Optica Applicata, Vol. XXXV, No. 2, 2005

Depth object recovery using a light line and a regression neural network

J. APOLINAR MUÑOZ RODRÍGUEZ[1], RAMON RODRIGUEZ-VERA[2], ANAND ASUNDI[2]

[1]Centro de Investigaciones en Optica, A.C., Leon, Gto, 37150 Mexico;
e-mail: munoza@foton.cio.mx

[2]Nayang Technological University, School of Mechanical and Production Engineering,
Nayang Avenue, Singapore 639798

A technique for measuring the objects shape is presented. In this technique, the object is scanned using a light line. From the scanning a set of images is captured by a CCD camera. By processing these images, the object surface is recovered. To determine the surface dimensions, a regression neural network is applied. This network is built using data from images of a light line projected onto the objects, with known dimensions. The data are extracted from the images by applying Gaussian approximation. By using the neural network in this technique, the surface measurement is determined without using the parameters of the set-up. It improves the accuracy of the techniques of light line projection for shape detection, because errors of parameters of the set-up are not introduced to the system. This technique is tested in an experimental way and its results are verified with a contact method.

Keywords: shape detection, light line projection, regression neural network, Gaussian approximation.

1. Introduction

In optical metrology and robotics vision, several techniques have been developed for object shape detection using optical methods. The use of structured illumination makes the system more reliable and the acquired data are easier to interpret. A particular technique is the light line projection [1–8]. When a light line is projected on an object surface at an angle, its position changes in the image due to surface height variation when viewed from a different direction. The main aim of this technique is the detection of the position of the light line in the image. From the position of the light line and the parameters of the geometry of the optical set-up, the object profile is deduced. In some cases, the line position is measured by using a peak detection method [1]. In this method, the position is assigned to the highest intensity pixel. This peak is obtained using a small window and applying a threshold crossing on the same area. Another method consists in measuring the thickness of a light line as a function of the

height [2]. This thickness is measured directly on a screen by using an eyepiece or a micrometer. Measurements of the line position using a coordinate measure machine (CMM) have been also used [3]. This technique uses gauge blocks as calibrators. The spatial coordinates of the line displacement can be determined by detecting the line position on the gauge block. Another peak method uses an electronic sensor for the detection of the peak position by means of a numerical integration [4]. In time delay and integration (TDI) methods, a camera is used to record line by line to obtain a fringe pattern [5–7]. The object shape is deduced by calculating the phase from the fringe pattern. In all of these methods, the distances of the geometry of the optical set-up are used for measuring the object shape.

In the technique proposed in this job, the object shape is measured using only the light line position and the distances of the geometry of the set-up are not needed. The position of the light line is related to the object height by a well-known geometric relationship [8]. In this technique, the geometrical relationship is represented by means of a regression neural network (RNN). In this manner, parameters of the set-up geometry are avoided. This RNN is implemented using images of the light line projected on objects with known dimensions. The topology of this RNN consists of an input vector, a hidden layer, two summation nodes and an output layer. The data of the input vector correspond to the position of the light line projected onto objects, whose dimensions are known. The hidden layer is formed by neurons of Gaussian shape. The two summation nodes perform the summation of the hidden layer neurons, which are multiplied by a weight. The output layer is determined after training with the division of the two summation nodes. The data of the light line position is measured with sub-pixel resolution by applying the Gaususian approximation method. From this approximation, the light line position is computed with a simple summation. It is an advantage in processing time with respect to the conventional methods, where the light line position is obtained detecting the intensity maximum by means of derivatives. In the set-up, the object is moved along the x-axis and scanned by a light line. In each step of the scanning, a light line image is captured using a CCD camera. Each one of these images is processed by the Gaussian approximation to determine the light line position. This position is processed by the RNN to determine the object shape dimensions. The information produced by each light line is stored in an array memory to obtain the complete object shape. The results obtained in this technique are achieved with very good repeatability.

2. Description of the set-up

In optical techniques for shape detection there are three kinds of geometric configurations. In the first configuration, both the camera and the line projector are aligned at an angle. In the second configuration, the line projector is placed perpendicularly to the object surface, and the camera is aligned at an oblique angle. In the last configuration, the camera is collocated perpendicularly to the object surface

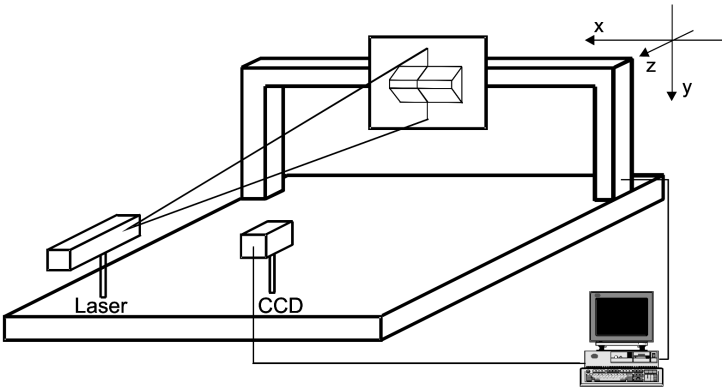


Fig. 1. Experimental set-up.

