Colorado School of Mines

Computer Vision

Professor William Hoff
Dept of Electrical Engineering & Computer Science
http://inside.mines.edu/~whoff/
Training Feature Recognizers

Some materials from:
• cv.snu.ac.kr/hyunxx/Seminar/2009/2009-08-21/CompactSignatures.ppt
• Comparative Evaluation of Random Forest and Fern classifiers for Real-Time Feature Matching, by I. Barandiaran
Training Feature Recognizers

- A recent development in feature tracking is to use learning algorithms to build special-purpose recognizers for features.
- You need to train classifiers on sample patches and their affine deformations.
- Then, extremely fast and reliable feature detectors can be constructed.

Real-time head tracking using the fast trained classifiers of Lepetit, Pilet, and Fua (2004)

- Keypoint recognition using randomized trees, V. Lepetit, P. Fua (PAMI 2006)
- Fast Keypoint Recognition using Random Ferns, M. Ozuysal, M. Calonder, V. Lepetit, P. Fua (PAMI 2009)
Randomized tree
Randomized tree

A simple classifier $f$

- Randomly select pixel position $a$ and $b$ in the image patch
- Simple binary test
  - The tests compare the intensities of two pixels around the keypoint:

\[
    f_i = \begin{cases} 
        1, & \text{if } I(a_{f_i}) > I(b_{f_i}) \\ 
        0, & \text{otherwise} 
    \end{cases}
\]

$I(*)$: Intensity of pixel position $*$

Invariant to light change by any raising function
Randomized tree

- Initialization of RT with classifier $f$
  - Define depth of tree $d$ and establish binary tree...
  - Each leaf has a $N$ dimensional vector, which denotes the probability for each class.

Number of leaves = $2^d$, where $d =$ total depth
Randomized tree

- **Training RT**
  - Generate patches to cover image variations (scale, rotation, affine transform, ...)

![Images of a person in different poses and perspectives]
Randomized tree

- Training RT
  - Implement training for all generating patches...
  - Update probabilities of leaves...

Depth 1

Depth 2

Leaf
Randomized tree

- Training RT
  - Implement training for all generating patches...
  - Update probabilities of leaves...

- Depth 1
- Depth 2
- Leaf
Randomized tree

- Training RT
  - Implement training for all generating patches...
  - Update probabilities of leaves...

depth 1

depth 2

leaf
Randomized tree

Classification with trained RT

- Implement classification for input patches...
- Confirm probability when a patch reaches a leaf...

depth 1

depth 2

leaf
Randomized tree

- **Random forest**
  - Multiple RTs (or RF) are used for robustness.
Randomized tree

- Random forest
  - Final probability is summed value of probability of each RT.

Final probability
Fern classifier
Fern classifier

- Randomized tree

Depth 1

Depth 2

Depth 3

\begin{itemize}
\item \( f_0 \) with children \( f_1 \) and \( f_2 \)
\item \( f_1 \) with children \( f_3 \) and \( f_4 \)
\item \( f_2 \) with children \( f_5 \) and \( f_6 \)
\end{itemize}
Fern classifier

- Modified randomized tree
  - In same depth, same classifier $f$ is used

Diagram:
- Depth 1: $f_0$
- Depth 2: $f_1$ and $f_1$
- Depth 3: $f_2$, $f_2$, $f_2$, $f_2$
FERNS

Classifier Training

\[ P(F_k \mid C = c_i) \]

Posterior Distributions (Look-up Tables)

\[ 2^3 \text{ Possible Outputs} \]
**FERNS**

**Classifier Training**

- **Fern 1**
  - Posterior Distributions (Look-up Tables)
  - Class 1
  - Class 2

- **Fern 2**
  - Posterior Distributions (Look-up Tables)
  - Class 1
  - Class 2

- **Fern n**
Example Classification.

\[ \text{Example class label} = \arg \max_f \prod_{i=1}^M \left( P(F_k \mid C = c_i) \right) \]
Summary

❖ Pros.
  ● Easily handle multi-class problems.
  ● Easily cover large perspective and scale variations.
  ● Classifier training is time consuming, but recognition is very fast and robust.

❖ Cons.
  ● Memory requirement is high.