On-Line Measurement of Wrinkle Using Machine Vision

Hoang Minh To[1], Dong Keun Shin[2], and Sung Lim Ko[2]

[1] Dept. of Advanced Technology Fusion, Konkuk University
[2] CAESIT (Center for Advanced E-System Integration), Konkuk University
minhtoh@yahoo.co.uk, fateful99@naver.com, slko@konkuk.ac.kr

Abstract. Roll to roll (R2R) manufacturing, also known as 'web processing', is the process of creating electronic devices on a roll of flexible plastic or metal foil. With the need for increased performance and productivity in the R2R industry, effective control and on-line supervision for web quality is essential. In this paper, we presents a system for on-line measurement of wrinkles, a defect inccurring due to compressive stresses developed in the web . This machine vision system, based on structured light ranging and multi-threaded processing is able to measure wrinkle height on a transparent web.

Keywords: roll to roll, wrinkle measurement.

1 Introduction

The promising R2R fabrication of electronic devices on continous plastic webs offers the possibility of great cost reduction. However, critical technical challenge for R2R also arises as how sub-micron features can be embosed on a large web having poorly controlled thickness and flatness. Research on modelling, control for web handling applications and the use of digital images for supervising the quality of the R2R process is being emerged. Many commercial machine vision systems are already developed, however as far as we've known, they are more focus on detection and reporting of the web's surface defects : dirts, spots and coating streaks… Meanwhile, other process defects such as wrinkle, web vibration, web break or fold are paid less attention as they're also hard to be inspected fully. In fact, wrinkling (figure 1) is a complex phenomenom that may induced by misalignment between rollers, anisotropic materials, variations in web tension across the web width or along the web length…and is difficult to be prevented [1]. Obviously, wrinkles do not all react in the same way to deformations, varying according to their shape and their alignment with the directions of elongation. Therefore, an effective measurement method is required as a mean of gathering information on wrinkle geometry to provide feedback to the control center.

Many problems were confronted due to the tranparency of the web (which is of 80% in our case). Triangulation laser sensors loose their intensity for most of the beam will go through the web. High velocity of web movement, ranging from 200 to 1000 mpm, also limit the use of these point sensors, which were previously reported in off-line wrinke measurement in textile processing [2] . When using a line scan

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4706, Part II, pp. 931–940, 2007.

camera with front light illumination, it is possible to detect the occurance of wrinkles, however the captured shape is not correct since it is based on shading when web surface is deformed. It's also difficult to distinguish with other features on the web surface in case it's already printed.

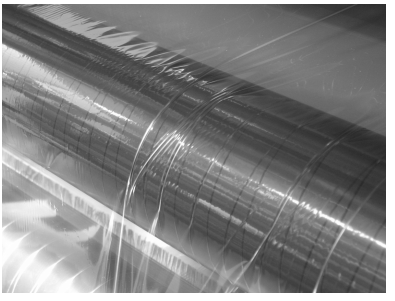


Fig. 1. Wrinkle on a web roller

The purpose of this paper is introducing a novel measurement method for wrinkling based on structured light vision. In section 2, we will describe measurement set-up including a measurement software buit on multi-threading mode for on-line inspection of the wrinkle heights and locations. Section 3 describes the processing pipeline. Section 4 gives some experimental evaluation and section 5 concludes the paper.

2 Exprimental Design

2.1 The Exprimental Set-Up

Figure 2 shows the measurement system which consists of a camera and a diode laser used as the structure light source projecting onto the web. The width of web is 300mm. A Basler Scout camera with resolution of 659 x 494 pixels, sensor size is of 10 μm is used. The camera has the maximum frame rate of 79 monochrome images per second at full resolution and connects to the computer via a Gigabit LAN card. The laser generates a plane of light, which in turn creates a light stripe when intersects the web surface. Though most of the laser penetrates the transparent web, the reflectance data from this light stripe can be used to reconstruct the shape of the wrinkles. When there is no wrinkle, the light stripe is a straight line. However, when wrinkles occur, the distortion of the light stripe reflects the wrinkle shape (fig. 3).