and the projector is aligned at an oblique angle. The proposed method can be used in these three kinds of configurations. In the set-up used in this technique, the line projector is placed perpendicularly to the object surface and the camera is aligned at an oblique angle. The experimental set-up is shown in Fig. 1. In this arrangement, the object is fixed on a platform, which moves it along the *x*-axis. This platform uses a stepping motor to move the object in steps with an accuracy of a fraction of a millimeter. To perform the scanning, a vertical light line is projected onto the object by means of a laser diode. In each step of the scanning, the vertical light line is deformed in *x*-axis according to the object topography. Every deformed light line is captured and digitized by a CCD camera and a frame grabber, respectively. By detecting the deformation position of the light line, object shape dimension is determined by the RNN. This RNN generates a continuous function, which calculates object shape dimensions. The information of each light line processed by the RNN corresponds to a transverse section of the object. These transverse sections are stored in an array memory to obtain the complete object shape.

To describe the relationship between a light line displacement and the object height, the geometry of the optical set-up is shown in Fig. 2. On the reference plane are located the *x*-axis and *y*-axis, and the object height is indicated by $h(x, y)$. The points *A* and *B* correspond to the light line projected onto the reference plane and object surface, respectively. When a light line is projected on the target, the light line position on the image plane is moved from x_A to x_B. This displacement can be represented as:

$$s(x, y) = x_A - x_B. \tag{1}$$

This light line displacement $s(x, y)$ corresponds to the object height $h(x, y)$. Using this displacement, the RNN determines its corresponding object dimension $h(x, y)$. The RNN generates a continuous function, which describes object dimension for a particular displacement $s(x, y)$ measured on the light line image. Therefore, with the displacement processed by the RNN, the height data are determined. To detect this displacement, light line position is measured in every row of the image. This position

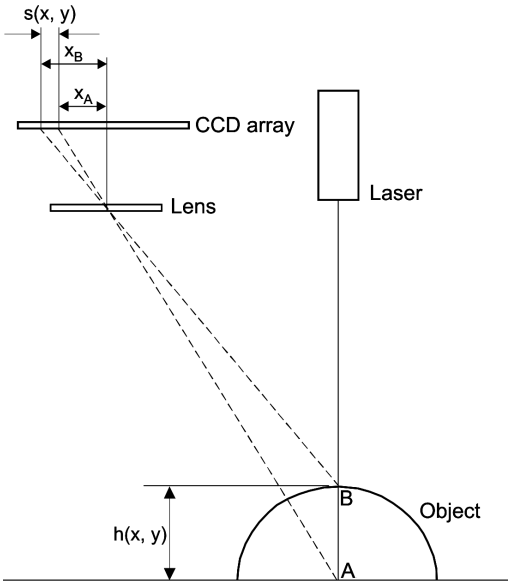


Fig. 2. Geometry of the experimental set-up.

is computed by calculating the mean from a Gaussian function by using the pixels of the light line image. With the processing of all light line images by the RNN, the complete 3D object shape is reconstructed. In Section 3, the measurement of light line displacement is described.

3. Light line image processing

The projected intensity by a laser diode is a Gaussian distribution in lateral direction (x-axis) [9]. The intensity values of the pixels of every row of the light line image are represented as: (x_1, z_1), (x_2, z_2), (x_3, z_3), ... , (x_n, z_n), where x_i is the pixel position and z_i represents the pixel intensity, for $i = 0, 1, 2, 3, ..., n$. A natural way to represent these intensity values is a Gaussian function. The equation of a Gaussian function is

$$f(x) = \frac{Ni}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{2}$$

where Ni is the area under the curve, μ is the mean of the function, and σ is the standard deviation [10]. To determine the position of the light line, the most important parameter is the mean μ of the Gaussian function. It is because μ represents the position of the maximum of the Gaussian function. Therefore, light line position is obtained by computing the mean μ in the image. The mean for a Gaussian function can be calculated using the set of pixels as

$$\mu = \sum_{i=1}^{n} z_i x_i \Big/ \sum_{i=1}^{n} x_i. \tag{3}$$

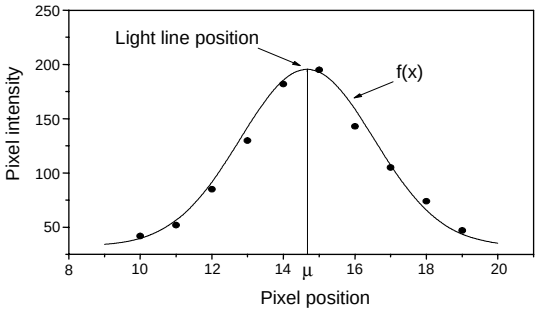


Fig. 3. Light line position from a set of pixels fitted to a Gaussian function.

