

Finding Templates Images with Masked Regions

Abstract

In the field of Dynamic User Interface, where regions of the Interface are constantly changing (such web pages), we want to find Template images (sub image) within a large Source image (the image in which we expect to find a match to the Template image), by excluding the dynamic regions from the Template image.

In such cases, we need to augment our template matching technology to regard only the static regions and exclude the dynamic parts.

We introduce a Template Match Algorithm intended to solve such problems, to find a Template image, where arbitrary regions are occluded with binary Mask, within a second large Source image. It has the ability to find as much as possible occurrences of the Masked Template image within the Source image, that match (are similar) only on the areas outside the occluded binary mask.

For example, when UI controls (such UI buttons) change their content (such text idiom), we need an efficient algorithm to find occurrences of the Dynamic UI controls excluding the UI button text.

Suppose we have a Source image that contains rectangles with text (UI buttons) and other shapes, and a Template image with text on it, we want to find all the rectangles regardless the text on it.

In this case, the text regions of the Template are masked out, and a matching against the Source Image is performed. The result should be all the rectangles on the Source Image (Fig. 1).

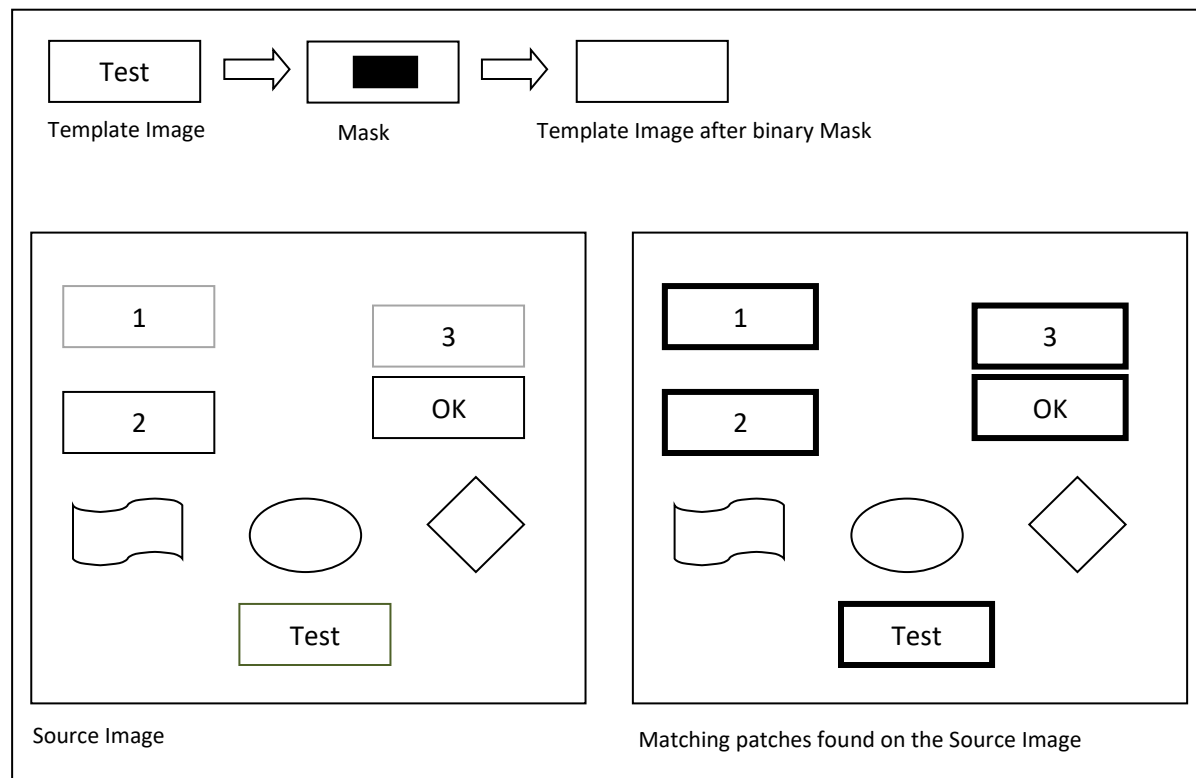


Fig. 1 – Finding Template image within a Source image.

The core of this Template Matching algorithm is based on the well known Feature matching theory of image processing. The matching method is based on Template Descriptor correlation rather than Image or Feature descriptors correlation currently used by most matching applications.

The Template Descriptor is a signature that represents the Template image by a vector of N-bytes, and can be changed in such a simple way that it can represent the Masked Template Image.

The correlation of the Template descriptor is performed against a set of Template patches Descriptors within the Source Image computed in selected feature points.