We set-up a cover behind the transparent web that curtain off all unnecessary information to reduce the work of image processing.. Therefore, the captured image is clear with bright pixels of the laser stripe on a dark background.

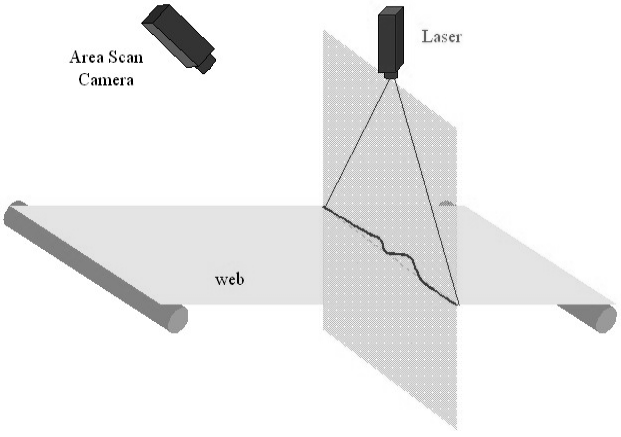


Fig. 2. Layout of the wrinkle measurement system

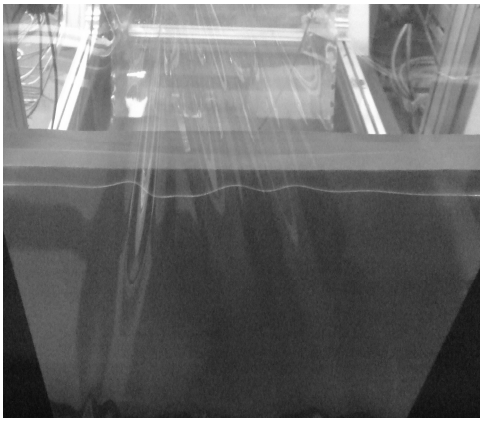


Fig. 3. Image of a laser stripe on the rolling web

2.2 System Calibration

The process of calibration consists of finding a relation between 3D points on the measuring surfaces with the projection of these points in the acquired image. Usually this is done in a two separate procedures to find the camera and the projector model. However in our case, the camera and light stripe plane are fixed in the world coordinate frame, the system can be calibrated in an one step procedure proposed by DePiero et al. In this method, a model $z = f(u,v)$ is used for relating height z to image coordinates (u,v), where f is found by using emprical calibration data. For example, when f is of second order, z is calculated as eq. (1). A similar model g is used to generate y coordinate. A variety of calibration models have been studied in details for both acquitision speed and ranging accuaracy in [4]. For our system, f is of 2^{nd} order and g is of 1^{st} order, which are described in eq. (1) and (2) :

$$\begin{bmatrix} u^2 & v^2 & uv & u & v & 1 \end{bmatrix} M = z \,. \tag{1}$$

$$\begin{bmatrix} u & v & 1 \end{bmatrix} N = y \,. \tag{2}$$

The parameters M and N are determined by measuring the calibration jig to find many triplets (z_i, u_i, v_i) and arrange them in an overdeterminded set of equations :

$$\begin{vmatrix} z_1 \\ .. \\ z_i \\ .. \\ z_n \end{vmatrix} = \begin{vmatrix} u_1^2 & v_1^2 & u_1 v_1 & u_1 & v_1 & 1 \\ .. & & & & & \\ u_i^2 & v_i^2 & u_i v_i & u_i & v_i & 1 \\ & & & & & \\ u_n^2 & v_n^2 & u_n v_n & u_n & v_n & 1 \end{vmatrix} \begin{vmatrix} m_1 \\ .. \\ m_6 \end{vmatrix} \,. \tag{3}$$

Figure 4 illustrates the calibration procedure for the model given in eq.(3). The calibration jig consists of a flatform of which a corner is chosen as the world coordinate center and the top surface is defined as *xy* plane. Several precise plates are stacked together on this surface to form different heights z_i that are imaged by the camera at positions (u_i,v_i). Calibrating the model $y = g(u,v)$ is done in similar manner with a series of vertical plates put into slots machined in the flatform. The world *y* positions of these slots are already known by measuring them with an coordinate measuring machine (CMM).

