POINT OF SALE SYSTEM, SCANNER, AND METHOD

FIELD

[0001] The present specification here relates in general to point of sales systems.

BACKGROUND

[0002] Barcode scanners are known and widely used to identify items. For example, items in a store can be labelled with a barcode for identification purposes. The barcode typically comprises a machine readable section representing a product identifier, such as a universal product code. The barcode is generally not placed on the product during manufacture or production, but added to the product, such as on a sticker and thus requires an additional step in the supply chain prior to display in the store. When the item is to be identified, such as at a point of sale, the barcode is scanned and a computer device automatically identifies the product based on its unique barcode.

[0003] Some items, particularly products that are not pre-packaged or that are sold in bulk, such as produce, grains, flours, aggregates (e.g., sand, gravel, etc.), liquids (e.g., honey), nails and other bulk hardware, lumber, and the like, generally either do not have a barcodes attached due to practical reasons or are initially given barcodes but are subject to handling that causes barcodes to become detached. Instead, such items generally use another product identifier such as a price look-up code unique to the item which is typically done by manually entering the code into the computer device using a keyboard or keypad.

SUMMARY

In accordance with an aspect of the invention, there is provided a point of sale system, scanner, and method for using the point of sale system and scanner.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Reference will now be made, by way of example only, to the accompanying drawings in which:

[0005] Figure 1 is a schematic representation of a point of sale system in accordance with an embodiment;

[0006] Figure 2 is a schematic representation of a scanner in accordance with an embodiment;

[0007] Figure 3 is a schematic representation of the scanner in in figure 2 showing internal components;

[0008] Figure 4 is a flow chart of a method in accordance with an embodiment;

[0009] Figure 5 is a flow chart of a method in accordance with another embodiment; and

[0010] Figure 6 is a schematic representation of a point of sale system in accordance with another embodiment.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0011] Referring to Figure 1, a schematic representation of a point of sale system is generally shown at 50. It is to be understood that the point of sale system 50 is purely exemplary and it will be apparent to those skilled in the art that a variety of point of sale systems are contemplated. In the present embodiment, the point of sale system 50 includes a terminal 55, a scale 60, and a scanner 65.

[0012] The terminal 55 is generally configured to carry out a point of sale transaction. The terminal 55 is not particularly limited. In the present embodiment, the terminal 55 is a cash register, or any other type of computing device capable of conducting a point of sale transaction. The terminal 55 includes an input device for receiving input from a user, such as a keyboard or keypad. The input is not particularly limited and can include data identifying a product such as keystrokes representing a product identifier, such as a universal product code (UPC) or a price look-up code (PLU), or selection of a product from a display screen. In other embodiments, the terminal 55 can be a personal computer, a personal digital assistant, a tablet computing device, cellular phone, or laptop computer configured to carry out similar functions.

[0013] Furthermore, the terminal 55 can also be in communication with a central server (not shown) in other embodiments. It is to be appreciated by a person of skill in the art with the benefit of this description that in embodiments where the terminal 55 is in communication with a central server, the steps to be carried out for a point of sale transaction can be divided between the central server and the terminal 55, or completely carried out by the central server.

[0014] The scale 60 is generally configured to measure weight of a product, such as produce or other products sold in bulk by weight. The scale 60 is not particularly limited and can be any type of scale configured to communicate with the terminal 55 and provide the terminal 55 with data corresponding to a weight. In the present embodiment, the scale 60 is a digital scale configured to continuously send readings in a unit of measure such as grams to the terminal 55. In other embodiments, the scale 60 can be configured to send a reading only upon receiving a request for a reading from the terminal 55 or when user input is received at the scale.

[0015] It is to be re-emphasized that the point of sale system 50 described above is a non-limiting representation only. For example, although point of sale system 50 includes a separate scanner 65 and scale 60, it is to be appreciated that the scale can be omitted for some applications and that products requiring a weight can be weighed by an independent separate scale manually entered into the terminal 55. As another example of the variation, the scale 60 can be combined with the scanner 65 such that products can be scanned at the same time as being weighed. As yet another example of a variation, the terminal 55, scanner 65, and scale 60 can be combined into a single unit.

[0016] Referring to figure 2, the scanner 65 is shown in greater detail. It is to be understood that the scanner 65 is purely exemplary and it will be apparent to those skilled in the art that a variety of scanners are contemplated including other embodiments discussed herein. The scanner 65 includes a head 70 having a light source 75, targeting aids 80, and a sensor 85. The scanner 65 also includes a handle 90 and an actuator 95. In the present embodiment, the scanner 65 is wireless and communicates with the terminal 55 via a wireless link, such as a BLUETOOTH connection, a WIFI connection. In other embodiments, the scanner 65 can be wired to the terminal 55, such as using a USB connection, RS232 connector or other suitable type of connector. In the present embodiment, the scanner 65 captures image data and transmits image data to the terminal 55. The the scanner 65 may also capture bar codes and transmit PLUs or other codes to the terminal 55. Further, it is to be appreciated that in other embodiments, the scanner 65 can transmit other types of product identifiers or transmit raw data captured by the sensor 85.

[0017] In the present embodiment, the head 70 is generally configured to house the internal components of the scanner 65. In particular, the head 70 provides a durable cover to protect the internal components from the external environment. The head 70 is not particularly limited and can be varied depending on the specific application of the scanner 65. For example, in applications where the scanner 65 is to be primarily used indoors, such as in a retail store, the head 70 can be designed to protect the internal components from damage due to falls or other gentle impact that may be associated with such use. Alternatively, if the scanner 65 is to be used outdoors, such as at a market where the scanner 65 may be exposed to the elements, the head 70 can be modified to be more durable to resist the elements such as by using a waterproof design. The head 70 is typically constructed from a rigid material. Some examples of suitable materials for the head 70 include plastics, metals, composites and other materials commonly used for portable scanning devices. The manner by which the head 70 is formed is not limited and can include 3D printing, injection molding, or other suitable processes.

[0018] The head 70 includes a light source 75 for illuminating the target product. It is to be appreciated by a person of skill in the art with the benefit of this description that the light source 75 is not particularly limited and can be omitted when the intended use is to be in an area having sufficient ambient light to reduce manufacturing costs. In the present embodiment, the light source 75 is a light emitting diode. In other embodiments, the light source 75 can be a laser or a flash bulb. In the present embodiment, the light source 75 provides lighting to the user in environments where the ambient light is low.

[0019] The head 70 also includes optional targeting aids 80 generally configured to project an image to assist in the aiming of the scanner 65. In the present embodiment, the targeting aids 80 project a bracket configured to identify the area from which the sensor 85 collects data. It is to be appreciated that the targeting aids 80 are not particularly limited and need not be a bracket. For example, targeting aids 80 can be modified to be a single circular or oval shaped projection.

[0020] The head 70 also includes a sensor 85 configured to capture data associated with the target product to identify the target product. The sensor 85 is not particularly limited and can include a variety of sensors capable of capturing data for identifying the target product. In the present embodiment, the sensor 85 is a camera configured to capture images in the visible spectrum. Although the sensor 85 can be a generic digital camera, the present embodiment includes a camera that does not automatically adjust the color, hue, saturation or tint of the captured image. In the present embodiment, the sensor 85 can be configured to automatically adjust the gain control for the captured image.

[0021] It is to be appreciated that the sensor 85 is not limited and can be modified. For example, the sensor 85 can be modified to include two digital cameras configured to capture a 3D image and/or to estimate the distance a target product is away from the sensor 85. For example, the two digital cameras can be positioned about 2-5 centimeters apart to capture a stereo image of the target product. As another example of a modification, the sensor 85 can also be substituted with a camera configured to capture images in the infrared spectrum. In further embodiments, it is to be appreciated that the sensor 85 is not necessarily a camera and can be a simple color sensor, such as a single point RGB sensor to detect colors within a target area, or a spectroscope to analyze the chemical makeup of a target product.

[0022] The handle 90 is not particularly limited and is generally configured to be held by a human hand during use. In the present embodiment, the handle 90 is extends from the head 70 and is formed from the same unitary body. In other embodiments, the handle 90 can be attached to the head 70 using various fastening methods. For example, the handle 90 can be manufactured from a different material, such as a resiliently deformable material to improve comfort. It is to be appreciated with the benefit of this description that the handle 90 can also be hollow to house some of the internal components of the scanner 65.

[0023] The actuator 95 is generally configured to operate as a trigger. In the present embodiment, the actuator 95 is biased toward an outward position. Upon depression of the actuator 95, such as with a finger, a contact can be made to provide a trigger to the scanner 65. The functionality of trigger is not particularly limited. For example, the actuator 95 can be used to capture data from the sensor 85 for further analysis. Alternatively, the actuator 95 can be used to operate the targeting aid 80 and/or the light source 75. It is to be appreciated that the actuator 95 is not limited and that the actuator 95 can be positioned at another location. It is also to be appreciated that the scanner 65 can include more than one actuator 95 or the actuator 95 can be omitted in applications were user input is not required, such as if the scanner continuously captures data using the sensor 85, for example, every 100ms. Furthermore, the actuator 95 can also be modified to be another type of input mechanism such as a touch key.

