Computers and Electronics in Agriculture 76 (2011) 80–88

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry

J. Rakun[a,*], D. Stajnko[a], D. Zazula[b]

[a] *University of Maribor, Faculty of Agriculture and Life Sciences, Pivola 10, 2011 Hoče, Slovenia*
[b] *University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova ul. 17, 2000 Maribor, Slovenia*

ARTICLE INFO

Article history:
Received 23 September 2010
Received in revised form 5 January 2011
Accepted 15 January 2011

Keywords:
Image segmentation
Texture analysis
Image registration
Affine transformation
Multiview geometry
3D reconstruction

ABSTRACT

This paper describes a computer vision based model for object detection that can serve as a preliminary step in fruit prognosis, which involves the estimation of the number, diameter and yield of apple fruits. In order to overcome the recognition unreliability in uncontrolled environments caused by uneven illumination conditions, partly occluded surfaces, and similar background features, we rely on a combination of the object's colour, texture and 3D shape properties. In our research, we apply colour segmentation to multiple scene snapshots to separate potential regions from the background and verify them first with texture analysis and second by reconstructing them to 3D space. By analysing all three distinct features (colour, texture and 3D shape) of possible areas, we can safely conclude if they represent fruits we are looking for. Once we detect and verify all areas representing fruits, we can measure their size and model estimated fruit yield.

1. Introduction

Object detection algorithms represent an important part of more complex systems, such as fruit picking robots (Juste and Sevilla, 1991; Tanigaki et al., 2008; Van Henten et al., 2003) or harvest prognosis systems (Stajnko et al., 2009) and even quality control systems (López-Garcíaa et al., 2010). In all three cases we can use digitised snapshots of natural scenes, captured by a common digital camera. By using 2D digitised representations of a given natural scene, we can detect regions that represent fruit and draw conclusions about quantity, quality and location. Various approaches have been studied and implemented to detect fruit on trees, on bushes, or on the ground, either to count them or to guide robotic arms towards them. In general, these approaches can be classified into three groups according to the techniques used, as summarised below.

The first group of authors relies on the use of range sensors or stereovision systems, as described by Jiménez et al. (1999) and Benady and Miles (1992). They offer a unique view of the scene, providing range data, where it represents measurements of distance from the sensor to the observed object. Based on range data the curvature of observed object can be determined. If the object in question has the right shapes, it probably represents objects we are looking for. The process is not as straight forward as it may seem, but tends to gain in complexity as scene complexity increases. Furthermore, range sensors represent specialized equipment that is rarely available on the field.

As an example of the first group, Tanigaki et al. (2008) presented a cherry harvesting robot that works with the help of 3D vision sensor. The sensor is made up of artificial light sources and two digital cameras that enable the robot to estimate the distance between the sensor and the object by using a stereovision approach. The robot is able to detect mature red cherries from the green leaves that are picked by using 4 degrees of freedom end effector.

Similar work has been done by Van Henten et al. (2003). In their work they describe a prototype of a robot that is able to pick cucumbers. It also uses stereovision to locate cucumbers, but still faces some problems. For example, it takes at best 53 seconds to detect and harvest a single cucumber, where authors report 74.4% success rate after several attempts.

The procedures described by the second group of authors (Sites and Delwiche (1988)) are based on capturing digitalized images while applying additional lighting to the scene. The illumination conditions are selected with care to promote regions of interest and to suppress surroundings. Some purposed approaches also use night-time conditions, when they are not interfered with other light sources. Presented solutions are therefore at least awkward and unpractical.

Instead of using an artificial light source and a digital camera, which works by capturing images in the visible spectrum, to detect fruits, a hyper spectral system could be used. It is more common to quality assessment systems as the one from Gómez-

[*] Corresponding author. Tel.: +386 2 320 90 88.
E-mail address: Jurij.rakun@uni-mb.si (J. Rakun).

0168-1699/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2011.01.007

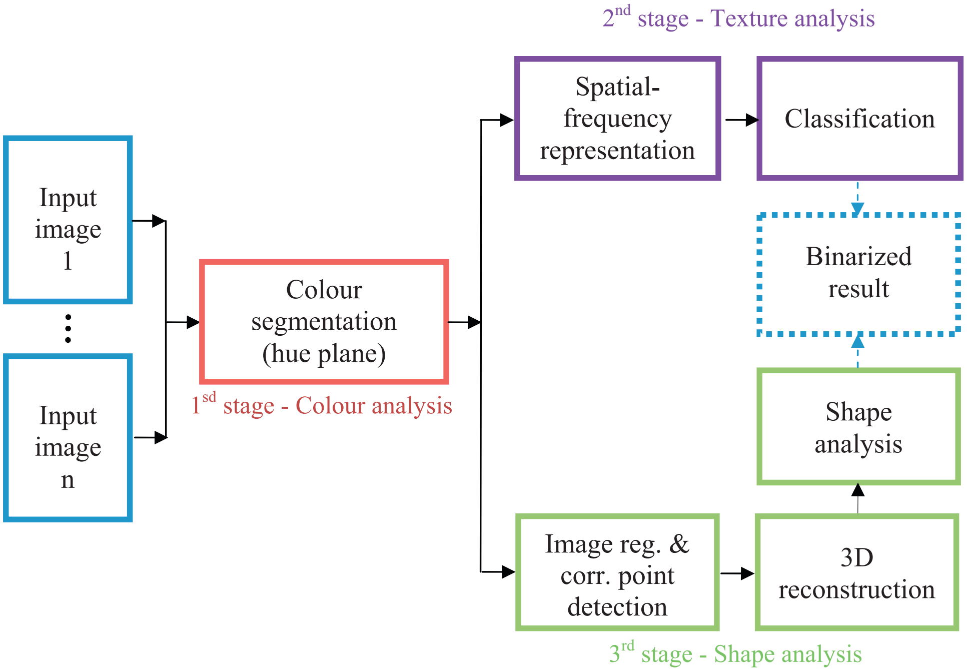
2nd stage - Texture analysis
Spatial-frequency representation
Classification
Binarized result
Input image 1
⋮
Input image n
Colour segmentation (hue plane)
1sd stage - Colour analysis
Shape analysis
Image reg. & corr. point detection
3D reconstruction
3rd stage - Shape analysis

Fig. 1. Flowchart for presented object detection algorithm.