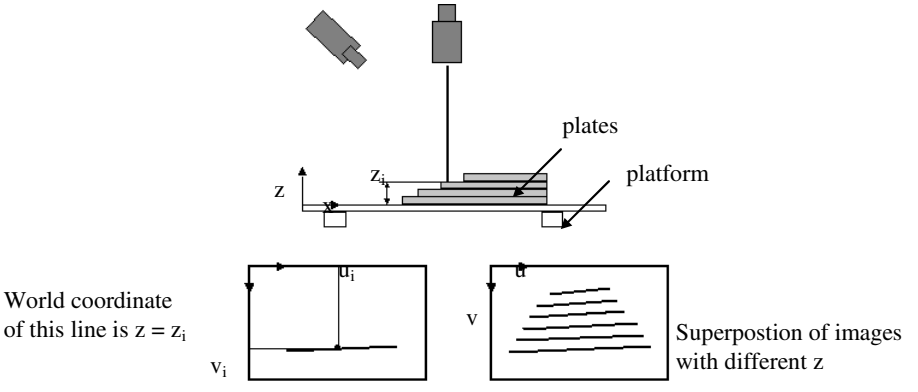


Fig. 4. Calibration of measured height by a series of horizontal plates

When M and N are solved (i.e by least square norm method), we can obtain the *y* and *z* coordinates from image data *(u,v)* using eq. (1,2). These two components combined with the *x* derived from the motion of the web using a wheel encoder form a 3D coordinates of the measured point.

2.3 The Measurement Software

The software is written in *Visual C++* and *Pylon*, the *Basler*'s camera interface wrapper. Two issues : grab/process and display of the data, which affect the overall system performance, will be discussed briefly below.

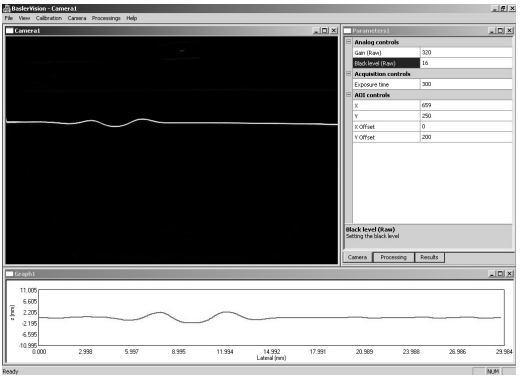


Fig. 5. The measurement software

Grab/Process. The program runs with multi-threads for acquiring and processing in overlapped mode. The grab thread, runs continuously to query the camera buffer's state. When this buffer is filled with an image, the grab thread copies data to the image buffer and triggers the processing thread, then continues to grab the next image. The processing thread manipulates the image buffer, and after finishing, it sends a message to notice the software interface updating new result. The message handling functions of the display windows will plot the corresponding data while the processing thread gets ready for the next image. Synchronization between the threads on shared buffers are guaranteed by using *Event* and *Critical Section* objects. As can be seen, rather than grabbing , processing and displaying consecutively, operating in this mechanism will minimize the overall time between two successive images.

Display. Data acquired from the camera are in form of a matrix of 8-bit gray values. To display them as an image, direct Windows GDI's drawing functions like the *SetPixel* or *FillSolidRect* are very slow and thus, inappropriate for real-time applications. A faster method without the need to use painful *DirectShow*'s functions is to treat the data as a memory bitmap and "copy" it onto the screen. This requires to manage the data in the bitmap format which contains a complex header describing the image information followed by the real data. To avoid calculation of this structure and reallocation of memory for every captured image, we implemented this method in a simple way : when the program starts, it will load a prior bitmap of the same size and color depth (8-bit) from the hard disk into a fix memory location. Each time the camera transmits the captured data, they only overwrite the image data section in this memory block while the header remains unchanged. Because all operations are done in memory, the displaying is fast enough.