[0024] It is to be re-emphasized that the scanner 65 described above is a non-limiting representation only. For example, the scanner 65 can include an optional proximity sensor (not shown) generally configured to measure the distance between the scanner 65 and the target product. It is to be appreciated that the proximity sensor can be used to facilitate data collection by the sensor 85, such as by focusing the digital camera. The type of proximity sensor is not limited and can include the miniature laser rangefinder / LIDAR module, ultrasonic sensor, or infrared proximity sensor. As another example of a variation, the scanner 65 can include an optional barcode scanner to identify target products having a barcode.

[0025] Referring to figure 3, a schematic block diagram showing various internal components of the scanner 65 is provided. It should be emphasized that the structure in figure 2 is purely exemplary and several different implementations and configurations for the scanner 65 are contemplated. The scanner 65 includes an input/output interface 105, a memory storage unit 110, and a processor 100.

[0026] The input/output interface 105 is not particularly limited and can include various network interface devices capable of communicating with the terminal 55 as described above. In the present embodiment, the input/output interface 105 is generally configured to connect to the terminal 55 wirelessly and send a PLU code to the terminal 55 upon identification of the target product.

[0027] The memory storage unit 110 can be of any type such as non-volatile memory (e.g.

Electrically Erasable Programmable Read Only Memory (EEPROM), Flash Memory, hard disk, floppy disk, optical disk, solid state drive, or tape drive) or volatile memory (e.g. random access memory (RAM)). In the present embodiment, the memory storage unit 110 is generally configured to store a database 210, such as a database correlating various characteristics and products with PLU codes. In addition, the memory storage unit 110 is configured to store codes for directing the processor 100 for carrying out computer implemented methods. For example, the codes can include the programming instructions 200 further described below.

[0028] The database 210 is generally configured to correlate image data from the sensor 85 with various products and a product identifier. It is to be appreciated that the database 210 not particularly limited and can be varied depending on the configuration of the sensor 85. For example, the database 210 can include color values associated with some or all products available for purchase using the point of sale system 50. It is to be appreciated that each product may also include more than one color value or a range of color values due to product variations. In some embodiments, the database 210 includes one or more lookup tables. In other embodiments, the database 210 can include a training data set used to train a classification engine using a statistical modelling method to identify a product. Cluster analysis may be used. K-means clustering may be used (https://en.wikipedia.org/wiki/K-means_clustering), In some embodiments, color histogram data of a captured image is compared to histogram data stored in the database 210. Histogram data stored in the database 210 is associated with various products. In such embodiments, errors between histogram data of a captured image and histogram data stored in the database 210, and one or more minimum errors are identified. Products associated with stored histogram data corresponding to the one or more minimum errors are identified as candidate products. Products can be sorted based on error. The scanner 65 can then send an error-ordered list of products to the terminal 55 for further processing or selection by a user. In other embodiment, the scanner 65 can be modified to include a display and input mechanism to allow for a user to select the products on the scanner 65.

[0029] In addition, it is to be appreciated by a person of skill in the art with the benefit of this description that environmental variations can be introduced in the data captured by the sensor 85. For example, the target product can be contained in a transparent or translucent plastic bag, such as the typical case for produce in a grocery store. Another example can be the environmental lighting in the vicinity of the point of sale system 50. The environmental variations can be taken into account in the database 210 with additional entries for each expected variation. For example, in applications where the target product is typically placed in a bag with a specific color tint or translucency, the modified color values can be stored in the database 210 in additional to or instead of the non-modified values.

[0030] Hence, the database 210 can contain a plurality of sets of color histogram data, or similar data, for various products and each product may correspond to more than one set of histogram data. That is, for example, bananas may be represented by four distinct sets of color histogram data: ripe bananas outside bag, green bananas outside bag, ripe bananas inside bag, and green bananas inside bag. All sets of color histogram data for a particular product map to the same code for the product.

[0031] The processor 100 is not particularly limited and is generally configured to execute programming instructions 200 for identifying target products based on data captured by the sensor 85. The manner by which the target products are identified is not particularly limited and will be discussed in greater detail below.

[0032] Referring to figure 4, a method of identifying a target product is represented in the form of a flow-chart and indicated generally at 500. In order to assist in the explanation of the method 500, it will be assumed that the method 500 is performed using the point of sale system 50. Furthermore, the following discussion of the method 500 will lead to further understanding of the point of sale system 50 and its various components, such as the scanner 65. In particular, it is to be understood that in one embodiment, the programming instructions 200 of the scanner 65 direct the processor 100 to carry out the methods discussed below. However, it is to be understood that the point of sale system 50 and/or the method 500 can be varied, and need not work exactly as discussed herein in conjunction with each other, and that such variations are within the scope of the present invention. Furthermore, it is to be emphasized, that method 500 need not be performed in the exact sequence as shown and that various blocks can be performed in parallel rather than in sequence; hence the elements of the method 500 are referred to herein as “blocks” rather than “steps”.

[0033] Block 510 comprises receiving, at the processor 100, data from the sensor 85 corresponding to an image of the target product. The manner by which the sensor data is captured and provided to the processor 100 is not particularly limited. In the present embodiment, the sensor 85 is a digital camera as described above and data captured is the image of the target product.

[0034] Block 520 comprises preprocessing the sensor data to facilitate the analysis of a characteristic of the target product, such as color. The manner by which the sensor data is preprocessed is not particularly limited and can involve various transformations carried out by the processor 100. Continuing with the present example of a captured image of the target product, the programming instructions 200 can direct the processor 100 to normalize the color of the image to reduce the varying effects of illumination. It is to be appreciated with the benefit of this description that by normalizing the color of the image, the accuracy of the subsequent identification steps can be improved. In addition, the preprocessing can involve image blurring, for example, with a Gaussian filter to provide a more consistent sampling of pixel colors and reduce white noise in the image.

[0035] Block 530 comprises identifying the target product using a characteristic from the preprocessed sensor data. The manner by which the identification is carried out is not particularly limited and can involve various methods. In the present embodiment, a specific color from the preprocessed sensor data is used to identify the target product using cluster analysis, such as k-means clustering. For example, for each specific color or range of colors, the database 210 can include a corresponding product and associated PLU code. It is to be appreciated with the benefit of this description that multiple products can have similar colors and that some ranges may overlap. Accordingly, other characteristics from the preprocessed sensor data can be used to improve the accuracy of the identification and the method 500 can be carried our iteratively for each characteristic. For example, the preprocessed sensor data can include data indicative of size, shape and texture. In the present embodiment, a confidence score is calculated for each characteristic and combined using Bayesian statistics. The product is then identified as the one having the highest combined confidence score and the associated PLU code would be sent to the terminal.

[0036] In other embodiments, a predetermined threshold for combined confidence scores can be set and PLU codes associated with products having a combined confidence score above the threshold can be sent from the scanner 65 to the terminal 55, where a user is prompted to make a selection from a small list of possible products. It is to be appreciated by a person of skill in the art with the benefit of this description that by presenting a small list instead of the list of all possible targets, the identification of the target product is substantially easier.

[0037] It is to be re-emphasized that the method 500 described above is a non-limiting representation only and that variations are contemplated. For example, although the method 500 was described to be carried out on the scanner 65, it is to be appreciated with the benefit of this description, that some or all steps can be carried out at the terminal 55 or at another server. For example, in embodiments, where the scanner 65 sends raw data from the sensor 85, the method 500 can be carried out at the terminal 55 or at another server.

[0038] Referring to figure 5, another method of identifying a target product is represented in the form of a flow-chart and indicated generally at 600. In order to assist in the explanation of the method 600, it will be assumed that the method 600 is performed using the point of sale system 50. Furthermore, the following discussion of the method 600 will lead to further understanding of the point of sale system 50 and its various components. It is to be understood that the point of sale system 50 and/or the method 600 can be varied, and need not work exactly as discussed herein in conjunction with each other, and that such variations are within the scope of the present invention.

[0039] Block 610 comprises receiving data from the sensor 85 corresponding to an image of the target product. It is to be appreciated that sensor 85 is a digital camera configured to capture an image of the target product with sufficiently high resolution.

[0040] Block 620 comprises sending the image from the scanner 65 to the terminal 55. The manner by which the image is sent is not particularly limited and can include methods of communications discussed above. It is to be appreciated that in this embodiment, the scanner 65 performs no further analysis and functions as a camera. Accordingly, in other embodiments the scanner 65 can be varied to be an internet camera or webcam.

[0041] Block 630 comprises identifying the product by applying neural networks to the image received from the scanner 65. In the present embodiment, the analysis using neural networks is carried out on the terminal 55. Since the application of neural networks can be resource intensive, it is to be appreciated that the terminal 55 can further send the image to a central server or a third party server to carry out the analysis. The manner by which the neural networks analysis is applied is not particularly limited. In the present embodiment, convolutional neural networks is used; however, other types of neural networks can also be used.

[0042] Referring to figure 6, another embodiment of a point of sale system is shown generally at 50a. The point of sale system 50a includes a scanner 65a and terminal 55a. Like components of the scanner 50a bear like reference to their counterparts in the point of sale system 50, except followed by the suffix “a”. In the present embodiment, the point of sale system 50a includes a scanner 65a and a terminal 55a. The scanner 65a includes a processor 100a, a sensor 85a, and an actuator 95a. The terminal 55a includes a processor 300a and a memory storage unit 310a.