Sanchis (2008), but as the computer power increases with time it will not be long until it reaches onboard solutions. The problem that hyper spectral systems face is the amount of data that needs to be processed. In order to reduce the quantity classification and regression trees (CART) or linear discriminant analysis is used.

The last group of authors (Juste and Sevilla, 1991; Zhao et al., 2005; Stajnko et al., 2009) purpose is to apply digitalized images taken by a common CCD digital camera without any artificial light sources. Images are captured under uncontrolled environments using everyday photographic equipment. In contrast to solutions from the previous section, digitalization step is simple, but detection step can be quite challenging.

By looking at the data captured using a digital camera, we can safely conclude that objects can be identified by detecting their prominent features such as colour, texture and shape. The colour property can be easily detected by converting the images to one of the colour spaces, such as HSI or HSV (Gonzales and Woods, 2001), and setting the threshold values according to the colours of the object we are trying to detect. The texture properties can be analysed in spatial or frequency domain, where we look for a familiar pattern that makes up the texture of an object. Shapes on the other hand are a bit trickier. By capturing pictures of the scene, we sample a 3D scene with 2D snapshots where we can only look for the right 2D shapes. However, when dealing with clustered or partly occluded objects, such as the ones in natural scenes, 2D analysis can fail, while the unoccluded area of the object is still (partly) the right shape in 3D space. 3D shape analysis, such as the one described by the first group of authors, would be more accurate. The third dimension, the depth, is needed but not directly available in 2D snapshots. Fortunately, the 3D data can be reconstructed by using 2D images from different viewpoints, as we will describe in the next section, and used as an additional verification stage.

In the following chapters we are going to present a multistage algorithm that was developed to detect the number of fruits using colour segmentation, texture analysis and 3D reconstruction of natural scenes. The specific objective of this work was to develop an algorithm suitable for detecting all kinds of natural objects, even those with colour similar to that of the background.

2. The algorithm

The three stage algorithm is summarized on Fig. 1, where the first stage represents a colour analysis, the second texture analysis and the third 3D shape analysis.

The algorithm starts by converting a set of image from RGB to HSI colour space, where we limited the colour segmentation step primarily to the hue plane. By setting the threshold values to include only the colour shades that could represent areas of interest, we make a first approximation step towards the end result.

Colour segmented areas are then further refined using texture analysis and 3D reconstruction. For the texture analysis we have chosen a spatial frequency domain representation, followed by a classification step based on neural networks. They help to produce a binary result that is later on compared with the result of the shape analysis step.

In parallel to texture analysis a 3D reconstruction step is applied. By using corresponding points of the registered images of the same scene, captured at different viewpoints, a reconstruction of a 3D space is made. 3D reconstructed shapes are then analysed where we look for familiar outline. Areas confirmed by shape analysis are then compared to the result of texture analysis and a final binarised result is produced, revealing all areas representing objects with the colour, texture and shape we are looking for.

2.1. *Texture analysis*

Texture analysis is made of two crucial steps. The first calculates a spatial-frequency representation of a given image, while the second looks for familiar patterns by using support vector machine (SVM) approach.

2.1.1. *Spatial-frequency representation*

In contrast to frequency representation, the time and frequency domain (Carranza et al., 2006; Baraniuk, 1995; Greitans, 2008) proves to be a better choice as it is not limited for non-stationary signals that change with time/location. We could use a short-time Fourier transformation, but it is sensitive to the chosen window size. Instead we used Wigner–Ville representation that offers good resolution in time as well as in frequency domain and its result is not affected by the size of the window.

As described by Wigner (1932) and Ville (1948), the Wigner–Ville distribution is defined as an auto-correlation function. Its result is time–frequency representation for every point in time and is calculated as shown by Eq. (2.1).

$$\mathrm{WV}_{\mathrm{1D}}(t,u)=\sum_{\tau=-N/2}^{N/2} f\left(t+\frac{\tau}{2}\right)f^{*}\left(t-\frac{\tau}{2}\right)e^{(-2j\pi u\tau)}, \tag{2.1}$$

where t and u represent time and frequency components, f^{*} the complex conjugate of the original signal and N its length. As we are working with images, the 2D version of Eq. (2.1) can be formulated in the following manner:

$$\begin{aligned}&\mathrm{WV}_{\mathrm{2D}}(n_1,n_2,k_1,k_2)=\\&\sum_{\tau_1=-N/2}^{N/2}\sum_{\tau_2=-M/2}^{M/2} I(n_1+\tau_1,n_2+\tau_2)I^{*}(n_1-\tau_1,n_2\\&-\tau_2)e^{-2j\pi/N\left[k_1(\tau_1-\frac{n_1}{2})+k_1(\tau_2-\frac{n_2}{2})\right]},\end{aligned} \tag{2.2}$$

by introducing parameters n_1 and n_2 as spatial domain coordinates, coordinates k_1 and k_2 for frequency domain representation, $\mathbf{I}$ as an input image, I^{*} its 2D complex conjugate and parameters M and N for image size.

A closer look at Eq. (2.1) and Eq. (2.2) reveals that the 1D version produces a 2D result, while the 2D version of the Wigner–Ville distribution produces a 4D solution. As the amount of data needed to be processed soon gets overwhelming, it is advisable to consider a sub-sampling approach and compute the distribution to every n-th point, where n is selected according to the Nyquist criteria.

The Wigner–Ville distribution is, in contrast, the period of 2π of the Fourier transformation, periodic at π. The result of Eq. (2.2) is therefore sampled at twice the frequency. It is, however, impossible to escape the aliasing in time as well as in frequency domain, but we can use two different approaches to minimize this property. The first approach consists of zero padding a signal in the time/spatial domain, while the second splits the frequency band. This can be done in one of two ways; to oversample (sample at four times the frequency) or to compute the analytic version of the input signal using Hilbert transforms (Zhua et al., 1990). It is not uncommon to use both approaches; especially for the 2D signal as the 2D Hilbert transform only computes an approximation of the analytic signal.

