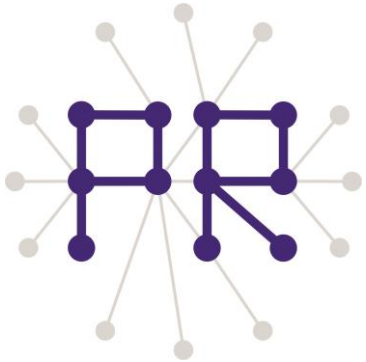
Generic Object Recognition

11 Extraction of BoVW Representation

Kimiaki Shirahama, D.E.

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany



Overview of Today's Lesson

- Implementing BoVW Extraction

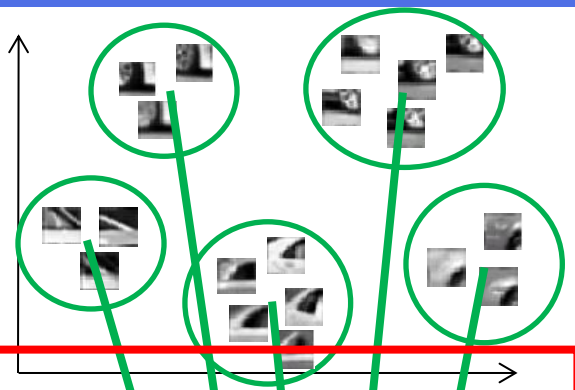
That's all ☺

Overview of Generic Object Recognition Using Bag of Visual Words (BoVW)

1. Visual word extraction: Organise local features into groups of similar features

➡ *The center of each group becomes a visual word*

More than 100,000 local features are organised into more than 1,000 groups. In other words, more than 1,000 visual words are extracted

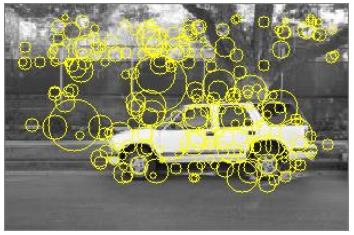


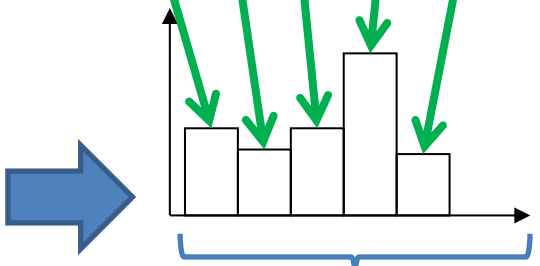
Today

2. BoVW representation: Assign local features in each image to the most similar visual words

➡ For each visual words, the number of assigned local features becomes the value of a bin

The number of dimensions of a histogram (vector) is more than 1,000.

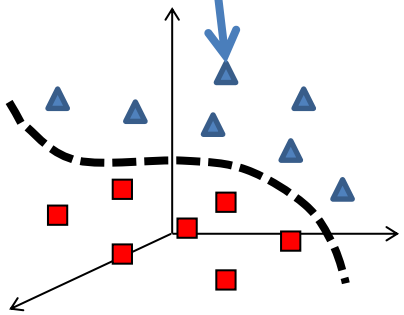




3. Classification: Extract the boundary between images where a certain object is shown and images where it is absent

➡ One image represented by BoVW is represented as a point in the high-dimensional vector space

Because of the high-dimensionality, simple similarity measures (e.g., Euclidian distance and cosine distance) do not work. Support Vector Machine (SVM) or other effective classifiers for high-dimensional data must be used.
I spent more than one year to find this point ☹

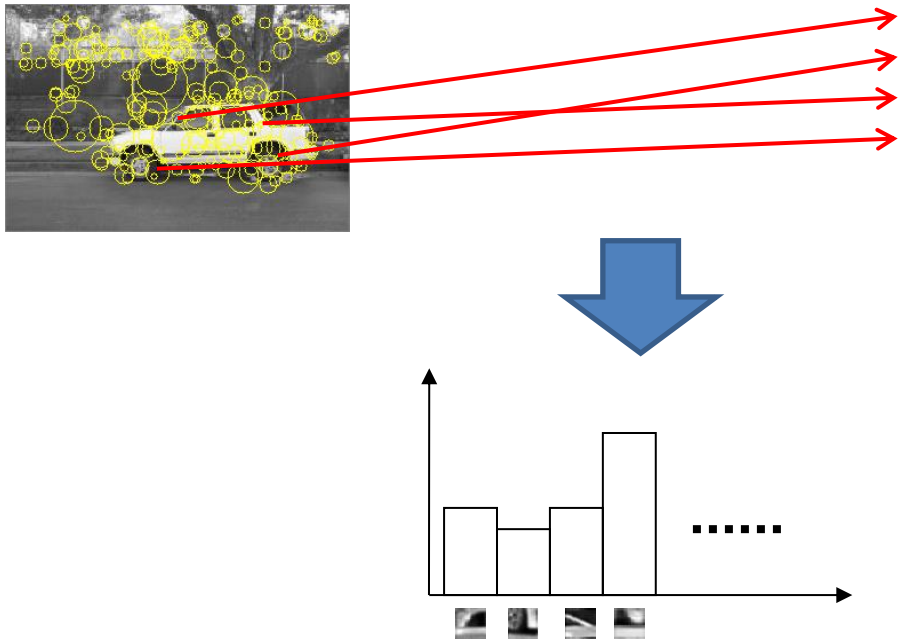


Overview of BoVW Extraction

1. Extract SURF features from an image

Create a histogram where each bin represents the frequency of a visual word

2. For each SURF feature,
   - Find the most similar visual word
   - Increment the bin corresponding to the most similar visual word



Pseudo Code of BoVW Extraction

```
vector< vector<double> > visual_words; // You can use any kind of matrix, such as cv::Mat
loadVisualWords(filename_of_visual_words, visual_words);

ofstream bovw_file(Text filename where BoVW representations of images are stored)

// To get image filenames, you can re-use a part of the code implemented in the 3-rd or 8-th lesson
For each image filename, do the following things:
    vector<double> bovw; // Histogram representing the frequency of each visual word (any one-dimensional array is OK)
    extractBoVWRepresentation(image filename, bovw, visual_words);
    saveBoVWRepresentation(bovw_file, bovw);
end of "For each image filename, …"

extractBoVWRepresentation(filename, bovw, visual_words){

    // Extract SURF features from the image specified by filename (see slides in the 7-th and 8-th lesson)
    // If no SURF feature is extracted, set all bin values in bovw to "0"

    for i (representing a SURF feature ID)
        // Compute the similarity between i-th SURF feature and a visual word as their Euclidian distance
        int visual_word_id = searchMostSimilarVisualWord(i-th SURF feature, visual_words);
        bovw[visual_word_id]++;
    end "for i"

    // Normalise bovw so that the sum of bin values becomes "1" (in the same way to histogram-based image retrieval)
    normaliseHistogram(bovw);

}
```

Please make sure that your code does not have any bug!
Even one small bug can cause a very disappointing result of generic object recognition.