

Feature Selection for Real-time Image Matching Systems

Quan Wang and Suya You
University of Southern California
Los Angeles, USA
{quanwang, suyay}@usc.edu

Abstract

This paper proposes a general feature selection approach for real-time image matching systems. To demonstrate the idea's effectiveness, we focus on the issue of rotational invariance. Most current image matching methods compute and align local image patches to a uniform dominant orientation, which are either too computationally expensive for real-time systems or insufficiently robust. In contrast to current approaches, we combine multiple-view training and feature selection into a unified framework. The most invariant features are selected during an offline training stage. Therefore, no additional computation is needed for online processing. Furthermore the proposed Rotation Invariant Feature Selection (RIFS) can be easily adapted to similar image matching problems such as scale invariance improvement and kernel selection in feature description.

Experimental results show the effectiveness of RIFS using only a small number of training views. The proposed approach is also successfully integrated into an augmented reality application for museum exhibitions.

1. Introduction and related works

Image matching under varying conditions is a challenge for virtually all intelligent vision systems including automated image registration, object recognition and tracking, content based database retrieval and image based modeling [4, 5, 6, 8, 9]. Many local feature based image matching systems utilize orientation alignment for rotation invariance. In [6], orientation histograms are computed from local circular regions of the relative scale. Although robust to image rotation and noise, histogram based methods are typically too computationally expensive to be a component of real-time image matching systems, because the process generally involves time-consuming steps such as relative scale searching, dominant orientation calculation, and pixel values extraction from irregular regions.

To tackle this problem, Lepetit and Fua [4] introduced a simplified orientation correction technique for real-time applications. The method only considers intensity changes along a fixed-size circular region centered on each keypoint. It is not, however, robust to scale changes or out-of-plane rotations. The rotation invariance of their proposed system still comes primarily from the multiple view training. Experimental results indicate a large number of training views are crucial to the system's reliability [5]. However, even the affine transformation space for simple planar objects has six parameters and thus demands a huge number of training views for stable and reliable performance. For example, if only 100 features are kept for each object, then 1000 training views will generate 100,000 feature vectors per object. Consequently, proper feature selection is crucial for fast and accurate performance of the whole system.

To enhance rotation invariance in real-time image matchings systems, we propose a different approach combining feature selection and multiple-view training into one unified framework. We first construct a small number of rotation-dominant views and obtain a set of descriptors for each view track. Then for those feature points with a high repeatability, raw ranking scores (RRS) are calculated based on feature distinctiveness and invariance. Finally, the raw ranking score is rescaled, weighted and combined with the other traditional feature selection criterion for the final ranking score (FRS). Features with high FRS are selected.

We integrated our feature selection approach into Multiple View Kernel Projection (MVKP) [9], but the approach could, in principle, work with any image matching system using vector-form descriptor. Additionally, as a general feature selection method, RIFS can be easily adapted to select scale-invariant parts of an object or select kernels for feature description.

Feature selection has been widely used to reduce computation time and improve accuracy. Multi-class SVM was used in [2] to select the most informative features for face recognition. The proposed SVM-DFS can

speed up classification without degrading the matching accuracy. Mahamud and Hebert [7] proposed discriminative object parts selection and used conditional risks as the distance measure in nearest neighbor search. Dorko and Schmid [1] introduced a method for selecting most discriminative object-part classifiers based on likelihood ratio and mutual information. None of these approaches focuses on rotation invariance or utilizes the additional information introduced by specifically designed and labeled training views.

2. Rotation Invariant Feature Selection

In order to select features that are distinctive and invariant under various rotations, we first generate a small number of rotation-dominant synthesized views (Fig. 1) during the pre-processing stage using affine transformation [3].

$$x' = H_A x = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x \quad (1)$$

$$A = R(\theta)R(-\phi)DR(\phi) \text{ and } D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (2)$$

The rotation parameter θ uniformly distributes from $-\pi$ to π . Other parameters ranges are: $\phi \in [-\pi/2, \pi/2]$, $\lambda_1, \lambda_2 \in [0.95, 1.05]$, $t_1, t_2 = 0$ or 1.

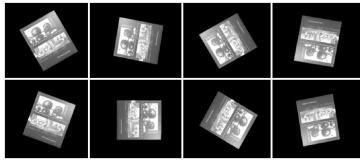


Figure 1. Rotation dominant views

The purpose of rotation-dominant views is to cover all the possible rotations while still containing small scale and skew changes because in the context of large view point changes, these issues are typically inter-related instead of independent. Besides rotation invariance, experimental results also indicate considerable improvement when dealing with general view point changes. Alternatively, we can also use scale-dominant synthesized views for scale invariant feature selection.

Our feature selection method's overview is given in figure 2. First, sets of local patches belonging to the same physical location (view tracks) are constructed and their corresponding feature descriptors computed. Then we compute three ranking scores for each view track based on distinctiveness, invariance, and stability. Finally, feature points with high weighted ranking scores are selected to build the object feature database.