The property of the Wigner–Ville distribution has, however, a drawback that must be accounted for. It produces so called cross-terms (Greitans, 2008, Debnath, 2002) between time–frequency or spatial-frequency responses that can affect the detection. They can be minimised by smoothing the signal in frequency as well as in frequency domain. Based on the recommendations, we can formulate the Smoothed Pseudo Wigner–Ville distribution as follows:

$$\begin{aligned}&\mathrm{PW}_{\mathrm{2D}}(n_1,n_2,k_1,k_2)=4\sum_{u=-P_2+1}^{P_2}\sum_{v=-P_1+1}^{P_1-1} h_{P_1,P_2}(u,v)\\&\sum_{r=-Q_2+1}^{Q_2-1}\sum_{s=-Q_1+1}^{Q_1-1} g_{Q_1,Q_2}(r,s)I(n_1+r+u,n_2+s+v)\\&I^{*}(n_1+r-u,n_2+s-v)e^{-j(2\pi uk_1/M+2\pi vk_2/N)},\end{aligned} \tag{2.3}$$

where we introduce the window functions $\mathbf{H}_{P_1,P_2}$ and $\mathbf{G}_{Q_1,Q_2}$, parameters P_1 and P_2 that designate one half of the dimensions size of the first window, Q_1 and Q_2 one half of the dimensions second window, and we use M and N for signal dimensions.

2.1.2. Support vector machines (SVM)

The computed Wigner–Ville distribution produces different responses for different textures and they must be analysed. To do so, we have chosen a support vector machine or SVM as described by Vapnik (1999), Haykin (1999) and Ivanciuc (2007). The SVM approach can be used for linear regression analysis or pattern classification, as in our case. To analyse an image, we use spatial-frequency distribution for each of the pixels and, based on the spatial-frequency response, conclude if the pixel belongs to an area with the texture we are trying to detect or not. This, in effect, produces a binary image, revealing pixels that make up a familiar texture pattern.

SVM works by applying a directed neural network in order to estimate a hyperplane that separates one group from the other. The process starts with a learning step that initializes the weights of the neural network in a way that the hyperplane is able to classify a member in one of two groups. Every group member is therefore presented with the help of a n-dimensional feature vector and, based on its contents, it is separated with the help of a liner classifier or $(n-1)$ dimensional hyperplane.

SVM can be defined according to Haykin (1999) in the following way. We use $\mathbf{x}_i$ to represent a feature vector, for which we select a response d_i, with values of 1 or -1 that classify the object according to the features in one of two groups. A hyperplane can then be written as:

$$\mathbf{W}^{T}\mathbf{X}+\beta=0, \tag{2.4}$$

where $\mathbf{w}$ represents a weight vector, β a bias and $\mathbf{x}$ an input feature vector. For an unknown feature vector $\mathbf{x}_i$ the equation (2.4) changes to:

$$\mathbf{W}^{T}\mathbf{X}_i+\beta\geq 0,\quad \text{for } d_{\mathrm{i}}=1 \tag{2.5}$$

and

$$\mathbf{W}^{T}\mathbf{X}_i+\beta<0,\quad \text{for } d_{\mathrm{i}}=-1 \tag{2.6}$$

The vectors that comply with Eqs. (2.5) and (2.6) fit the hyperplane perfectly and we name them support vectors. In order to make a classification, we need to calculate the weight parameter W and a bias β. To do so, we use a quadratic optimization (Haykin, 1999) and Lagrange multiplayer approach (Haykin, 1999) that are covered elsewhere and will not be recapitulated here.

Once the support vectors are known, we can proceed with a classification of an unknown object, based on its feature vector. In our case, the object is represented as a pixel in spatial domain, while its feature vector is made of spatial-frequency responses of the 2D Smoothed Psevdo Wigner–Ville distribution. However, the classification of an object will only be as good as the learning set of objects defined by its feature vectors used during the learning step.

2.2. Multiview geometry

In order to successfully reconstruct a scene in 3D space, we must use at least two snapshots taken from different viewpoints. A group of corresponding points, seen on all images, needs to be detected and then reconstructed to 3D space, based on the location discrepancies between corresponding pixel pairs.

The selection of corresponding points is far from simple. The usual approach would be to detect distinct areas of an image, such as edges, extremes, etc. For this, Harris edge detector (Harris and Stephens, 1988) could be used. Of course, spatial relationship between detected markers should be estimated. For example we could have used RANSAC method (Kong et al., 2010) to produce putative matches between corresponding pairs. However, as not all natural areas, like the surface of the fruit, consist of specific markers, this is not always an option. Therefore we propose an approach using image registration.

First, images should be registered, so their overlapping contents is covered. The areas with the same contents are then detected by a simple subtraction method, where results close to 0 reveal areas similar/same on all images. If an area is visible on another image and is of the right colour, it represents a corresponding pair suitable for 3D reconstruction.

2.2.1. Image registration

Several approaches to rigid or nonrigid image registration have been reported (Zitová and Flusser, 2003; Goshtasby, 2005). Our search for corresponding points relies on an image deformation matrix that must reliably mirror the translational relationship between pairs of pixels from two images. An empirical conclusion suggests that a rigid registration of images must first be applied to the extent where no better matching can be obtained by global affine transformations. From this stage on, nonrigid registration must align local vicinities of pixels as accurately as possible. However, the adaptation of their intensity and contrast must be restricted, otherwise this registration may end up without additional shifts of pixels, or with minor changes in pixel intensity and contrast.

For this reason, we chose a registration method that can operate in rigid or in nonrigid mode (Periaswamy and Farid, 2006). The method incorporates eight parameters: six define an affine transform, while the remaining two regulate the pixels' intensity and contrast. The following model is optimised:

$$E(\mathbf{m}) = \sum_{x=1}^{M}\sum_{y=1}^{N}[m_7 I(x,y,t) + m_8 \\ -I(m_1x + m_2y + m_5, m_3x + m_4y + m_6, t-1)]^2, \quad (2.7)$$

where parameters m_1 to m_6 stand for affine transform, m_7 and m_8 describe differences in contrast and illumination, and the notation $I(x,y,t)$ means a selected intensity image identified by parameter t. Eq. (2.7) can be applied in two different modes: affine transformation is obtained if only the parameters from m_1 to m_6 are activated globally, whereas nonrigid registration involves all 8 parameters involved iteratively on local subimage regions.