The presented method allows applications to perform efficiently matching tasks with arbitrary mask regions.

Prior Solutions

A variety of solutions had been presented to solve the problem of finding a match of a Template image within a source image; however most of them do not involve the problem of finding Templates with arbitrary masked regions. These methods use image correlation on the pixel or frequency domains, or feature based correlation, they tend to fail when occluded regions are present.

The feature based correlation methods are used for correlation of un-occluded images and most of them are used for tracking feature points in image motion field.

The pixels correlations are of poor performance in high quality images, they compute the correlation of two images based on all data.

Harold Stone et al. describe methods for using binary occlusion Mask, the correlation is performed in the frequency domain. Stone and Shamoon patented a method for computing correlation on partially occluded data also based on correlation on the frequency domain.

Algorithm Description

Step 1. Detect Template image and Source image Feature points using FAST or HARRIS feature detectors.

Feature detection is a low-level image processing operation that examines every pixel of the image to see if there is an "interesting" part of the image present at that pixel (the Feature point).

We use the FAST (Features from Accelerated Segment Test) or HARRIS feature algorithm that derives from the definition of what constitutes a "corner".

The FAST Feature detector is based on the image intensity around a putative feature point, and has been specifically designed to allow very fast detection of interest points in an image.

If the Template image is Masked out, then the feature points under the Mask are discarded (Fig. 2) and not added to the list of valid feature points of the Template that will be use in next steps for the calculation of the descriptors.

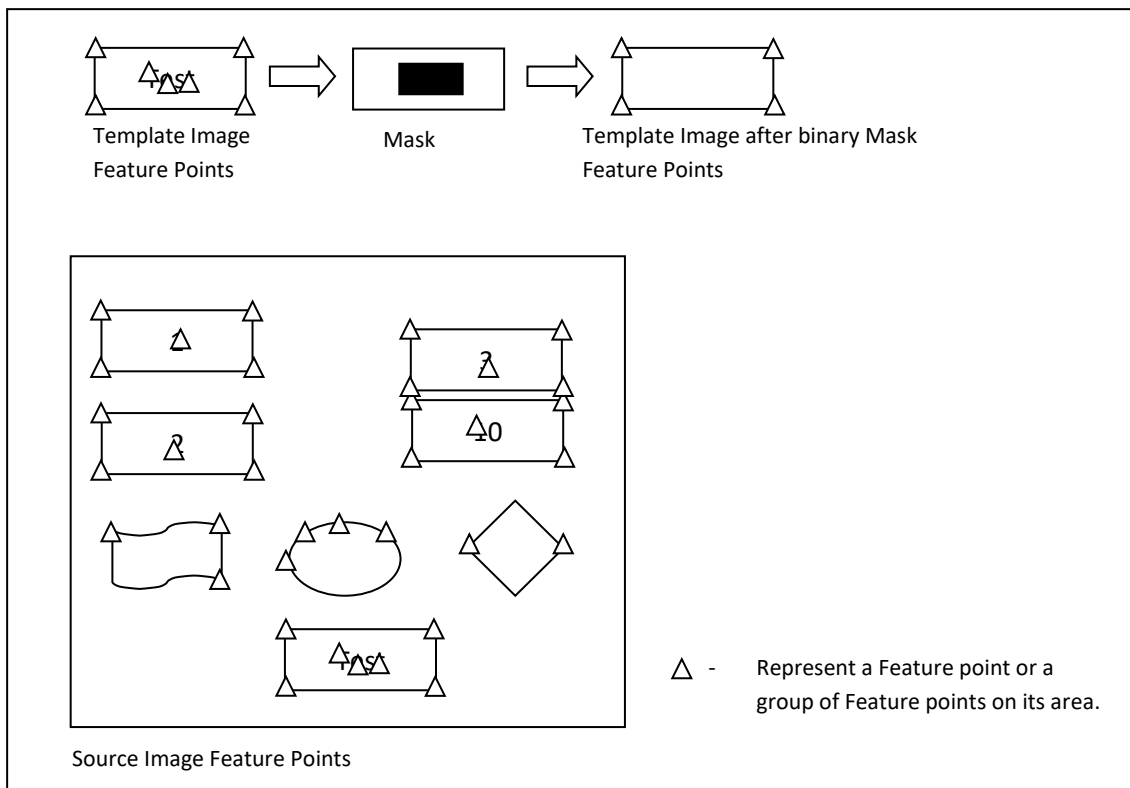


Fig. 2 – Template and Source Images Feature Points.

Step 2. Calculate Template Image Descriptor.

The core of the presented algorithm is based on the definition and calculation of the TID descriptor.

