# Local Feature Based Salient Region Detection

Anonymous CVPR submission

Paper ID ****

## Abstract

*Local feature descriptors have become the most important part in image / video retrieval systems. But considering the great amount of local features, thousands of local features in one HD photo, it's hard to compute them efficiently in a realistic system. In our research, we overcome this obstacle with a straightforward local feature reduction processing by using a algorithm named LFSR (Local Feature based Salient Region). With no additional computation for salient regions, this algorithm help to improve both the performance and accuracy of local feature descriptors. In our evaluation, we also compare LFSR algorithm with a state-of-the-art salient region algorithm. And the results shows that LFSR provides a thousands of times computation speedup, with an acceptable precision loss. Furthermore, when integrated with the SURF algorithm, LFSR can provide a overall 1.6X speedup for the whole processing [**?**].*

## 1. Introduction

Our society has entered a data-centric era and a huge amount of data are transferred and processed on the Internet. Among them, multimedia data, such as image and video, has become one of the major data types being processed. As analyzed by CISCO Inc., video data occupies 50% of network traffic in 2011 and will increase to 90% in 2013 [1]. According to a report [6], as one of the most popular video sharing sites, more than 20-hour new videos are uploaded to *YouTube* every minute. Moreover, as two most popular photo sharing sites, *Facebook* and *Flickr* host billions of user-uploaded images respectively.

With the rapid increase of multimedia data, one of the most significant challenges is to understand and interpret such a huge amount of multimedia data. Currently, more and more retrieval applications are emerging to process these multimedia data, such as video recommendation [10], travel guidance systems [5] and content-based TV copy identification [7]. In these systems, a fundamental step is to extract feature information from images.

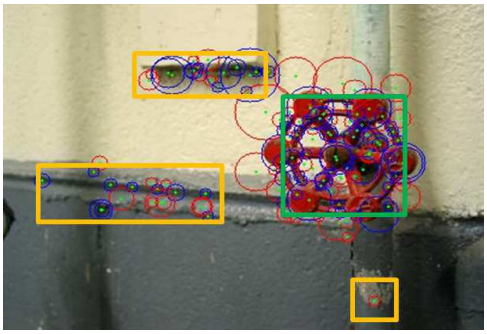
Image features can be divided into two domains – local features and global features. Global image features tend to describe the image as a whole, such as contour representations, shape descriptors and texture features. On the other hand, local image features represent image patches, computed at multiple points in the image. For example, SIFT [8] and SURF [3] are two most widely-used local image feature descriptors [9] [2]. They use histograms of gradient orientations to extract feature points and describe them using high-dimension feature vectors.

Compared to the global features, local feature descriptors are more robust, both scale-invariant and rotation-invariant. But even the SURF descriptor, which is an optimized algorithm for SIFT, is still very slow in a practical usage – the processing speed of SURF is about 2.6 frame per second on a 3.3GHz Core i7 CPU [4], far from the requirement of real time. Actually local feature descriptor should extract enough feature points from one image, and these features can be more than thousands in a standard VGA ($640 \times 480$) image. Considering the computation for describing each local feature, the great amount of local features means a relatively high overhead.

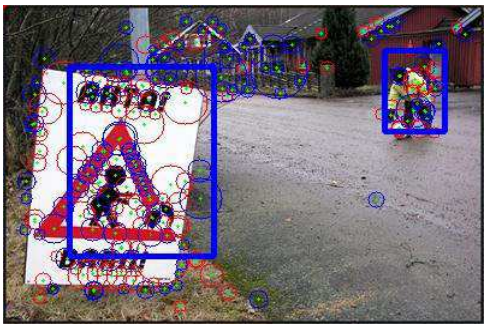
In general, there exist three major computation phases when processing local image features in a typical image retrieval system. First, the system detect all feature points from images. Then, with some specific formats and algorithms, each feature point is described as a high dimension vector. At last, all extracted feature are compared to each other according to the distance of their vectors.

According to a previous research [4], the computation of feature describing are obviously greater than the feature detecting in the SURF algorithm, which is caused by the great amount of local features to be described in one image. Furthermore, in a realistic image retrieval system, the performance is dominated by the number of features in the database.

So, it's possible and necessary to improve the performance of the whole system by the reduction of local features extracted from each image. In this paper, we present a algorithm named LFSR (Local Feature based Salient Region) to eliminate unimportant local features efficiently with no obvious precision loss for a image retrieval system.



(a) Salient region has the densest local features.



(b) Several salient regions in one image.

Figure 1. Example images with local feature based salient regions

The main idea of LFSR algorithm is extracting salient regions of images and only describing the local features in those regions. Without involving any other salient region algorithm, it is only based on the feature points extracted from the first phase of a typical local feature algorithm. This approach has two remarkable advantages: no additional computation for salient region detection; totally integrated with local feature algorithm to compute the salient region efficiently. The details of LFSR can be found in Section 2.

We have evaluated LFSR against a stat-of-the-art salient region algorithm. The evaluation results show that our approach has a much better performance with compatible precision and recall. Furthermore, withing a realistic image retrieval system, our evaluation shows that LFSR can help to improve both performance and accuracy.

## 2. Observations

We propose LFSR algorithm based on three major observations:

- **Observation 1:** Local feature in the salient region are close to each other, while noisy and unimportant features are far away from them. As shown in Figure 1(a), local features in the salient region (green box) gather together and many obvious unimportant local features locate far from that box.

- **Observation 2:** The region with most local features is the preferred salient region. Also from Figure 1(a), the preferred salient region in the green box has much more local feature than other two yellow boxes.

- **Observation 3:** There may exist several salient regions in one image. For example, the photo in Figure 1(b) contains two .....