Plotting the real-time 2D profile graph is rather simple by drawing a series of line segments. However instead of using the command *LineTo* drawing from one point to another, we use the *Polyline* connecting all points in the graph which is much faster and since the graph is updated very fast, we used *CMemDC* rather than direct *CDC* to avoid flickering.

3 Data Processing

3.1 Image Processing

Image processing for edge detection in this case is simple thanks to the system set-up that already well distinguishes the laser line and the background. We applied a Sobel operator with 3x3 mask which is separable:

$$f(x,y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = g(x) \times h(y) . \quad (4)$$

Hence, the convolution can be performed faster by executing two 1-dimensional masks *g(x)* and *h(y)*. Only the Area of Interest (AOI) where the laser line occupies is convoluted to reduce the number of computations. The resulting image (gradient magnitude) is then binarized by means of an histogram-based thresholding. The result is shown in figure 6

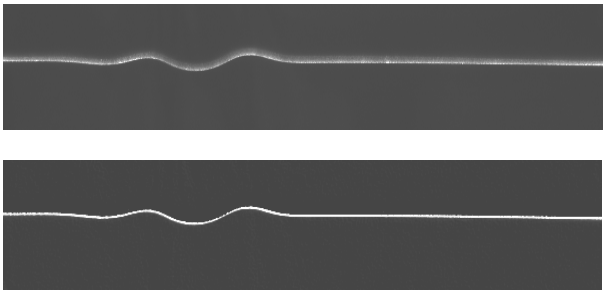


Fig. 6. The image before and after being applied the Sobel operator

3.2 Profile Filtering

The shape of the extracted profile depends on the applied tension on the web. When the tension is high, one can regard the web surface as a plane and the profile as a straight line respectively (except for local peaks and valeys created by wrinkles). The mean line of the profile is found by simply fitting the least square line and wrinkle heights are measured as the distances from the peaks to this mean line. In general, the profile is a curve with certain unsmoothness due to the digital image's limit

resolution. A natural way to find the mean line in this case is using the Gaussian filter (ISO 11562) twice with cutoff λ_c (small) and λ_f (large) to filter the short and long wavelength components. However, similar to any convolution based filters in image processing technique, this filter suffers the border distortion. Over half of the first and last cutoff length are discarded to eliminate end effects, i.e using λ_f = 80mm on a 300mm profile will result a loss of 50mm at each ends and hence, the obtained mean line is reduced to 200mm [7,8,9].

To overcome this problem, we applied the Gaussian regression filter proposed by Brinkmann et al. [11] and being discussed in ISO/TR 16610-10:2000(E). To eliminate the end effects, the weighting function of this filter is scaled up in such a way that the enclosed area under the Gaussian bell curve always equals 1 :

$$s_r(p-i) = \frac{s(p-i)}{\Delta x.\sum_{i=0}^{n-1} s(p-i)} \cdot \tag{5}$$

where :

$$s(p-i) = \frac{1}{\sqrt{2\pi}\alpha\lambda_f}\exp\left[-0.5.\left(\frac{(p-i)\Delta x}{\alpha\lambda_f}\right)^2\right] \cdot \tag{6}$$

$\alpha = \pi^{-1}.\sqrt{(\ln 2/2)}$, λ_f : the cutoff wavelength of the filter, *n* is the number of points, *p* is the index of profile points, *i* is the index for the location of weighting function.

