Active and Interactive Vision

T-K Kim
Computer Vision and Learning Lab
EEE, ICL
http://www.iis.ee.ic.ac.uk/ComputerVision/

• We have tackled pose estimation of
  – Hands, Face, Body as structured label estimation problems
  – 6D Object Pose

• Active and interactive Vision
  – Interaction among Human-Computer-Object
Object Pose and Next-Best-View Estimation

- **Problem** - estimating objects’ 3D location and pose
- **Application** - e.g. picking and placing for logistics
- **Challenge** - highly crowded scenes, active camera planning

- **Problem** - estimating clothes types, grasp points and pose
- **Application** - autonomously unfolding clothes
- **Challenge** - highly deformed objects, multi-view solution, active planning
Object Pose and Next-Best-View Estimation

- Estimating objects’ 3D location and pose

Latent Hough Forest (ECCV14):
- novel template-matching based splitting, one-class learning

6D Object Detection and Next-Best-View Prediction in the Crowd (ongoing):
- deep-features, a novel active solution on Hough Forests, joint registration

- Estimating clothes types, grasp points and pose

Autonomous unfolding clothes (ICRA14, best paper award):
- regression forests, probabilistic active planning

Active Forest (ECCV14):
- multi-task learning, next-best view learning in RF
Latent-Class Hough Forests for 3D Object Detection and Pose Estimation

ECCV 2014
Latent-Class Hough Forests for 3D Object Detection and Pose Estimation

ECCV 2014

Alykhan Tejani, Danhang Tang, Rigas Kouskouridas and Tae-Kyun Kim

Imperial College London

https://www.youtube.com/watch?v=idY3Q7wg5rk
Challenges and Proposed Ideas

• Challenges
  – Foreground occlusions, Multi-instances, Large scale changes

• Main ideas
  – Integration of LINEMOD \((S. \text{ Hinterstoisser, et al. PAMI12})\) Template Matching into Hough Forests \((J. \text{ Gall, et al. PAMI11})\) : Efficient data split at node levels
  – Making LINEMOD scale-invariant
  – Inference of occlusion masks: Iteratively updating class distributions (latent variable, one-class learning)
Template-matching Split Functions

- A random patch \( T \) (with red frame) is chosen. All other patches are compared with \( T \).

\[
S(\mathcal{X}, T) = \sum_{r \in \mathcal{P}} g(\text{ori}(\mathcal{X}, r), \text{ori}(\mathcal{O}, r))
\]

- They go to e.g. a right child node if the similarity is greater than a threshold, otherwise to a left child node.
- This achieves more discriminative (nonlinear) yet fast splits.
Split function model in Decision Forests

Examples of split functions

Split fn: axis aligned
Split fn: oriented line
Split fn: conic section

Slide credit to Criminisi and Shotton, ICCV11 tutorial
Template-matching Split Function

Examples from Pedestrian Detection

2 pixel test (axis aligned splits): efficient but less discriminative
Example of pedestrian detection using template matching:

Template matching (nonlinear splits):
- discriminative but cost-demanding?
Template-matching Split Function using Binary Bit Operations

\[
F(S, T) = \sum_{\substack{P^S_d \in S \\ P^T_d \in T}} \delta(P^S_d \otimes P^T_d \neq 0), d = 1, \ldots, n
\]

\[
h_i(S) = \begin{cases} 
0, & F(S, T_i) \leq \tau_i \\
1, & F(S, T_i) > \tau_i 
\end{cases}
\]

Template matching split is highly accelerated by binary bit operations.
Split Function Properties

- The split function with an efficient z-value check:
  \[
  S(\mathcal{X}, T) = \sum_{r \in P} f(\mathcal{X}, \mathcal{O}, c, r)g(\text{ori}(\mathcal{X}, r), \text{ori}(\mathcal{O}, r)),
  \\
  f(\mathcal{X}, \mathcal{O}, c, r) = \delta((D(\mathcal{X}, c) - D(\mathcal{X}, r)) - (D(\mathcal{O}, c) - D(\mathcal{O}, r))) < \tau
  \]

  Blue patch: true positive match
  Red patch: false positive match

- Scale invariance:
  \[
  S(\mathcal{X}, T) = \sum_{r \in P} f(\mathcal{X}, \mathcal{O}, c, r)g(\text{ori}(\mathcal{X}, \frac{r}{D(\mathcal{X}, c)}), \text{ori}(\mathcal{O}, \frac{r}{D(\mathcal{O}, c)})),
  \\
  f(\mathcal{X}, \mathcal{O}, c, r) = \delta((D(\mathcal{X}, c) - D(\mathcal{X}, \frac{r}{D(\mathcal{X}, c)})) - (D(\mathcal{O}, c) - D(\mathcal{O}, \frac{r}{D(\mathcal{O}, c)}))) < \tau
  \]
Inference with Iterative Refinement
Inference with Iterative Refinement
F1-Scores for the 13 objects in the dataset of Hinterstoisser et al. (1,100 RGBD images)
Results

Average Precision-Recall curve over all objects in the dataset of Hinterstoisser et al.
Computer Vision & Learning Lab
Imperial College London

Multi-instance Object Detection and Pose Estimation in 1 fps

Latent-Class Hough Forests for 3D Object Detection and Pose Estimation
A. Tejani, D. Tang, R. Kouskouridos, T-K. Kim, ECCV 2014
Optimised by Andreas Doumanoglou