In this equation x_i represents the pixel position and z_i – the pixel intensity. To measure the light line position using Eq. (3), the position of the intensity maximum is calculated for every row of a light line image. To describe this procedure, Fig. 3 shows a set of pixels of a row of a light line image. From these pixels, the central position μ of the Gaussian function is calculated. To carry it out, the pixel position x_i and intensity pixel z_i are substituted into Eq. (3), for i = 1, 2, 3, ..., n. In this case, n = 10 and the result is μ = 14.664. This central position μ corresponds to the maximum intensity position, with a resolution of a pixel fraction. Therefore, the position of the light line is x_B = 14.664 pixels. Also, the area under the curve Ni and the standard deviation σ are calculated to fit the Gaussian function $f(x)$ shown in Fig. 3. Usually the center for a Gaussian function is calculated using the arithmetic mean [11]. It means, that the maximum position is always the middle point. However, the camera or laser diode is placed at an oblique angle. Therefore, maximum is on left hand side or right hand side of the middle point. The arithmetic mean assumes that all pixels values are of equal importance [11]. The result of the arithmetic mean for the set of pixels of Fig. 3, is μ = 14.5 pixels. This position is slightly different from the result obtained by Eq. (3), which is called weighted arithmetic mean. For the weighted arithmetic mean, a weight

▲
Fig. 4. Light line projected onto an object surface.

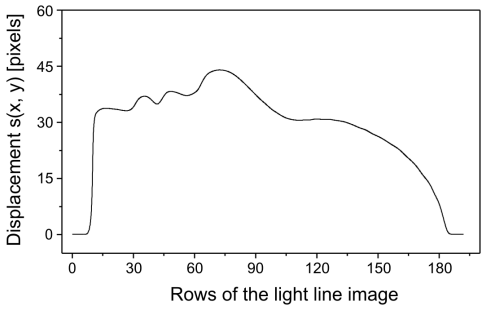


Fig. 5. Displacement profile corresponding to the light line of Fig. 4.

is assigned to every point. In this case the weight is the pixel intensity. It means that a pixel with major intensity is more important than a pixel with low intensity. Therefore, the weighted arithmetic mean is used to measure the light line position. The procedure to detect the position of the intensity maximum is applied to all rows of the image to obtain the displacement $s(x, y)$ of the light line position. A light line image is shown in Fig. 4; for each row the mean μ is computed to find the line position. This line position is substituted in Eq. (1) to obtain the displacement, which represents the object profile measured in pixels as shown in Fig. 5. This profile is extracted from each one of the captured images in the scanning step. This image processing is performed with very few operations via Eqs. (3) and (1). The algorithm for processing a light line image to measure the displacement of a light line can be described by the next steps:

– Step 1: choose threshold from light line intensity, y = number rows, x = number column;

– Step 2:

```
Repeat, for i = 1 to y
   Sum_z_i x_i = 0; Sum_z_i = 0;
   Repeat, for j = 1 to x
      If image(j, i) > threshold
        Compute: Sum_z_i x_i = Sum_z_i x_i + j*image(j, i);
                 Sum_z_i = Sum_z_i + image(j, i);
        end
      end
   Compute: μ(i)= Sum_z_i x_i/Sum_z_i;
end.
```

4. RNN architecture

The RNN was generalized by Specht in 1991 for systems modelling and identification [12]. The RNN is an architecture that can solve any approximation problem. The basic idea of a RNN is that a data set can be approximated by a continuous function. Assuming that a function has x inputs from x_1 to x_n and one output y, the expected mean value of the output for a particular input x can be found using the next equation:

$$\overline{y} = \frac{\int_{-\infty}^{\infty} y f(x, y) \mathrm{d}y}{\int_{-\infty}^{\infty} f(x, y) \mathrm{d}y}. \tag{4}$$

In this equation $f(x, y)$ is a joint probability density function (PDF), the probability of the output being y and input being x. The PDF can be approximated by a sum of

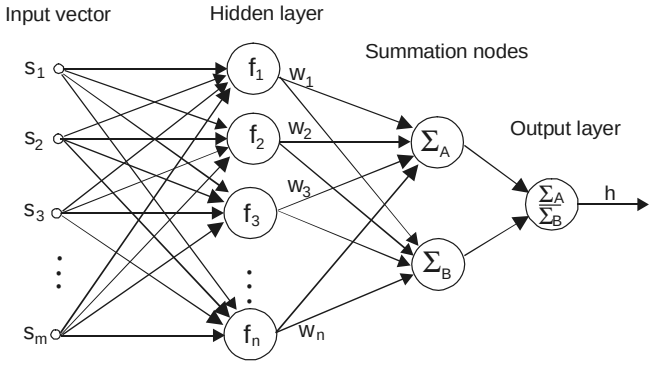


Fig. 6. Structure of the RNN used in this technique.

