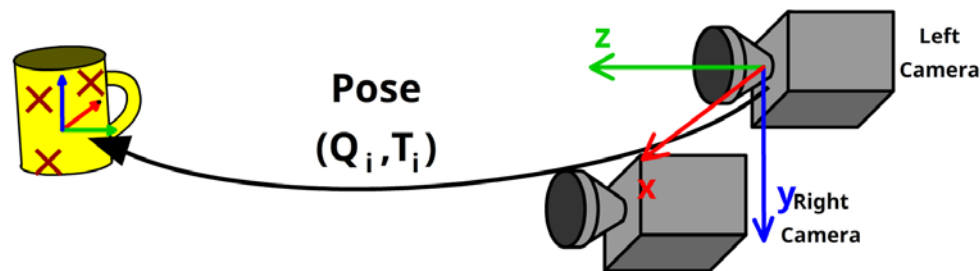


2D Keypoint matching

- A pool of SIFT[1], SURF[2], MSER[3] and Harris[4] is created
- Filtering using epipolar lines
- Matching and triangulation
- 3D points are transformed to reference coordinates using pose



Keypoint matching samples for stereo image pairs



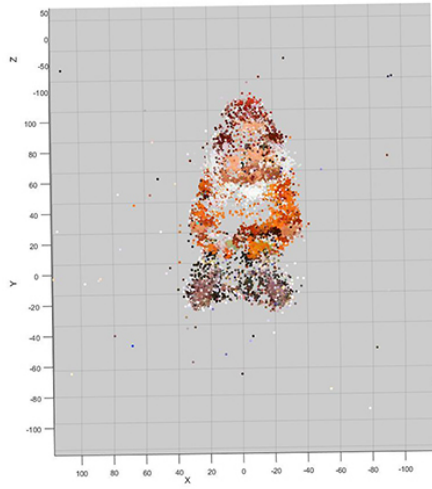
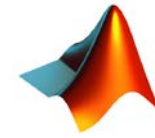
[1] Lowe et al. **Distinctive image features from scale-invariant keypoints**. ICCV 2004.

[2] Bay et al. **Speeded Up Robust Features**, ECCV 2006

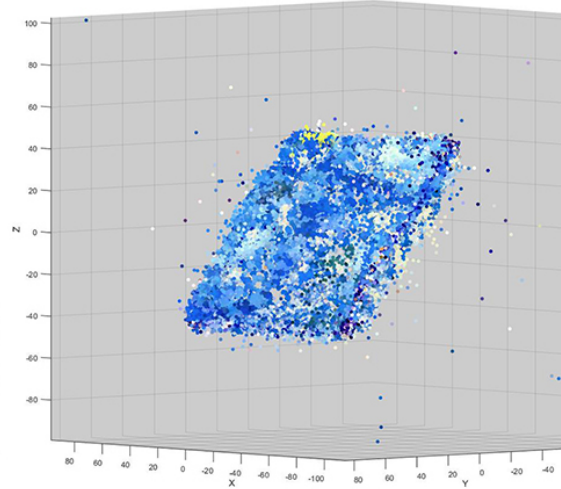
[3] Matas et al. **Robust wide baseline stereo from maximally stable extremal regions**, BMVC 2002

[4] Harris et al. **A combined corner and edge detector**. Proceedings of the 4th Alvey Vision Conference, 1988.