[0043] The scanner 65a is not particularly limited and can include various network interface devices (not shown) capable of communicating with the terminal 55a as described above. In the present embodiment, scanner 65a is generally configured to connect to the terminal 55a with a standard serial cable and send data from the sensor 85a to the terminal 55a. It is to be appreciated by a person skilled in the art with the benefit of this description that the data sent to the terminal 55a can be preprocessed (eg. blurred or color corrected), or can be raw data directly from the sensor 85a.

[0044] The memory storage unit 310a can be of any type such as non-volatile memory (e.g. Electrically Erasable Programmable Read Only Memory (EEPROM), Flash Memory, hard disk, floppy disk, optical disk, solid state drive, or tape drive) or volatile memory (e.g. random access memory (RAM)). In the present embodiment, the memory storage unit 310a is generally configured to store a database 210a, such as a database correlating various characteristics and products with PLU codes. In addition, the memory storage unit 310a is configured to store codes for directing the processor 300a for carrying out computer implemented methods.

[0045] The database 210a is generally configured to correlate image data from the sensor 85 with various products and a product identifier. It is to be appreciated that the database 210a not particularly limited and can be varied depending on the configuration of the sensor 85a. For example, the database 210a can include color values associated with some or all products available for purchase using the point of sale system 50a. It is to be appreciated that each product may also include more than one color value or a range of color values due to product variations. In some embodiments, the database 210a includes one or more lookup tables. In other embodiments, the database 210a can include a training data set used to train a classification engine using a statistical modelling method to identify a product. Cluster analysis may be used. K-means clustering may be used (https://en.wikipedia.org/wiki/K-means_clustering), In some embodiments, color histogram data of a captured image is compared to histogram data stored in the database 210a. Histogram data stored in the database 210a is associated with various products. In such embodiments, errors between histogram data of a captured image and histogram data stored in the database 210a, and one or more minimum errors are identified. Products associated with stored histogram data corresponding to the one or more minimum errors are identified as candidate products. Products can be sorted based on error.

[0046] In addition, it is to be appreciated by a person of skill in the art with the benefit of this description that environmental variations can be introduced in the data captured by the sensor 85a. For example, the target product can be contained in a transparent or translucent plastic bag, such as those typically used for produce in a grocery store. Another example can be the environmental lighting in the vicinity of the point of sale system 50a. The environmental variations can be taken into account in the database 210a with additional entries for each expected variation. For example, in applications where the target product is typically placed in a bag with a specific color tint or translucency, the modified color values can be stored in the database 210a in additional to or instead of the non-modified values.

[0047] Hence, the database 210a can contain a plurality of sets of color histogram data, or similar data, for various products and each product may correspond to more than one set of histogram data. That is, for example, bananas may be represented by four distinct sets of color histogram data: ripe bananas outside bag, green bananas outside bag, ripe bananas inside bag, and green bananas inside bag. All sets of color histogram data for a particular product map to the same code for the product.

[0048] The processor 100a is not particularly limited and is generally configured to execute programming instructions 200a handling the data captured by the sensor 85a. For example, the programming instructions 200a can be configured to direct the processor 100a to carry out preprocessing on the data received from the sensor 85a prior to transmitting the preprocessed data to the terminal 55a.

[0049] The processor 300a is not particularly limited and is generally configured to execute programming instructions 220a for handling the data received from the scanner 65a. For example, the programming instructions 220a can be a driver configured to carry out image recognition using the database 210a and generate an error sorted list of products. The terminal 55a can then offer the user the error sorted list of products to make a selection. In addition, it is to be appreciated that the programming instruction 220a can be configured to recognize a barcode that may be in the data received from the scanner and utilize the barcode identifier to positively identify the product. In further embodiments, the histogram analysis discussed above can be used to verify barcode identifier to improve accuracy and/or prevent product tampering or fraud, such as by a customer attaching an unrelated barcode to a product.

[0050] Various advantages will now be apparent to a person of skill in the art. Of note is the ability to identify target products automatically without the use of barcodes. Applications such as at grocery store checkouts are particularly advantageous since the transaction time can be substantially reduced and the accuracy can be substantially increased by reducing the number of keystrokes required per transaction as well as reducing decisions by a human user. In particular, training for store employees can be reduced and the point of sale system 50 can also be particularly useful for self-checkout kiosks where users typically have no training and are unaware of PLU codes. Although examples presented above are primarily directed to produce in a grocery store, it is to be appreciated that the applications are not limited and that applications can include identification of a wide variety of products typically without barcodes. For example, the point of sale system 50 can be used in bakeries, deli shops, steel shops, hardware store, and restaurants.

[0051] Another advantage of the scanner 65 is that it can be sold separately from the point of sale system 50 and be compatible with existing point of sale systems since the scanner 65 can identify a target product and provide PLU code to the existing point of sale system using known formats. In addition, the scanner 65 can be used in other identification applications as well, such as warehouse and/or inventory management.

[0052] While specific embodiments have been described and illustrated, such embodiments should be considered illustrative only and should not serve to limit the accompanying claims.

APPENDIX

Handheld Point-of-Sale Electronic Produce Identification Description

This document offers a complete description of Ladon Labs' handheld point-of-sale electronic produce identification (PEPI) device, including all considered configurations and implementations.
The device is similar to handheld barcode scanners; however, in addition to resolving barcodes, it can identify commodities by analyzing sensor data that is collected from scanned objects. It then compares the sensor data against a database of expected properties for the scanned items.
The application of the device is to facilitate automation for commodity identification, typically when barcodes are not practical or expensive, but when commodities bear unique (and typically visual) characteristics. For instance, produce (at farmer's markets, grocery stores, etc.) and items sold in bulk can be visually identified by a cashier, whereupon a product look-up (PLU) code that is specific to the type of commodity can be keyed into a point-of-sale (POS) system. The device is designed to eliminate the need for manual keying of PLU codes by automatically identifying commodities.
This document considers designs that allow the device to function with a retailer's legacy POS equipment without the need for much modification.

1 Physical Components

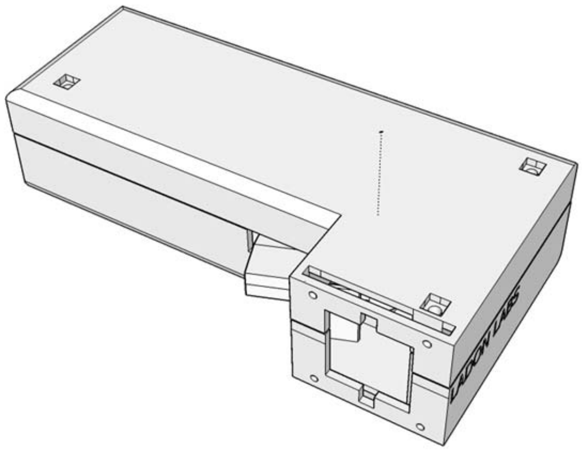
The device may encompass any of the following components for the handheld PEPI solution:

- Frame/case and activation switch
- Sensors
- Light source
- Data transceiver and channel
- Power source
- Display, output, and inputs
- Memory
- Logic device

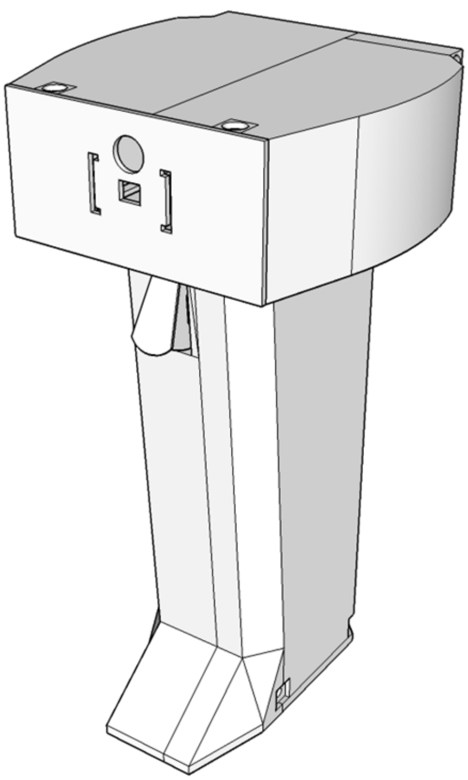
1.1 Frame/Case and Activation Switch

The rigid frame is designed to be handheld and aimed at commodities for scanning. The frame encases some or all of the electronics required for operation and is made out of plastic molded parts that are screwed together.
The handle of the device is designed to fit in a human hand. At the top of the handle is a head, which has a face panel that holds one or more sensors. The panel is intended to be aimed towards commodities for scanning.
Near the top of the handle, but below the head (in the location where the user's index finger is expected to rest), there is a single mechanical trigger. When pressed, it either compresses an internal pushbutton

