Monthly Report– November 2013

Hossein Malekmohamadi

January 13, 2014

Contents

List of Figures

Chapter 1

Introduction

This is the November report for the project Trainable Vision-based Anomaly Detection and Diagnosis (TADD). The aim of this project is to deliver a software system capable of detecting and recognising intelligently different anomalies in food contents and packages, e.g. potato or food container. Currently, this system can do semantic segmentation, barcode region detection and foreground extraction based on some image processing techniques and the Adaboost learning. These features can be used in different parts of quality control task which is an industrial aim in TADD. However, further improvements to TADD can be accomplished by bringing 3D vision systems and enhancing learning capabilities. In this report, the focus is on candidate techniques in machine vision and machine learning for TADD systems. This is covered in Chapter 2 where we investigate photometric stereo as the candidate 3D vision system and random forests and deep learning as the candidate learning algorithms. Finally, Chapter 3 is a summary of this report.

Chapter 2

Candidate Techniques for TADD

Here, an overall review is given for further improvements in TADD systems both in machine vision techniques and learning algorithms. For the vision part, there are several possibilities from expensive solutions to cost-effective ones. We had visitors from Cognex to show us their scattering laser imaging system. We plan to visit Sick UK to see their demo in the near future. However, another candidate to obtain more details of potato and food packaging is to build a photometric stereo rig. Photometric stereo can be built cost effectively and it will reveal 3D dense information of surfaces under investigation. Here, we give a brief overview on photometric stereo algorithm and some initial results for the data gathered in Kingston university. Furthermore, improvements to TADD system can be accomplished in the machine learning side of the project with topics like decision forests or deep learning. In this chapter, a brief introduction to random forests and deep learning algorithms are given.

2.1 Photometric Stereo

To recover highly dense surface information, we employ the Photometric Stereo (PS) technique which is based on high-resolution image acquisition using a number of different light source directions and a single 2D camera as shown in Figure 2.1. PS is used to estimate the dense normal map of a potato or a package, from which the gradient field is then computed. This enables robust separation of the 2D (albedo) and 3D (height map) components of the potato or package at a significantly high level of accuracy.

PS has been available for many years but only recently has affordable technology become available for improving camera resolution and algorithm execution speed, allowing synchronised light switching at the fast rates needed to avoid inter-frame motion. We use 4 photometric images (4 source PS system) to recover the 3D surface information from potato. While fewer photometric images could be sufficient, a 4 source PS system allows better recovery in the presence of highlights and shadows . We assume a Lambertian reflectance model together with intensity variation at each pixel to estimate local surface orientations. The integration of these surface orientations results in a highly detailed estimate of the surface geometry. The captured images per each LED illumination are shown in Figure 2.2.

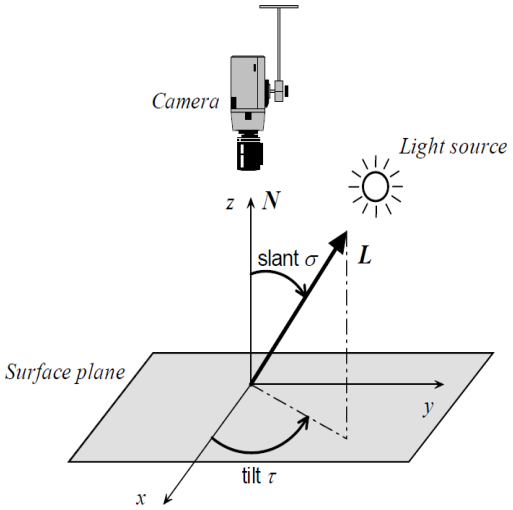

Figure 2.1: Typical PS Set-up and the Surface Under Investigation

Figure 2.1 shows the set-up for the capture of photometric images. τ denotes tilt angle and represents the angle that an illuminant vector projected onto the surface plane makes with *x-axis*. The slant angle, σ, represents the angle the illuminant vector makes with the *z-axis*. The function to represent the photometric images at individual illumination direction is as follows:

$$i(x,y) = \frac{-p(x,y)cos\tau sin\delta - q(x,y)sin\tau cos\delta + cos\delta}{\sqrt{p(x,y)^2 + q(x,y)^2 + 1}} \tag{2.1}$$