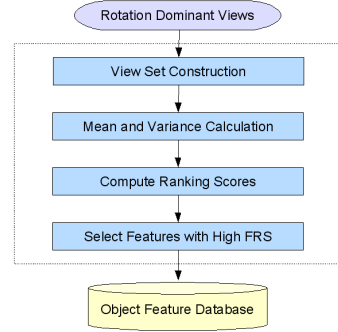


Figure 2. System Overview

The first step is *View Set Construction*. Suppose N_V and N_F are the numbers of training views and feature points independently detected in those views. Let V_k ($k = 0 - N_V$), F_i ($i = 1 - N_F$) represent training view and descriptor (N_D -dimensional vector) for those feature points respectively. Because the affine transformations from V_0 (the input object image) to all the synthesized training views are known, we are able to identify subsets of F_i belonging to the same physical locations. Such a subset is called a view track of the object, represented by T_j ($j = 1 - N_T$). Formally we can define:

$$\delta_{i,j} = \begin{cases} 1 & \text{if feature } i \text{ belongs to view track } j \\ 0 & \text{otherwise} \end{cases}$$

All features are described by 32×32 local patches around feature points. Our strategy is to treat each patch as a vector composed of 1024 pixel intensities. These vectors will be projected onto a space with much lower dimensionality (N_D) using the Walsh-Hadamard kernels. Experimental results [8, 9] demonstrate that the W.H. kernels projection approach remains reliable even under very noisy conditions and fast enough for real-time systems. The outputs of view set construction are the compact feature representations and $\delta_{i,j}$ values for all the feature points and view tracks.

After all the view tracks are constructed, each corresponding to a physical object location, our goal is to select view tracks that are distinctive, invariant and stable. The stability is measured by the feature repeat rate across all the training views, equivalent to the size view tracks. The stability score for view track j is defined as: $SS_j = \sum_{1 \leq i \leq N_F} \delta_{i,j}$. Besides eliminating those feature points with stability scores lower than a certain threshold (L_{SS}), we also propose additional criteria by utilizing further information provided by view tracks.

Features that are not distinctive from one view track to another are more likely to come from small repeated elements of the scene, which cause confusion to image

matching systems. When training by rotation-dominant views, features that have large variance within one view track have a high possibility to come from boundaries of a region, forming more diverse descriptors under rotation and potentially compromising the system’s robustness. Therefore, our goal is to select those feature points that are distinctive and invariant, and consequently to improve the geometric invariance of the whole system.

The distinctiveness is measured by the expected value of pair-wise distances between view track mean vectors. We measure the variance of each view track by the expected value of single dimension variances. Thus, *Mean and Variance Calculation* is needed for evaluating feature ranking scores.

To measure the distinctiveness of one view track T_j , we first compute the mean vector of all the feature vectors belonging to T_j :

$$M_j = \frac{\sum_{i=1}^{N_F} \delta_{i,j} F_i}{\sum_{i=1}^{N_F} \delta_{i,j}}$$

Let $Dist_{j,j'}$ represent the mean vector distance between view track T_j and $T_{j'}$. The *Distinctiveness Score* for view track T_j is defined as: $DS_j = \frac{1}{N_T} \sum_{1 \leq j' \leq N_T} Dist_{j,j'}$

Because each view track corresponds to a number of high dimensional feature vectors, an intuitive way to compute the variance property is to use co-variance matrix. However, computing co-variance matrices for each view track is time-consuming and not necessarily needed. We simplified variance computation by independently handling each dimension of one view track’s all feature vectors, which provides a N_D -dimensional variance vector for each view track:

$$VV_{j,l} = \frac{\sum_{i=1}^{N_F} \delta_{i,j} (F_{i,l} - M_{j,l})^2}{\sum_{i=1}^{N_F} \delta_{i,j}}$$

where $l = 1 - N_D$ is the dimension of vectors.

The *Variance Score* for view track T_j is defined as the expected value of all VV_j ’s components: $VS_j = \frac{1}{N_D} \sum_{1 \leq l \leq N_D} VV_{j,l}$

Good features should have high distinctiveness and low variance. Therefore, the *Raw Rank Score* (RRS) for view track T_j is defined as: $RRS_j = \frac{DS_j}{VS_j}$.

The unified formula for RRS expressed by original

feature descriptors is:

$$RRS_j = \frac{\frac{1}{N_T} \sum_{j'=1}^{N_T} \left| \sum_{i=1}^{N_F} \delta_{i,j} F_i - \sum_{i=1}^{N_F} \delta_{i,j'} F_i \right|}{\frac{1}{N_D} \sum_{l=1}^{N_D} \frac{\sum_{i=1}^{N_F} \delta_{i,j} (F_{i,l} - \sum_{i=1}^{N_F} \delta_{i,j} F_{i,l})^2}{\sum_{i=1}^{N_F} \delta_{i,j}}}$$

Finally, RRS is rescaled to the same range as stability score and combined together through a weight parameter ($\alpha = 0 - 1$) to form *Final Ranking Score*. $FRS_j = \alpha RRS_{rescaled,j} + (1-\alpha) SS_j$. Here $\alpha = 0$ is the traditional criterion when only repeatability is considered while $\alpha = 1$ is the extreme case using only RRS.