Our search for corresponding points gives optimum results if image registration is implemented in three steps:

Step 1: a coarse rigid transform is based on the alignment of image centroids.
Step 2: affine transform is applied according to Eq. (2.7) in order to rigidly adjust the registered images, parameters m_7 and m_8 equal 1 and 0, respectively.
Step 3: a nonrigid implementation of Eq. (2.7) gives final, elastic stretches to image regions.

The first step of coarse rigid image registration has been published in (Rakun, 2006; Rakun and Zazula, 2007). It is straightforward and will not be recapitulated here.

Step 2 introduces affine application of Eq. (2.7) to register two images rigidly. It is based on the following global alignment error estimation:

$$E(\mathbf{m}) = \sum_{x=1}^{M}\sum_{y=1}^{N}[I(x,y,t) \\ -I(m_1x + m_2y + m_5, m_3x + m_4y + m_6, t-1)]^2 \quad (2.8)$$

where the parameters m_1 to m_6 describe the affine transformation. Eq. (2.8) unfortunately does not reveal explicit affine parameters, but these can be estimated by using a Taylor series (Periaswamy and Farid, 2006) which produces the following approximation:

$$E(\boldsymbol{m}) = \sum_{x=1}^{M}\sum_{y=1}^{N}(k - \mathbf{c}^T\boldsymbol{m})^2 \quad (2.9)$$

Scalar k and vector $\mathbf{c}$ from Eq. (2.9) have the following form:

$$k = p_t + xp_x + yp_y \quad (2.10)$$

and

$$c = [\, xp_x \quad yp_x \quad xp_y \quad yp_y \quad p_x \quad p_y\,]^T, \quad (2.11)$$

where p_x, p_y, and p_t denote partial derivatives in image pixels $I(x,y,t)$ with respect to x, y, and t. For reasons of simplicity, we omitted the co-ordinates (x,y) in all the derivatives. By solving Eq. (2.9) in the minimum-square-error sense, the parameters **m** yield as follows (Periaswamy and Farid, 2006):

$$\boldsymbol{m} = \left[\sum_{x=1}^{M}\sum_{y=1}^{N}\mathbf{c}\mathbf{c}^T\right]^{-1}\left[\sum_{x=1}^{M}\sum_{y=1}^{N}\mathbf{c}k\right] \quad (2.12)$$

The global parameters **m** from Eq. (2.12) registers a pair of images rigidly. By applying successive iterations, new registration steps can decrease mean-square error until a preselected tolerance is reached.

Of course, when dealing with images captured from different viewpoints, the images include areas that do not perfectly match, and rigid registration is not enough. That is why we decided to apply additional, non-rigid registration steps. In contrast to the rigid version, the non-rigid one works locally, using windows of partial image contents. This, in our experience, can completely miss proper alignment without the preceding rigid registration. The non-rigid registration applied follows an extended scheme from Eq. (2.12):

$$\boldsymbol{m} = \left[\sum_{x=1}^{U}\sum_{y=1}^{V}(\mathbf{c}\mathbf{c}^T)\boldsymbol{w}\right]^{-1}\left[\sum_{X=1}^{U}\sum_{y=1}^{V}(\mathbf{c}k)\boldsymbol{w}\right], \quad (2.13)$$

where U and V denote smaller spatial neighbourhoods in $M \times N$ images. Scalar k and vector $\mathbf{c}$ in Eq. (2.13) have, according to (Periaswamy and Farid, 2006), the following form:

$$k = p_t - p + xp_x + yp_y \quad (2.14)$$

and

$$c = [xp_x \quad yp_x \quad xp_y \quad yp_y \quad p_x \quad p_y - p - 1]^T, \quad (2.15)$$

where designations have the same meaning as in Eq. (2.12) and the parameter p stands for pixel intensity values, again with co-ordinates (x, y) omitted.

The final solution is computed through several iteration steps. In order to estimate **m** in Eq. (2.13), we need weights **w**, and to estimate **w** we need an approximation of affine parameters **m**. The problem can be solved by using the maximum likelihood approach described in (Periaswamy and Farid, 2006), which works by first setting an approximation of weights **w** based on rigid registration results as described in step 2 above. Once the approximation of the current **m** parameters **m** is known, we can re-evaluate weights **w** and minimize the error. The procedure is repeated iteratively until the changes in **m** and **w** fall below a preselected threshold.

2.2.2. 3D shapes and reconstruction

Reliable corresponding points, with a difference of intensity close to zero for a given window, can be used to reconstruct selected regions in 3D space. The proposed algorithm for detection of objects of irregular shape needs a 3D reconstruction that preserves shape. We decided on projective reconstruction and followed the procedure from Forsyth and Ponce (2002) and Mahamud et al. (2001).

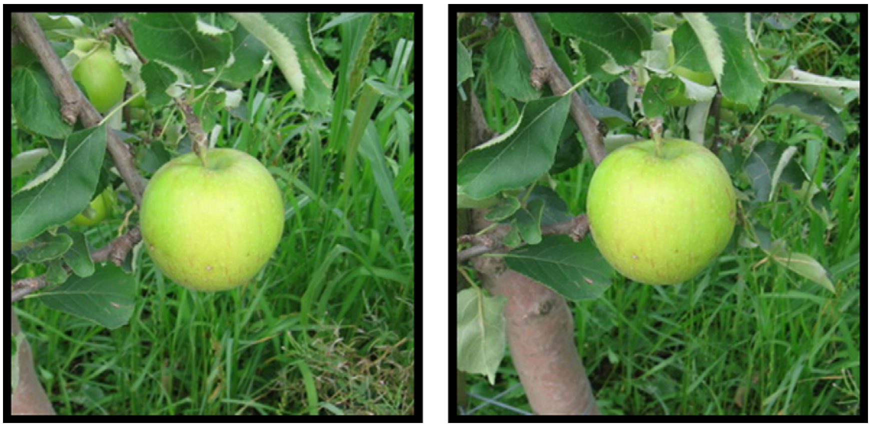

Fig. 2. Two snapshots of the same scene, taken from different viewpoints.

