# Finding complex patterns using template matching

G. Jasvilis [a, *], C. Weise [b], B. Zenger-Landolt [b]
[a] Trimble, Geospatial Division, Vingriu st. 6A, 01141 Vilnius – gediminas_jasvilis@trimble.com
[b] Trimble, Geospatial Division, Arnulfstr. 126, 80636 München, Germany – (christian_weise, barbara_zenger-landolt)@trimble.com

**KEY WORDS:** template matching, object-based image analysis, correlation coefficient, omission and commission error

## ABSTRACT:

Complex patterns that combine pixels of different brightness and colour in a fixed spatial layout pose a challenge for the object-oriented approach to image analysis. This is because the pixels that make up the pattern cannot readily be linked together into an object as they do not share spectral or textural properties. The goal of this study was to determine whether template matching techniques can be used to identify those patterns without a prior segmentation of the image.

We studied three different scenarios that pose specific challenges: the application of template matching is most straightforward in (1) the detection of *symbols in scanned maps* where different occurrences of a given pattern strongly resemble one another; most notably, symbols always have the same size and orientation. (2) The detection of *trees in aerial images of oil palms* is more challenging, since the characteristic tree pattern exhibits significant variations from tree to tree. Finally, in the (3) detection of *pavement markings in aerial images* objects could appear at arbitrary orientations in the image.

Our results show that template matching successfully identifies many target regions. Once target locations are known, objects can be created and the full power of object-based image analysis approaches can be applied to eliminate remaining false positives.

## 1. INTRODUCTION

For object-oriented image analysis to exhibit its full strength, meaningful objects have to be created (Baatz, Hoffmann & Willhauck 2008). This can be challenging for patterns characterized by a specific spatial layout that cannot easily be described with shape or texture features. What is needed is a method that provides information about the likelihood of pattern presence before the segmentation step. In this study, we investigated whether classical template matching techniques (e.g. Nixon & Aguado 2012, Bradski & Kaehler 2008) can be used to solve this problem.

There are a variety of different algorithms in common use that differ mostly in how the similarity between a template and an image region is quantified (Bradski & Kaehler 2008).

One simple method for measuring the similarity *s* is to use the sum of the squared differences between a template *t* and an image region *I:*

$$s = \sum_{x,y} \left( t(x,y) - I(x,y) \right)^2 \tag{1}$$

Note that for this measure, a simple overall shift in mean intensity in the image can harm the match quality as much as any real change in the shape of the pattern. Therefore, the sum of squared differences may not be a good option for many common use cases, where natural variations in illumination cannot be eliminated.

Another very simple measure for match similarity *s* is the *correlation* between a template and an image region *I*.

$$s = \sum_{x,y} t(x,y) \cdot I(x,y) \tag{2}$$

Unfortunately, this measure is also affected heavily by shifts in mean illumination. The disadvantage can be eliminated though, when the Pearson's *correlation coefficient r* is used as similarity measure.

$$r = \frac{\sum_{x,y} (x-\bar{x})(y-\bar{y})}{\sqrt{\sum_{x,y} (x-\bar{x})^2} \cdot \sqrt{\sum_{x,y} (y-\bar{y})^2}} \tag{3}$$

The correlation coefficient can be interpreted as a correlation between a template and an image *after both the template and the image have been z-normalized* (i.e. rescaled so that the mean is zero and the standard deviation is 1). Illumination and contrast differences are thus eliminated before match quality is evaluated — making the correlation coefficient an ideal measure of match when we want to ensure robustness for variations of pattern brightness and contrast.

Our study shows that template matching based on the correlation coefficient *r* can successfully identify potential target regions in a number of real life use cases.

## 2. USE CASES

Our study included three different use cases, each highlighting a different challenge.

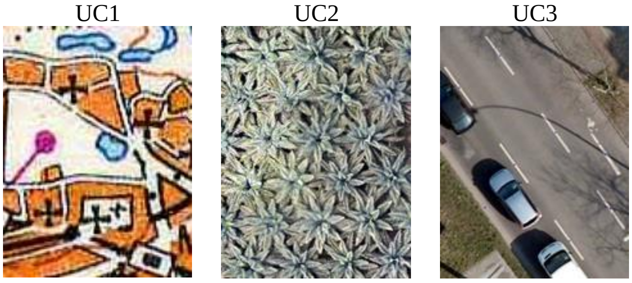

Figure 1. Image thumbnails for the different use cases.

[*] Corresponding author

### 2.1 Use case 1: symbols in scanned maps

Symbols in scanned maps come at a fixed size and a fixed orientation, thus perhaps representing an ideal problem to address with template matching, as we need worry neither about scale nor rotation invariance.

### 2.2 Use case 2: trees in aerial images of oil palm plantations

The detection of palm trees in a plantation is already more complex. The trees have a very characteristic shape, but they do not exactly match one another. There is a significant amount of natural variation or *pattern heterogeneity* that needs to be taken into account.