The mean line *w(p)* calculated by the convoluting the whole profile

$$w(p) = \sum_{i=0}^{n-1} z(i).s_r(p-i)\Delta x \cdot \tag{7}$$

The shorter wavelength components (including wrinkles) are obtained by :

$$r(p) = z(p) - w(p) \cdot \tag{8}$$

However, a straightforward implementation of this filter following the ISO guideline might be too slow for on-line inspection. From equations (5,6,7) one can see that to obtain the mean line of *n* points, it is required to compute the exponent function n^2 times for each image and thus, $m.n^2$ times if the camera transmits *m* images per second. In fact, the weighting function can be calculated prior to the convolution. Instead of *n* values, the *Gauss* function shown below calculates for *2n-1* values of the weighting function in equ.(6) and fills the values to a global array. These values are used in the function *Conv* implementing equations (5,7,8) for each profile. We can reduce evaluation times of the exponent functions even more by taking the symentry condition of *s*, however it is not necessary since *Gauss* is called only once, i.e when the program starts. In short, we only need to calculate the exponent function *2n-1* times for all the profiles, this greatly speeds up the convolution algorithm.

List 1 : Function computes the Gauss weighting function

```
void Gauss(int n,double cutoff,double  spacing,double
s[])
{
   const double pi = 3.14159,
                alpha = sqrt(log(2.0f)/ 2)/pi;
   n -= 1;
   double  a,b;
   for (int i = 0; i<=2*n+1; i++)
   {
         a = 1/alpha/cutoff;
         b = (n-i)*spacing*a;
         b*=b;
         s[i] = 1/sqrt(2*pi)*a*exp(-0.5*b);
   }
}
```

List 2 : Function convolves the profile with the Gaussian weighting function

```
void Conv(int n,double s[],double z[],double r[])
{
   double a,b,w;
   n-=1;
   for (int p=0;p<=n;p++)
   {
      a = 0;
      b = 0;
      for (int i = 0; i<=n; i++)
      {
         a +=z[i]*s[n+p-i];
         b +=s[n+p-i];
      }
         w = a/b;
         r[p] = z[p]-w;
   }
}
```

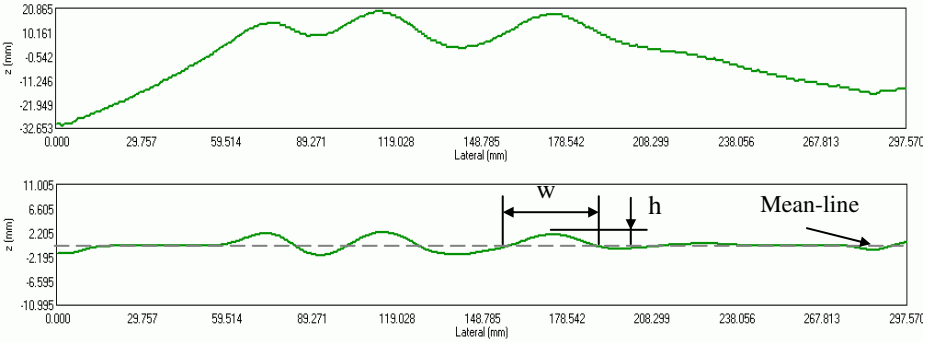


Fig. 7. A profile before and after filtered by the Gaussian regression method

Figure 7 show an example of a profile before and after filtering with this method by applying λ_c =1.5mm and λ_f = 6mm of which small cutoff λ_c =1.5mm is used to smooth the profile for better visualisation and could be obmitted in real processing. Notice that the profile length are kept unchanged without edge distortion and its curvature is suppressed to a negligible degree.

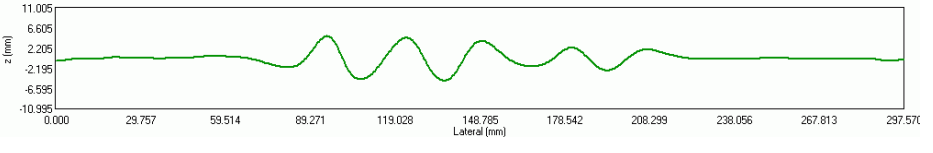
3.3 Wrinkle Characterization

Some parameters for characterizing wrinkle geometries are proposed : wrinkle heights W_z (distance between the peaks to the mean line) and their locations W_y , wrinkle density W_d (number of peaks), wrinkle sharpness W_s (the ratio $k=h/w$ as described in fig.7). An example of measurement and its analysis results are given in fig. 8.