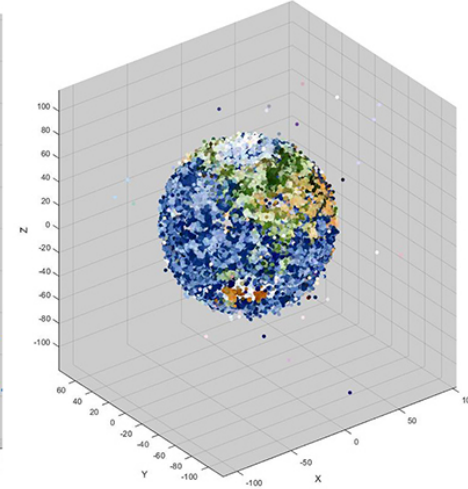
Keypoint Extraction



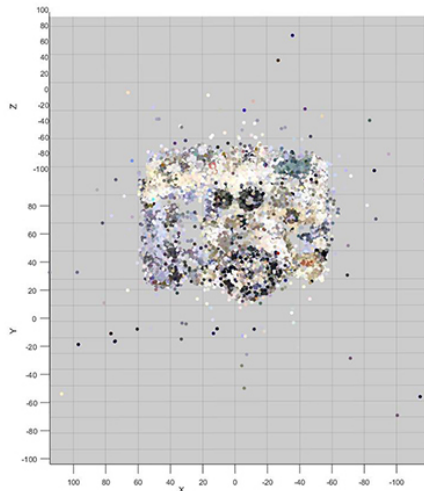
Dwarf



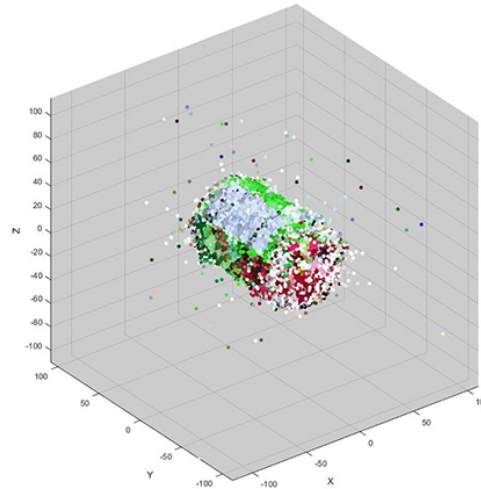
Pillow



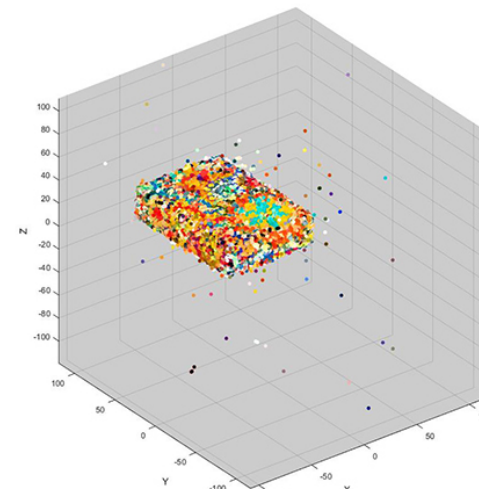
Earth



Camera



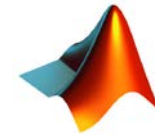
Beets



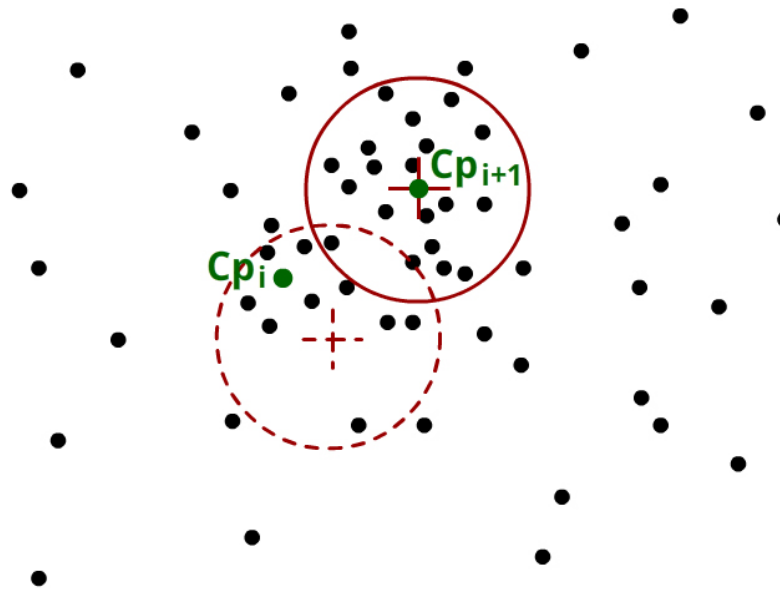
Box

Projection of extracted keypoints for the 6 dataset objects



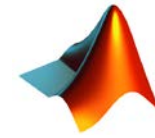


- We cluster keypoints in 3D using mean-shift algorithm
 - Cheng et al. **Mean Shift, Mode Seeking, and Clustering**. PAMI 1995
- Mean-shift is parameterless
- Clustering is done in 6D-space (color+position)