This approach builds on the inverse projection from 3D to 2D, which can be written as follows:

$$\boldsymbol{D} = \boldsymbol{MP} \tag{2.16}$$

here **M** designates a camera matrix having a size of 3 m × 4m, a matrix **P** of size 4 × *n* that stands for a matrix of 3D corresponding points, and **D**, a matrix of size 3 m × *n* that describes a 3D-to-2D transformation; *m* and *n* stand for the number of views and the number of corresponding point pairs, respectively. Matrices **D**, **M** and **P** have the following form:

$$D = \begin{bmatrix} z_{11}\boldsymbol{p}_{11} & \cdots & z_{1n}\boldsymbol{p}_{1n} \\ \vdots & \ddots & \vdots \\ z_{m1}\boldsymbol{p}_{m1} & \cdots & z_{mn}\boldsymbol{p}_{mn} \end{bmatrix} \tag{2.17}$$

is composed of corresponding points of a form $p_{ij} = (x_{ij}\ y_{ij}\ 1)^T$ and initially unknown projective parameters z_{ij};

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 \\ \vdots \\ \boldsymbol{M}_m \end{bmatrix} \tag{2.18}$$

is a set of camera matrices that project the following 3D points:

$$\boldsymbol{P} = [\boldsymbol{P}_1\, \boldsymbol{P}_2\, \ldots\, \boldsymbol{P}_m] \tag{2.19}$$

The 3D points have the following form: $\mathbf{P}_t = [x_t\ y_t\ z_t]^T$. Since **M** and **P** are unknown, we cannot set projective depth parameters z_{ij} directly. Instead, we can use an iterative scheme that minimises the error between projective depths and matrices **M** and **P**, as described by Forsyth and Ponce (2002) and Mahamud et al. (2001). Matrix **D** of Eq. (2.16) can be factorised by singular value decomposition (SVD), and the obtained left unitary matrix will correspond to the camera projection matrices, while the right unitary matrix multiplied by singular values corresponds to the 3D reconstruction points. A weak projection estimate must be provided, an initial estimate with unknowns z_{ij} set to 1, which minimizes the error and iterates towards a global minimum, as suggested by Forsyth and Ponce (2002).

3. Validation of the algorithm

The images used in this experiment were captured on August 7th 2008 in the research orchard (46.50°N, 15.63°E), owned by the Faculty of Agriculture and Life Sciences, University of Maribor. The capturing stage was performed from 9 am till 3 pm on a clear, sunny day. We captured images of 4-year old ‘Gala’ and ‘Golden delicious’ variety of apples, grafted onto M9 rootstock at 0.7 m spacing and a row spacing of 3.2 m. On the selected date, the apples were mostly green (Golden delicious) with some shades of yellow, and red (Gala). All selected scenes were captured from different viewpoints, on average 2 m from the tree at around 20 ± 5° angular difference.

As this paper is limited in space, the validation of the method will be summarized with one of the examples, depicted in Fig. 2, while at the end of the section an average estimate will be given for our whole test pool.

Fig. 2 depicts an area of an apple tree where initial colour segmentation detected shades of colour that might represent an apple fruit. It is not necessary to analyze the whole apple tree image but

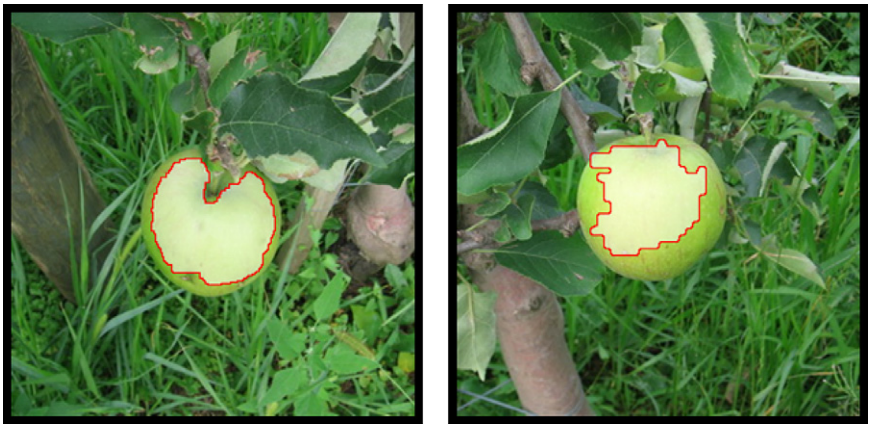

Fig. 3. Two similar images; were the left serves as a learning set (with positive area selected by hand) and the right shows the result of SVM classification.

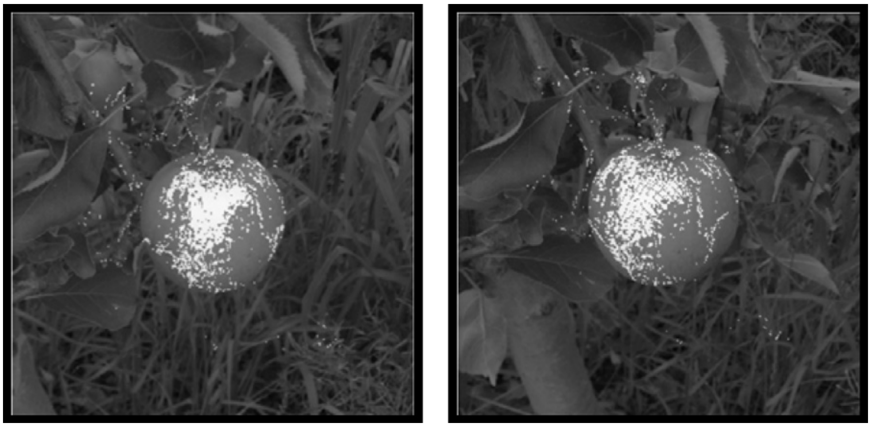

Fig. 4. Detected corresponding points, marked as white pixels.