### 2.3 Use case 3: pavement markings in aerial images

The detection of specific pavement markings (here the white bars that separate different lanes) pose a new challenge: the markings may appear at an arbitrary number of different orientations in a given image.

## 3. METHODS

### 3.1 Template Generation

Before template matching can be applied to a problem, an adequate template for a given pattern of interest needs to be generated. In this study, templates were generated from manually selected samples. To avoid overlearning, samples were collected in regions outside the designated test region.

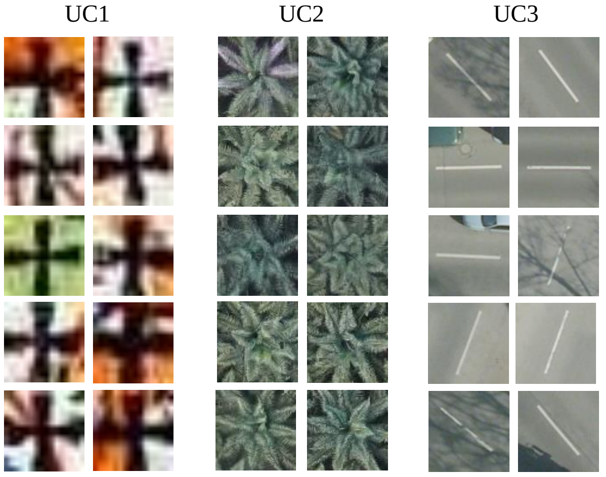

Figure 2. A selection of samples used for template generation

Perhaps the most straightforward way of generating a template is averaging all the samples, and this is the basic approach we used (see e.g. Figure 3/UC1).

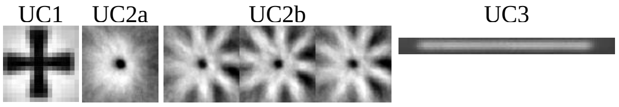

Figure 3: The templates generated for the different use cases. For the detection of palm trees, the mean template (UC2a) misses characteristic aspects of palm trees and a template group (UC2b) was used instead.

However, when samples show considerable variation (like in the example of palm trees) a simple mean template does not capture the characteristic aspects of the samples (Figure 3/UC2a). To address this, we used a variant of factor analysis to extract more than one template. The variety in the sample space was thus captured not by a single template, but by several distinct templates, which we refer to as *template group* (Figure 3/UC2b).

Another obvious situation where simple averaging would fail is when samples differ in orientation, as they do in the case of pavement markings. To prevent this, we rotated the sample images so they would all be rotationally aligned before averaging them to obtain a single template image (Figure 3/UC3).

### 3.2 Generation of a similarity map

For reasons already discussed in the introduction section, we used the correlation coefficient *r* to quantify the quality of the match between a target and an image region.

A similarity map was generated by sliding the template over the image (sliding window), and computing the correlation coefficient *r* at each location. The result was assigned to the central pixel of the current window. Note that the final similarity map is thus smaller than the original image, with shrinkage depending on the template size.

For the more challenging use cases we applied more than one template to each location. This could be either the different templates of a template group (use case: trees in aerial images of oil palm plantations), or several rotated versions of a single template (use case: pavement markings in aerial images).

### 3.3 Target localization

Once the similarity map is obtained, there still remains the task of localizing the targets. Clearly, a certain similarity threshold must be reached to suggest presence of a target. In addition, we want to avoid finding many “hits” in a very small region. We thus impose two criteria for target presence at a specific location:

a) The similarity map value of the corresponding pixel must exceed a defined threshold $thr_{r.}$ This threshold was set to obtain a small overall error rate while keeping omission and commission errors (see next section) relatively balanced.

b) The match quality of the pixel must represent a local maximum in a neighbourhood corresponding ($d_{min}$ is the radius of this neighbourhood).

### 3.4 Accuracy Assessment

To evaluate the quality of the template matching algorithm we selected for each use case a test image for which we manually created a ground truth. A target counted as *correct detection* when the distance to a ground truth element was below a defined tolerance. This tolerance was set to a value of about 10% to 20% of the template size, depending on the use case. Tolerance was largest for the tree detection (corresponding to 1.5 m on the ground), as the tree centres cannot be allocated precisely, reflecting an inherent noise in the ground truth.

We also evaluated the number of *misses*, i.e., locations marked in the ground truth, for which no target was identified in the neighbourhood defined by the tolerance. Finally, we counted also *false positives* — targets identified by our algorithm that were absent from the ground truth.

Following standard convention, we used those measures to evaluate the omission and commission errors:

$$Err_{\text{omission}} = \frac{\text{misses}}{\text{misses} + \text{correct detections}} \tag{4}$$

$$Err_{\text{comission}} = \frac{\text{false positives}}{\text{false positives} + \text{correct detections}} \tag{5}$$

Those error rates were used to evaluate the power of our template matching algorithms in the different use cases.