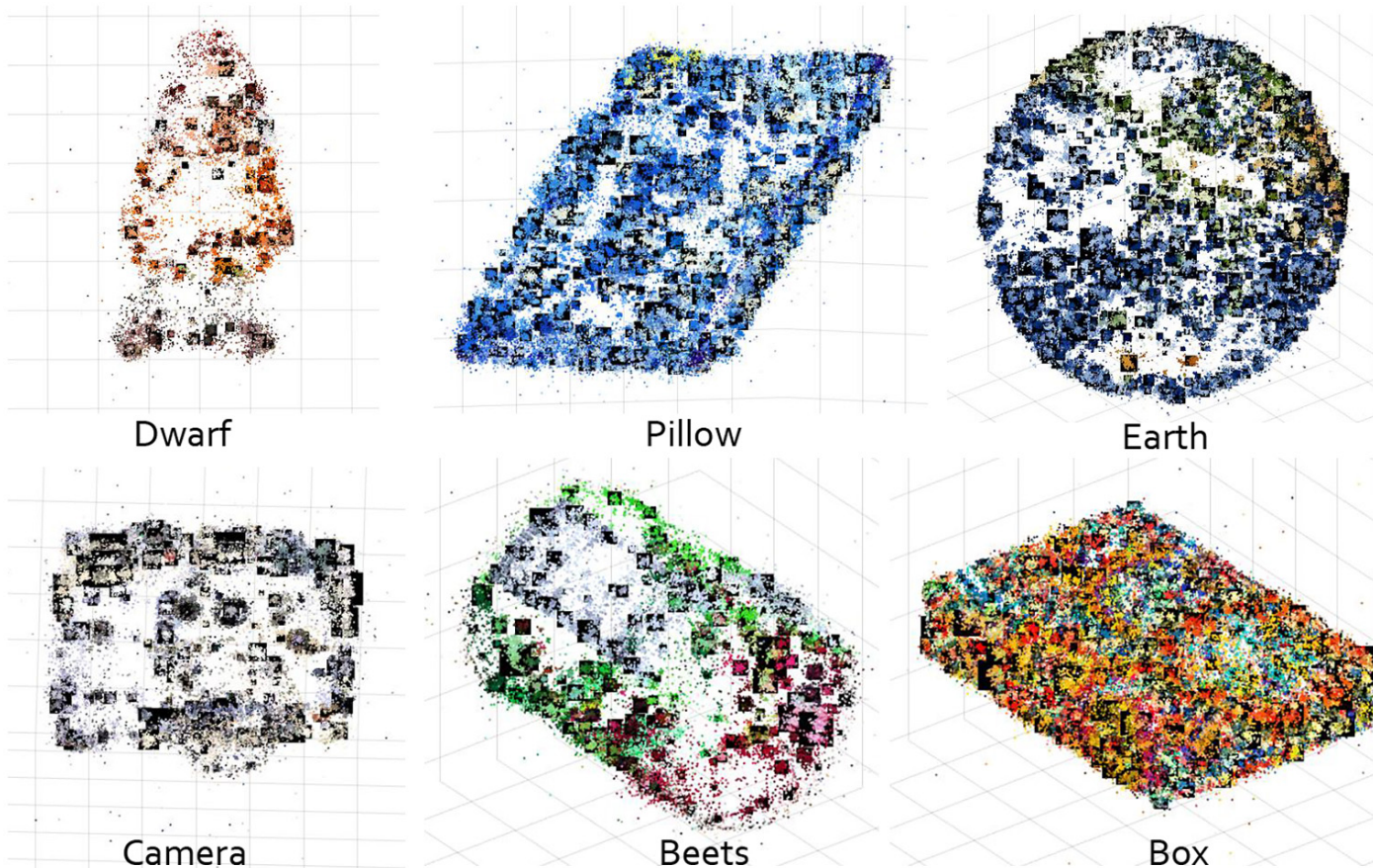


Mean-shift update schema



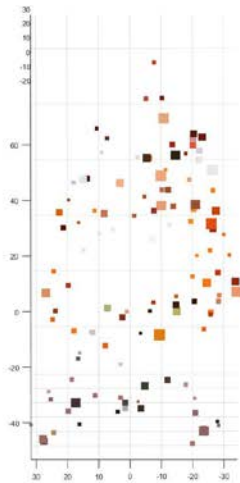
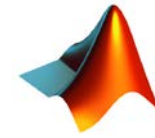


- Clusters with few points are eliminated
- Mean-shift centroids are keypoints
- Mean color and mean FREAK feature is computed
 - Alahi et al. **Freak: Fast retina keypoint**. *CVPR 2012*