Table 1
Average texture analysis measurements for 30 randomly selected examples classified in to four groups.

	Group 1	Group 2	Group 3	Group 4
Average fruit coverage	3%	1%	66%	33%

only the regions that need to be verified. If more possible areas would be detected they would be analysed separately, but in a way shown by our example.

3.1. Results of texture analysis

For the texture validation step we will use the right image from Fig. 2 and a new, similar image that will serve as a learning set. The training set image is depicted on Fig. 3(left). Both images are of 300 × 300 pixels in size. As described in Sections 2.1.1 and 2.1.2, we first compute their spatial-frequency representation, where we use the training set to learn the SVM. For this purpose each pixel was described by its feature vector comprised of 50 × 50 spectral components that corresponded to ±25 pixels contents around each pixel. Regularized support vector classification was performed using the radial basis function as the kernel type.

The area marked with partly transparent white colour and circled with red serves as one (positive), while the rest of an image serves as the other (negative) learning set. Based on the learning sets, a classification of a new image can be done as shown by the right image of Fig. 3. The classification in this case was successful, where we note that we do not expect to detect the whole area we are looking for but partial detection is enough to confirm a region produced by colour segmentation.

3.2. Results of multiview geometry analysis

Images from Fig. 2 are now used for 3D reconstruction. As described in Sections 2.2.1 and 2.2.2, we first proceed with image registration. Based on the registered images we can then conclude which of the pixels are visible on both images and how good is the registration. We do so by comparing the contents in the close proximity of each corresponding pixel, where a comparison window of size 5 × 5 pixels is suggested. Good corresponding pairs that also have the right shades of colour are selected. Fig. 4 shows the selected corresponding points marked as white pixels for our test example from Fig. 2.

By taking a look at Fig. 4 we see that not all corresponding pairs are on the fruit surface. This is to do with colour segmentation that is rarely perfect, when dealing with green fruits against green leaves, and proves additional verification steps, such as texture or 3D analysis, must be carried out.

Fig. 5 depicts the result of our test example from Fig. 2. The left image depicts a set of reconstructed voxels in 3D space, while the right image includes a sphere that best fits the reconstructed voxels. The sphere was determined by calculating the circular Hough transformation (Gonzales and Woods, 2001). Based on the number of the voxels on the surface of the sphere or its vicinity we can eas-

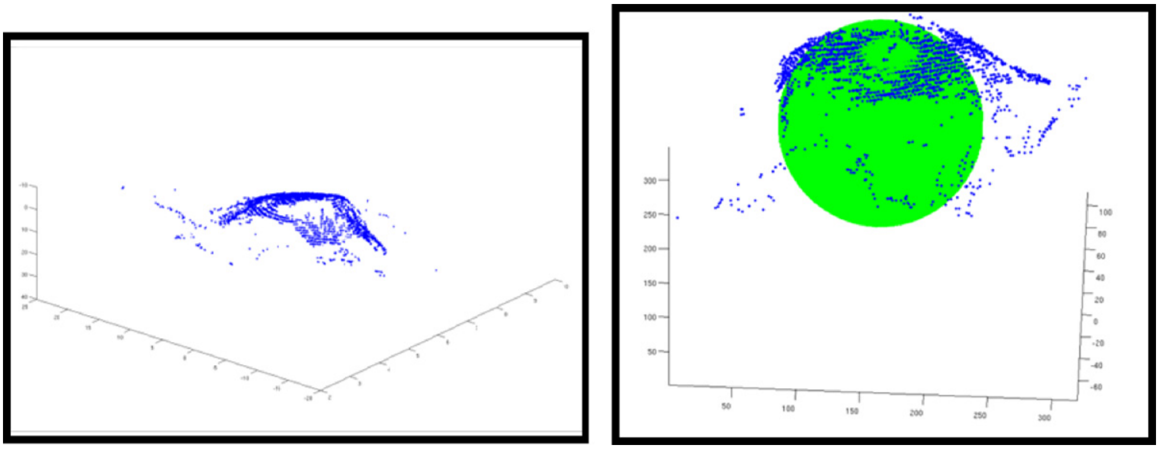

Fig. 5. Reconstructed voxels for our test example from Fig. 2; the left image depicts a 2D view of the reconstructed 3D space, while the right also includes a reconstructed sphere that best fits the reconstructed voxels.

Table 2
Average 3D shape analysis measurements for 30 randomly selected examples classified in to four groups.

	Group 1	Group 2	Group 3	Group 4
The number of detected corresponding pairs	3763 (∼31% of all fruit pixels)	3751 (∼46% of all fruit pixels)	3536 (∼25% of all fruit pixels)	3801 (∼32% of all fruit pixels)
Num. of voxels that align perfectly	577 (15%)	416 (11%)	386 (11%)	151 (4%)
Num. of voxels n close proximity	1983 (53%)	2076 (55%)	1997 (56%)	2280 (60%)

ily conclude if it is of a nearly spherical shape and it could represent an object such as an apple of nearly spherical shapes.

4. Discussion

For the final subsection, we have selected 30 random examples, tested them and summarised the results in Tables 1 and 2. All test examples were classified in one of four groups; the first representing colourful objects, such as yellow–red fruits against green background, the second colourful object with partial occlusions due to other objects in the scene, the third less colourful objects, such as green fruits against green background, and the fourth less colourful objects with partial occlusions. For the first group we used 7 examples, for the second 5, for the third 7 and for the fourth 5.

The results of texture analysis are summarized in Table 1, where average results for all our test cases are summarized. For each result, confirmed pixels were counted and compared to expected manually counted number of pixels. It should be noted that the results are influenced by the training set used. If the learning set is not representative enough, the texture analysis may miss (as it happened to 3 of our test examples from group 2). However, only a small positive response of texture analysis is enough to prove or disprove a region. Based on the measurements, we can conclude that texture analysis based on Wigner–Ville distribution and SVM works as an additional verification step.