It is important that the three photometric images provide enough change in illumination gradient so that the partial derivatives for the surface (p and q) can be estimated (Woodham, 1980). Based on the assumption that

Figure 2.2: 4 Captured Images

Lambert's law is preserved, we model these reflectance functions for the 4 PS images as

$$i_d(x, y) = \rho(l_d n) \qquad \forall d \in \{1, 2, 3, 4\} \tag{2.2}$$

With known reflectance intensities i_d and illumination directions l_d (which are fixed in the camera coordinate), the unit surface normal at a given position (x, y) in the surface plane is given by $n = \frac{(p,q,-1)^T}{\sqrt{p^2+q^2+1}}$ with p and q being the partial surface derivatives. Having surface normal components, it is time to reconstruct the surface as shown in Figure 2.4. Among different integration algorithms to generates integrable surface from gradients, we apply Frankot-Chellappa method(Frankot and Chellappa, 1988).

Following the extraction of surface gradients, we need to derive some features to be fed in to machine learning algorithm. One candidate for the feature extraction stage is to use the partial derivatives of the 3D surfaces (p,q) to extract surface curvature information. We then compute the first order derivatives along the *x-axis* and *y-axis* for p and q: p_x, q_y, p_y. These derivatives are used to compute the surface curvature. The Gaussian curvature, K, is given by

$$K = \frac{p_x q_y - {p_y}^2}{(1 + p^2 + q^2)^2} \tag{2.3}$$

Mean curvature is

$$H = \frac{(1 + p^2)q_y + (1 + q^2)p_x - 2pqp_y}{2(1 + p^2 + q^2)^{\frac{3}{2}}} \tag{2.4}$$

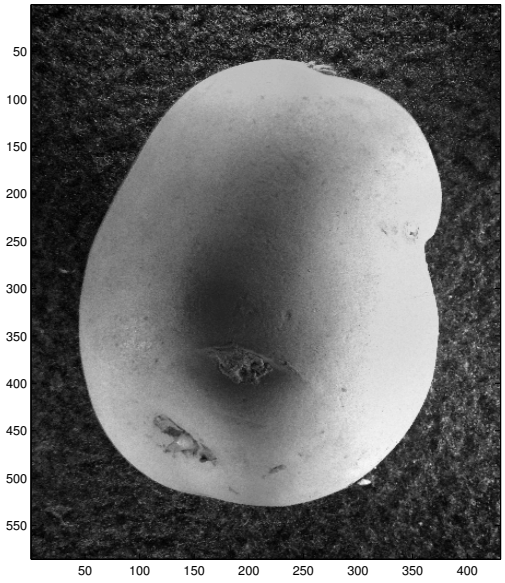

Figure 2.3: Albedo Image from Kingston Data (Code is implemented in MATLAB)

The principle curvatures $k_{1,2}$ are

$$k_{1,2} = H \pm \sqrt{H^2 - K} \tag{2.5}$$

The resultant principal curvatures as shown in Figures 2.5 and 2.6 are used to compute a shape index (SI) map for each 3D surface. This provides a characterisation of topography using a continuous angular representation. Shape index is defined as

$$S = \frac{2}{\pi} arctan(\frac{k_2 + k_1}{k_2 - k_1}) \tag{2.6}$$

In Figure 2.7 we illustrate the shape index maps of a potato from the Kingston dataset[1]. The shape index maps presented appear to be able to capture significant amount of 3D surface texture information. From the shape index map, we can clearly see that the texture information of blemish area.