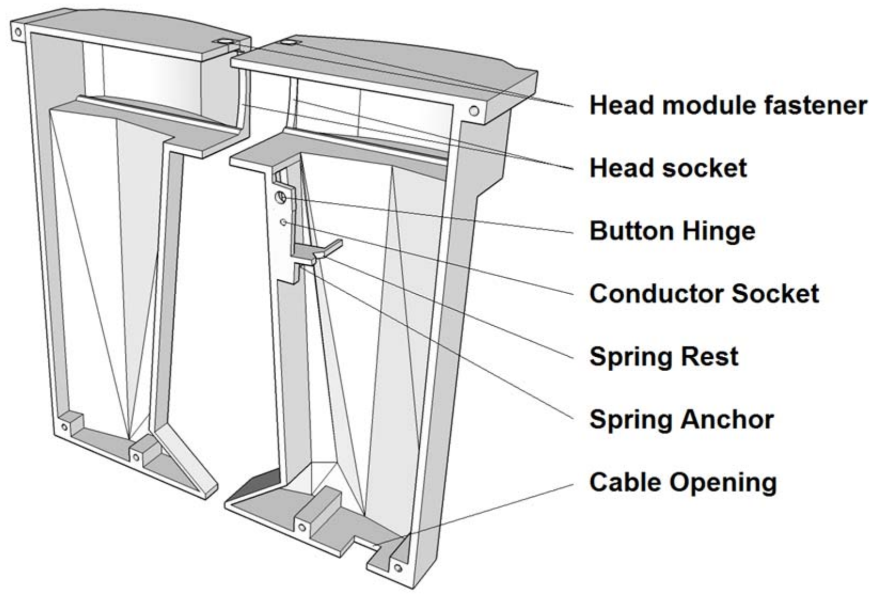
or shorts out two electrical conductors that feed into the onboard computer/logic device, resulting in the device's scanning subroutine to activate.
The following models are early stage alphas that have been manufactured by a 3D printer. The final case designs and manufacturing are to be outsourced.



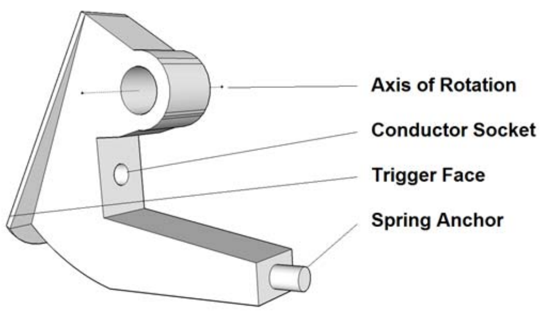
Closed casing for the A1 model (initial prototype). This prototype's face has an opening where a specific model of a digital camera (and supporting electronics) can be installed.



Casing for the A2 model, which features a removable face.



Opened casing for the A2. The opening on the base is for a USB cable.



The trigger used in the A2.

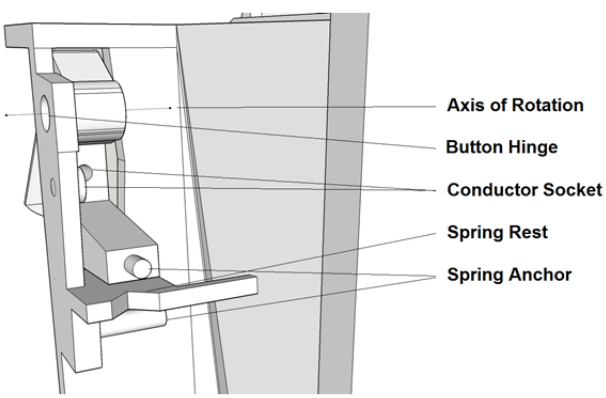
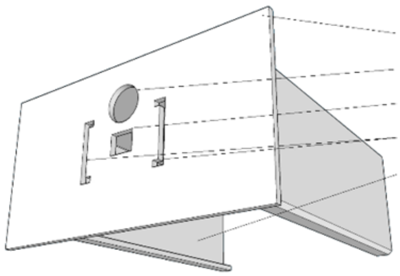


Figure 1 – A closer look of the inside of the casing for the A2 (with the trigger switch in position). When the trigger is depressed, a conductor on the trigger switch comes into contact with a conductor on the case (both labeled “conductor socket”), which activates the device. A spring resets the depressed trigger.



The removable head module for the A2. It simplifies the experimentation with different sensors and light sources without the need of manufacturing many different (and costly) prototypes. This will not be a feature of the final product. The electronics fit inside this module.

1.2 Sensors

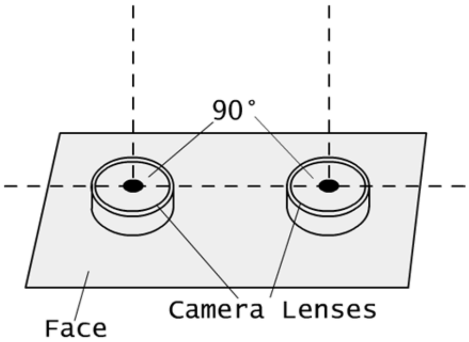
Sensors are fixed behind the device's face. We have considered variations of the device with a variety of sensor configurations. In particular, the following sensors were considered: proximity sensor, digital camera(s), colour sensors, spectroscopes, and specialized barcode sensors. Section 2 lists out the configurations in detail, and section 6 explains how sensor data is used in identification algorithms.

1.2.1 Proximity Sensor

Although a proximity sensor does not collect data on the characteristics of scanned objects, it is an optional asset to the design that could increase the overall accuracy of the system. The proximity sensor would be used alongside another sensor. This sensor would be capable of gathering more useful (and precise) data.
In addition, the proximity sensor reading can be used to assist in the focusing of a camera's lens (if one is present in the device).
The current prototype does not include a proximity sensor. However, we may produce some models that are equipped with proximity sensors. Examples of proximity sensors that we may use are the miniature laser rangefinder / LIDAR module, ultrasonic sensor, and infrared proximity sensor.

1.2.2 Digital Camera

A digital camera is used to capture coloured images of commodities. For practical applications, the camera must be calibrated to focus on objects that are typically within 10–30 cm of the camera lens. The lens itself may be specially designed for taking images at close range.
Digital cameras should also be configured to disable features that automatically adjust colour, hue, saturation, or tint. However, automatic gain control might be alright and can be configured.
The device may house two adjacent cameras instead of one in order to achieve stereo vision, which can be used to create a depth map without the need of a proximity sensor. In this configuration, the cameras would be positioned side by side from anywhere between 2–5 cm apart, mounted on the same plane, and parallel to one another.



Stereo vision camera configuration.

The current prototype does not use stereo vision, nor does it have two cameras. However, we intend to design a prototype that uses stereo vision in the future, since there is a lot of potential for increasing the device's accuracy.

We have looked at a number of different options for the digital camera. One consideration is the OVM9724, which is a compact colour CMOS-based image sensor chip (it is modularized and includes a lens) that can take 720p images and is in line with our requirements.

1.2.3 Colour Sensors

A naive method for produce identity estimation is accomplished by using simple colour sensors. The technique is explained in section 6.2. An early prototype used the colour sensor technique as a proof of concept.

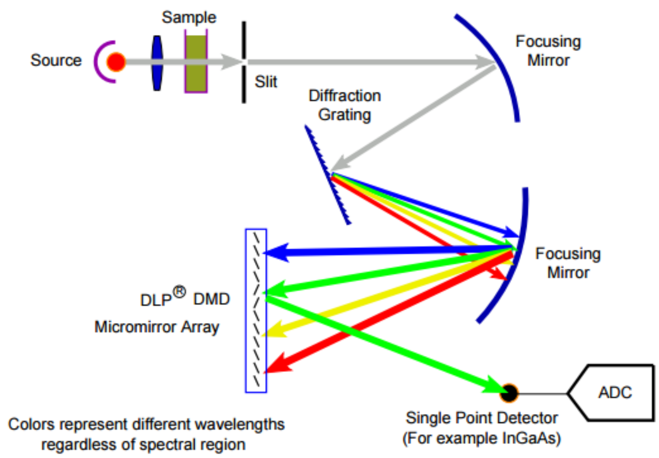
A module is used to read RGB data for a point directly in front of the sensor. Colour sensors may require an analog to digital converter circuit to digitize the colour brightness readings.

An optical lens may be included with the optical sensor so that only the point that is being aimed at affects the reading.

A variation of the device that uses colour sensors would typically not include a digital camera (but would sacrifice some accuracy, which is acceptable for some applications).

1.2.4 Spectroscopes

There are modular chips available on the market (such as the TI DLP chipset) that are capable of doing spectroscopy.



