The following manuscript was published in

Measuring 3D surface using spatial distance computation

Yung-Sheng Chen

Department of Electrical Engineering, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li 320, Taiwan, ROC.

eeyeschen@ee.yzu.edu.tw

Bor-Tow Chen

Opto-Electronics & Systems Laboratories, Industrial Technology Research Institute, Bldg. 51, 195-8, Sec. 4,
Chung-Hsung Rd., Hsin-Chu 310, Taiwan, ROC.
This paper deals with the correspondence problem of two grabbed images, obtained by projecting a structured light on the measuring surface, when 3D information of a given surface is needed. In our system, the constraint that codifies the pattern projected on the surface has been simplified by using a random speckle pattern, thus the correspondence problem is reduced to the local matching between two grabbed images and solved by a spatial distance computation technique. The performance of our approach including disparity error analysis, search range suggestion, and disparity gradient limit, are investigated and discussed. Some parameters, say, percentile constraint, sampling interval, and subpixel compensation, used properly in this approach are suggested. Experiments have shown the feasibility of the proposed method. © 1902 Optical Society of America

**OCIS codes:** 110.6880, 100.6890, 100.3010

1. **Introduction**

In order to obtain 3D information of a given surface, the approach may be considered in between a *passive* scheme and an *active* scheme. The widely known passive scheme is stereovision, which is useful to measure surfaces with well-defined boundary edges and vertexes. An algorithm to recognize singular points may be used to solve the problem of correspondence between points on both image planes. However the stereoscopic system becomes rather inefficient to measure continuous surfaces, where there are not many reference points. It has
also several problems in textural surfaces or in surfaces with lots of discontinuities. Under such an environment, the abundance of reference points can produce matching mistakes. Thus, an active system based on a structured light concept will be useful [1].

The scheme of using structured light may be reviewed briefly as follows. Let the first stereo camera be replaced by a light source, which projects a known pattern (regular patterns are usually adopted) of light on the measuring surface, and let the second stereo camera image the illuminated scene. Then the desired 3D information can be obtained by analyzing the deformations of the imaged pattern with respect to the projected one. Most of the proposed structured light methods obtain 3D information from the geometric constraint propagation, especially from the epipolar constraint, and some of them are rather limited to measure surfaces with depth discontinuities [2]. In recent, a new structured light technique has increased in importance [3]-[9]. This technique is based on a unique codification of each token of light projected on the measuring surface. When the token is imaged by the camera, this codification allows us to obtain the correspondence and it is not necessary to use geometrical constraints to get it. Several coded structured light techniques have been proposed in the past, which have been discussed and compared in a comprehensive survey [1], [5].

The codified patterns can be classified as: single scanned dot, slit line, grid and dot matrix [3], light strip [4], color strip [5], and random speckle [6]-[11]. Rocchini et al used a video projector to project a set of RGB strip patterns, produced by recursive subdivision onto the measured surface [5]. Their improved color bands allow to reconstruct the gray code and the indexing of the single vertical lines (the green ones). The shape reconstruction is operated by extracting from the image the center line of a thin green line instead of detecting the discontinuity between two regions of different color. As the system projects color on the
scene, according to the comments of Batlle et al. [1], the use is limited to a neutral color scene, as highly saturated color objects may produce lost pattern region in the segmentation step and posterior decodification. For the projection of random speckle texture, it has been also a feasible and exciting scheme for the topic of 3D surface reconstruction [6]-[9]. An early work on speckle projection can be found in [10] (regarding stereophotogrammetry as an anthropometric tool) and [11] (investigating a practical real-time imaging stereo matcher). The idea of projecting a random texture pattern onto the scene is that if the projected pattern is random such that it does not repeat, thereby ensuring that there is always a unique pattern on which to correlate in the field of view of the cameras. In order to find the correspondence, Siebert and Marshall [6] designed a scale-space stereo-matching using image pyramids, whereas D’Apuzzo [7]-[9] adopted the adaptive least squares method [12]. On a Pentium III 600 MHz machine, with this method, about 20000 points are matched in approximately 10 minutes [9]. Along such a potential research direction, in this paper, we also adopt a random speckle pattern to simplify the constraint that codifies the pattern projected on the measuring surface, and use a technique of spatial distance computation to solve the correspondence problem. Projecting random speckle pattern on an object may produce a random dotted image, it can be regarded as a CBP (color blindness plate) problem [13]-[15], and the spatial distance computation technique used in CBP segmentation may be applied to the current consideration of correspondence problem.