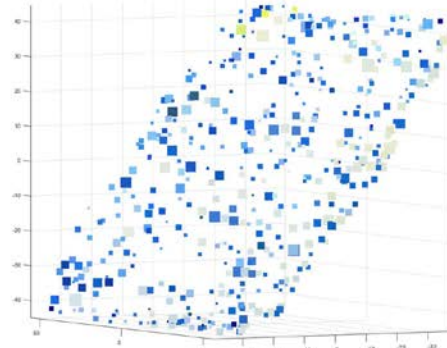


Mean-shift clusters and their including points

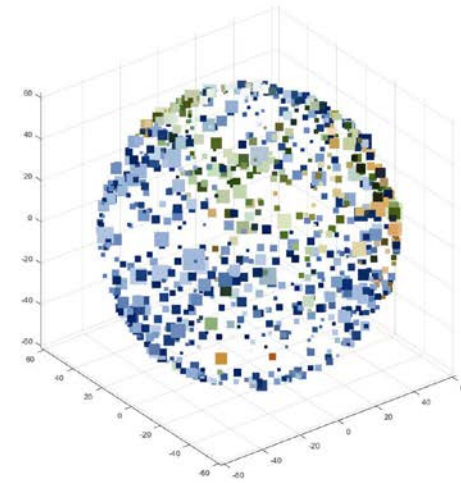




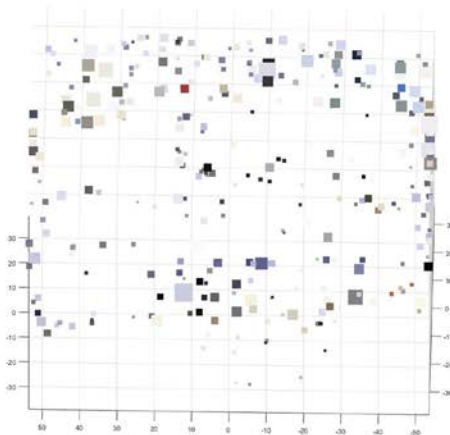
Dwarf



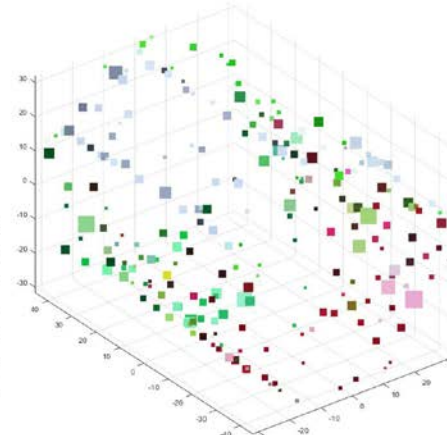
Pillow



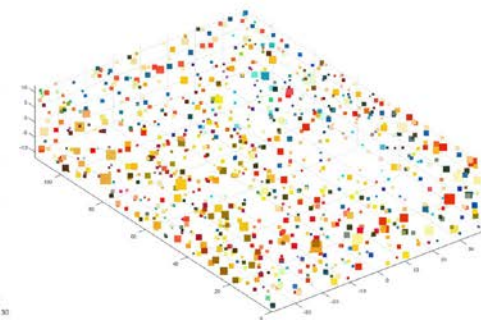
Earth



Camera



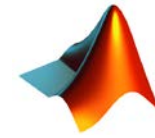
Beets



Box

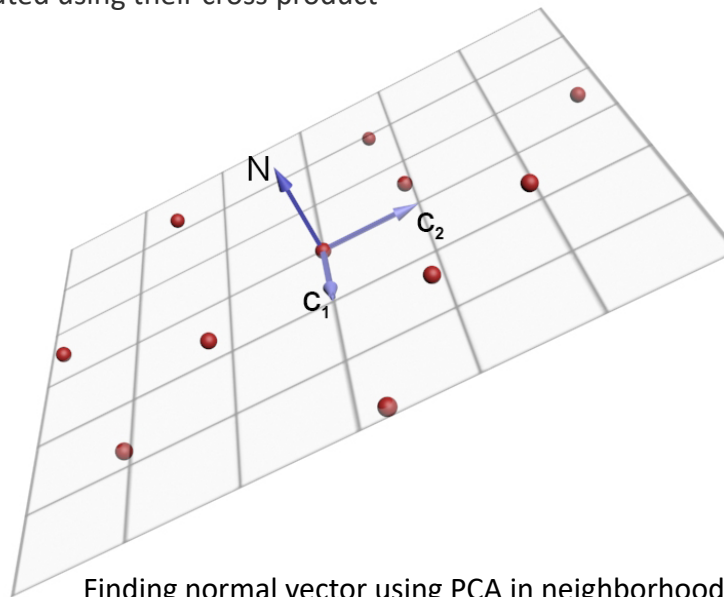
Mean-shift cluster centroids as our marker candidates point cloud for 6 dataset objects





Keypoint reprojection

- In order to find the number of keypoint occurrences on images
- Rendering sparse point cloud is challenging
- Big uncertainty whether a keypoint is not visible
- Different checks based on pose, feature, pose,... failed
- Estimate normal vectors using PCA[1]
 - A neighborhood of point is found using K-Nearest-Neighbors (KNN) [2]
 - A surface is fitted to the points using first two principle components
 - Normal vector is computed using their cross product

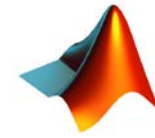


Finding normal vector using PCA in neighborhood



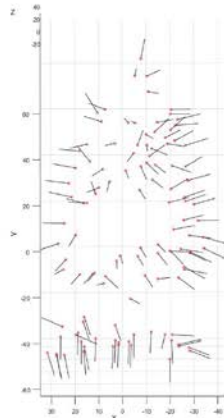
[1] Pearson, K. **On Lines and Planes of Closest Fit to Systems of Points in Space**. Philosophical Magazine, 1901

[2] Altman, N. S. **An introduction to kernel and nearest-neighbor nonparametric regression**. The American Statistician, 1992

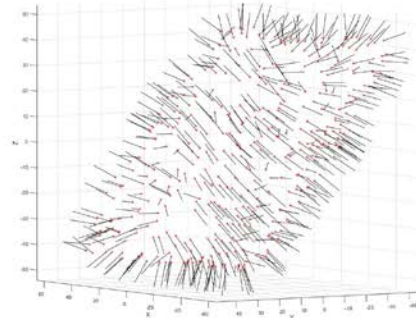


Keypoint reprojection

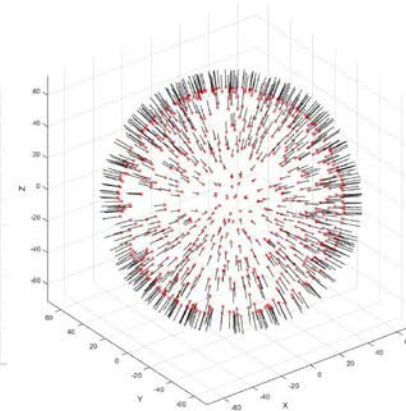
- Normal vector is computed for the point cloud
- Keypoints are projected on stereo images