***Source: TI**. Shows how a DLP chip can be used for spectroscopy. Columns of mirrors are turned on or off to select specific wavelengths.*

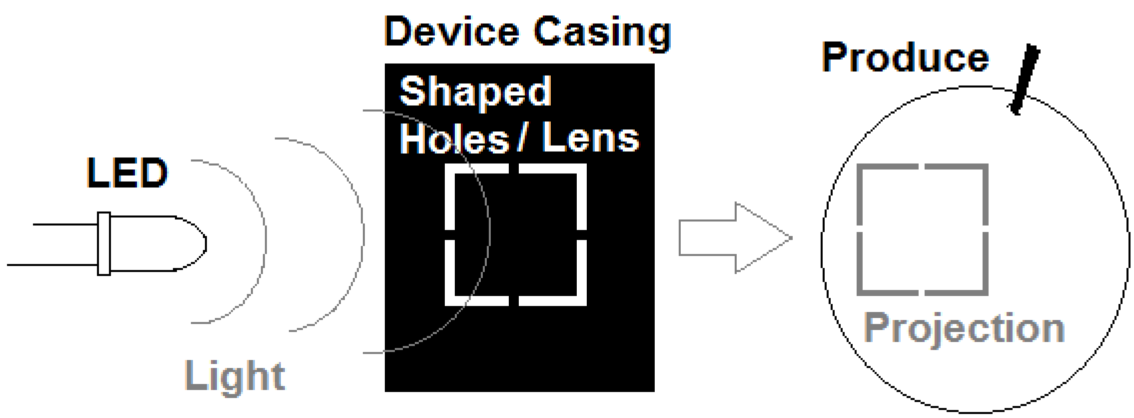
This entire setup (i.e. lenses, mirrors, light sources, drivers, etc.) are available from TI as a packaged module (see http://www.ti.com/lsds/ti/dlp/advanced-light-control/wavelength-control.page).
The spectroscope would be set up to do near-infrared spectroscopy (NIRS), allowing equipped devices to get a reading from deep within the target sample. The NIRS spectrograph can be passed onto a computer for analysis of the sample's chemical makeup. It is our intention to build spectroscopic-enabled units when some of the issues are worked out with the technology.

1.2.5 Specialized Barcode Scanners

The device is able to identify produce without the use of barcodes. However, it needs to function seamlessly for items with barcodes.
In order to accomplish barcode scanning, a device may be equipped with a laser barcode scanner.
A specialized barcode scanner module is not required if the device is equipped with a digital camera, since barcode information can be extracted from photographs.

1.3 Light Source

The device's face may be equipped with a light source (i.e. white LED) in order to illuminate commodities or barcodes.
Since many of the sensors are directional, a targeting light may be used to project light onto the target area or point. This can be achieved by using a laser or a coloured LED that is behind a special lens. Since many of the sensors are colour sensitive, the targeting light would need to be electronically deactivated just before an object is scanned.



Shows the projection of the scanner's sample area onto a produce item.

1.4 Data Transceiver and Channel

Two options exist for device-computer communication: cabled and wireless. All devices come equipped with at least one cable port. However, high-end variations may include the ability to communicate wirelessly with POS computers.

1.4.1 Cabled

Communication with a POS computer is established by using any cable that can carry serial data, including a USB 2.0 (or 3.0) cable, 10P10C/RJ45 to USB, or RS232 serial connector.
Many models of traditional barcode scanners can be connected to a computer by using a PS2 keyboard wedge. The wedge is a three-ended cable. One end connects to the barcode scanner (typically with a

male 10P10C or RJ45 jack), another end connects to the PS2 port on a computer, and the last end connects to the PS2 cable of the keyboard. This, in effect, splits the PS2 port so that the barcode scanner can be used as a second keyboard while an actual keyboard is also connected and can be used. Then, when a barcode is scanned by the scanner, the scanner automatically types the code into a field within the POS software's user interface. This would be a good way for our device to work seamlessly with legacy systems; however, as explained later in this document, data collected from the device's sensors is not always enough to narrow down the selection of potential PLU codes to just one, and it may be prone to some degree of error that requires manual intervention to rectify. If a high-end device were to include a graphical user interface directly on the scanner itself, then it would conceivably be possible to allow the device to work with a keyboard wedge. Of course, sensor data cannot be sent through PS2, so all of the device databases and sensor data processing would have to happen on the device itself. This is not an optimal setup.
The current prototype is designed to use a USB cable.

1.4.2 Wireless

High-end units can connect with PCs over high speed Bluetooth or Wi-Fi (IEEE 802.11).

1.5 Power Supply

Device models that are capable of wireless connectivity will have an onboard power supply in the casing. The equivalent of a minimum of 4 AA batteries connected in series is needed to power the device's onboard electronics. The casing for a battery-powered device has not been modeled yet (since we are currently working with wired prototypes); however, we have considered adding a hatch on the bottom of the device where batteries can be slid into a special battery compartment. Alternatively, we have also considered using rechargeable power sources. For instance, two in-series lithium polymer cells. The rechargeable cells would require some additional circuitry (specifically, a lithium polymer cell charger).
Cabled devices can be powered directly from a computer (except for devices connected via RS232, which does not supply adequate power). For this reason, we are also considering any variation of the device that uses an external AC power adapter (even though it is unlikely that we will build this variation).
The device will house a capacitor on its circuit board to provide a small amount of charge in case the device is unplugged unexpectedly. The extra charge can be used to safely wrap up any writes to internal flash memory.

1.6 Display / Output / Inputs

The device may include additional indicator LEDs that are used to inform the user of the device's state (such as power conditions, status of communication with the computer, etc). In addition, an internal piezoelectric speaker generates audible feedback when the device is used (but could also be used to indicate errors). We have considered adding a dot matrix display or even an LCD (touch) screen for high-end units and additional buttons that will be positioned on the unit's head or handle, which will allow for on-site configuration. The device may include the following four buttons:

- Up
- Down
- Menu / Enter / Confirm
- Escape / Back / Cancel

1.7 Memory

Units contain flash memory to store some of the programmable operational parameters for the device.

1.8 Logic Device

A digital logic device bridges the sensor data, inputs, outputs, and memory for the scanner. The logic device may also be responsible for some or all of the sensor data processing. The current prototype uses a CPU (the Atmega328); however, we are looking at possible CPLD or even ASIC options, and it is unlikely that the Atmega328's 20 MHz processor will be suitable for quickly reading data from attached cameras.

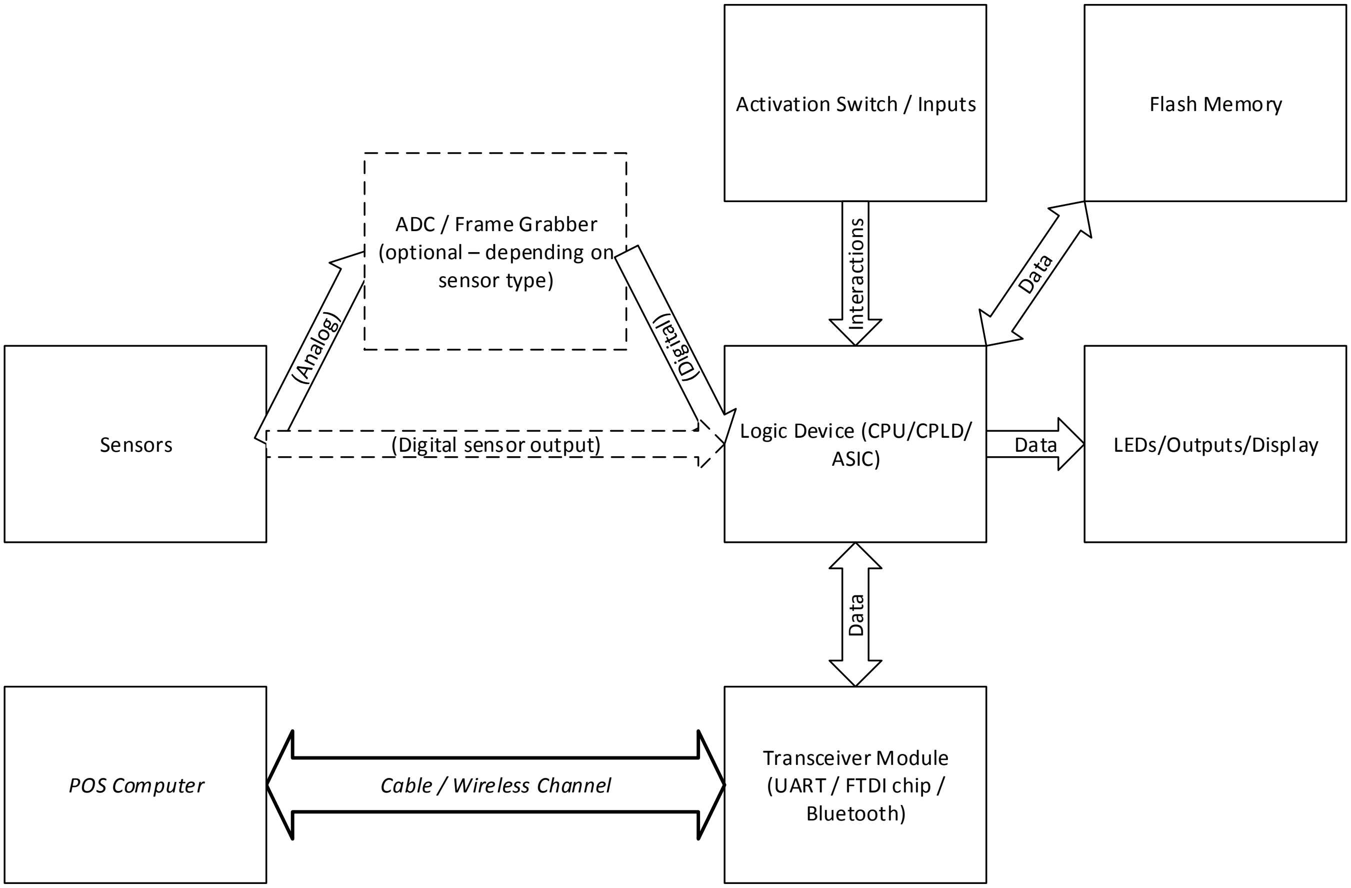
2 Sensor Configurations

The following distinct sensor configurations are supported:

Primary Sensors (Object Classification)	**Secondary Sensors (Optional)**	**Barcode Scanning (Optional)**
Camera (mono or stereo)	Proximity Sensor	Using Camera or Laser Sensor
Spectroscope	Proximity Sensor & RGB Colour Sensor	Laser Sensor
Spectroscope & Camera (mono or stereo)	Proximity Sensor	Using Camera or Laser Sensor
RGB Colour Sensor	Proximity Sensor	Laser Sensor

The version that has both a spectroscope and a camera would be a high-end unit that can use both sensors to make very informed predictions. The camera can also be used for barcode scanning.