Figure 1(a) shows our 3D measurement system including two CCD cameras and a video projector, where the used random speckle pattern shown in Fig. 1(b) is sent from the computer and projected via the video projector on the measuring object. In this paper, three objects (object 1, object 2, object 3) will be used to illustrate our method, the corresponding
grabbed images using the designed system shown in Fig. 2 are of size $640 \times 480$. After the correspondence vectors obtained by our method, they are sent to the TriD system [16], which is a 3D data processing software developed by Opto-Electronics & Systems Laboratories, Industrial Technology Research Institute, to reconstruct the 3D surface.

2. Proposed Method

Let a gray block image be defined as $[g]$ having the size of $m \times m$. In this paper, $m = 16$ is used for illustrations and experiments. The correspondence problem solved in our approach is based on the local matching between two binary block images. Hence it is very important to determine the thresholding value, $TH$, for obtaining the binary block image, $[b]$. To overcome the uneven-brightness and out-of-focus problem (see Fig. 2 for reference) arising from the lighting environment and different CCD cameras, the brightness equalization and image binarization are used in our approach. Any the following concerned block images (we call them an effective block image) for computation should satisfy the condition: \( g_{\text{max}} = \max \{ \forall z \in [g] \} > \text{too dark} \), where “too dark = 40” used in our experiments to filter out the too dark block images, which will not be taken for spatial distance computation. Let \( m^2 \) be the total number of pixels of a block image, and \( cdf(z), z = 0 \sim 255 \) (the gray value index, where each pixel is quantized to a 8-bit data), be the cumulative distribution function of $[g]$, then a threshold controlled by the percentile $p = 0 \sim 100\%$ is defined

\[
TH_p = \{ z_p | cdf(z_p) \approx pm^2 \}.
\]  

(1)
Thus for a percentile \( p \) each gray block image \([g]\) will have a thresholding value \( TH_p \) to obtain its corresponding binary block image \([b]\), and we have

\[
b(x, y) = \begin{cases} 
1 & \text{if } g(x, y) \geq TH_p, \\
0 & \text{otherwise}. 
\end{cases}
\]  

(2)

Where “1” and “0” denote the nonzero (“white”) pixel and the zero (“black”) pixel respectively. Note here that the higher the \( p \) is, the smaller the data amount having nonzero pixels. Figure 3(c) illustrates a gray block image \([g]_{(150,150)}^T\), which is boxed in Fig. 3(a), converted into a binary block image \([b]_{(150,150)}^T\) with \( p = 50\% \) (the corresponding \( TH_p = 74 \), and \( g_{max}^T_{(150,150)} = 231 \). Because the accuracy of finding the correspondence (or quality of surface reconstruction) may be dependent on the data amount having nonzero pixels, the further investigation of \( p \)-factor is necessary. This will be presented in next section.

In the following, we describe our method to find the correspondence vector by means of the spatial distance computation. First we let \([b]_{(x_0,y_0)}^T, x_0 = 0, s, 2s, \ldots; \) and \( y_0 = 0, s, 2s, \ldots, \) be a binary block image in the left grabbed image starting at the location \((x_0, y_0)\), where \( s \) is the sampling interval from the grabbed image. The searched block image, \([b]_{(u_0,v_0)}^r\) starting at the location \((u_0, v_0)\) in the right grabbed image, will be in the range of \( u_0 \in [x_0 - R_x, x_0 + R_x] \) and \( v_0 \in [y_0 - R_y, y_0 + R_y] \), where \( R_x \) and \( R_y \) depend on the system configuration. In the searching range, if a right binary block image \([b]_{(u_f,v_f)}^r\) has the minimum spatial distance \( d([b]_{(x_0,y_0)}^T, [b]_{(u_f,v_f)}^r) \) between it to \([b]_{(x_0,y_0)}^T\), then the vector from \((x_0, y_0)\) to \((u_f, v_f)\) is defined to be the found correspondence vector (or the detected disparity). The spatial distance \( d([b]_{(x_0,y_0)}^T, [b]_{(u_0,v_0)}^r) \) between any two binary block images, \([b]_{(x_0,y_0)}^T\) and \([b]_{(u_0,v_0)}^r\), is defined as follows.