## 4. RESULTS

Parameters used for template generation, template matching and target localization are shown in Table 4, together with the results of the accuracy assessment.

| | UC 1 | UC 2 | UC3 |
|---|---|---|---|
| Parameters | | | |
| # Samples | 278 | 394 | 60 |
| Template size (pxl) | 17x17 | 69x69 | 95x7 |
| $thr_r$ | 0.69 | 0.18 | 0.76 |
| $d_{min}$ (pxl) | 13 | 50 | 60 |
| Tolerance (pxl) | 2 | 12 | 10 |
| Results | | | |
| Elements in Ground Truth | 127 | 436 | 68 |
| Correct Detection | 107 | 399 | 55 |
| False Positives | 17 | 18 | 14 |
| Misses | 20 | 37 | 13 |
| Commission Error | 0.137 | 0.043 | 0.203 |
| Omission Error | 0.158 | 0.085 | 0.191 |

Table 4. Parameter settings and accuracy assessment results

Error rates vary between around 6% for the detection of oil palm trees and around 20% for the detection of pavement markings. Image thumbnails highlighting the detected targets are shown in Figure 5.

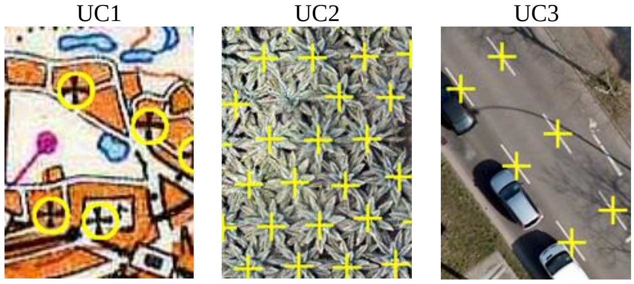

Figure 5: Image thumbnails with identified target locations.

## 5. DISCUSSION

It is difficult to directly compare our results to the literature, as obtained accuracies do not only reflect the power of the algorithms applied. Results are also heavily influenced by the testing method, and of course, the nature and quality of the test images. Nevertheless, it is noteworthy that in spite of the conceptual simplicity of our approach, the detection quality achieved here for tree detection lies well within the range of other published work on the detection of palm trees (Malek, Bazi, Alajalan, AlHichric and Melgani, 2014; Srestasathiern & Rakwatin, 2014; Manandhar, Hoegner & Stilla, 2016).

While we have set the threshold $thr_r$ in this study to achieve a low and balanced error rate, software-supported workflows will typically warrant a different type of threshold adjustment. This is because the omission error is usually associated with a higher cost than a commission error. Consider, for instance, a manual quality assurance step: finding additional targets in an image is by far more time consuming than “clicking away” a selection of false positives.

Clearly, the specific implementations of template matching we used here need not be the final word on the optimal approach. For example, in the symbol detection it appears that not all pixels in the cross template are created equal — some contribute more to the actual pattern than others. Indeed we found that applying a mask which prevents certain pixel locations to enter the correlation computation can lower error rates considerably, from around 15% to around 10% in the symbol detection studied here. Furthermore, we observed significant improvement of oil palm tree detection when pre-processing the data layers in a carefully designed manner, before applying template matching.

But even when taking further improvements into account, it appears questionable that template matching will be able to provide an answer to all pattern matching tasks. In our view, template matching simply represents a useful first step. It enables identification of target regions that can then be processed further by classical object-oriented approaches (Baatz, Hoffmann & Willhauck 2008).

In summary, we have applied template matching to three different use cases of practical relevance. We found that template matching can successfully isolate many targets, in spite of its conceptual simplicity. To leverage this power, template matching capabilities are now made available in eCognition Developer, where the user can choose from a variety of convenient algorithms and tools for generating and applying templates.

## REFERENCES

Baatz, M., Hoffmann, C., and Willhauck, G. (2008). Progressing from object-based to object-oriented image analysis. *Object Based Image Analysis*, ed. Blaschke, T., Lang, S. and Hay, G. New York.

Bradski, G. and Kaehler, A. (2008). “Histograms and Matching” in *Learning OpenCV: Computer Vision with the OpenCV Library* (1st ed.), O'Reilly Media, Inc., pp. 193–221

Malek, S., Bazi, Y., Alajlan, N., AlHichri, H. and Melgani, F., 2014. Efficient framework for palm tree detection in uav images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12), pp. 4692–4703.

Manandhar, A., Hoegner, L. and Stilla, U., 2016. Palm tree detection using circular autocorrelation of polar shape matrix. (abstract accepted at ISPRS 2106)

Nixon, M. and Aguado, A. 2012. Template matching. In: *Feature Extraction & Image Processing for Computer Vision* (3rd ed.). Academic Press, pp. 222-235

Srestasathiern, P. and Rakwatin, P., 2014. Oil Palm Tree Detection with High Resolution Multi-Spectral Satellite Imagery . *Remote Sensing*, 6, pp. 9749-9774