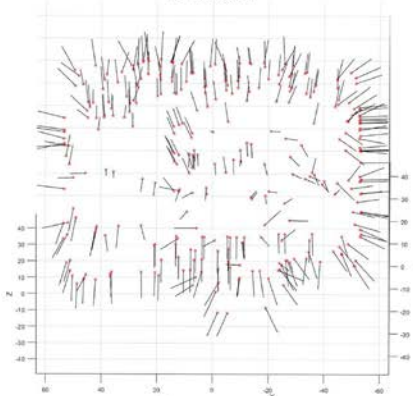
Dwarf



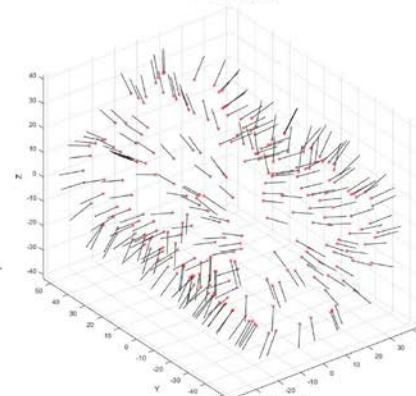
Pillow



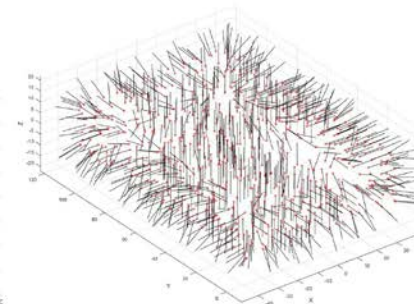
Earth



Camera



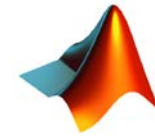
Beets



Box

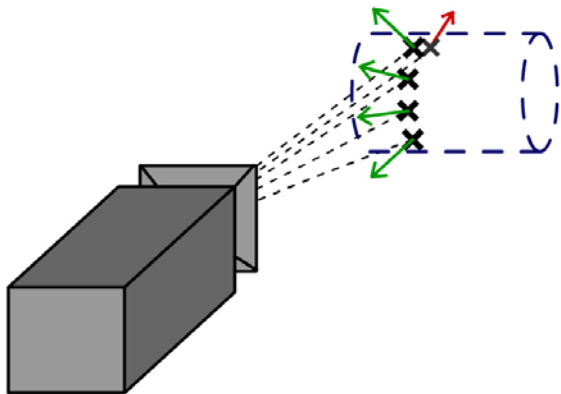
Marker normal vectors for 6 objects of dataset





Keypoint reprojection

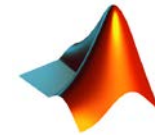
- Keypoints are projected on stereo images
- Visibility of keypoint is assessed
 - Normal vector angle with camera ray
 - Left right color check



Raytracing for rendering using normal vectors

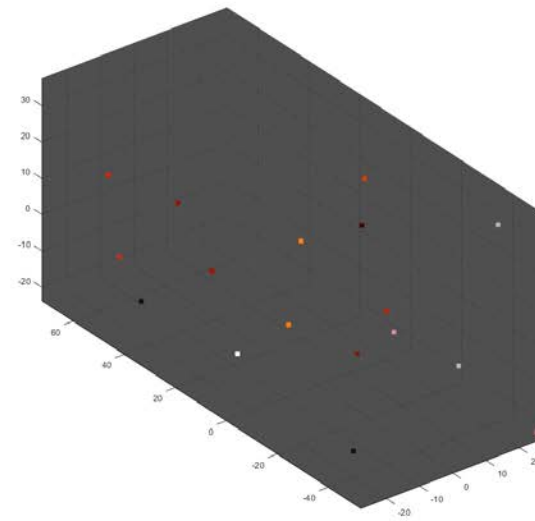
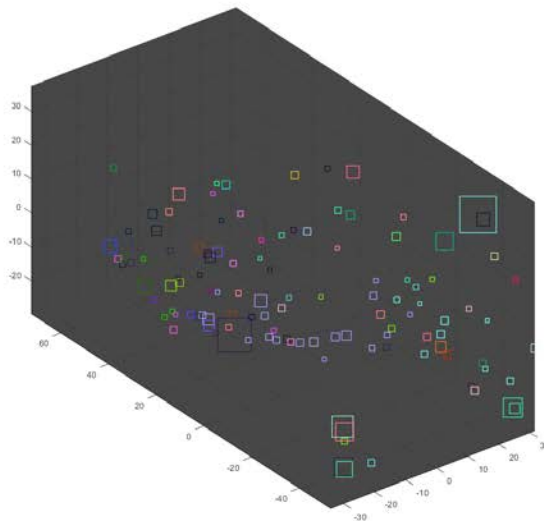


Marker reprojections for some frame samples of Dwarf object



Scoring Criteria

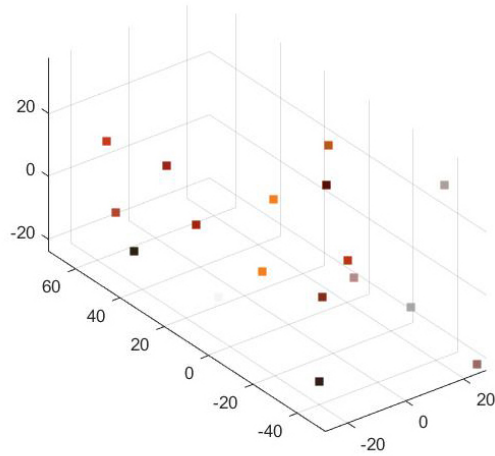
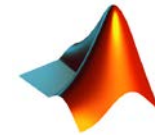
- **Distinctiveness:** A metric to privilege non-similar salient keypoints
 - Sum of L2 Distances to other FREAK[1] descriptors
- Each frame has a single vote:
 - The summation of discriminations for every frame is the same
- **Repetitiveness:** Privilege the keypoints that are seen in a wide range of view
 - Reprojection occurrences are counted
- **Widely distributed:** a minimum number of keypoints should be seen in a view
- Take locally best scored keypoint



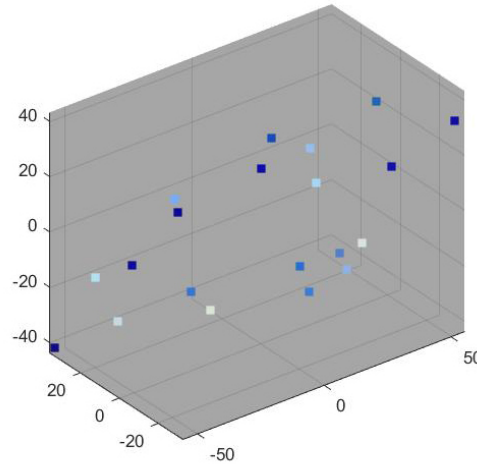
Marker candidates point cloud, their scoring and the selected best markers for the Dwarf object