[1]Images are captured by Dr Vasileios Argyriou at Kingston University in November 2013

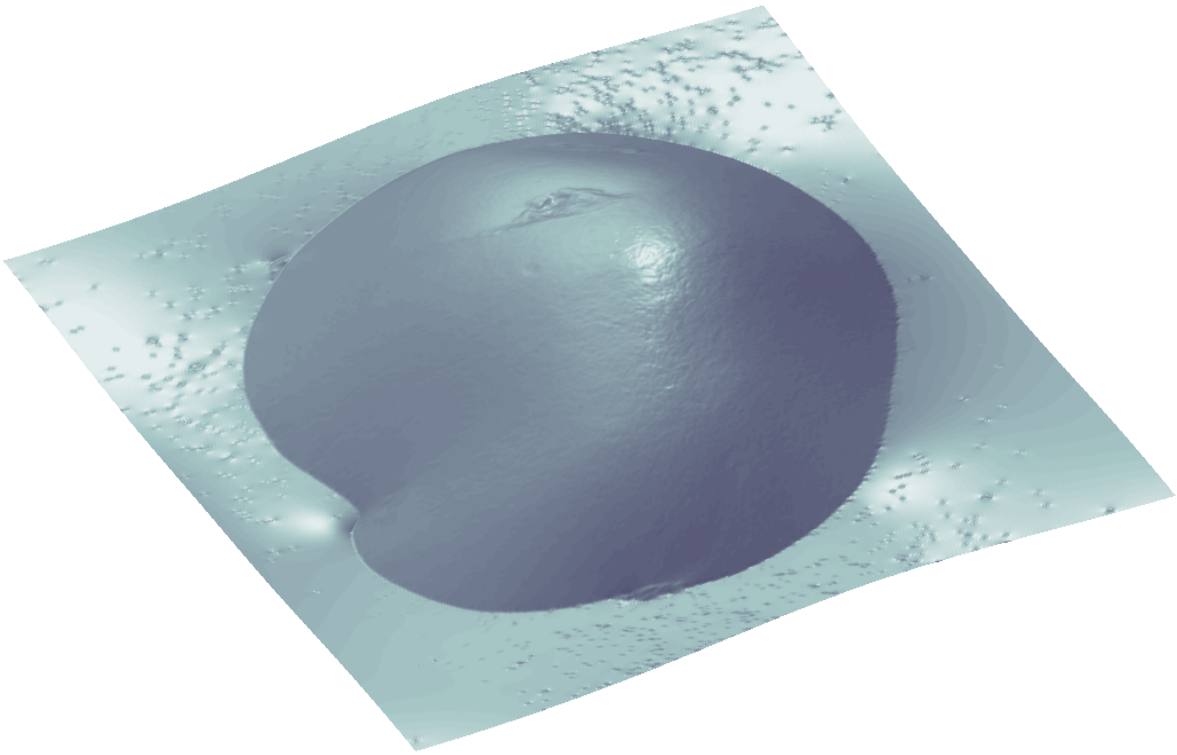

Figure 2.4: Reconstructed Surface Height Map (Code is implemented in MATLAB)

We have implemented the PS technique and 3D surface information in MATLAB. In this implementation, the inputs are 4 images as shown in Figure 2.2 and the outputs are surface normals, height map, principle curvatures and shape index. Briefly, the PS technique can be applied in our current TADD system to model 3D objects. Its applications are various from potato disease detection to seal integrity identification. However, the choice for white LEDs as point source lights and a high resolution camera should be considered within the next quarter.

2.2 Potential Improvements to the Learning Section of TADD

Current system benefits from Adaboost classification algorithm however we are looking for increasing speed and accuracy. In this part, we give a brief overview for two learning candidates for this project: 1) Decision forest and 2) Deep learning. Both techniques have absorbed a lot of attention in the

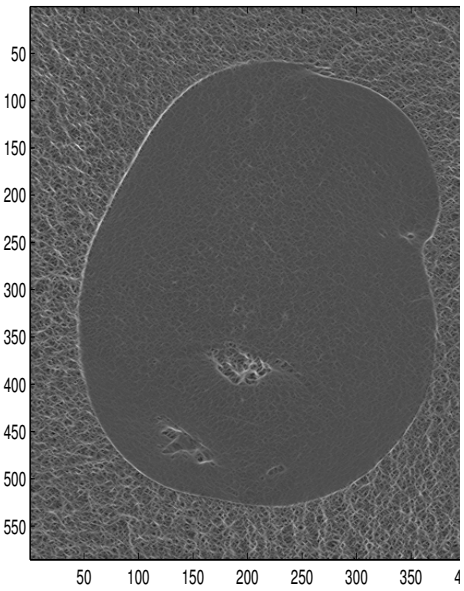

Figure 2.5: Principle Curvature 1 (Code is implemented in MATLAB)

