From images to descriptors and back again

Patrick Pérez

FGMIA 2014
Visual search

- Searching in image and video databases
- One scenario: *query-by-example*
  - Input: one query image
  - Output
    - Ranked list of “relevant” visual content
    - Information on object/scene visible in query
- Some existing systems
  - Google Image and Goggles / Amazon Flow / Kooaba (Qualcom)
Large scale image comparison

- Raw images can’t be compared pixel-wise
  - Relevant information is lost in clutter and changes place
  - No invariance or robustness

- Meaningful and robust representation
  - Global statistics
  - *Local descriptors aggregated in a global signature*

- Efficient approximate comparisons
Local descriptors

- Select/detect image fragments, normalize and describe them
  - Robust to some geometric and photometric changes
  - Most popular: SIFT $\in \mathbb{R}^{128}$

- Precise image comparison: match fragments based on descriptors
  - Works very well ... but way too expensive on a large scale

[Mikolajczyk, Schmid. IJCV 2004]
[Lowe. IJCV 2004]
Forget about precise descriptors
- Vector-quantization using a dictionary of $k$ “visual words” learned off-line

Forget about fragment location
- Counting visual words

BoW: *sparse fixed size signature by aggregation* of a variable number of quantized local descriptors

[Sivic, Zisserman. ICCV 2003][Csurca et al. 2004]
Efficient search with *inverted files*
- Search only images that share words with query
- Short-listing based on histogram distance

[Sivic, Zisserman. ICCV 2003]
Bag of “Visual Words” pipeline

- Geometrical post-verification
  - Match local features
  - Infer most likely geometric transform
  - Rank short list based on goodness-of-fit

[Sivic, Zisserman. ICCV 2003]
Limitations and contributions

- Precise search requires large dictionary \((k \sim 20,000-200,000 \text{ words})\)
  - Difficult to learn
  - Costly to compute \((k \text{ distances per descriptor})\) on database
  - Memory footprint still too large \((\sim 10KB \text{ per image})\)
    - With 40GB RAM, search 10M images in 2s
    - Does not scale up to web-scale \((\propto 10^{11} \text{ images})\)

- Contribution*
  - Novel aggregation of local descriptors into image signature
  - Combined with efficient indexing
    - Low memory footprint \((20B \text{ per image, 200MB RAM for 10M images})\)
    - Fast search \((50ms \text{ to search within 10M images on laptop})\)

*[Jégou, Douze, Schmid, Pérez. CVPR 2010]*
Beyond cell counting

- Vector of Locally Aggregated Descriptors (VLAD)
  - Very coarse visual dictionary (e.g., $k = 64$): $\mathbf{C} = \{c_1, \cdots c_k\} \in \mathbb{R}^{128}$
  - But characterize distribution in each cell

\[ \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{bmatrix} \]

\[ \mathbf{v} \propto \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^{128k}, \quad \mathbf{v}_i = \sum_{x \in \text{cell } i} (x - c_i), \quad \|\mathbf{v}\|_2 = 1 \]
■ Vectors of size $D = 128 \times k$, $k$ SIFT-like blocks

<table>
<thead>
<tr>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Fisher interpretation

- Given parametric family of pdfs \( \{p_\theta, \theta \in \Theta \subset \mathbb{R}^u\} \)
  - Fisher information matrix (size \( u \))
    \[
    F_\theta = \mathbb{E}_{p_\theta} [\nabla_\theta \ln p_\theta \nabla_\theta^T \ln p_\theta]
    \]
  - Log-likelihood gradient of sample \( \{x_n\}_{n=1}^N \)
    \[
    G_\theta(\{x_n\}) = \frac{1}{N} \sum_{j=n}^N \nabla_\theta \ln p_\theta(x_n)
    \]
- Fisher kernel: given \( \theta \), compare two samples
  \[
  K_\theta(\{x_m\}, \{y_n\}) = G_\theta(\{y_m\})^T F_\theta^{-1} G_\theta(\{x_n\})
  \]
  \[
  = \langle F_\theta^{-\frac{1}{2}} G_\theta(\{y_m\}), F_\theta^{-\frac{1}{2}} G_\theta(\{x_n\}) \rangle
  \]
  - Dot product of Fisher vectors (FV)

Example: spherical GMM with parameters $\theta = (\{\pi_i, \mu_i, \sigma_i\})_{i=1 \ldots k}$

- *Approximate* FV on mean vectors only

$$
G_{\mu_i}(\{x_n\}) = \frac{1}{N \sqrt{\pi_i}} \sum_{n=1}^{N} \kappa_n(i) \sigma_i^{-1}(x_n-\mu_i), \ i = 1 \ldots k
$$

with *soft assignments* $\kappa_n(i)$. FV of size $D = d \times k$

- If equal weights and variances, hard assignment to code-words, FV = VLAD

$$
G_{\mu_i}(\{x_n\}) \propto v_i(\{x_n\}), \ i = 1 \ldots k
$$
Additional tricks