[1] Alahi et al. **Freak: Fast retina keypoint**. *CVPR 2012*

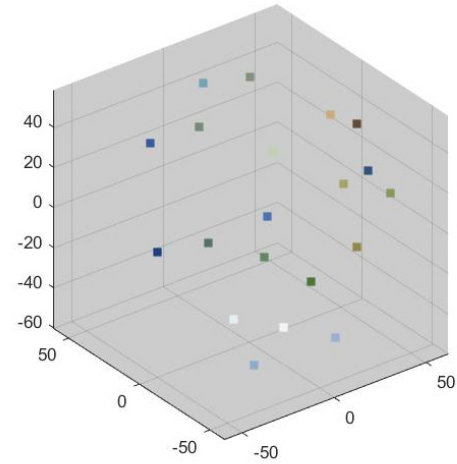
Keypoint Scoring



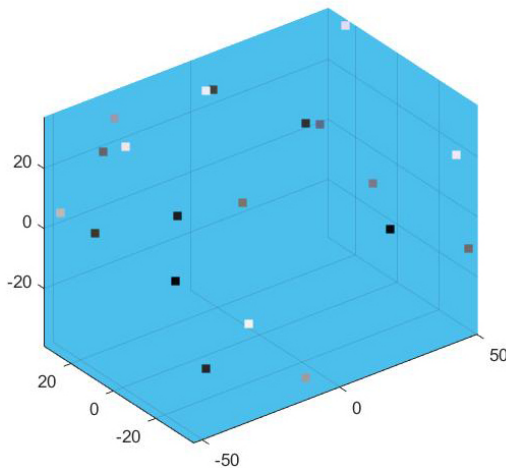
Dwarf



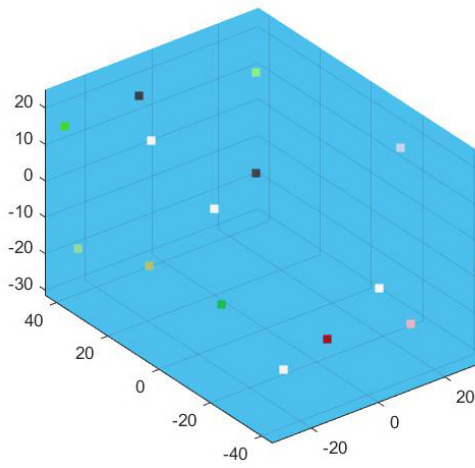
Pillow



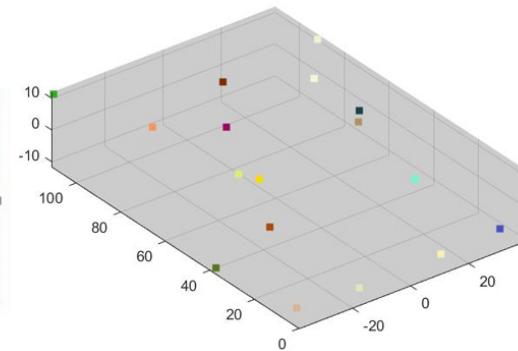
Earth



Camera



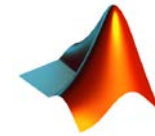
Beets



Box

Selected marker for 6 objects of dataset





Dataset Creation

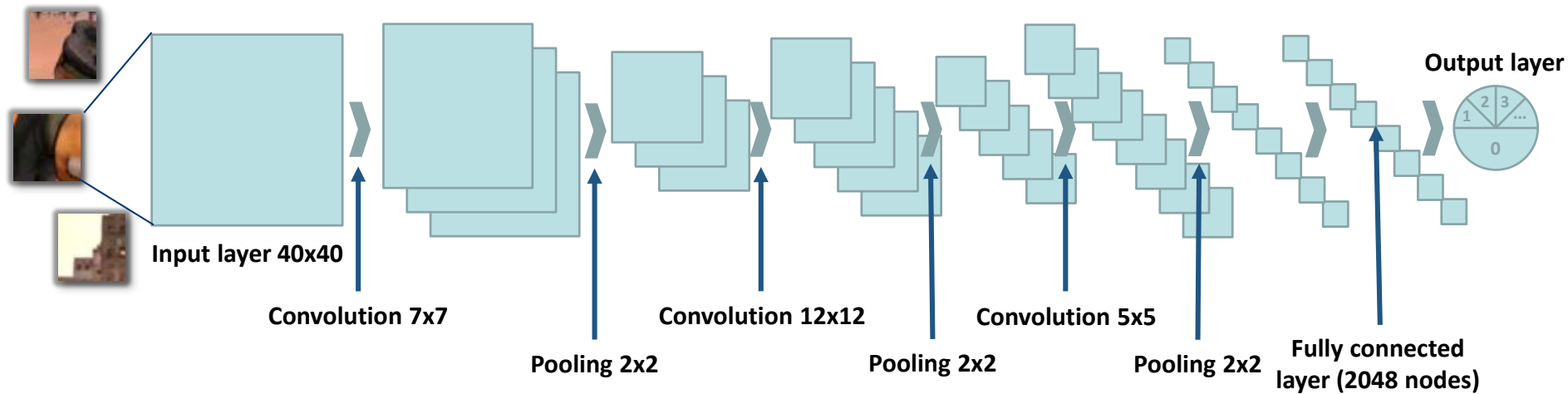
- Selected markers are backprojected
- 40x40 patch is created around them
- Data augmentations
 - Shifting, rotation and scaling
 - Brightness, contrast, saturation
- Marker labels are stored $\{1, \dots, n\}$
- Markerless patches are in class 0
 - 70% outside object region
 - 30% inside object region
- Nearly 200k images for 15 markers
- 80% training set and 20% test set
- Evaluation set on some other sequence with known pose
 - 8 pixel interval
 - Pose estimation