Demonstrated at Imperial College Science Festival in May 2015

https://www.youtube.com/watch?v=dh2VtnnsGuY
Directions

- Object pose in the **crowd** (or bin-picking)
  - Better Feature Learning (deep convolutional networks)
  - Active vision (moving cameras, manipulators interacting objects)
  - Joint multiple object pose estimation (global optimization)

A complete pipeline including, sparse autoencoders, 6D hough voting, a novel next-best-view estimation based on Hough Forests (ongoing work)
Autonomous Active Recognition and Unfolding of Clothes using Decision Forests

A. Doumanoglou, A. Kargakos, T-K. Kim, S. Malassiotis
ICRA 2014 (best service robotics paper award)

A. Doumanoglou, T-K. Kim, X. Zhao, S. Malassiotis
ECCV 2014
Clothes Recognition

- How to reduce the large configuration space?

- Grasp Lowest Hanging Point First:

- Training Database:
  - 40 depth images
  - Rotate 360°
  - 28,800 Training Images
  - 40 depth images

- RF training by pixel-tests in depth/curvature channels, and class entropy

6 non-symmetric lowest points → 6 Classes Total
Grasp Point Detection

- Desired grasp Points:

- Hough Forest

\[ f(V, C_i) \]

Minimize:

\[ H = - (|L| \log \frac{p_L}{p_M} + |N| \log \frac{p_N}{p_R}) \]

or

\[ D = \sum_{\text{samples}} d(p_s, p_M) \]

Variance in each child

Optimization

Point Prediction
Active Planning

Single view success ~ 90%

Crucial Decisions

How can other views help?

Approach

Keep looking sequential views

Until we reach a certain degree of confidence
Active Planning

POMDP (Partially Observable Markov Decision Processes) solution

Active Recognition

Actions (A): Rotate Cloth / Take Final Decision
States (S): Clothes Classes
Observation $P(O | S, A)$
Probabilities: Measured Experimentally

Active Point Estimation

Actions (A): Rotate Cloth / Grasp Garment at (i, j)
States (S): 65 — 8x8 grid quantization, or (INV)
Observation $P(O | S, A)$
Probabilities: Measured Experimentally

POMDP solution policy: $A(\text{current belief state}) \rightarrow \text{Optimal Action}$
Unfolding Process

1. Initial Random Configuration
2. Cloth Recognition POMDP
3. Choose Hough Forest
4. 1st Grasping Point POMDP
5. 2nd Grasping Point POMDP
6. Template Matching
Results

- 28,800 training images and 1,440 testing images, captured with Xtion
Comparison with State-of-the-Art

*Bringing clothing into desired configurations with limited perception*, ICRA 2011 — M. Cusumano-Towner *et. al*

- Grasp lowest point twice
- Unfolding using table (slow)
- Baby clothes

- Grasp lowest point once
- Unfolding in the air (fast)
- Regular-sized clothes
https://www.youtube.com/watch?v=YpD-ip6g5lY
Active Random Forests

Improve POMDP solution

Create a *Generic Active Vision Framework*

Extend objectives – Estimate Garment Pose

\( g, \text{ grasp point} \)
\( v, \text{ viewpoint} \)
\( p, \text{ pose} \)
Active Random Forests

One Forest for all objectives (Classification, Regression, Pose Estimation)

\[(I(v), c, g(v), p(v))\]

Hierarchical coarse to fine quality function \(Q\)

Classification (\(Q_c\))

\[Q = \alpha Q_c + (1 - \alpha) Q_r\]

Regression (\(Q_r\))

(Desired Grasp point, Pose)

Pixel tests
ARF Training

Training

Training Set Validation Set

Stage 1

\( V_{\text{seen}} = V_{\text{current}} \)

Stage 2

\( V_{\text{seen}} = V_{\text{seen}} \cup V_{\text{selected}} \)

‘Action-Selection’ Node Insertion Criteria

a) Hellinger Distance

\[
HL(S_T^1 || S_D^1) = \frac{1}{\sqrt{2}} \sqrt{\sum_{c=1}^{C} \left( \sqrt{P_{S_T^1}(c)} - \sqrt{P_{S_D^1}(c)} \right)^2} > t_\Delta
\]

b) Jeffrey Divergence

\[
JS(S_T^1 || S_D^1) = \frac{1}{C} \sum_{c=1}^{C} P_{S_T^1}(c) \log \frac{P_{S_T^1}(c)}{P_{S_D^1}(c)} + P_{S_D^1}(c) \log \frac{P_{S_D^1}(c)}{P_{S_T^1}(c)} > t_\Delta
\]
### ARF Training

#### Training

- **Split Node**
- **Action-Selection node**
- **Leaf node**
- **Random Split**

#### Grasp Point Visibility
- Calculated from training
- Random sampling for next best view in action-selection nodes

#### Cost of actions
ARF Testing
ARF Results

Self Comparisons

Comparison with state of the art

Qualitative Results
Directions

• Various benchmarks/methods have been collected.
• A comparative study (using the challenge results) will be done.
  – Feature comparison, active vision, multi-object registration, multi-view registration, real-time performance, texture-less, articulated objects, highly occluded scenarios, etc.
• Deep learning + RF
  – learning representation, conditional computing, efficiency
• Active RF classifiers
  – action as a learning parameter

Chao et al. ICCV15