- Power-law\(^1\)  \(v_j \leftarrow \text{sign}(v_j)|v_j|^\alpha, \ j = 1 \cdots D, \ \alpha \in (0, 1)\)
- Residue normalization (“RN”\(^2\))
  \[
  v_i = \sum_{x \in \text{cell } i} \frac{x-c_i}{\|x-c_i\|_2}, \ i = 1 \cdots k
  \]
- Intra-cell PCA local coordinate system (“LCS”\(^2\))
  \[
  v_i = R_i \sum_{x \in \text{cell } i} \frac{x-c_i}{\|x-c_i\|_2}, \ i = 1 \cdots k
  \]
- RootSift (“\(\sqrt{SIFT}\)”\(^3\))

---

1 [Jégou, Perronnin, Douze, Sanchez, Pérez, Schmid. PAMI 2012]
2 [Delhumeau, Gosselin, Jégou, Pérez. ACM MM 2013]
3 [Arandjelovic , Zisserman. CVPR 2013]
## Exhaustive search

- Comparisons to BoW on Holidays (1500 images with relevance GT)

<table>
<thead>
<tr>
<th>Image signature</th>
<th>dim</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW-20K</td>
<td>20,000</td>
<td>43.7</td>
</tr>
<tr>
<td>BoW-200K</td>
<td>200,000</td>
<td>54.0</td>
</tr>
<tr>
<td>VLAD-64</td>
<td>8192</td>
<td>51.8</td>
</tr>
<tr>
<td>+ $\alpha = 0.2$</td>
<td></td>
<td>54.9</td>
</tr>
<tr>
<td>+ $\sqrt{SIFT}$</td>
<td></td>
<td>57.3</td>
</tr>
<tr>
<td>+ RN</td>
<td></td>
<td>63.1</td>
</tr>
<tr>
<td>+ LCS</td>
<td></td>
<td>65.8</td>
</tr>
<tr>
<td>+ dense SIFTs</td>
<td></td>
<td>76.6</td>
</tr>
</tbody>
</table>
Towards large scale search

- PCA reduction of image signature to $D' = 128$
- Very fine quantization with *Product Quantizer (PQ)*
- Results on *Oxford105K* and *Holydays+1M Flickr distractors*

<table>
<thead>
<tr>
<th>Image signature</th>
<th>Ox105K</th>
<th>Hol+1M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best VLAD-64 (8192 dim)</td>
<td>45.6</td>
<td>–</td>
</tr>
<tr>
<td>Reduced (128 dim)</td>
<td>26.6</td>
<td>39.2</td>
</tr>
<tr>
<td>Quantized (16 bytes)</td>
<td>22.2</td>
<td>32.3</td>
</tr>
</tbody>
</table>

*[Jégou, Douze, Schmid. PAMI 2010]*
Quantized signatures

- Vector quantization on $k_f$ values
  \[ w \approx q(w) \]
- For good approximation, large codes
  - e.g., 128 bits ($k_f = 2^{128}$)
- Practical with product quantizer*
  \[ w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}, \quad q(w) = \begin{bmatrix} q_1(w_1) \\ \vdots \\ q_m(w_m) \end{bmatrix} \]
  with $k_r$ values per sub-quantizer
  - yields $k_f = (k_r)^m$ with complexity $k_r \times m$

* [Jégou, Douze, Schmid. PAMI 2010]
Quantized signatures

\( w \in \mathbb{R}^{128} \Rightarrow (w_1, w_2, w_3, \ldots, w_{15}, w_{16}) \)

256 quantized values

16 Bytes index ⇔ 1 Byte

\[
D' = 128 \\
m = 16 \\
k_r = 2^8 \\
k_f = 2^{128}
\]
Asymmetric Distance Computation (ADC)

- Given query signature $\mathbf{v}$, distance to a basis signature $\mathbf{w}$:

  \[
  \| \mathbf{v} - \mathbf{w} \|^2 \approx \sum_{i=1}^{m} \| \mathbf{v}_i - q_i(\mathbf{w}_i) \|^2
  \]

  $k_r$ possible values

- Exhaustive search among $N_b$ basis images

  $mk_r$ distances + $(m - 1)N_b$ sums
ADC with Inverted Files (IVF-ADC)

- Two-level quantization of signatures
  - Coarse quantization (e.g., $k_c = 2^8$ values)
  - One inverted list per code-vector
  - Compare only within lists of $w$ nearest code-vectors to query
  - Fine PQ quantization of residual signatures (e.g., $k_f = 2^{128}$)

- Search among $N_b$ basis images

  \[ mk_r \text{ distances} + w(m - 1)N_bk_c^{-1} \text{ sums} \]

  \[ w = 16, m = 16, k_r = k_c = 256 \Rightarrow \text{one sum only per image with almost no accuracy change!} \]
# Performance w.r.t. memory footprint

