Paper Reading Seminar

Yan Wang

August 31, 2012

Kernel descriptors for visual recognition

- Problem: a distance metric for various visual features
  - Orientation histogram (HoG, SIFT)
  - Color
  - Texture (binary patterns)
- Efficiency

Approach

- Orientation histogram
  - Input: orientation histogram of two patches P, Q

$$K_{\text{grad}}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} \tilde{m}(z)\tilde{m}(z') k_o\big(\tilde{\theta}(z), \tilde{\theta}(z')\big) k_p(z, z')$$

  - $\tilde{m}(z)\tilde{m}(z')$: magnitude of the gradient as weights
  - k_o, k_p are Gaussian kernels, $k_p(z, z') = \exp\big(-\gamma_p \|z - z'\|^2\big)$
  - $\tilde{\theta}(z) = [cos\big(\theta(z)\big), \sin\big(\theta(z)\big)]$. L2 distance $\Rightarrow$ difference of gradient orientations
  - k_p measures the differences of pixel positions (for SIFT)

Approach

- Color

$$K_{\text{color}}(P,Q) = \sum_{z \in P} \sum_{z' \in Q} k_c\big(c(z), c(z')\big) k_p(z, z')$$

- Shape

$$K_{\text{shape}}(P,Q) = \sum_{z \in P} \sum_{z' \in Q} \tilde{s}(z)\tilde{s}(z') k_b\big(b(z), b(z')\big) k_p(z, z')$$

- $\tilde{s}(z) = s(z) / \sqrt{\sum_{z \in P} s(z)^2 + \epsilon_s}$
- $s(z)$: standard derivation of pixel values in the 3×3 pixels neighborhood
- $b(z)$: binary pattern of the neighborhood

163	155	124
168	139	187
171	135	130

$\rightarrow$

1	1	0
1		1
1	0	0

Learning compact features

- Kernel $k(x, y) = \psi(x)^T \psi(y)$
- Project to a low dimension space given bases $H = [\psi(z_1), \psi(z_2), ..., \psi(z_D)]$
- Compute the coefficients with close form solution

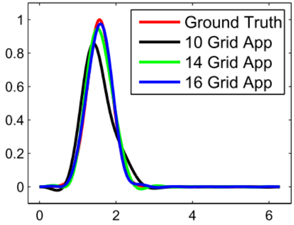
$$v_x^* = \arg\min_{v_x} \|\psi(x) - Hv_x\|^2$$

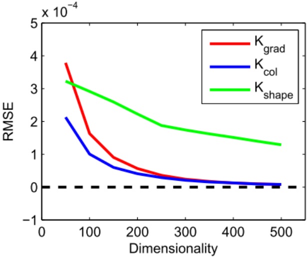
$$v_x^* = \left(H^T H\right)^{-1} \left(H^T \psi(x)\right)$$

- Approximate the kernel distance with the projection coefficient as the feature map

Learning compact features

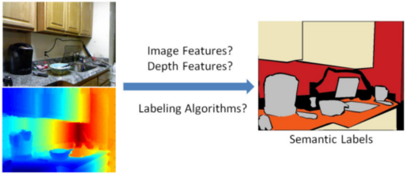
- How to get the basis?
  - Uniform dense sampling (like LSH?) from the support region (feature space)
  - Effective





RGB-(D) Scene Labeling: Features and Algorithms

- Problem: indoor scene, optical photo + depth image ⇒ pixel-wise label



- Evaluation: NYU Depth Dataset (13 categories), Stanford Background Dataset (8 categories, no depth info), Mean AP.

Intuition

- Kernel Descriptor + Efficient Matching Kernel: pixel level features in different domains $\Rightarrow$ superpixel level feature
- Segmentation tree: different scales of superpixel
- Contextual refinement

Approach

- Segmentation trees
  - gPb: local + global contrast cues $\Rightarrow$ pixel-level probability-of-boundary map
  - Extend to depth frames
  - Linear fusion for RGB-D frames

$$\mathrm{gPb}_{\mathrm{rgbd}} = (1-\alpha)\cdot \mathrm{gPb}_{\mathrm{rgb}} + \alpha \cdot \mathrm{gPb}_{\mathrm{d}}$$

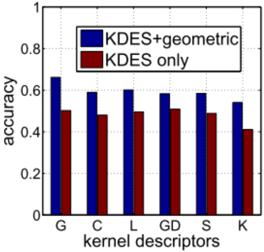
- Feature design
  - Gradient, color, local binary pattern, depth gradient, spin/surface normal, KPCA/self-similarity

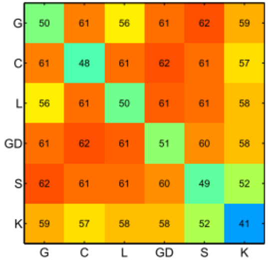
Approach

- Kernel descriptors
  - Intuition: pixel features $\Rightarrow$ superpixel

$$F^t_{\text{grad}} = \sum_{z \in Z} \tilde{m}_z k_o(\tilde{\theta}_z, p_i) k_s(z, q_j)$$

  (p_i, q_j are randomly sampled from the superpixel)





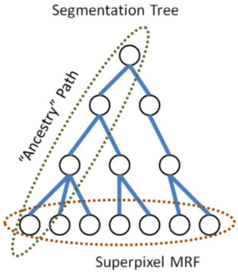
  - Use image gradient + spin/normal

Approach

- Classification
  - Efficient Match Kernel for fixed-length features on superpixels
  - Linear SVM
  - Normalize on superpixel area (A_s)

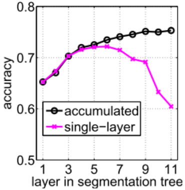
$$A_s / (\sum_{q \in Q_c} A_q)^p$$

- Segmentation tree
  - Different level (t) of segmentation tree $\Leftrightarrow$ different scale of superpixels
  - Tree(s) $= \{f_{t,c}(s_t)\}, t, c$

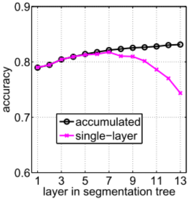


Approach

- Segmentation tree
  - Accumulate features along paths for better accuracy



(a) NYU Depth



(b) Stanford

- Superpixel MRF with gPb
  - Data term: $-f_{c,t}$
  - Smoothing term

$$V_{s,r} = \beta \exp(-\gamma \cdot \mathrm{gPb}_{\mathrm{rgbd}}(s,r))$$

  - Solve with graph-cut

Interesting insights

- Superpixel-level feature beats pixel-level feature + MRF
- High-dimensional features + Linear SVM pick up important features
- Segmentation hierarchy helps a lot