Chapter 4

TADD Technical Report for Oct 2013

4.1 Abstract

This report is twofold. One the one hand, we summarise our new improvement on the current TADD system, which is based on the well-known Scale Invariant Feature Transform (SIFT) algorithm. On the other hand, we propose an efficient method for detecting barcode region within a food tray from a complicated background and explain the related technical details.

4.2 Review

In our last report, we show an improved TADD system which can automatically, intelligently and efficiently partition a food tray image into background and foreground regions, even if the background is inconsistent, non-black or with noisy objects. After such a background segmentation, the system visualises the position and the orientation of the tray by detecting the contour and the bounding box of the tray.

The segmented tray region is actually the region of interest (ROI) where we can (i) more efficiently implement other demanding operations within it (e.g., tickbox detection, label detection and seal detection, etc) since we just need to

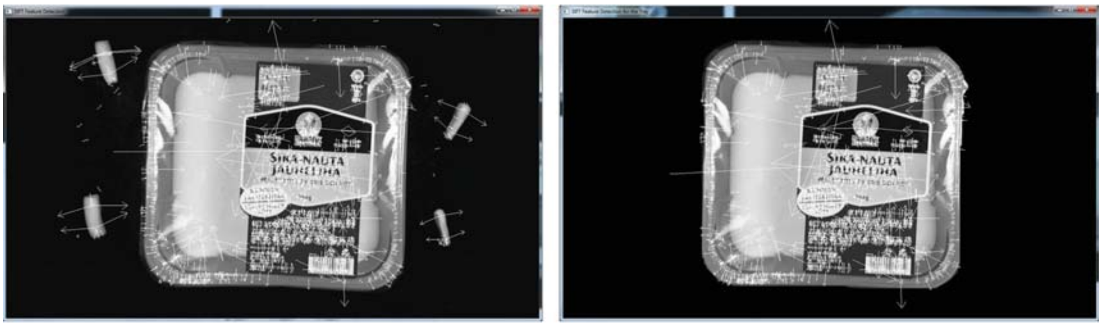

Figure 4.1: SIFT feature detection. Left: without background segmentation; Right: with graph-based background segmentation

handle a smaller number of pixels by ignoring the pixels in the background region and (ii) more reliably implement these operations because it is highly possible that there are some noisy pixels in the background region which have an effect on the implementation of the operations in the tray region.

Thus in this work, we apply SIFT algorithm [Lowe, 2004] on the input images and integrate the codes with the current TADD system. The detected features can be potentially used for various tasks in the future development of the TADD system.

4.3 SIFT feature detection

As one of the most famous and widely-used computer vision algorithms, SIFT algorithm transforms image data into scale-invariant coordinates relative to local features. Typically, it can generate a large number of features that densely cover the image over the full range of scales and locations. Each feature has three visualised properties: scale, orientation and location. In fact, typically, each feature corresponds to a vector of high dimensionality (which represents the image gradients within a local region of image where the size of the local region is defined by a Gaussian) and thus the SIFT features are very distinctive and can be matched reliably. Experiments implemented in [Lowe, 2004] demonstrate that SIFT features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint.

Fig. 4.1 shows two images of SIFT features produced by the improved TADD system. In Fig. 4.1, we use an arrow to represent a detected SIFT feature. The starting point of the arrow denotes the location of the SIFT feature; the size of the arrow denotes the scale of the feature; and the orientation of the arrow is the orientation of the feature. It can be observed that the background segmentation effectively removes the features detected on the noisy objects within the background region. This is particularly important for the following object recognition (tickbox detection, label detection, etc) since these noisy objects could have similar features to those within the ROI and they can thus lead to unreliable matching.

4.4 Barcode region detection

Barcode region detection, or more accurately, automatic barcode region detection in complex background, is a technique very different from the current prevalent barcode reading techniques. Barcode reading techniques usually rely on devices with laser-scanners and assumpts that the barcode region are dominant over the entire image. However, this is not the case in our project. As shown in the top row images of Fig. 4.2, the barcode regions are quite small in size compared to the entire image. To find the barcode region in complex image background efficiently and reliably, we propose a new algorithm. The basic idea is to capture the difference between the horizontal gradient and the vertical gradient of the image. Then the barcode region should correspond to the largest differences due to the specific shape and intensity distribution within a barcode region. The details of the algorithm is shown in Algorithm 1.

According to Algorithm 1, the proposed method for barcode region detection is quite fast. Also, as demonstrated in Fig. 4.2 (b), this method has some degree of robustness to rotation. If the rotation angle is quite large (although it is extremely rare in our project), we perhaps need to first rotate the tray according to its orientation information captured by the background segmentation.

One significant limitation of our current method is that the boundary of the barcode region is typically not well defined. One solution to this problem is to give a tolerance to the detected region (e.g., ± 50 horizontally and vertically) and

Algorithm 1: Barcode region detection in complex image background

Data: A 2D image I
Result: A saliency map I
begin
  if *I is an RGB image* **then**
    convert I into an intensity image;
  Compute the horizontal and the vertical gradients of I, writen as I_x and I_y;
  Measure the difference $D = abs(abs(I_x) - abs(I_y))$ where abs denotes the computation of taking the absolute value of each element in a matrix;
  Construct an average filter kernel H with the size of 60×60;
  Do convolution using H and the difference map D: $C = H \otimes D$;

then to calculate the rectangular bounding box of the region. The image content within the bounding box is in fact a new image where the barcode region should be dominant and thus general barcode readers can work well with it.

Certainly there are some other more advanced techniques which can more accurately detect the boundary of the barcode region. For instance , [Fang & Xie, 2010] achieved this through region-based image analysis. Nonetheless, our method is more efficient. Typically it can detect the barcode region in a complex 3000×2048 image within 1.5 seconds on a standard PC (Duo core, 3GHz CPU, 2GB RAM).