<table>
<thead>
<tr>
<th>Image signature</th>
<th>bytes</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW-20K</td>
<td>10,364</td>
<td>43.7</td>
</tr>
<tr>
<td>BoW-200K</td>
<td>12,886</td>
<td>54.0</td>
</tr>
<tr>
<td>FV-64</td>
<td></td>
<td>59.5</td>
</tr>
<tr>
<td>- Spectral Hashing* 128 bits</td>
<td>16</td>
<td>39.4</td>
</tr>
<tr>
<td>- PQ, $m = 16, k_r = 256$</td>
<td>16</td>
<td>50.6</td>
</tr>
</tbody>
</table>

*Weiss et al. NIPS 2008*
Large scale experiments

- Holidays + up to 10M distractors from Flickr

\[ k = 64, \text{ exact, 7s} \]

\[ k = 256, \text{ 320B} \]

\[ k = 64, \text{ 16B, 45ms} \]

BoW-200K
Larger scale experiments

- Copydays + up to 100M distractors from Exalead

Crop 50% of image surface

Strong transformations

- GIST
- GISTIS
- Fisher
- Fisher+IVFPQ

[64B, 245ms]

[64B, 160ms]

GIST: Oliva, Torralab. PBR 2006][GISTIS: Douze et al. AMC-MM 2009]
Beyond Euclidean distance

- Kernel-based similarities
  - Other better but costly kernels
  - For histogram-like signatures: Chi2, histogram intersection (HIK)
- Explicit embedding recently proposed for learning\(^1\)
  - Given PSD kernel function \( K(x, y) = \langle \phi(x), \phi(y) \rangle \)
  - Find an explicit finite dim. approximation of implicit feature map
    \[
    K(x, y) \approx \langle \tilde{\phi}(x), \tilde{\phi}(y) \rangle
    \]
  - Learn linear SVM in this new explicit feature space
  - KCPA\(^2\): a flexible data-driven explicit embedding
    \[
    [K(x_i, x_j)] = U \Lambda U^\top \approx U_E \text{diag}(\lambda_1, \ldots, \lambda_E) U_E^\top, \ E < N
    \]
- What about search?

\(^1\)[Vedaldi, Zisserman. CVPR 2010][Perronnin et al. CVPR 2010]
\(^2\)[Schölkopf et al. ICANN 1997]
Approximate search with short codes

- Simple proposed approach* ("KPCA+PQ")
  - Embed database vectors with learned KPCA
  - Efficient Euclidean ANN with PQ coding
  - Kernel-based re-ranking in original space

- Competitors: binary search in implicit space
  - Kernelised Locally Sensitive Hashing (KLSH) [Kulis, Grauman. ICCV09]
  - Random Maximum Margin Hashing (RMMH) [Joly, Buisson. CVPR11]

- Experiments
  - Data: 1.2M images from ImageNet with BoW signatures
  - Chi2 similarity measure
  - Tested also: “KPCA+LSH” (binary search in explicit space)

* [Bourrier, Perronnin, Gribonval, Pérez, Jégou. TR 2012]
Results averaged over 10 runs

Recall@R

\[ E = 128, B = 256 \text{ bits, } M = 1024 \]

Recall@1000

\[ B = 32 \rightarrow 256 \text{ bits} \]
Reconstructing an image from descriptors

- If sparse local descriptors only are known

- Better insight into what local descriptors capture, with multiple applications
Reconstructing an image from descriptors

- Possible to some extent

[Weinzaepfel, Jégou, Pérez. CVPR’2011]
Inverting local description

- Local description, severely lossy by construction
  - Color, absolute intensity, spatial arrangement in each cell are lost
  - Non-invertible many-to-one map
  - Example-based regularization: use key-points from arbitrary images

- Patch collection must be large and diverse enough (e.g., 6M)
Inverting local description

\[ j^* = \arg \min_j \| d_i - d_j \| \]

\[ A_i^{-1}(q_{j^*}) \]
Assembling recovered patches

- **Progressive collage**
  - Dead-leaf procedure, largest patches first

- **Seamless cloning**
  - Harmonic correction: smooth change to remove boundary discrepancies

- **Final hole filling**
  - Harmonic interpolation

*Pérez, Gangnet, Blake. Siggraph 2003*
Reconstruction
Reconstruction
Outlook

- New: reconstruction from dense local features
  - HOG (Hoggles)$^2$

- Human-understandable images can be reconstructed
  - Visual insight into information exploited by detectors and classifiers
  - *Visual information leakage* in image indexing systems: privacy?

---

$^1$ [D’Angelo, Alahi, P. VanderGheynst. ICPR 2012]

$^2$ [Vondrick, Khosla, Malisiewicz, Torralba. ICCV 2013]