To perform the distance computation, the coordinates of \((x_0, y_0)\) and \((u_0, v_0)\) are adjusted
to the same basis. Intuitively, the distance between two nonzero pixels \( b^l_{(x_0, y_0)}(x, y) \) and \( b^r_{(u_0, v_0)}(u, v) \) may be defined to be Euclidean

\[
d(b^l_{(x_0, y_0)}(x, y); b^r_{(u_0, v_0)}(u, v)) = \sqrt{(x - u)^2 + (y - v)^2},
\]

and the distance between the nonzero pixel \( b^l_{(x_0, y_0)}(x, y) \) and the binary block image \( [b]^r_{(u_0, v_0)} \) is defined as

\[
d(b^l_{(x_0, y_0)}(x, y); [b]^r_{(u_0, v_0)}) = \min_{\forall b^l_{(x_0, y_0)}(u, v) \neq 0} d(b^l_{(x_0, y_0)}(x, y); b^r_{(u_0, v_0)}(u, v)).
\]

Although the Euclidean distance is suitable in our design, it is time-consuming due to the need of higher complexity arithmetic operations such as the “square root.” Moreover since only the minimum distance is mainly considered in our method, most nonzero pixels \( \in [b]^r_{(u_0, v_0)} \) are not necessary to be searched for computation. Therefore, we present an alternate scheme to simplify the distance computation as described in Algorithm 1.

**Algorithm 1: Simplifying the distance computation of equation (4)**

1. Set the coordinates of \((x_0, y_0)\) and \((u_0, v_0)\) to be the same basis. For each nonzero pixel \((x, y) \in [b]^l_{(x_0, y_0)}\) do Steps 2-5.

2. Set \( k = 0 \).

3. Search the image space with coordinates \((u, v)\), where (3-1) \( u \in [x - k, x + k] \) and \( v = y - k \); (3-2) \( u = x + k \) and \( v \in [y - k, y + k] \); (3-3) \( u \in [x - k, x + k] \) and \( v = y + k \); and (3-4) \( u = x - k \) and \( v \in [y - k, y + k] \). Once a nonzero pixel \((u, v) \in [b]^r_{(u_0, v_0)}\) or boundary of the block image is met, go to Step 5. Otherwise go to Step 4.

4. Increment \( k \) by 1 and go to Step 3.
5. $k$ is the desired distance between the nonzero pixel $b'_{(x_0, y_0)}(x, y)$ and the binary block image $[b]_{(u_0, v_0)}^r$.

Based on this algorithm, for a nonzero pixel belonging to the left block image, once the closest nonzero pixel belonging to the right block image is found in the $k$th search the searching process stops. Thus the most pixels in the right block image will not be searched. In addition, the searching steps contain no high complexity arithmetic operations as Eq. (3) used. This leads to an efficient search for finding the desired spatial distance. Now the distance between $[b]_{(x_0, y_0)}^l$ and $[b]_{(u_0, v_0)}^r$ may be expressed as

$$d([b]_{(x_0, y_0)}^l, [b]_{(u_0, v_0)}^r) = \sum_{\forall b'_{(x_0, y_0)}(x, y) \neq 0} d(b'_{(x_0, y_0)}(x, y); [b]_{(u_0, v_0)}^r). \quad (5)$$

Thus within the searching range, the best-matching block image $[b]_{(u_f, v_f)}^r$ should have the minimum distance such that

$$\begin{align*}
(u_f, v_f) &= \{(u_0, v_0) \mid u_0 \in [x_0 - R_x, x_0 + R_x], v_0 \in [y_0 - R_y, y_0 + R_y] \\
&\quad \min_{v_0 \in [y_0 - R_y, y_0 + R_y]} d([b]_{(x_0, y_0)}^l, [b]_{(u_0, v_0)}^r) \}\end{align*} \quad (6)$$

As a result, the whole procedure to find the correspondence vector from left grabbed image to right grabbed image may be summarized as the following algorithm.

\textbf{Algorithm 2: Finding the correspondence vector}

1. For any sampled gray block image $[g]_{(x_0, y_0)}^l$ in the left grabbed image, convert it to a binary block image $[b]_{(x_0, y_0)}^l$ using Eqs. (1) and (2). If $g_{max}^l(x_0, y_0) \leq \text{too dark}$, then go to Step 4.