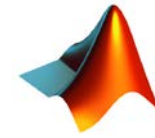
Patch samples for some classes of dwarf object

CNN learning

- Microsoft CNTK used for training
 - Seide et al. **CNTK: Microsoft's Open-Source Deep-Learning Toolkit**, SIGKDD 2016
- A mini-batch SGD with decreasing learning rate is used
 - Li et al. **Efficient Mini-batch Training for Stochastic Optimization**, ACM, 2014
- Architecture inspired from LeNet-5* with some changes inspired from TILDE
 - LeCun et al, **Gradient-based learning applied to document recognition**, Proceedings of the IEEE, 1998
 - Verdie et al. "**TILDE: A Temporally Invariant Learned DEtector.**" *CVPR 2015*

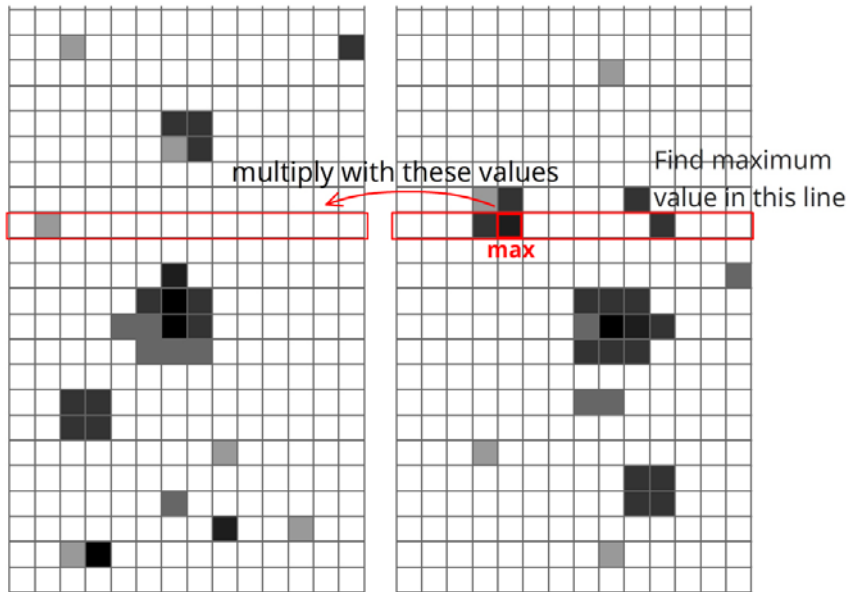


Deep network architecture

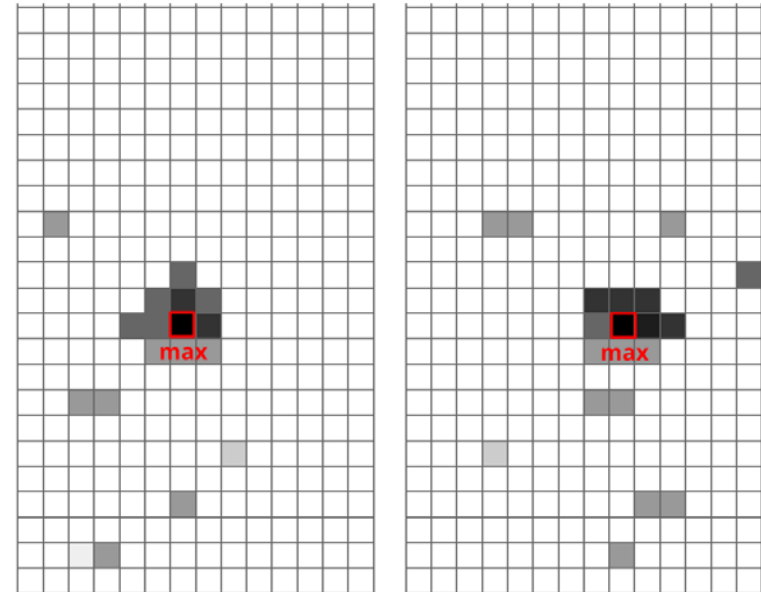


Point Cloud build-up

- Likelihood maps are created for each keypoint
- Map resized to full size image
- Using epipolar lines the maps are swept
- Likelihoods without match diminished
- Maximum of the two maps is triangulated
- Applied to all marker likelihood maps