Given all the ranking scores, the final *Feature Selection* step is to select object features (corresponding to view tracks) with scores higher than a threshold or select a certain percentage of high scoring features.

3. Experimental results

We use synthesized images with known correspondences and challenging real images. During our experiments, MVKP with feature selection use only 50 views for general training and 72 rotation-dominant views for feature selection part, while other methods [5] typically require 1,000 training views for similar performance.

The proposed approach is also applied to a museum application [8] and improves the system’s performance.

3.1. High ranking feature evaluation

We apply different weight parameters to the same set of training views. $\alpha = 0$ is the traditional repeat rate criterion and $\alpha = 1$ corresponds the extreme case using only the proposed ranking scores.

Figure 4 visualizes the difference in selected features. For both α values, only the 100 top-ranking features are marked as white plus symbols. It can be clearly observed that: high ranking features selected by our approach (1) are more likely to distribute within high texture region instead of on boundaries of such regions. Features on the region boundary introduce large variances when rotated and have high probability of being mixed and affected by background textures; (2) have a smaller chance of appearing on the repeated elements of the scene (the top left part of figure 4(b)). Features describing repeated elements are less distinctive and can confuse image matching systems.

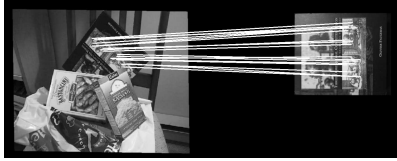


Figure 3. Image matching result. $\alpha = 0.5$. The original image was adapted from [6]

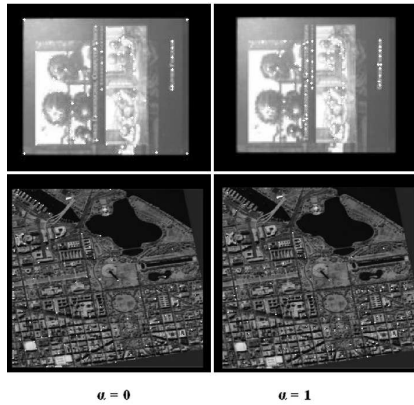


Figure 4. Feature selection visualization

3.2. Improvement from feature selection

We carry out a number of independent pair-wise image matching tests (with large view point and lighting changes) and only keep the 30 percent top-ranking feature points. The test indexes are ordered by the increasing error of $\alpha = 0.5$ case. The average error is defined as the average Euclidean distances from all the reported matches to the corresponding ground truth matches.

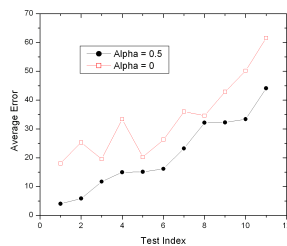


Figure 5. Feature selection improvement

Since our feature selection approach only occurs during training, it will not prolong online processing time. If a smaller number of features are selected, processing time can even be reduced. When the same percentage

of features is selected, figure 5 clearly indicates that the new feature selection method improves accuracy.

3.3. Influence of the weight parameter

During this test, we change the weight parameter from 0 to 1 but always select the 30% top-ranking features. Optimal performance is observed around $\alpha = 0.5$.

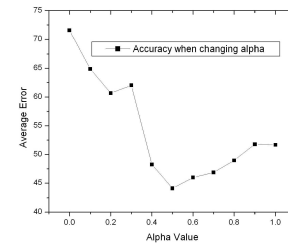


Figure 6. Changing weight parameter

We observe small changes of optimal weight for different kinds of images. If the image category is given, a good weight parameter can be found through machine learning techniques such as holdout or cross validation.

References

- [1] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *IEEE International Conference on Computer Vision*, pages 634–639, 2003.
- [2] Z. Fan and B. Lu. Fast recognition of multi-view faces with feature selection. In *IEEE International Conference on Computer Vision*, pages 76–81, Oct. 2005.
- [3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [4] V. Lepetit and P. Fua. Towards recognizing feature points using classification trees. Technical report, IC/2004/74, EPFL, 2004.
- [5] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465 – 1479, 2006.
- [6] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91 – 110, 2004.
- [7] S. Mahamud and M. Hebert. The optimal distance measure for object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 245–255, June 2003.
- [8] Q. Wang, J. Mooser, U. Neumann, and S. You. Augmented exhibitions using natural features. *International Journal of Virtual Reality*, 2008.
- [9] Q. Wang and S. You. Real-time image matching based on multiple view kernel projection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.