2. Search the block images in the right grabbed image, $[g]_{(u_0, v_0)}^r$, $u_0 \in [x_0 - R_x, x_0 + R_x]$ and $v_0 \in [y_0 - R_y, y_0 + R_y]$, where $R_x$ and $R_y$ depend on the system configuration. For each
processed gray block image, \( [\mathbf{g}]_{(u_0,v_0)}^r \) is also converted to a binary block image \( [\mathbf{b}]_{(u_0,v_0)}^r \) using Eqs. (1) and (2). If \( g_{max}^r_{(u_0,v_0)} \leq \text{too dark} \), then disregard this case and do next block image in the searching range.

3. Perform the spatial distance computation between \( [\mathbf{b}]_{l(x_0,y_0)}^L \) and \( [\mathbf{b}]_{l(u_0,v_0)}^r \), \( \forall u_0, v_0 \), using Eq. (5). Then select the best-matching correspondence location \((u_f,v_f)\) using Eq. (6). The found correspondence vector is from \((x_0,y_0)\) to \((u_f,v_f)\), whose difference is the detected disparity.

4. Perform this procedure until all the sampled gray block images in the left grabbed image are processed.

---

By means of the above procedure, for the case \((x_0,y_0) = (150,150)\) given in Fig. 3(c) and the right searching range boxed in Fig. 3(b), the searching result of the found correspondence \((u_f,v_f) = (144,148)\) is shown in Fig. 3(d), where \( p = 50\% \), \( TH_p = 80 \), and \( g_{max}^r_{(u_0,v_0)} = 214 \). In order to illustrate the behaviour of our spatial distance computation, the distance distribution for a partial searching range in the current example is given in Fig. 3(e), where the minimum value \((0.242188)\) is just located at the correspondence \((144,148)\). The effectiveness of performing our spatial distance as a similarity measure can be further investigated by Root Mean Square Error, which will be presented in next section.

Since the corresponding information used in the stereoscopic system are usually represented with subpixel level, which provides a more accurate measurement, in our system a simple averaging is used to obtain the desired subpixel coordinate \((u_f^*, v_f^*)\). Such a post-processing is named as subpixel compensation in this paper and described by the following
Algorithm.

**Algorithm 3: Averaging for subpixel compensation**

1. For a sampled block image indexed with \((x_0, y_0)\) in the left grabed image, create an array \(A\) of size \(w \times w\) containing all the correspondence results, \((u_f, v_f)\)'s, obtained from itself and its neighborhoods.

2. Set \(N\) to be the number of nonzero correspondence vectors \(\in A\).

3. Perform

\[
u_f^* = \frac{1}{N} \sum_{(u_f, v_f) \in A} u_f,
\]

\[
v_f^* = \frac{1}{N} \sum_{(u_f, v_f) \in A} v_f.
\]

4. \((u_f^*, v_f^*)\) is the desired subpixel correspondence vector. Perform this algorithm until all the sampled block images in the left grabbed image are processed.

---

In this designation, \(w = 3\) means that \(N \leq 9\) correspondence vectors (the maximum number of vectors includes the current correspondence vector and its 8 neighborhoods) are used for supporting the subpixel compensation. Similarly, \(w = 5\) means \(N \leq 25\) correspondence vectors are used for supporting the subpixel compensation. Note here that \(w = 1\) means the subpixel compensation is disabled.

### 3. Analyses and Experiments

The algorithms of the proposed method were implemented with Visual C++ 5.0 and executed on a Pentium II 450 MHz personal computer. In this section, a set of synthetic stereo-pairs
(for planar surfaces) containing known disparity fields under the introduction of additive noise are first used for error analyses. Then the search range \( R_x \) and disparity gradient limit attainable by our algorithm are validated. Finally, the suitable \( p, s, \) and \( w \) values used in our algorithm are suggested by experiments on real cases.