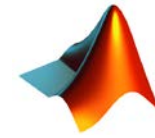


Stereo filtering algorithm using epipolar lines



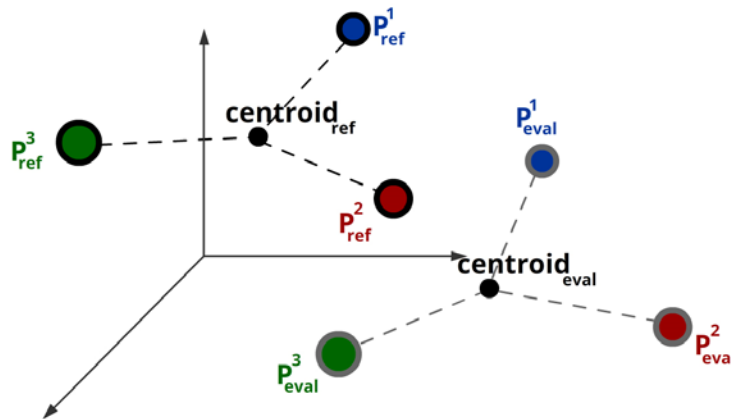
Stereo filtering output



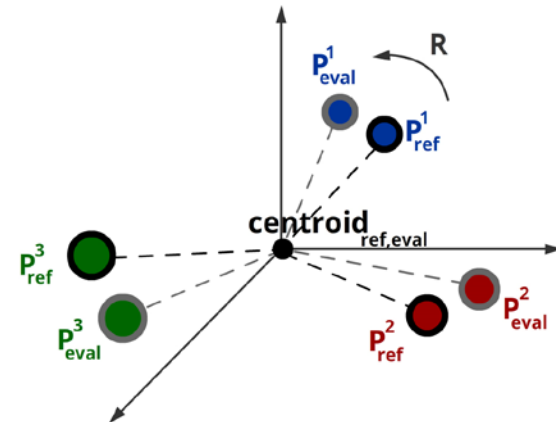


Pose estimation

- Registering reference point cloud to evaluated point cloud
- The $[R, t]$ is the estimated pose
- Different combinations of evaluated point cloud is tried
 - $[R, t]$ is calculated using accumulated covariance matrix H
 - Rotation is calculated using SVD
 - t is estimated after R
- The pose of combination with least error is selected



Reference and evaluated point cloud and their centroid









Finding R by bringing point clouds to a reference coordinates









Marker point cloud evaluation

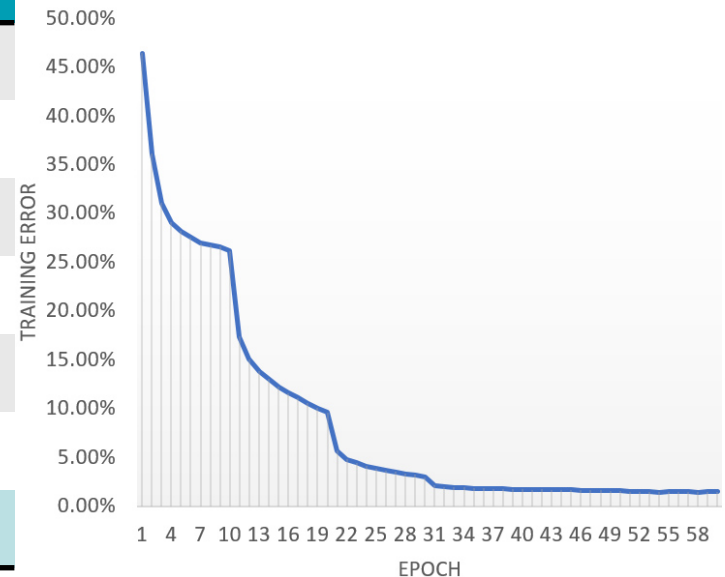
- No benchmark to evaluate stereo keypoint extraction
- Average speed to build a point cloud from video sequence <1 hour on Intel CORE-i7 6700 CPU
- Quantitative evaluation is done
- Marker calculation is done with different sequences
- The resulting point clouds are compared
- The RMS error between point clouds is calculated

Object	Number of markers	Mean # of visible keypoints per frame	RMS error [mm]	Outliers
 Dwarf	17	6.86	2.4354	2
 Pillow	20	7.34	2.3171	4
 Earth	14	7.84	3.2647	1
 Camera	20	8.86	2.8927	3
 Beets	15	6.82	3.1193	1
 Box	17	8.78	2.7622	2
Average	17.16	7.75	2.7985	2.16

Marker training evaluation

- Training is done for 60 epochs
- Test is done using a different set
- Average speed to train markers around 2 hours on NVIDIA GeForce GTX 970 GPU







Object	Training error	Test error	Cross entropy
 Dwarf	1.48%	12.76%	0.0845
 Pillow	0.79%	11.11%	0.0729
 Earth	0.88%	11.88%	0.0769
 Camera	1.41%	15.13%	0.0981
 Beets	5.81%	14.78%	0.2109
 Box	0.64%	14.03%	0.0657
Average	1.83%	13.28%	0.1015



Training error for Dwarf object

Pose estimation errors

- Evaluated on a set of frames with known pose (30 frames)
- Average speed to estimate pose from network output is <30 sec on Intel CORE-i7 6700 CPU

Object	Angle error [degree]	Distance error [mm]	Rejection rate
 Dwarf	5.4862	7.0438	33.3%
 Pillow	3.8803	5.0497	10.0%
 Earth	4.3718	3.9942	50.0%
 Camera	6.2057	4.9467	23.3%
 Beets	8.2237	5.9485	20.0%
 Box	4.2041	5.9312	46.6%
Average	5.3953	5.4854	30.5%



- A lot of personal learning and challenges
- Pose estimation results promising

Future works

- Marker dataset is prone to error
- Pitfalls like repeating texture and monotonous textures
- Long and slow pipeline
- Pose estimation procedure can be improved
- Adapting real dataset for the pipeline



Thanks for your attention



Questions?