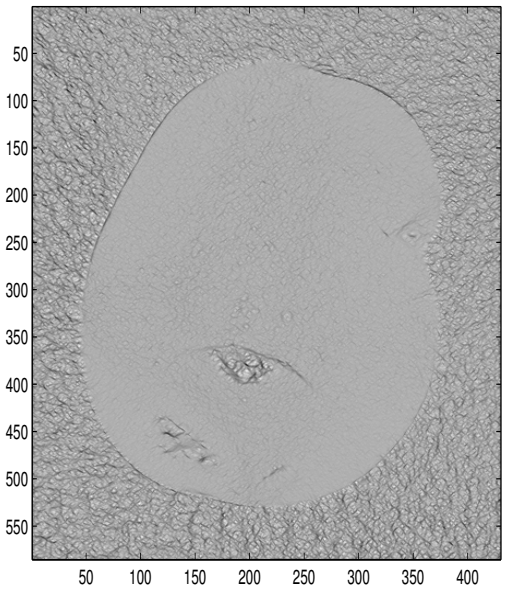

Figure 2.6: Principle Curvature 2 (Code is implemented in MATLAB)

research community in the last decade. Decision forests have light-weight implementations and operate fast. They are suitable for tasks like segmentation. Deep learning is to able learn multiple levels of representation and abstraction that help to make sense of data like images of sealed trays with food content.

- Decision Forests are amongst estimative methods that can classify data. Decision forests consists of several decision trees (DTs). A decision tree uses training data to construct a classifier, which models the relations between data attributes and class targets. The learning process of decision trees is an inductive process that uses particular facts from data attributes to make more generalised conclusions (Witten et al., 2011). Once it has learnt, the decision tree is able to estimate the output based on new inputs. The final model is made of a set of Boolean tests, which is less complex than a one-stage classifier for implementations and real-time tasks. Representing a decision tree can be performed two ways. In the first method, a decision tree is modelled with the influence diagram (ID) through nodes and arcs. There are three types of nodes: 1) decision nodes represent points at which the

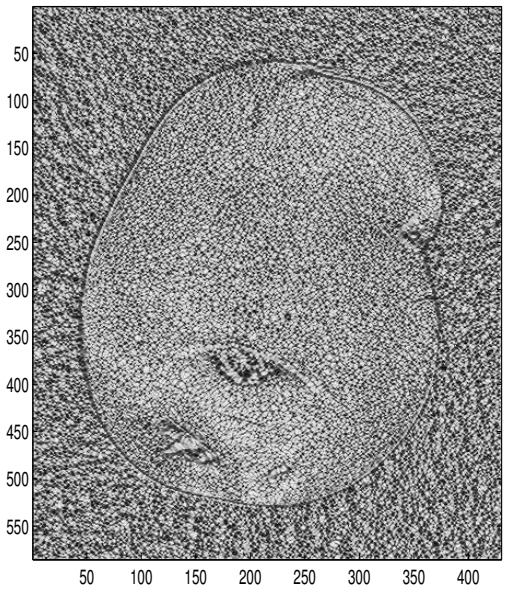

Figure 2.7: Shape Index Map (Code is implemented in MATLAB)

system has to make a choice of one alternative from a number of possible alternatives; 2) chance nodes represent points at which chance, or probability, plays a dominant role and reflect alternatives over which the system has no control; and 3) terminal nodes represent the ends of paths from top-down through the decision tree. Both decision and chance nodes are interior nodes and can have child nodes, but terminal nodes have no child nodes. Arcs are in three types: functional which ends in a decision node, conditional and informational arcs. Second, mathematically, decision trees are If- Then rules, where different rules apply sequentially to get a specific output.