A. Error Analysis of Correspondence Results

Consider the elementary stereo geometry illustrated in Fig. 4(a), assume we have a bird’s-eye view of two cameras with parallel optical axes separated by a distance \( 2b \) and the focal length is \( f \). The images they provide, together with one point \( P \) with co-ordinates \( (x, y, z) \) in the planar surface, show this point’s projection onto left \( (P_l) \) and right \( (P_r) \) images. The co-ordinates in Fig. 4(a) have the \( z \) axis representing distance from the cameras (at which \( z = 0 \)) and \( x \) axis representing horizontal distance (the \( y \) co-ordinate, into the page, does not appear). In this subsection, we make the synthetic stereo-pairs satisfying the epipolar constraint, this reduces the potential 2D search into 1D search along \( x \) axis. That is, search range \( (R_y) \) in \( y \) direction is ignored. \( x = 0 \) will be the position midway between the cameras; each image will have a local co-ordinate system \( (x_l \) on the left, \( x_r \) on the right). Based on this configuration, the relationship between distance \( z \) and disparity \( P_r - P_l \) is

\[
z = \frac{2hf}{P_r - P_l} \tag{7}
\]

If \( z \) is given, the disparity \( P_r - P_l \) will be known, and vice versa. Similarly, based on the geometry relationships in Fig. 4(a), if the point \( P(x, y, z) \) is known, \( P_l \) and \( P_r \) may be found by the following equations,

\[
P_l = \frac{-(h + x)f}{z}, \text{ and } P_r = \frac{(h - x)f}{z}. \tag{8}
\]
Figure 4(b) and 4(c) illustrate two synthetic stereo-pairs, where we use the random speckle pattern shown in Fig. 1(b) as a planar surface. The parameters are set to be \( h = 35 \), \( f = 5 \), \( z = 6 \) (disparity = 58.333374) for Fig. 4(b) and \( z = 14 \) (disparity = 25.0) for Fig. 4(c), respectively. In order to investigate the effect of adding noise, the specific percent “white” noise, generated by a random number generator in C language, may be involved in the stereo-pairs as exemplified by Fig. 5 with 50% “white” noise. Thus if the original disparity fields of a synthetic stereo-pair are known, the detected disparity fields may be obtained by applying our algorithm to the stereo images for comparison. Based on the original disparity fields and the detected ones, the *Root Mean Square Error* is used to analyze the presented algorithm.

Three cases of planar surfaces with different distances \((z = 6, 10, 14)\) as well as different noise added (Noise Ratio = 0 \(\sim\) 100%) were analyzed by Root Mean Square Error to investigate the behaviour of our algorithm. Here percentile \( p = 50\% \) for binarization, sampling interval \( s = 4 \), and subpixel compensation \( 5 \times 5 \) support were used in these analyses. The measuring results are plotted in Fig. 6(a). Three properties can be observed from this plot. First, the smaller the disparity (or the farther the distance \( z \) is), the lower the Root Mean Square Error. Secondly, after the noise ratio about 55\%, the Root Mean Square Errors are converged to about 5.0 for all cases. This phenomenon may be explained as that the larger “white” noise may inherently make the original random speckle pattern *anti-noisy* such that the proposed algorithm works again but not better than the low noise cases. Finally, under some noise ratio below, e.g, noise ratio \( \leq 28\% \) for the plot of \( z = 14 \), a very low Root Mean Square Error \((\leq 0.337923)\) may be achieved.

In order to further explore the effect of subpixel compensation, we also apply the proposed approach without involving Algorithm 3 to the same set of synthetic stereo-pairs.
Figure 6(b)-6(d) show the Root Mean Square Error measuring results with different cases. All these plots demonstrate that the accuracy of using subpixel compensation will be higher than that of without using subpixel compensation support in our approach.

As a result, since the noise does usually not appear in a normal system, based on this analysis it is expected that the proposed algorithm can provide an acceptable result for surface reconstruction, which will be presented later.

B. Search Range and Disparity Gradient

Search range $R_x$ (under the epipolar constraint, the $R_x$ is mainly concerned, whereas $R_y$ is ignored) and disparity gradient are other two factors, which may be used to investigate the algorithm’s performance. In usual, the search range strongly depends on the disparity fields of measured surface and should be large enough to cope with the largest disparity so that all the disparities can be detected and used for surface reconstruction. Based on this characteristic, the effective search range ($ESR$) may be properly defined by

$$ESR = \max_{\delta} \{\text{disparity}_i\} + \delta,$$

where $\delta$ is a some positive number, which will guarantee the effectiveness of $ESR$ for finding the correspondence.