Gaussians [12]. This process is done by placing the center of Gaussian over the input data and multiplying the Gaussian by the corresponding output. The Eq. (4) can be approximated by the next equation

$$\hat{y} = \frac{\sum_{i=1}^{n} w_i \exp\left(-d_i^2 / 2\sigma^2\right)}{\sum_{i=1}^{n} \exp\left(-d_i^2 / 2\sigma^2\right)}. \tag{5}$$

In this equation d_i is the distance between each input, n is the number of input/output, σ is the width of the Gaussian function, and w_i is a weight that multiplies at Gaussian function [12]. The architecture to construct a RNN to obtain the Eq. (5) is shown in Fig. 6. In this structure, the output h is the object dimension and s_i is its corresponding light line displacement. This RNN is characterized by an input vector, a hidden layer, two summation nodes and an output layer. The data of the input vector $\mathbf{S} = (s_1, s_2, s_3, ..., s_m)$ correspond to the displacement $s(x, y)$ of the light line, which is projected onto objects with known dimensions $\mathbf{H} = (h_1, h_2, h_3, ..., h_m)$. The hidden layer is constructed by a set of neurons, whose shape is a Gaussian function centered at the input data. The effective range of the Gaussian function is determined by the values allocated to the center and width of it. The function used to obtain each neuron is determined with next equation

$$f_i = \exp\left[-\frac{\|s - s_i\|^2}{2\sigma^2}\right] \tag{6}$$

where $||s - s_i||$ is the distance d_i used in Eq. (5), s_i and σ are center and the width of the Gaussian function, respectively. The two nodes contain the summation of the neurons

of the hidden layer, which are multiplied by a weight. The output layer is determined with the division of the two summation nodes. In this manner, the output of the architecture of this RNN is defined by the next equation

$$h = \frac{\sum_{i=1}^{n} w_i f_i}{\sum_{i=1}^{n} f_i}. \tag{7}$$

To construct the RNN via Eq. (7), neuron width and weights should be determined. The width of the Gaussian neuron is the smother parameter σ, which is calculated as

$$\sigma = \frac{d}{\sqrt{2n}} \tag{8}$$

where d is the distance between each couple of inputs $\|s_i - s_{i-1}\|$ for $i = 1, 2, 3, ..., n - 1$ (n is the number of inputs). To determine the weights, the ADALINE training process is applied. The name ADALINE is derived from the words ADaptive LINear Elements [12]. This kind of training is called supervised because the outputs are known and the network is being forced into producing the correct outputs. This learning method is the proper method for training this network, because the RNN is forced to produce the desired outputs. The requirements for this training process are a suitable weight adjustment mechanism. To carry it out, the input vector $\mathbf{S} = (s_1, s_2, s_3, ..., s_m)$ is connected to hidden layer to produce the desired $\mathbf{H} = (h_1, h_2, h_3, ..., h_m)$. The RNN output is defined as the division of two summations nodes. Therefore, for each input s_i an output is generated as a linear equation system:

$$\begin{aligned}
h_1 &= \frac{w_1}{k_1}\exp\left(-\frac{\|s_1 - s_1\|^2}{2\sigma^2}\right) + \frac{w_2}{k_1}\exp\left(-\frac{\|s_1 - s_2\|^2}{2\sigma^2}\right) + \ldots + \frac{w_n}{k_1}\exp\left(-\frac{\|s_1 - s_n\|^2}{2\sigma^2}\right),\\
h_2 &= \frac{w_1}{k_2}\exp\left(-\frac{\|s_2 - s_1\|^2}{2\sigma^2}\right) + \frac{w_2}{k_2}\exp\left(-\frac{\|s_2 - s_2\|^2}{2\sigma^2}\right) + \ldots + \frac{w_n}{k_2}\exp\left(-\frac{\|s_2 - s_n\|^2}{2\sigma^2}\right),\\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \vdots\\
h_m &= \frac{w_1}{k_m}\exp\left(-\frac{\|s_m - s_1\|^2}{2\sigma^2}\right) + \frac{w_2}{k_m}\exp\left(-\frac{\|s_m - s_2\|^2}{2\sigma^2}\right) + \ldots + \frac{w_n}{k_m}\exp\left(-\frac{\|s_m - s_n\|^2}{2\sigma^2}\right)
\end{aligned} \tag{9}$$