There are different algorithms for the learning process of a decision tree. Common to all the algorithms is that a decision tree chooses the best attributes to split the instances based on some measures like information gain or gini index (Breiman, 1993). If the stopping criterion is met, the splitting attribute represents a decision node. The stopping criterion is when there is no attribute or training sample left or all samples have the same target value. Due to numerous attributes and targets, handling error in training a decision tree is performed mainly by pruning; e.g. if two decision trees employ the same attributes and have the same accuracy, the one with fewer leaves (nodes) is selected for the final model. Pruning can be performed either as

pre-prune (forward pruning) or as post-prune (backward pruning). In pre-pruning, it is achievable to stop adding new attributes and hence hinder over sizing through some measures like information gain. Post-pruning is a recursive task that it waits for the full decision tree has built and then prunes the attributes by two major concepts: 1) sub-tree replacement; and 2) sub-tree raising. Post-pruning is an inherent feature of many decision tree learning algorithms to obtain optimum tree with an acceptable rate of accuracy.

There are different methods for learning process of decision trees like C4.5, ID3, and CHAID (Witten et al., 2011). In this research C4.5 is selected. Based on some information theory concepts like entropy, C4.5 builds decision trees from a set of training samples considering error rate and speed. C4.5 is a greedy, recursive and top-down algorithm for decision trees and has an appropriate combination of speed and accuracy. C4.5 starts with the root node where it selects an attribute more capable of separating the instances. Following the root node, C4.5 decides on other attributes recursively until it meets the stopping criterion and where all training instances are classified. The selection measure is based on information theory and that is information gain. Lastly, post-pruning is accomplished to obtain the optimum tree with fewer leaves and higher accuracy.

Random forests is another machine learning techniques which consists of several weaker classifiers (decision trees) (Breiman, 2001). Weaker classifiers can be combined by voting (for classification) or averaging (for regression) to form strong classifier. It is applicable for classification, regression, clustering, density estimation, outlier and anomaly detection. A random forest has some specific rules for tree growing and combination, self-testing and post-processing (Breiman, 2001). Consequently, they can deliver higher prediction accuracy for even very large volume of data and unlike decision trees they are resistant to the over-fitting problem.

-Deep Learning is a set of machine learning algorithms that attempt to learn in multiple levels, e.g. neural network layers, corresponding to different levels of abstraction (Ng, 2013). In multi-level learning, higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. In deep learning, an instance, e.g. an image, can be represented in several ways depending on the extracted features, but some representations make it easier to learn tasks of interest. Features extracted based on SIFT, Spin image, HoG, RIFT, Texton and GLOH are widely applied to represent vision features. For the first stage in deep learning which is feature extraction, each image is represented