Table 2 introduces the average results of the 3D shape analysis for 30 randomly selected examples. The second row lists the number of successfully detected corresponding points on an object we are trying to detect, while the number in brackets represents an approximate value of how much of the objects pixels that is. In the third row the number of voxels that perfectly fits the reconstructed sphere is introduced, while the fourth row the number of voxels in close proximity, that is less or equal to 2 voxels, of the reconstructed sphere can be observed. In both cases the percentage is calculated against the value from the second row.

The proposed approach was implemented using a Matlab 7.9.0 package with LIBSIM (Chang and Lin, 2001) and 2-D image registration library (Periaswamy and Farid, 2004). Due to memory

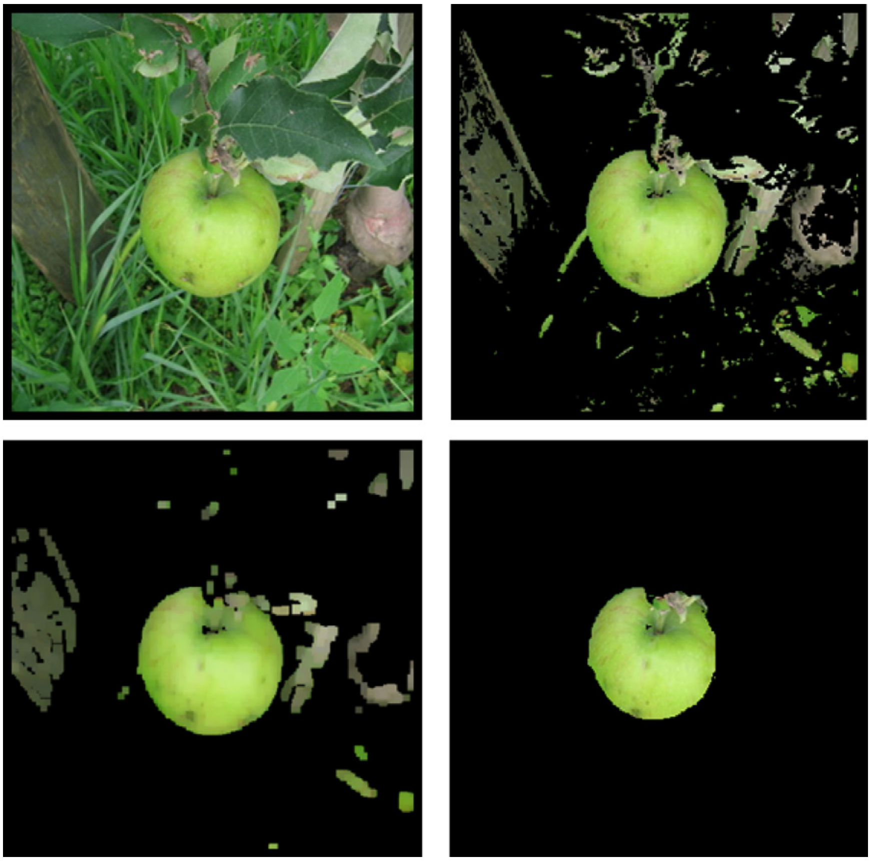

Fig. 6. A test example – the upper left image presents one of the original images of natural scene, on its left is its colour segmented version, in the bottom left its cleaned version (by using morphological operators) and finally, the bottom right image, the result that was confirmed with texture as well as 3D shape analysis. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

restrictions of our workstation (6 GB of RAM) and rather unoptimized implementation, we had to limit the reconstruction to max. 4000 corresponding pixel pairs. This was however enough to confirm the shapes we were looking for. In all test groups we managed to find voxels that fit perfectly to the reconstructed sphere. Even more compelling is the number of voxels in close proximity of the sphere. As the object is of a nearly spherical shape it is ambiguous to expect the reconstructed voxels will align perfectly. By setting the distance threshold value of no more than 2 voxels from the sphere we managed to include more than half of all reconstructed voxels. This is certainly a strong proof to confirm the region and can be used as a threshold value (if less than 50% of reconstructed voxels describe the spherical shape, it does not represent an object we are trying to detect and if it does, it is of the right shape).

We conclude that texture analysis as well as 3D shape analysis offer additional steps to prove or disprove a region. In our final Fig. 6, we depict a set of images. The upper left is one of the original images, on its right a colour segmented version, where we used HSI colour representation and the following thresholds: $H = [0.15,0.25]$, $S = [0.1,0.75]$, $I = [0.25,0.8]$. The bottom left image represents the colour segmented image after using morphological operators with which we can clean the area of small doubtful areas. Areas that remain after this phase are then checked with texture as well as 3D shape analysis. The final result can be seen on the bottom right image of Fig. 6.

5. Conclusion

In this paper we presented a novel approach to detect natural objects of partly spherical shapes such as fruit under natural light conditions in the orchard. It consists of colour segmentation, texture segmentation based on spatial-frequency distribution as well as 3D reconstruction from 2D images of the natural scenes. The approach could be used as a preliminary step to detect natural objects, such as apples, of the fruit prognosis system.

Texture analysis based on Wigner–Ville distribution and SVM offers a reliable tool for image segmentation or, as in our case, region confirmation. Once the neural network of the SVM is trained with well-representative examples, it offers a quick classification tool for spatial-frequency responses of the input signal. The only drawback is the space requirements of the 2D Wigner–Ville distribution that can quickly consume much of the workstation's available memory. It is therefore advisable to limit the number of pixels in spatial domain for which we compute its spatial-frequency representations.

Additional verification is introduced by the 3D reconstruction and consequent 3D shape analysis. As the visible natural objects maybe partly occluded with branches, leaves and other object, they are more easily distinguishable in the third dimension, the depth. The key component of accurate 3D reconstruction is determining good corresponding points. Instead of relying on the geometric, Euclidean properties of a given scene, we apply image registration procedures. In this way, we construct a set of deformation matrices that can be used to determine correspondence ties for, potentially, all registered pixel pairs. Points with the best correspondence are involved in the construction of a projective 3D representation of the observed scene. Thus, the reconstructed objects can be assessed according to their 3D shapes. This means an additional validation step for objects of interest whose snapshots were taken in uncontrolled natural environments.