where

$$k_1 = \exp\left(-\frac{\|s_1 - s_1\|^2}{2\sigma^2}\right) + \exp\left(-\frac{\|s_1 - s_2\|^2}{2\sigma^2}\right) + \ldots + \exp\left(-\frac{\|s_1 - s_n\|^2}{2\sigma^2}\right),$$

$$k_2 = \exp\left(-\frac{\|s_2 - s_1\|^2}{2\sigma^2}\right) + \exp\left(-\frac{\|s_2 - s_2\|^2}{2\sigma^2}\right) + \ldots + \exp\left(-\frac{\|s_2 - s_n\|^2}{2\sigma^2}\right),$$

$$\vdots \qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \vdots$$

$$k_m = \exp\left(-\frac{\|s_m - s_1\|^2}{2\sigma^2}\right) + \exp\left(-\frac{\|s_m - s_2\|^2}{2\sigma^2}\right) + \ldots + \exp\left(-\frac{\|s_m - s_n\|^2}{2\sigma^2}\right).$$

The linear system of Eq. (9) can be represented as

$$\begin{aligned} h_1 &= w_1 f_{1,1} + w_2 f_{1,2} + \ldots + w_n f_{1,n}, \\ h_2 &= w_1 f_{2,1} + w_2 f_{2,2} + \ldots + w_n f_{2,n}, \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ h_m &= w_1 f_{m,1} + w_2 f_{m,2} + \ldots + w_n f_{m,n}. \end{aligned} \tag{10}$$

This equation can be rewritten in a matrix form as the product between the matrix of the input data and the matrix of the corresponding output values **H** = **FW**. In ADALINE method the weights are calculated by a linear equation system. The linear system represented as a matrix form is described as:

$$\begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{m,1} & f_{m,2} & f_{m,3} & \cdots & f_{m,n} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{bmatrix}. \tag{11}$$

By solving linear system Eq. (11), the weights w_i are obtained. This system is solved by Cholesky method [13]. This method uses a lower (**L**) and upper (**U**) decomposition to transform the matrix system. In **LU** descomposition a symmetric and positive definided matrix can be efficiently decomposed into a lower and upper triangular matrix. In this **LU** decomposition **F** = **LU** is determined as

$$\begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{m,1} & f_{m,2} & f_{m,3} & \cdots & f_{m,n} \end{bmatrix} = \begin{bmatrix} l_{1,1} & 0 & 0 & \cdots & 0 \\ l_{2,1} & l_{2,2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{m,1} & l_{m,2} & l_{m,3} & \cdots & l_{m,n} \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & u_{m,n} \end{bmatrix} \tag{12}$$

where $u_{1,1} = \sqrt{f_{1,1}}$, $u_{i,1} = \dfrac{f_{i,1}}{u_{1,1}}$ for $i = 1, 2, 3, ..., n$ and $\sum_{k=1}^{i} u_{i,k}^2 = f_{i,i}$, $f_{i,j} = \sum_{k=1}^{j} u_{i,k} u_{j,k}$, for $j < I$.

The Cholesky factorization can be described by the next steps:

for k = 1 to n
 for i = 1 to k–1
  compute:

$$f_{k,i} = u_{k,i} = \left(f_{k,i} - \sum_{j=1}^{i-1} u_{i,j} u_{k,j} \right);$$

$$l_{k,k} = u_{k,k} = \sqrt{f_{k,k} - \sum_{j=1}^{k-1} u_{k,j}^2};$$

 end
end.

By using the lower and upper triangular matrix, linear system Eq. (11) is transformed into **LUW** = **H**. Then, the triangular system is solved as **LY** = **H** and **UW** = **Y**. In this manner, **Y** is calculated by using **L** and **H**. Then, **W** is solved using **Y** and **U**. With this step the weights w_1, w_2, w_3, …, w_n are calculated in a fast form. In this manner, the learning procedure has been completed and the RNN has been constructed. The result of this RNN is a continuous function, which describes object dimension according to the light line displacement The RNN obtained using Gaussian neurons is shown in Fig. 7. In this figure, the displacement s_i is indicated by the symbol "△" and continuous line is the function obtained by the RNN. In this manner, for a particular given s_i an h_i is calculated by the RNN. To construct this RNN, different functions can be used as neuron, such as: cubic, inverse quadratic, inverse multi-quadratic and Gaussian [14]. To select the suitable shape of each neuron, the criterion MSE (mean squared error) is used [15]. This criterion provides information on the error of the RNN output with respect to the desired output. The criterion MSE is defined by

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (hc_i - h_i)^2 \tag{13}$$

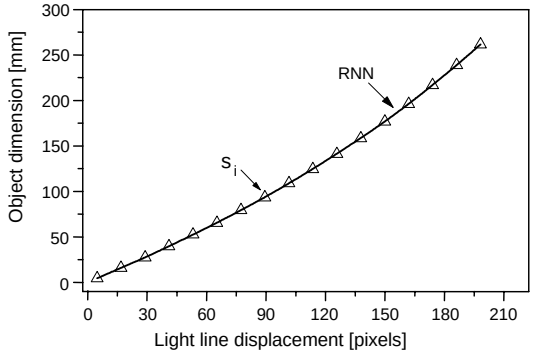


Fig. 7. RNN obtained with the structure of Fig. 6.