For the aspect of disparity gradient, Burt and Julesz [17] have provided evidence that if both dots are to be binocularly fused simultaneously by the human visual system, then the ratio of the disparity difference between the dots to their cyclopean separation must not exceed a limit of about 1. This ratio is known as the disparity gradient and the existence of a disparity gradient limit implies that even small absolute disparity differences will not produce fusion if the spatial separation between dots is also small. A well-known application of using
disparity gradient limit constraint is the PMF algorithm proposed by Pollard et al.[18], which is used to deal with the feature-based stereo correspondence. The disparity gradient may be further briefly introduced as follows. Consider a stereogram comprising the left and right images. When $A_l$ is matched to $A_r$, and similarly for $B$, the disparity gradient between them is the difference in their disparity divided by their cyclopean separation. The latter is given by the distance between the midpoints of the two pairs of dots. In order to investigate the disparity gradient limit attainable by our algorithm, we rotate the planar surface about the center point $C$ as illustrated in Fig. 7(a) to produce the other set of synthetic stereo-pairs for further analyses. In such a designation, disparity fields varying linearly which are also suitable for exploring the effective search range. For each $y$th horizontal line, the disparity gradient may be simplified and expressed by

$$\text{disparity gradient} = \frac{2|X_r - X_l|}{|X_r + X_l|}$$

(10)

From this designation, the larger the slope of planar surface is, the larger the disparity gradient; whereas the nearer the symmetrical pair of points $A$ and $B$ centering with $C$ is, the smaller the disparity gradient. If the measured Root Mean Square Error is raised and the reconstructed surface is distorted above some disparity gradient, then the disparity gradient limit can be detected.

Consider the planar surface used in the case of Fig. 4(c), Figure 7(b) shows a synthetic stereo-pair by rotating this planar surface (to produce a sloping surface as the Pollard et al.'s work [18]), in which the disparity fields vary linearly from 17.8 to 33.1. There are seven cases are analyzed. For each case, we use different $R_x$ value as the search range and compute the corresponding Root Mean Square Error. The results plotted in Fig. 7(c) report that when the
search range is larger than $ESR$ for each case, the lower bound of Root Mean Square Error may be achieved. This plot also shows that the wider the range of disparity fields is, the higher the Root Mean Square Error. This means that the case of larger disparity (gradient) may lead to more errors for finding stereo correspondence. Based on this analysis, we can suggest that if the disparity fields for a measured surface are known previously, which can be easily tested via the system settings, the $ESR$ will be readily determined for the system use.

Since the test images are size of $640 \times 480$, based on the designation of Fig. 7(a), the disparity gradient computed for a symmetrical-pair may be measured from center (i.e., the horizontal position, 320) to both sides along the horizontal axis according to Eq. (10). For each case, an appropriate $ESR$ selected based on the results of Fig. 7(c) is used. Figure 7(d) reports that the five cases of disparity range, 25.0, [17.8, 33.1], [12.6, 39.2], [10.8, 42.4], and [8.7, 45.1] have a linear change (observe the change from right to left) of the measured disparity gradients (the largest disparity gradient is about 1.07 in these cases), where the corresponding surface reconstruction shown in Fig. 8 displays a good result. For the cases of disparity range, [7.2, 46.8] and [6.9, 48.9], the measured disparity gradient plot displayed in Fig. 7(d) still has a linear change from right to left until a corner (the corresponding disparity gradient is about 1.09 and 1.12, respectively) appears. The large amount of mismatches of finding stereo correspondence occur after the corners for these two cases. This phenomenon can be easily observed from Fig. 8. As a result, by our approach, stereo images of sloping surfaces will satisfy the well-known disparity gradient limit of about 1.0 (even a higher limit, say about 1.09, is achieved) provided the slope they depict is not too great. This result agrees with that obtained by Pollard et al [18].
C. Suggestion of $p$, $s$, and $w$ Values