4.5 Future work

Since the current TADD system can correctly detect SIFT features in the tray region, tasks like tick box detection can be done by matching the detected SIFT features with those saved as template features. Because SIFT features are invariant to image rotation, this strategy should be robust as long as the template features can be extracted. Once the feature matching is implemented, the system can estimate whether the positions of the tickboxes are correct or acceptable by checking the locations of the features within those tickboxes.

Another way to do semantic object recognition is to decompose the tasks.

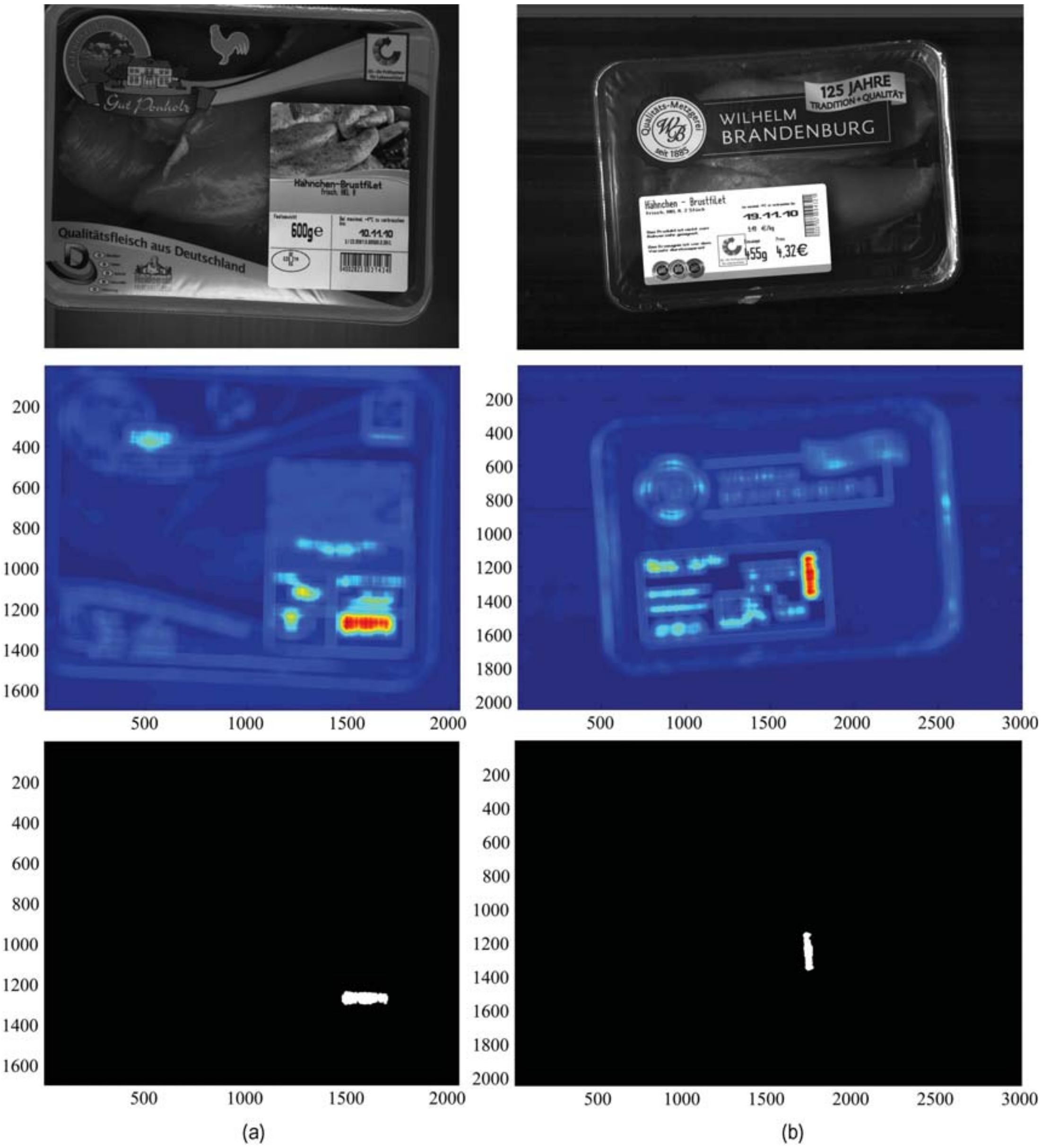

Figure 4.2: Results of our method for barcode region detection. (a) A horizontal barcode (b) A vertical barcode. Top row: Original input images; Middle row: Colour maps (C in Algorithm 1) produced by our method where warm colour represents strong response to potential barcode region; Bottom row: Binarisation of the colour maps which clearly locate the barcode regions (The binarisation threshold is set as $T = 0.7 \times C_{max}$ where C_{max} is the maximum value of C).

At each step, the system just focuses on one specific tickbox. For instance, the system can first read in the image with a blank tray (with patterns but without tickboxes) and then read in a tray with only one tickbox. Next, by comparing the difference of SIFT features detected in the two trays, the system can know what the tickbox is. The potential drawback for the latter method is its low speed. It needs to perform several rounds of SIFT algorithms since each newly added tickbox or label will change the content of the tray surface and consequently change the result of SIFT detection. Considering that SIFT does not have a real time performance, the entire processing time might be too long. Therefore, perhaps the next step is to find and test some other feature detection algorithms (such as SURF: speeded-up robust features [Bay *et al.*, 2006]) faster than SIFT using the specific tray images taken by our TADD system.

Another interesting work is to improve our current barcode region detection algorithm. In particular, we hope to detect the boundary of the barcode region more accurately at a low computational cost.