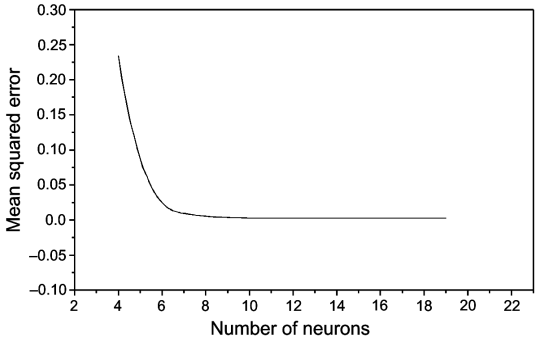


Fig. 8. MSE value of the RNN according to the number of neurons.

where hc_i is the output response of the RNN, h_i is the desired output and m is the number of analysed data. Based on MSE, the function used as neuron that provides the best approximation with respect to the desired output is the Gaussian function. The MSE values for the RNN using each one of these functions are shown in the Table. In the first column, the definitions of the functions to be compared are placed. Second column shows the names of each one of these functions. Third column shows the number of neurons used to obtain the RNN. Fourth column shows the MSE value for each one of these functions. Based on the Table, the Gaussian function provides the best approximation. Therefore, the Gaussian function Eq. (6) is used to construct the neurons of the hidden layer of this RNN. The number of neurons used in the hidden

T a b l e. Neuron shape definition and number of neurons are given to determine the MSE criterion. In the first column, the definitions of the functions to be compared are placed. Second column shows the name of each one of these functions. Third column shows the number of neurons used to obtain the RNN. Fourth column shows the MSE value for each one of these functions.

Function shape	Function name	Number of neurons	MSE
d^3	Radial cubic function	11	0.00394
$(d^2+\sigma^2)^{0.5}$	Inverse multi-cuadratic function	11	0.01813
$1/(1+d^2)$	Inverse cuadratic function	11	0.00324
$\exp(d^2/2\sigma^2)$	Gaussian function	11	0.00221

layer for this RNN is determined again by the MSE criterion. This criterion also provides the error of the output response according to the number of neurons in the hidden layer. Figure 8 shows the behaviour of the RNN error, according to the number of neurons by using the MSE criterion. For this network architecture, MSE criterion indicates that the error is minimum and stable with more than ten neurons. Therefore, eleven neurons can be chosen to construct this RNN.

5. Experimental results

The main motivation to use a regression neural network in this technique is based on good adjustment to a set data. A RNN generates a better output function than those in the classical approximation methods. In the classical approximation methods, a set of data is adjusted to an *n*-degree polynomial function. To determine the difference between RNN and classical approximation methods, MSE criterion is used. The MSE criterion indicates the best adjustment of a function to a data set. For comparison, a data set is adjusted by RBF neural network and least squares approximation method. The fitted functions obtained by the RNN and least squares approximation method are shown in Fig. 9. The MSE value for a neural network is 0.00294 and MSE for least squares method is 0.423. It indicates that RNN network is the better adjustment. It has a great influence on the accuracy of the final result. Therefore, the neural network is used to apply at the light line technique. Figure 1 shows the experimental set-up. The object is moved by steps of the 1.27 mm along *x*-axis. A vertical light line is projected on the surface target by a 5 mW laser diode. The light line is deformed according to the object topography at each movement. These deformed lines are captured by a CCD camera and digitized by a frame grabber with 320×200 pixels and 256 gray levels. The object dimension *h* is recovered substituting the displacement $s(x, y)$ into RNN Eq. (7). From each processed image a profile of the object is extracted. All profiles are stored in array memory to obtain the complete 3D object shape. The experiment is performed with three objects. The first object to be profiled is the surface of a dummy bell, shown in Fig. 10. Each captured light line is processed to obtain its corresponding profile. To extract each one of these profiles, every image is processed by the Gaussian

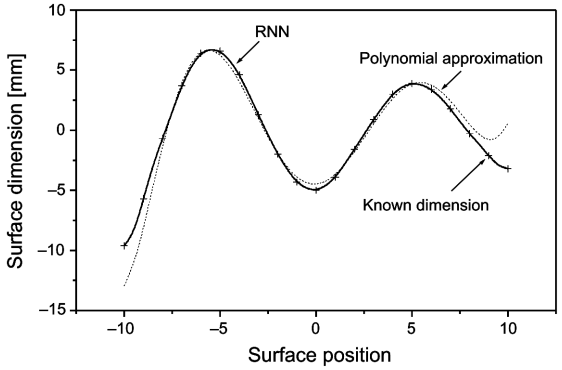
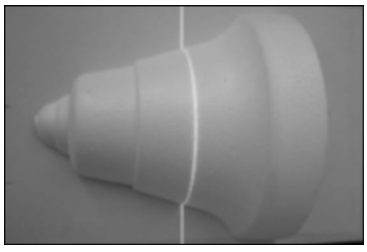