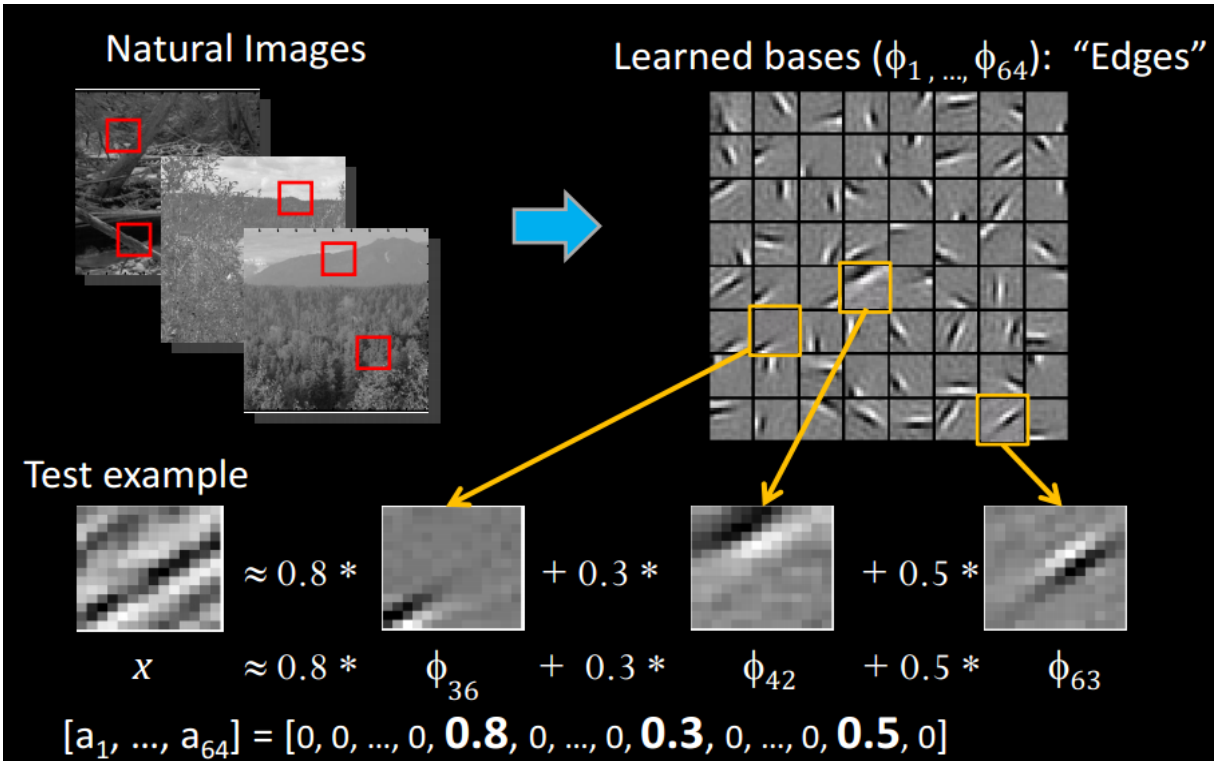

Figure 2.8: Sparse Coding Illustration (Image is taken from (Ng, 2013))

by a dictionary of features, e.g. edge detection in a patch of image, using sparse coding. As shown in Figure 2.8, images can be represented in higher level than the raw pixels. In the next higher levels, combination of edges are mixed to represent object parts, e.g. eye. This hierarchical set-up can be implemented with neural networks to classify image contents, e.g. car, chair, plane, etc.

In the TADD system, depending of application layer, we can implement any of the above-mentioned algorithms to increase accuracy, speed and robustness. For example, deep learning can be helpful in semantic segmentation or random forests can be applied in blemish detection.

Chapter 3

Summary and Future Work

In this report, we investigated several potential techniques to be implemented in the TADD systems. These techniques are either for image acquisition and set-up like adding 3D vision or enhancing the learning side of the software package. For the vision systems, there are some expensive sensors which can be bought within project budget like the sensor we had been shown by Cognex. These sensors recovers surface and then based on that we can estimate surface normals. However in some cases, we need to investigate 3D surface normals first and then, if necessary recover the height map. This can be achieved by basic mathematical model called photometric stereo. In this method, surface normals are extracted based on a set-up containing at least three LED light sources and a high definition 2D camera. For the learning part of TADD, we can enhance the speed and accuracy by other algorithms. In this report, deep learning and random forests are presented. Deep learning can be more suitable for tasks like barcode region detection or date detection in different languages. Decision trees can be more useful in potato blemish detection where speed is more important.

The aim in the next quarter of this project is to investigate the aforementioned techniques. For this purpose, initial studies and research on Kingston University data have been done and we are investigating the building of our own PS rig for 3D surface reconstruction of potato and food containers. Moreover, manipulating the current codes in TADD, we can have access to data before current learning algorithm (Adaboost) and then we can test random forest learning.

Bibliography

Breiman, L. (1993). *Classification and regression trees*. CRC press.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Frankot, R. T. and Chellappa, R. (1988). A method for enforcing integrability in shape from shading algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):439–451.

Ng, A. (2013). Deep learning.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Elsevier.

Woodham, R. J. (1980). Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139–191139.