3 Electronics Layout – Block Diagram



A block diagram for the electronics within the device. The ADC or frame grabber is only necessary for sensors that require those components. Otherwise, digital sensor data can be sent directly to the logic device.

4 Operational Concepts and Considerations

4.1 Where do the Identification Algorithms Process Sensor Data?

There are three places that sensor data may be processed:

1. On the device's logic chip.
2. On the POS computer that uses special driver software (after receiving raw data from the device).
3. On the cloud (after receiving data from an internet-connected POS system, which gets data from the device by using special driver software).

Our ultimate business goal is to maximize the algorithm's speed, accuracy, and compatibility while minimizing cost and size. A fast CPU or GPU on the device may be able to do all of the data processing very quickly, but may make for a very expensive and bulky unit. The POS computer is the next best choice. We can send images or other sensor data to drivers that are running a computer by using a virtual RS232 over USB. That data can then be processed by our drivers. The challenge with that comes when we try to process data spectroscopy. Currently, the leading algorithms available for analyzing NIRS data require the use of artificial neural networks that run on fairly beefy computers. Therefore, it may be more practical to do this type of analysis on the cloud (especially when the neural network is optimized to run on GPUs).

With all of that being said, we can also do combined processing between the device and the POS computer. The device's logic unit could contain specialized hardware for performing certain pre-processing. This would simplify one or more computationally expensive steps in the identification algorithm that would otherwise have to run on (possibly very slow / single-threaded) POS computers. For instance, it may be more efficient to do continuous barcode scanning on the device and only send images over the serial connection when the trigger is pressed (for produce scanning).

The most recent prototype does produce identification on the cloud, but we are moving it to the drivers. Therefore, other sections in this document will assume that produce identification occurs within the drivers.

4.2 Database of Commodities and PLU Codes

POS systems contain databases of commodities with corresponding PLU and UPC codes. The device has its own database of commodities and PLU codes. Although, UPC numbers are read directly from barcodes (and do not require the device to know what it is scanning), we need to arbitrarily assign PLU codes to each commodity. Various conventions exist for PLU codes, but not all retailers adhere to the same standard. Also, although the device's database may someday support thousands of classes of commodities, a retailer typically will not need the device to offer codes for every possible commodity (as this would make the device less accurate and possibly slower).

For these reasons, a configurable database must be used to store a list of enabled commodities and variations for that retail location. A map between the device's PLU codes and the PLU codes of the retailer is another configurable database, but one that interacts with the layer of the software stack that is responsible for output (which may be bundled software or user-written software).

The database would be accessible by the device drivers and configurable through software tools.

4.3 Interfacing with POS Software

The device will be delivered with drivers that bridge communication between the scanner and the (typically legacy) POS software. The drivers provide a "device interface" that allows developers to access the device, configure settings, receive notification of device activation or barcode detection, and read the results of identification algorithms.

Units will also include optional software that has been written by Ladon Labs. This software will use the device drivers to perform certain default configurable functions.

4.4 Dealing with Virtually Identical Varieties of Produce

When a non-barcoded commodity is scanned, candidates for the item's PLU code must be interpreted from sensor data. Sometimes more than one code can be attributed to a single class (in which class refers to a distinct commodity from the perspective of the identification algorithms) in the case of perfectly identical commodities or commodities that have unnoticeable variations (for instance, organic vs. non-organic commodities, GMO vs. non-GMO, chocolate covered nuts vs. chocolate covered raisins, etc.).

To mitigate this problem, the device's PLU database may contain subvarieties for certain classes. Then, when an item is scanned and selected from the user interface, the interface will show options for each of the subvarieties for the selected class. For instance, a cashier might scan an orange and be prompted to confirm the scanner's guess that an orange was scanned. Upon confirmation, the cashier would be prompted to select between "Organic," "GMO," and "Regular."

4.5 Dealing with Error

The device needs to be commercially viable despite not being able to achieve 100% accuracy. However, at the same time, it must never tally the wrong item. This motivated us to conceptualize one of the most important aspects of the device: the commodity ranking system. Instead of trying to build a perfect technology with super-high accuracy, the identification algorithms assign a rank to each commodity in the database, then display the ordered list to the user with the most relevant items on top. The user may confirm which item in the list is correct. The correct item should typically be somewhere near the top of the list, and, if not, the user can scroll down (which may involve a swipe if it is a touch screen) or visit the next page of results. Alternatively, if the data collected was of a very low quality, the user can simply rescan the commodity to refresh the list of suggestions onscreen. If, for whatever reason, the device's sensors are malfunctioning, or the item being scanned is not in the device's database (or the wrong results are showing for other reasons), the user can always revert back to the old method of looking up the commodity's PLU code and entering it.

4.6 Scanning has to Work Through Plastic Bags

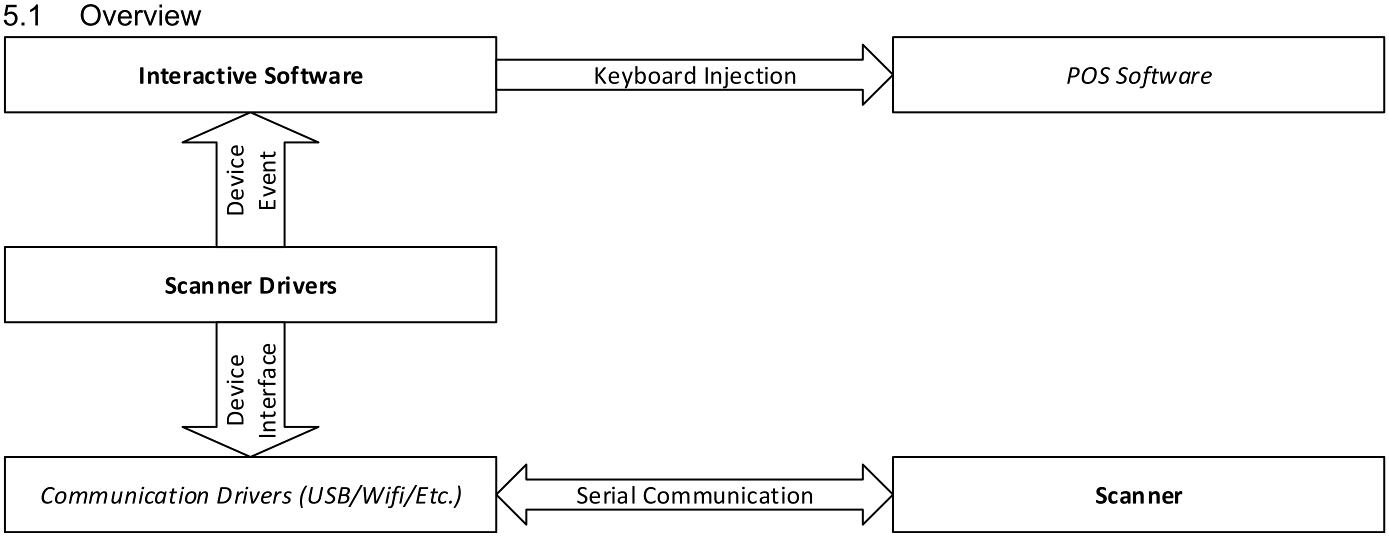
In many grocery stores, produce items are placed in clear plastic bags. Most produce bags are designed so that any barcoded commodities inside will be visible. Each algorithm described in this document works for bagged items and non-bagged items. The NIRS described later in this document are able to see right through plastic and obtain readings from underneath the surface of scanned items. Colour-based algorithms are trained on both bagged items and non-bagged items.

4.7 Automatic Camera Functions

Some cameras have automatic saturation, tint, or white balance for enhancing the images taken. Unfortunately, these features interfere with the identification algorithms and must be disabled. Some other features are useful such as automatic focus and (maybe) gain control.