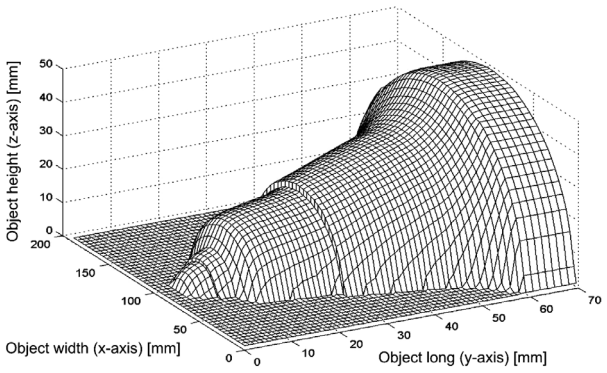
Fig. 9. Fitted function by RNN and polynomial approximation.

approximation method as is described in Sec. 3. To carry it out, the central position μ of the Gaussian function is computed in each row of the line image by using Eq. (3). With this central position μ, the displacement is computed for every row of the image by using Eq. (1). The result of this step is a vector $\mathbf{S} = (s_1, s_2, s_3, ..., s_n)$, which contains the displacements of every row of the light line image. Then, every vector is substituted into the RNN to obtain its corresponding height dimension h of the transverse section of the object. The data obtained from each light line are stored in array memory to construct the complete 3D object shape. To know the accuracy of these calculated values, the root mean squared error (rms) is calculated using these height data and data obtained by a coordinate measure machine (CMM). The rms is described by the following equation [16]:

$$\text{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(ho_i - hc_i)^2} \tag{14}$$

where ho_i is the data obtained CMM, hc_i is the calculated data $h(s)$ by RNN and n is the number of data. The rms is calculated for this method and the result is rms = 0.193 mm. Sixty-four lines are processed to obtain the complete 3D object shape of the bell, which is shown in Fig. 11. In this figure, the scale of the axis is in mm. The second object profiled is a trapezium shown in Fig. 12. By applying the Gaussian approximation, the displacement information is obtained from the light line images. By processing this displacement with the RNN, the height data of a transverse section is obtained. In this case, forty–eighty lines were processed to determine the complete 3D object shape shown in Fig.13. The rms is calculated for this object and the result is rms = 0.152 mm. The third object to be profiled is a dummy face, which is shown in Fig. 14. The object shape is reconstructed by applying the Gaussian approximation and RNN. In this case, sixty–eighty lines were processed to determine the complete 3D object shape shown in Fig. 15. The rms is calculated for this object and the result

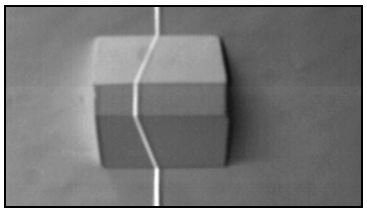


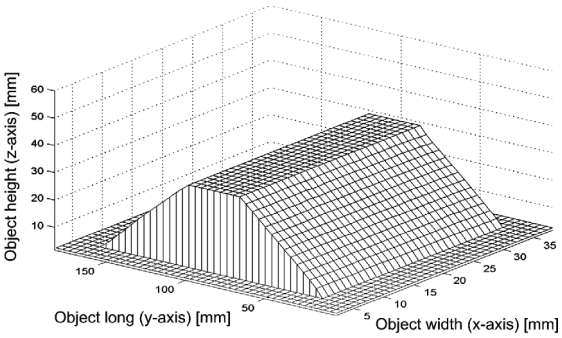


▲

Fig. 10. Bell used to extract its 3D shape object.

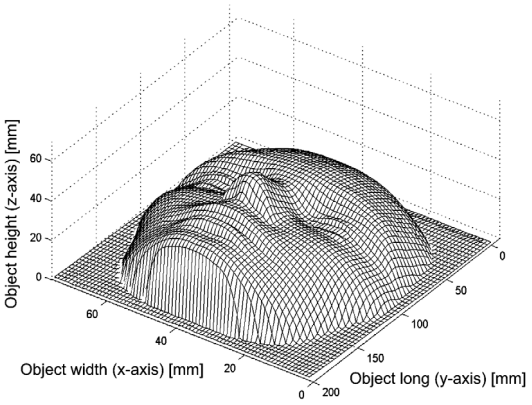
Fig. 11. 3D shape of the bell corresponding to Fig. 10.