Two possible future improvements to the proposed approach are logical. The first is to register several images of the same scene with smaller changes of viewpoint and, thus, circumvent the problem of regions that are occluded in one pair of images. Since it is impossible to prevent all occlusions, the second improvement suggests a partial reconstruction of clustered shapes. Furthermore, we could also improve the texture analysis to determine the quality of the harvest with a set of known defects and illnesses as well as make a more detailed shape analysis with the help of normal vectors.

Acknowledgements

This research was partly funded by the Slovenian Research Agency, project number 1000-07-310134Z. The funding is gratefully acknowledged. The authors also acknowledge the vital contributions made Mr. Storey Lindsay, B.B.A. in Accounting for his proofreading of the manuscript.

References

Baraniuk, R.G., 1995. Nonlinear Wigner–Ville spectrum estimation using wavelet soft-thresholding. In: Proceedings of SPIE, vol. 2491 , pp. 661–670.

Benady, M., Miles, G.E., 1992. Locating melons for robotic harvesting using structured light. Transaction of the ASAE, 92–7021.

Carranza, N., Sroubek, F., Cristóbal, G., 2006. Motion estimation by the pseudo-Wigner Ville distribution and the hough transform. In: Eusipco.

Chang, C.-C., Lin, C.-J., LIBSVM: a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.

Debnath, L., 2002. Recent developments in the Wigner–Ville distribution and time–frequency signal analysis. PINSA 1, 35–56.

Forsyth, D.A., Ponce, J., 2002. Computer Vision—a Modern Approach. Prentince Hall.

Gonzales, R.C., Woods, R.E., 2001. Digital Image Processing, Second ed. Prentice Hall PTR, Upper Saddle River.

Goshtasby, A.A., 2005. 2-D and 3-D Image Registration. Wiley-Interscience.

Gómez-Sanchis, J., 2008. Hyperspectral system for early detection of rottenness caused by Penicillium digitatum in mandarins. Journal of Food Engineering 89, 80–86.

Greitans, M., On discrete Wigner–Ville distribution in nonuniform sampling case, http://web.auth.gr/sampta07/MainPage/Abstracts/Greitans,%20Modris.pdf, 20.11.2008.

Harris, C.G., Stephens, M.J., 1988. A combined corner and edge detector. In: Proceedings Fourth Alvey Vision Conference , pp. 147–151.

Haykin, S., 1999. Neural Networks, second ed. Prentice Hall, 318–350.

Ivanciuc, O., 2007. Applications of Support Vector Machines in Chemistry, Reviews in Computational Chemistry, vol. 23. Wiley-VCH, 291–400.

Jiménez, A.R., Ceres, R., Pons, J.L., 1999. A machine vision system using a laser radar applied to robotic fruit harvesting, computer vision beyond the visible spectrum: methods and applications. In: (CVBVS '99) Proceedings , pp. 110–119.

Juste, F., Sevilla, F., 1991. Citrus: a European project to study the robotic harvesting of oranges. In: 3rd International Symposium on Fruit, Nut and Vegetable Harvesting Mechanization , pp. 331–338.

Kong, H., Audibert, J.-Y., Ponce, J., 2010. Detecting abandoned objects with a moving camera. IEEE Transactions on Image Processing 19, 2201–2210, num. 8.

López-García, F., Andreu-Garcíaa, G., Blascob, J., Aleixosc, N., Valientea, J.-M., 2010. Automatic detection of skin defects in citrus fruitsnext term using a multivariate image analysis approach. Computers and Electronics in Agriculture 71 (2), 189–197.

Mahamud, S., Hebert, M., Omori, Y., Ponce, J., 2001. Provably convergent iterative methods for projective structure from motion. In: IEEE Conference on Computer Vision and Pattern Recognition , pp. 1018–1025.

Periaswamy, S., Farid, H., 2-D image registration library, Software available at http://www.cs.dartmouth.edu/farid/research/qr_0.2.tar.gz, 2004.

Periaswamy, S., Farid, H., 2006. Medical image registration with partial data. Medical Image Analysis 10, 452–464.

Rakun, J., 2006. The Computer-aided detection of inferior printing quality and errors. In: 13th IEEE Mediterranean Electrotechnical Conference, May 16–19, Málaga, Spain. Electronic proceedings. [Piscataway]: IEEE , pp. 1236–1240.

Rakun, J., Zazula, D., 2007. Optimization of image registration for print quality control. IWSSIP & EC-SIPMCS: CD Proceedings of 2007 14th International Workshop on Systems, 454–458.

Sites, P., Delwiche, M., 1988. Computer vision to locate fruit on a tree. Transactions of the ASAE 1 (31), 257–263.

Stajnko, D., Rakun, J., Blanke, M., 2009. Modelling apple fruit yield using image analysis for fruit colour, shape and texture. European Journal of Horticultural Science 74 (6), 260–267.

Tanigaki, K., Fujiura, T., Akase, A., Imagawa, J., 2008. Cherry-Harvesting Robot, Computers and Electronics in Agriculture, vol. 63, 65–72.

Van Henten, E.J., Van Tuijl, B.A.J., Hemming, J., Kornet, J.G., Bontsema, J., Van Os, E.A., 2003. Field test of an autonomous cucumber picking robot. Biosystems Engineering 86, 305–313.

Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Transaction on Neural Networks 10, 988–999.

Ville, J., 1948. Théorie et applications de la notion de signal analytique. Cables et Transmission 2A, 61–74.

Wigner, E.P., 1932. On the quantum correction for thermodynamic equilibrium. Physical Review 40, 749–759.

Zhao, J., Tow, J., Katupitiya, J., 2005. On-tree fruit recognition using texture properties and color data. In: Conference on Intelligent Robots and Systems, Edmonton , Alberta, Canada, pp. 263–268.

Zhua, Y.M., Peyrina, F., Gouttea, R., 1990. The use of a two-dimensional hilbert transform for wigner analysis of 2-dimensional real signals. Signal Processing 19 (3), 205–220.

Zitová, B., Flusser, J., 2003. Image Registration Methods: a Survey, Image and Vision Computing, vol. 21. Elsevier, 977–1000.