5 Software

5.1 Overview



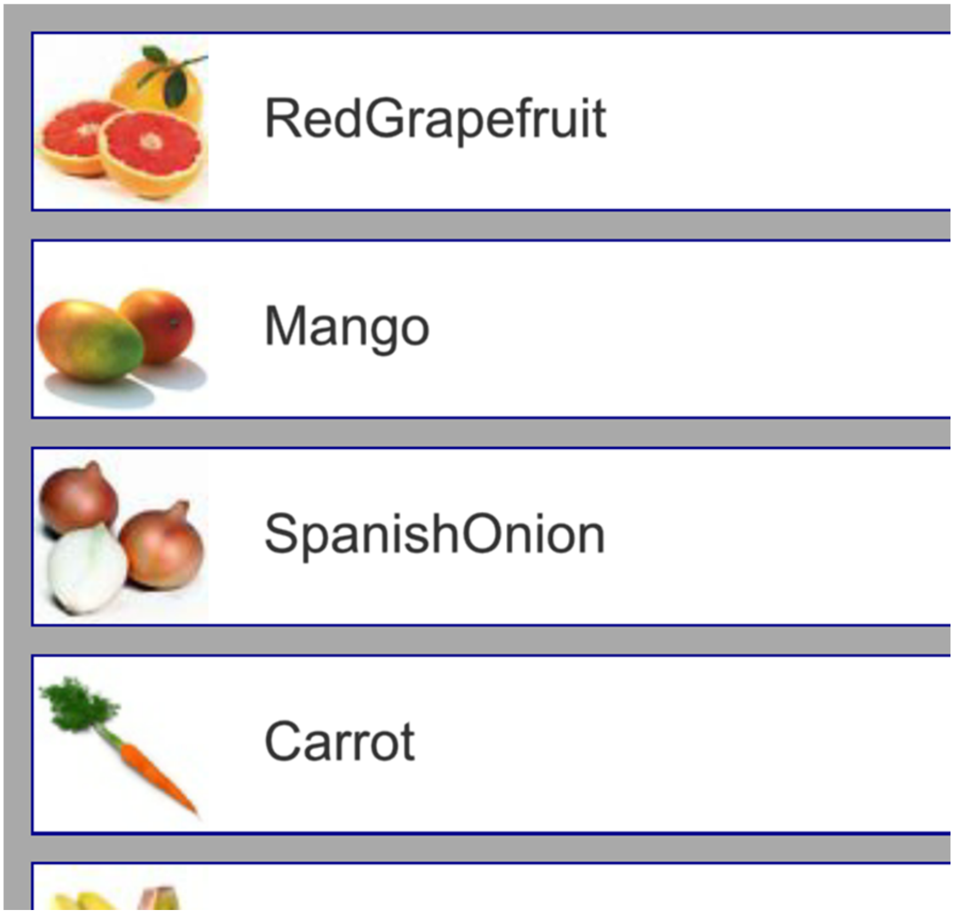
The driver polls connected scanners via USB (or other) system drivers. Sensor data is returned to the driver when the device is activated. The driver analyzes the sensor data and fires a device event, which is picked up by the interactive software. The *software confirms the identity for the scanned item with the user, and the code for that commodity is injected into the POS software. Variations follow.*

5.2 Drivers

Drivers need to be installed on a POS computer in order to communicate with the device.
When active (and a scanner is connected), the driver software constantly polls the device approximately every 100 milliseconds for updates. When the device has been activated via a user-activated scan, or a barcode being identified, the driver software will be notified in the device's response to the poll.
If the particular model of device is able to do any preprocessing of sensor data (or full processing while offering commodity identity suggestions) on the device, then that data is sent over the communication channel to the driver. Otherwise, the device just sends raw sensor data.
In addition to sensor data that is retrieved from the scanner, the drivers accept certain parameters that can be passed down from the software layer. Although some of these parameters include settings on how to communicate with the device, which items to scan for, and others, the drivers also accept clues that will assist with produce identification (for instance, the weight reading from a POS-connected scale). The drivers are also expected to have access to the host system's date and time, which are useful external data.
The drivers process all of the available data by using the appropriate identification algorithm, or (as a variation) the driver may send the data over the internet to the cloud for processing (especially for spectroscopic data).
Once the algorithms return their results, the driver fires a device event, which notifies listening applications of the algorithm outputs.

5.3 Default POS Software

The default software included with the scanner listens to the device drivers for an activation event (i.e. a user-activated scan or a barcode scan). If the event is a barcode being scanned, the interactive software automatically passes the scanned UPC code into a field within the user's configured POS software, which can be accomplished in a variety of ways such as memory editing or emulating the keyboard to type the code in (similar to the way many regular barcode scanners work). For non-barcoded commodities, a user interface will appear on top of all other windows on the computer's screen, displaying the ordered and ranked list of potential identities. The list can be scrolled down in case of an anomalous reading, or an option can be selected by clicking on it or typing a keyboard shortcut (1–9, which will be displayed next to each option). The Escape key can be used to cancel the activation event non-barcoded item selection interface, and if the drivers indicate that a second scan was processed before the selection could be completed, the selection interface is refreshed with the results of the new scan, and the old data is discarded. Below is a screenshot of the interface for context.



*Screenshot of the software's popup interface, which lists suggestions for a grapefruit that was just scanned. The top-most item is the identity of the object, which is most probably correct, and each subsequent item down the list are less and less likely to be correct until the very bottom of the list where the absolute least likely option lies (however, the list can also be safely trimmed for items that are extremely improbable). *The keyboard shortcuts for the top nine items do not appear in this screenshot*. This was taken with a select database of only 27 types of produce. Note that the produce photographs are **not licensed and are for internal use only.***

Once the item is selected, the PLU (or UPC) is communicated to the POS software, similar to how the UPC for barcodes would be when a handheld barcode scanner is used (explained above).

5.4 Configuration Software

This software provides a convenient interface for reading and writing device settings, including the list of commodities supported. The software can be used to configure settings in the device drivers and on the device itself (via the drivers). In the future, the software will most likely provide an interface to map PLU codes (however, until then, it will have to be done in a config file).

5.5 Variations not Requiring Software or Drivers on POS Computers

Although this would be difficult to achieve, it is possible if the following technological improvements are made (and we can expect these improvements to be made within the lifetime of the project due to trends in computers):

- Digital processors become efficient and powerful enough to run all of the identification algorithms on the device's processor

- The accuracy of the algorithms becomes high enough that the correct identity is suggested within the top five results nearly every time (allowing for the user to select the correct item by using a user interface on the device itself)

The POS computer would not need any special drivers to interface with the scanner, and the scanner could be used with a keyboard wedge for legacy equipment. In addition, we would not need any special software on the POS computers; this would all be built into the device's firmware.
Since a keyboard wedge uses a PS2 connector (which is inconvenient to develop a custom interface for), a device of this variation would also have the option of connecting with a USB cable in case users want to upload their own custom settings.

6 Identification Algorithms

We may use any one or more of the outlined methods, depending on available memory, computing power, and the level of accuracy for each algorithm, which is expected to increase over time, along with an expected increase in the density and affordability of computing power.

6.1 Concepts

6.1.1 Preprocessing

Algorithms looking at images are optimal when the images have been preprocessed as follows:
Colour normalization is a technique that reduces the varying effects of illumination, increasing the effectiveness of colour-based computer vision techniques.
Images are blurred by convolving the image with a Gaussian filter. Blurring images results in more consistent sampling of pixel colours by reducing the effects of white noise, which is essential for any type of feature detection.

6.1.2 Handling Variations within a Commodity

Consider mangos. Ripe mangos look very different from unripe mangos and even contain different concentrations of chemical compounds. This poses a challenge to identification algorithms that try to model commodities by using a set of consistent expected features.
To solve this, we train each sufficiently unique variety of the same commodity behind the scenes as a separate class. However, each of those classes will map back to the same commodity in our database, which means that the user interface will not show duplicate results for variations (such as "ripe mango," "unripe mango," etc. Instead, it will just show "mango" if either is scanned).

6.1.3 Supervised Learning

Supervised machine learning is used in all of our algorithms because there are a known number of classes of produce. Supervised learning algorithms are trained when they look at many examples of sensor inputs for each class. The larger the training set, the better the algorithms typically perform. Training techniques also involve them looking at example inputs that are not in the class to be learned so that they can effectively learn what the sensor readings for a class do not look like.
Once trained, any given algorithm can take digital/digitized sensor data and run a test that will compute the most likely classification of the scanned item.

6.1.4 Combining Multiple Algorithms

The ranking of commodities is the combination of all of the available sensors and data. This may involve the use of several algorithms. If each algorithm outputs a probability that each commodity is the correct one, all of the probabilities can be computed into one final ranked list of commodities by using Bayesian statistics.

6.2 Using an RGB Colour Sensor

High accuracy is not necessary for all applications, and using a single-point RGB colour sensor is much cheaper than using a digital camera or spectroscope. This is the simplest algorithm that can solve the identification problem. However, in tests, the correct identity is suggested within the top five items (out of a set of 27 commodity classes) about 95% of the time, and the algorithm and database are both small enough to fit onto the scanner itself.

In this algorithm, a training set is built by analyzing numerous photographs of each class of commodity. The images are blurred by using a Gaussian filter and are sampled for (R, G, B) values at different points. Each point is normalized. The new normalized RGB values are recorded in memory, and Gaussian distributions are applied to the normalized colours to model the class. Those models are saved in a database.

During testing, the colour sensor takes RGB readings from a point that is directly in front of the sensor. Not all of the sensors are created equally, so some corrections may be required to the sample. This is so that the sample is consistent with the camera that is used in the training set. The corrected colour is then normalized by using the formula above. Then, the point is compared with each Gaussian model in the device's database, and an error % is generated. The classes are ranked from the lowest to highest error.