▲
Fig. 12. Trapezium used to extract its 3D shape object.

Fig. 13. 3D shape of the trapezium corresponding to Fig. 12.



▲
Fig. 14. Dummy face used to be profiled.

Fig. 15. 3D shape of the dummy face corresponding to the Fig. 14.

is rms = 0.179 mm. In this figure, the scale of the axis is in mm. The employed computer in this process is a PC to 333 MHz. Each light line image is processed in 0.031 s. This time processing is given because the data from the image is extracted with a few operations via Eq. (3). In this procedure, the object shape is obtained without using the parameters as projection angle, distances of the camera and laser line. Therefore, the procedure is easier than in those techniques that use distances of the components of optical set-up. Therefore, in this technique all steps are performed by computer processing and measurements of optical steps are avoided. In this manner a good repeatibility is achieved in each measurement.

6. Conclusions

A technique for shape detection based on a RNN of a light line has been presented. The described technique here provides a valuable tool for shape detection in inspection

industrial. The proposed technique avoids the measurements of the components of the experimental set-up, as is common in the light line projection methods. In this technique, the parameters of the set-up are obtained by computer processing. It improves the accuracy of the technique because measurement errors are not introduced in the system. In this job, the ability to measure the light line position with sub-pixel resolution has been achieved by Gaussian approximation in a very fast manner. It is achieved because the light line position is determined with a few operations. By using this experimental set-up a good repeatibility in every measurement is achieved, therefore this technique is performed in a good manner.

Acknowledgments – J. Apolinar Muñoz Rodríguez would like to thank for the financial support of CONACYT Mexico and CONCYTEG Guanajuato.

References

[1] CHENG X.X., SU X.Y., GUO L.R., *Automated measurement method for 360 profilometry of 3-D diffuse objects*, Applied Optics **30**(10), 1991, pp. 1274–8.

[2] THOMAS L.P., GRATTON R., MARINO B.M., SIMON J.M., *Measurements of free-surface profiles in transient flow by a simple light-slicing method*, Applied Optics **33**(13), 1994, pp. 2455–8.

[3] TAI W.CH., CHANG M., *Non contact profilometric measurement of large form parts*, Optical Engineering **35**(9), 1996, pp. 2730–5.

[4] BABA M., KONISHI T., KOBAYASHI N., *A novel fast rangefinder with non-mechanical operation*, Journal of Optics **29**(3), 1998, pp. 241–9.

[5] MAROKKEY S.R., TAY CH.J., SHANG H.M., ASUNDI A.K., *Time Delay an integration imaging for inspection and profilometry of moving objects*, Optical Engineering **36**(9), 1997, pp. 2573–8.

[6] SAJAN M.R., TAY C.J., SHANG H.M., ASUNDI A., *Improved spatial phase detection for profilometry using a TDI imager*, Optics Communications **150**(1-6), 1998, pp. 66–70.

[7] ASUNDI A., ZHOU W., *Mapping algorithm for 360-deg profilometry with time delay integration imaging*, Optical Engineering **38**(2), 1999, pp. 339–44.

[8] MUÑOZ-RODRIGUEZ J.A., RODRIGUEZ-VERA R., SERVIN M., *Direct object shape detection base on skeleton extraction of a light line*, Optical Engineering **39**(9), 2000, pp. 2463–71.

[9] HERZOG W.D., UNLU M.S., GOLDBERG B.B., RHODES G.H., HARDER C., *Beam divergence and waist measurement of laser diodes by near field scanning optical microscopy*, Applied Physics Letters **70**(6), 1997, pp. 688–90.

[10] DIXON W.J., MASSEY JR. F.J., *Introduction to Statistical Analysis*, McGraw-Hill, New York 1969.

[11] MONTGOMERY D.C., RUNGER G.C., *Applied Statistics and Probability for Engineers*, Wiley, New York 2001.

[12] PICTON P., *Neural Networks*, Polgrave, U.S.A. 2000.

[13] PRESS W.H., FLANNERY B.P., TEUKOLSKY S.A., VETTERLING W.T., *Numerical Recipes in C*, Cambridge University Press, Cambridge 1993.

[14] ROJAS I., POMARES H., GONZALEZ J., BERNIER J.L., ROS E., PELAYO F.J., PRIETO A., *Analysis of the functional block involved in the design of radial basic functions networks*, Neural Processing Letters **12**(1), 2000, pp. 1–17.

[15] LEUNG H., DUBASH N., XIE N., *Detection of small objects in clutter using a GA-RBF neural network*, IEEE Transactions on Aerospace and Electronic Systems **38**(1), 2002, pp. 98–118.

[16] MASTERS T., *Practical Neural Network Recipes in C++*, Academic Press, Boston 1993.

Received December 9, 2004