Observation 1 indicates a method to rewrite our a


Figure 2. An overview of Local Feature based Salient Region algorithm

## 3. Local Feature Based Salient Region

### 3.1. Algorithm Overview

The basic idea of LFSR is to compute salient regions for local feature reduction. Since we are not concerned with the precise region boundary in our task, it's possible to get approximate salient regions with much less computation. According to our observation, the distribution of features in one image is related to the salient region, which means most concerned local features located in one major region and other noises located apart from them with a much larger distance to the salient region center. Thus, a approximate salient region can be regarded as a region expanded from the geometry center of local features. Considering the distribution of local features in a image's X-axis and Y-axis, the local feature based salient region should has a width and height ratio computed from the standard deviation of feature's positions in both X-axis and Y-axis.

In some images with more than one major objects, there may exist several local feature dense region, result in multiple LFSR in one image. To identify and compute these scenario efficiently, we involve a preprocessing to do a simple segmentation on all local features.

As shown in Figure 3.1, there exist two major stages in LFSR algorithm. First, a segmentation is performed on all local features to identify whether exist multiple salient region in that image. Second, for each image segmentation, LFSR computes that segmentation's salient region individually.

### 3.2. Local Feature Based Segmentation

One image may have multiple objects to construct a whole topic. When performing LFSR on this kind of images, we found that it's necessary to avoid computing the salient region across all local features in one run, which may lead to a significant precision loss.

There have been a lot of research about image segmentation, e.g. . But in our research, we prefer to do a approximate segmentation only relying on the geometric meaning of local features. LFSR solve this problem by performing scan operations in both X-axis and Y-axis of a image. In each scan, a cut-point may be found by following these two constraints:

1. No local feature should be divided into multiple parts. Every local feature can be recognized as a dot with a radius that equals its scale and no cut-point should locate on that dot. This constraint is based on an observation that one local feature should contribute to only one object, not several objects.

2. The cut-point should be located as near as possible to the center of image. Each scan is performed from the center of a image, in order to find the nearest cut-point for that image. This constraint is also based on an observation that major objects in one real photo always locate in the center, not far away from it.

The detailed steps are shown in Figure 2. The segmentation is started from the center of each axis. When a cut-point satisfying the above two constraints is found, LFSR stops scanning and takes that cut-point for the image segmentation. If a scan exceeds a threshold distance to the image center, for example 1/4 of the image with, the scan should stop and announce that there exists no valid segmentation on that dimension. After scanning on both X-axis and Y-axis, at most four image segmentations are found. If necessary, this kind of segmentation can be done recursively. But according to our observation, one run of segmentation is enough in most situations with almost no image with more than two major objects.

### 3.3. Local Feature Based Detection

As discussed in Section 3.1, a precise salient region detection is not necessary for local feature reduction. Thus, LFSR employ geometric meaning of local features to compute a approximate salient region.

Local feature can be represented as points in a image. After careful observations on the distribution of these points, we find that local features of one salient region locate near to each other while noises locating far away from them. To simplify this problem, LFSR regards the geometric center of feature points as the center of the salient region.

$$C(x,y) = \sum_{i}^{N} (P_i(x,y)) \tag{1}$$

Where $C(x,y)$ means the geometric center of every local feature point $P(x,y)$.

After locating the center, LFSR build the final salient region by expending the region as rectangle with a particular length-width ratio. And the ratio can also be computed directly from the distribution of features:

$$Ratio = \sqrt{\frac{\sum_i^N (x_i - x_c)^2}{\sum_i^N (y_i - y_c)^2}} \tag{2}$$

Where $s_i$ and $y_i$ is every feature's position, while $x_c$ and $y_c$ is the center position computed by Equation 1. To get the final salient region, LFSR grows the region size until the number of local feature in that area exceeds a threshold, for example 50 percent of the original local features.

In general, LFSR detects the salient region with one mean value and two standard deviation computations, and also a few additional loops to expand the detected region. Since these computations are only performed on local features, the cost should be very slight when integrated with local feature descriptors. Furthermore, we find that this simple approach also achieves an obvious performance advantage when compared to other precise salient region algorithms.

## 4. Evaluation

### 4.1. Experimental Comparison

### 4.2. Integration with SURF Descriptor

## 5. Conclusion

## Acknowledgement

## References

[1] Cisco Visual Networking Index: Forecast and Methodology, 2010-2015. 2011. 1

[2] J. Bauer, N. Sunderhauf, and P. Protzel. Comparing Several Implementations of Two Recently Published Feature Detectors. *International Conference on Intelligent and Autonomous Systems*, 2007. 1

[3] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *European Conference on Computer Vision*, pages 404–417, 2006. 1

[4] Z. Fang, D. Yang, W. Zhang, H. Chen, and B. Zang. A Comprehensive Analysis and Parallelization of an Image Retrieval Algorithm. *ISPASS*, 2011. 1

[5] Y. Gao, J. Tang, R. Hong, Q. Dai, T. S. Chua, and R. Jain. W2GO: a travel guidance system by automatic landmark ranking. *ACM Multimedia*, pages 123–132, 2010. 1

[6] C. Jansohn, A. Ulges, and T. M. Breuel. Detecting pornographic video content by combining image features with motion information. *International Conference on Multimedia*, 2009. 1

[7] A. Joly, C. Frelicot, and O. Buisson. Robust Content-Based Video Copy Identification in a Large Reference Database . *International Conference on Image and Video Retrieval*, pages 511–516, 2003. 1

[8] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1

[9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 1

[10] B. Yang, T. Mei, X.-S. Hua, L. Yang, S.-Q. Yang, and M. Li. Online video recommendation based on multimodal fusion and relevance feedback. *International Conference on Image and Video Retrieval*, pages 73–80, 2007. 1