This technique can be improved by taking many samples from the commodity over an interval and minimizing the mean sample-class error for all samples.

6.3 Using Image Classification (Camera)

6.3.1 Artificial Neural Networks for Image Classification

Image classification can be solved quite easily by using artificial neural networks. Most of our current research is using convolutional neural networks, which excel at feature detection. Since neural networks are computationally and memory intensive, they are well-suited for running on the cloud or a beefy POS computer.

Our convolutional neural networks work on GPUs and are massively parallelized. However, the GPU-optimized version requires that the computer have a powerful graphics card.

Neural networks are trained by using supervised learning (in our case, usually with over 1000 positive examples and the same—or more—negative examples). In both training and testing, the inputs to the neural net can be passed through a pyramid function so that the neural net can look at each input under different resolutions or zoom levels. This may increase the accuracy of feature detection when it is uncertain of how near or far away the scanned object is from the camera (at the cost of increased computation).

6.3.2 Image Segmentation

Image segmentation is used to determine which areas of the image are of commodities. The technique can be accomplished by using edge detection and combining it with colour clustering (and if available from stereo vision, depth mapping) to determine the area and border of scanned objects in the image.

6.3.3 Bag of Features

The bag of features (or bag of words) model for image classification is a common technique that breaks test images into a histogram of features, and then uses the feature-space representation of each image to perform classification.

6.3.4 Using Stereo Vision

In addition to one of the above methods, a model with two adjacent cameras can use stereo vision techniques. Images from stereo cameras can be used to calculate depth data and distances from the object.
Stereo vision is a technique that uses a known distance between two cameras—as well as a perceived distance (in pixels) between two similar-looking regions of the two images that are nearby to each other—to calculate the distance between the camera plane and the point that is being looked at.
The depth data can aid in image segmentation. For instance, it can use a region-growing algorithm to determine segments for bagged goods and the background.

6.4 Spectrometry

Near-infrared spectroscopy (NIRS) can be used to determine the chemical composition of scanned items, but raw spectroscopic data from complex samples is influenced by many variables, which requires a bit of work to determine the chemical makeup of the object.
Existing methods exist for extracting relevant information from NIRS.
http://www.sciencedirect.com/science/article/pii/S0169743907000925# compares several methods for spectral data analysis. Most importantly, it discusses an approach that uses variations of partial least squares regression as well as an approach that uses neural networks. These techniques are typically computationally difficult, which means that they are best suited for remote servers or a beefy POS computer. However, we are considering the effectiveness of using a custom application-specific integrated circuit (ASIC) or complex programmable logic device (CPLD). Both of these can be mounted on the scanner's internal circuit board, which is specifically designed for this particular calculation. However, we have not designed one as of yet, and such a chip would constitute its own patent.
After processing, the chemical properties (including concentrations of the sample) will have been determined.
Once the data has been examined by either of the above techniques, the unique chemical properties of the sample need to be tested against each commodity class in the database to find a match. This can be solved by using simple statistical classification techniques such as the analysis of feature vectors (in which detected chemical compounds are the features).

6.5 Barcode Scanning

Barcode scanning has been around for a long time. However, it seems necessary that this produce scanner is also able to read barcodes (otherwise, many of our target customers might not want to adopt our product). One way to think of the entire device is that it is an improved handheld barcode scanner that also has the ability to scan produce.
The use of a typical laser barcode has been used commercially for many years now. This technology may involve reflecting light from a laser diode onto a commodity, focusing the returned light to a linear light receiver, and processing the output of the receiver.
Since our scanners will likely be equipped with a camera, images that are taken can be processed for barcodes. This processing is simple enough to be done on the device itself. There is a two-step process: the first step is detection/localization, and the second step is reading. Note that processing images for barcodes does not require the image to be colour-normalized because barcodes are typically black and white, but it does help if the image is converted to greyscale first.
Detection is determining whether or not an image contains a barcode. Localization is the problem of finding the position and orientation of barcodes in an image. Detection and localization are done simultaneously and can be accomplished by passing the photograph through several filters that

emphasize barcode-like shapes while suppressing other shapes. This is typically done by using edge detectors.
Once localization is complete, reading the barcode is accomplished by scanning each pixel along the center of the rectangle that contains the barcode widthwise and measuring the relative length of dark or bright regions.
Barcodes contain self-check codes that must be examined on the scanner's computer to verify that the correct number was read.

6.6 Other Clues

6.6.1 Date

Knowing the current date can increase the accuracy of the system because it factors in the seasonal popularity of each commodity. For instance, squash is most popular during the fall or late summer.

6.6.2 Time and Weekday

A heuristic for considering the trends of purchase frequency for types of commodities during hours of the day and days of the week may be able to marginally increase accuracy. For instance, people may be slightly more likely than normal to purchase vegetables (for supper) if it is evening during a weekday.

6.6.3 Weight

The weight of the commodity that is being scanned can be approximated from image data (especially stereo vision) by considering a database of commodity densities. However (in the absence of all other data), if a weight reading is available from a scale, the choice of commodity that minimizes the (absolute) difference between the estimated weight and the measured weight is more likely to be correct than one that does not.

7 Programming and Interfacing with the Device

7.1 List of Configurable Features or Variables in the Device Drivers

- Database flags for commodities and their varieties to scan for
- Which device to use (if multiple ones are connected)
- Connection method and related parameters (especially for wireless connectivity)
- Whether or not cloud services (reporting or cloud algorithms) are enabled, and how those connections should be made
- Whether or not barcode scanning should be enabled in the drivers (which requires barcode scanning to not be enabled on the device), and, when enabled, any images that are received from the device will be processed for barcodes
- Whether or not barcode scanning should occur automatically (continuous), when the device is activated via the trigger, or not at all
    - If continuous, a repeated signal will be sent at a variable interval to poll the device for barcode information
    - If barcode reading is supported and enabled by the device, the device may return the value of a barcode (otherwise, the device may return an image to be processed by the drivers—if that is supported)

    - Any images that are received by the drivers for the scanning of barcodes will not be considered for produce unless the device is configured to scan barcodes when the trigger is activated
- The time interval at which continuous barcode scanning should occur
- Whether or not to use the weight of a commodity to help identify it, which is useful if a scale is connected
- The current value of a connected weight scale (volatile)
- The flag that indicates that the scale is not stable (if enabled, it pauses the device's image capture)

In addition, the drivers can be used to set parameters on the device itself.

7.2 List of Configurable Features or Variables in the Device's Memory

- Whether or not barcode scanning should be enabled on the device—if supported (if enabled, images taken on the device will be processed for barcodes)
- Whether or not to pause activation (which can optionally be set when a connected scale is balancing)
    - If activated, the device trigger will not capture the image, and a notification LED on the device will indicate that the device is waiting for an activation signal

8 Usage

8.1 Connecting to a Scanner at the Driver Lever

The drivers are designed to interface with a single device that is connected to the computer (however, depending on the customer needs, this could change). The drivers can be polled to get a list of all of the currently connected USB scanners that are supported. The drivers can also contain subroutines to set a device as the active device. Alternatively, the drivers also contain subroutines to search for wireless scanners and connect with them.

If no device is currently set as the active device, a new supported device will automatically be set as the driver's active device (and will be unset when the device is unplugged). Therefore, the only time that the drivers need to be used to choose an active device is when multiple scanners are connected to the same computer.

8.2 Updating the Device Settings or Variables

All updating of the device settings or variables should be done by communicating with the device drivers. The communication protocol with the drivers will eventually be documented so that developers can configure both the drivers and settings on the device itself.

8.3 Connecting a Scale

The drivers offer support for an external weight scale to be plugged in. However, the weight values must be passed into the drivers in order to be used. There are two variables to consider.

Whenever the weight on the scale changes, the drivers should be notified that the scale is not stable. This will prevent the device from doing produce identification (even when the button is pressed). Once the scale is balanced, the weight value should be sent to the drivers, and the "unstable scale" flag should be cleared. This will resume normal device operation, and if the drivers are configured to use the weight to help identify the commodity, they (the drivers) will do so.

8.4 Programming the Scale by Using Included Software

The device and drivers will be programmable by using software that interfaces with the drivers. It is possible for developers to write their own software or use the included software. The configuration software has not been written yet.

What is claimed is:

1. An point of sale system for identifying a target product, the point of sale system comprising:

    a terminal;

    a scale for measuring a weight of the target product; and

    a scanner for measuring a characteristic of the target product, wherein the characteristic is used to identify the target product.

2. An scanner for identifying a target product, the scanner comprising:

    a sensor for capturing data associated with the target product; and

    a processor in communication with the sensor, the processor analyzing a characteristic of the data to identify the target product.

3. A method for identifying a target product, the method comprising:

    receiving data from a sensor, the data associated with the target product;

    preprocessing the data to facilitate analysis of a characteristic;

    analyzing the characteristics to generate results; and

    identifying the target product using results of the analysis of the characteristic.

4. A point of sale system, scanner, or method for identifying a target products as described in the specification and figures.

ABSTRACT OF DISCLOSURE

A point of sale system, scanner, and method are provided.