In feature matching, feature descriptors are usually N-dimensional vectors that describe a feature point, ideally in a way that is invariant to change in lighting and to small perspective deformations. In addition, good descriptors can be compared using a simple distance metric (for example, Euclidean distance). Therefore, they constitute a powerful tool to use in matching algorithms.

The TID descriptor is however, a N-dimensional vector that describe the entire Template image. It is a 64 Bytes vector, for which 63 Bytes are used to store a mask of each feature point angle, and one byte is used to store the mean of the image pixels values.

For each Template feature point calculated in previous Step, calculate the angle between the feature point and a base point on the Template image (we take this base point to be the upper left corner of the image). The corresponding bit on the 63-Byte vector is set (Fig 3).

The resolution of the angle mask is $63\text{Bytes} \times 8\text{bit} / 90\text{Deg.} = 5.6$.

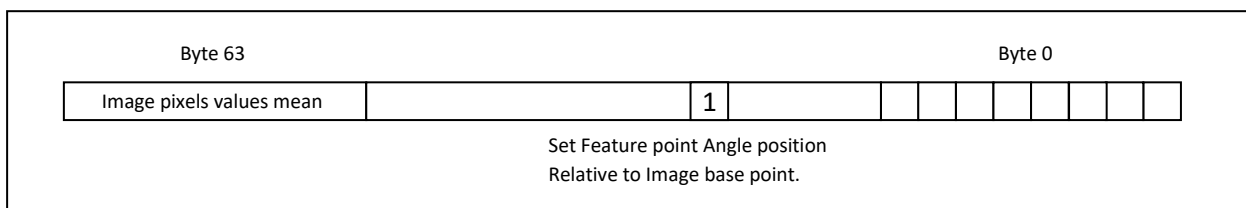


Fig. 3 – TID Image Descriptor.

Step 3. Calculate Source Image TID descriptors.

For the Template Image, calculate the distance D_t of the Top-Left most feature point relative to the Top-Left corner of the Template Image.

For each Feature point $x - FP_x$ on the Source image, set a rectangle of interest (*Roi*) of size as Template image size, where the distance between the selected Feature point to the Top-Left *Roi* corner is D_t .

Find all the Feature points inside this *Roi* and calculate the TID descriptor $Desc_x$ as specified in Step 2 where *Roi* is now the Template image (again, feature points under the Mask are discarded).

Step 4. Matching TID descriptors.

For each TID descriptor $x - Desc_x$ on the Source image, perform bitwise operations with the Template image TID descriptor to retrieve descriptor similarity, for the first 63 Bytes of the descriptors.

Moreover, by using the 64 Byte, compare the mean value of the images pixels.

Return the N most similar Source image Feature points, according to pre-setting thresholds.

Step 5. Calculate Moments descriptors.

It is possible that in the previous Step we receive much false identifications because we rely on Feature points angles comparisons, where, in non-similarity between Template image and Source patch image, the TID descriptors may be with high similarity.

For this reason, we perform a second matching for only the N most similar Source image Feature points returned in Step 4. This matching is based on calculating image Moments as follow.

For each Image patch of the Feature points returned in Step 4, and for the Template image, calculate the Moments of the masked image and store the Central and Normalized central Moments in a 64 Byte vector descriptor.

Step 6. Matching by Moments descriptors.

Once the Moments descriptors were created, perform matching of the Template image and Source Feature points patches images Moments descriptors and return the N most similar Source image Feature points.

Step 7. Return *Roi* for the matched Source Feature point.

Each Feature point returned in Step 6 represents a match for the Template image.

We calculate the *Roi* patch around each Feature Points and calculate its Location in the Source image.

Each of these coordinates represents a match of the Template image over the Source Image.

References

E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking", in IEEE International Conference on Computer Vision, October 2005.

E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", in European Conference on Computer Vision, pp. 430-443, May 2006.

C. Harris and M. Stephens (1988), "A combined corner and edge detector", in Proceedings of the 4th Alvey Vision Conference. pp. 147–151.

Robert Laganière, "OpenCV2 Computer Vision Application Programming Cookbook", Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK, May 2011.

Jan Flusser, Tomáš Suk and Barbara Zitová, "Moments and Moment Invariants in Pattern Recognition", A John Wiley and Sons, Ltd, Publication, 2009.

McGuire, M., Stone, H.S., "Techniques for multiresolution image registration in the presence of occlusions", in Geoscience and Remote Sensing, IEEE Transactions, Volume 38 , pp 1476 – 1479, May 2000.

Shamoon et al., "Method for Computing Correlation Operations on Partially Occluded Data", United State Patent, US005867609A, Feb. 1999.