4 System Testing

The system's performance were evaluated in term of time effiency. The computer used for the measurement system is a Pentium IV 2.4GHz, 2GB RAM with 128MB graphics card. The exposure time of the camera is set at a level allowing maximum frame rate of 79 fps can be achieved (with full resolution). When the program ran with a single thread (grab, process and display consecutively), the frame rate is reduced to 44 fps (or 22ms for each frame). However when the program executed with two worker threads (grab, process), maximum frame rate of 79 fps was reached. That's because the processing time, measured as 6ms, is smaller the fastest image cycle time, 12.7ms. Restricting AOI to half of the camera resolution increased the frame rate to 86 fps. The CPU usage in all case is 50%. As can be seen, our thread based processing scheme is satisfactory since it does not reach the limit of execution time. This also creates the possibility for further improvements to be made without violating the timming performance, such as: wrinkle classification, using multiple cameras at different web locations, 3D measurement…

(a) The bitmap image (shown in inverted color)



(b) The filtered profile

Fig. 8. A typical measurement. W_d = 5, (W_y, W_z) = P_1(95, 4.80), P_2(123, 4.74), P_3(149, 3.68), P_4(180, 2.32), P_5(207, 2.09), *max* W_s = 3.6

5 Conclusions and Future Work

In this paper, we presented a machine vision system based on structured light ranging for measurement of wrinkle geometry. The device set-up is simple, low cost comparative to line scan camera systems that widely used in web manufacturing. Another advantage of this method is the capability of measuring the wrinkle height, which is a more useful parameter to feedback to the control center than its projective shape. A measurement software was built on multi-threading technique with image processing and filtering algorithms tailored to meet the requirements of on-line visual inspection. Exprimental results showed that the software operated well at the camera's maximum acquisition frame rate.

However, the system is still in early stage of development. Many issues, i.e the accuracy of algorithms, range of measurement … need to be studied more in details. The measurement resolution should be improved by replacing new lens to increase the detection capability. In future, the 3D measurement will be implemented by combining a sequence of profiles in a specific amount of time. This allows further analysis of the 3D shape or flow of wrinkle with respect to web tension, velocity, time… to be carried out. The final objective is able to predict wrinkling and to understand what steps can be taken to prevent it.

References

1. Duane Smith, R.: Roll and Web Defect Terminology. The Winding Committee of the Finishing and Converting Division. Tappi Press (1995)
2. Xu, B., Cuminato, D.F.: Evaluation of Fabric Smoothness Appearance Using A Laser Profilometer. Textile Research Journal 12(68), 900–906 (1998)
3. Chen, C.H., Kak, A.C.: Modelling and calibration of structured light scanner for 3-D robot vision. In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 807–815. IEEE Computer Society Press, Los Alamitos (1987)
4. De Piero, F.W., Trivedi, M.T.: 3-D Computer Vision using Structured light: Design, Calibration and Implementation Issues. Advances in Computers 43, 243–278 (1996)
5. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C. Cambridge University Press, Cambridge (1998)
6. Myler, H.R., Weeks, A.R.: Computer Imaging Recipes in C. Prentice-Hall, Englewood Cliffs (1993)
7. Krystek, M.: A fast Gauss filtering algorithm for roughness measurements. Precision Engineering 19, 198–200 (1996)
8. Raja, J., Muralikrishnan, B., Fu, S.: Recent advances in separation of roughness, waviness and form. Precision Engineering 26, 222–235 (2002)
9. ISO 11562: Geometrical product specification (GPS)—surface texture: profile method—metrological characteristics of phase correct filters. International Organization for Standardization (1996)
10. ISO/TR 16610-31: Geometrical product specification (GPS)—Filtration. Part 10. Robust and spline filtering (2002)
11. Brinkmann, S., Boshwinna, H., Lemke, H.W.: Accessing roughness in three- dimensions using Gaussian regression filtering. Int. J. Machine Tools and Manufacture 41, 2153–2161 (2001)