In order to investigate the factors of $p$ (for binarization), $s$ (for sampling interval), and $w$ (for subpixel compensation used in Algorithm 3), we present some experiments for the grabbed images of object 1 as given in Fig. 2(a) and 2(b), in this subsection. According to our system settings for real cases, the searching range is set to be $R_x = 20$ and $R_y = 1$. Here the 3D surface is reconstructed by using the TriD system [16] which can process the correspondence vectors obtained by our method. First, we fix $s = 16$ with no subpixel compensation ($w = 1$) for each $p \in [10\%, 90\%]$, we display the corresponding 3D surface reconstruction and inspect its visual quality. Figure 9 shows some reconstructed 3D surfaces with different $p$ values under this condition and suggests the better results can be obtained when $p$ is selected in the neighborhood of $p = 65\%$. Since the found correspondence is under pixel level, it is unavoidable that the 3D reconstructed surface presents the coarseness effect. To reduce the coarseness effect, the subpixel compensation performed in Algorithm 3 may be involved in the postprocessing. Figure 10 displays the results by adding the $3 \times 3$ support for subpixel compensation and confirms the effectiveness of the subpixel compensation. These experiments illustrate that the subpixel compensation can effectively refine the visual quality of the reconstructed 3D result. To present the subpixel compensation more clearly, for the cases of $s = 16, p = 65\%$ in Fig. 9 and 10, the correspondence vector maps (or the detected disparity fields) of before and after compensation are illustrated respectively in Fig. 11(a) and 11(b). Furthermore, the results using sampling interval $s = 8$ combining with $w = 3$; and $s = 4$ combining with $w = 5$, are given in Fig. 12 and 13, respectively, where left side is without subpixel compensation and right side is with the compensation. Figure 11(c) and
11(d) display the corresponding vector maps for comparison. Based on our observations on these experiments, the suggested value \( p = 65\% \) can provide an acceptable 3D surface reconstruction by our method. In addition, three suggested pairs for \((sampling\ interval, subpixel\ supports)\) are \((s, w \times w) = (16, 3 \times 3), (8, 3 \times 3),\) and \((4, 5 \times 5)\).

Figure 14 shows some results for the grabbed images of object 2 and object 3 given in Fig. 2. They also suggest the better results may be obtained with \( p = 65\% \). Figure 15 displays other viewing results for the reconstructed 3D surfaces of the three objects, with \( p = 65\%, s = 4\), as well as \( 5 \times 5\) subpixel compensation support. The execution time for each case is listed in Table 1. Our experiments show that the smaller the sampling interval is, the longer the execution time is and the more vivid the reconstructed 3D surface should be, and \textit{vice versa}. The better reconstructed 3D surface suggested by the value \( p = 65\% \) may be purchased by a reasonable computation cost.

4. Conclusions

To solve the correspondence problem of measuring 3D surface, in this paper, a random speckle pattern is adopted to simplify the constraint that codifies the pattern projected on the surface, and the technique of spatial distance computation is applied to find the correspondence vector. The spatial distance computation scheme is only based upon binary block images, its current computing cost takes under 100 sec for a \( 640 \times 480 \) image on a Pentium II 450 personal computer. A set of synthetic stereo-pairs containing known disparity fields under the introduction of additive noise have been used for error analyses. The search range and disparity gradient limit attainable by our algorithm have also been validated. In addition, the main parameters, \( p = 65\% \) for thresholding and \((s, w \times w)\) for subpixel
compensation, have been suggested for providing an acceptable 3D surface reconstruction by our method. Although only the shaded surface reconstruction is presented in the body of this paper, by a suitable combination on the shaded surface plus the corresponding texture image (which can be captured by a CCD camera without projecting the random speckle pattern), it is possible to obtain the photorealistic surface reconstruction [19]. For example, Fig. 16(a) shows the captured texture image of the second author of this paper (B. T. Chen), by some manipulations on the TriD system [16], the photorealistic visualization may be shown in Fig. 16(b)-(d). Accordingly, our results have confirmed the feasibility of the proposed method. Since the presented method is very simple, in the near future, a hardware system may be designed along the VLSI technique to reduce the distance computation time under 1 sec; the size of the system can be reduced by using the consumer digital camera; and the projection of random speckle pattern may be achieved by combining a random speckle slide in the flashlight system of a digital camera. As a conclusion, with the measurement system based on random speckle texture projection stereophotogrammetry now developed and ongoing R&D promising real-time, low-cost and flexible 3D imaging, there is an exciting future for new and creative applications based on the 3D data and models [6].
References


Fig. 1 (a) Our 3D measurement system. (b) The used random speckle pattern, which is sent from the computer and projected via the video projector.
Fig. 2. (a) Left grabbed image and (b) right grabbed image for object 1. (c) Left grabbed image and (d) right grabbed image for object 2. (e) Left grabbed image and (f) right grabbed image for object 3.
Fig. 3 (a) and (b) show the grabbed left image and right image, respectively. Let the block image size be $16 \times 16$, and percentile $p = 50\%$ used in Eq. (1). (c) shows the left gray block image and its binary image, where $(x_0, y_0) = (150, 150)$. Based on our distance computation scheme, the best-matching right block image shown in (d) is found with $(u_f, v_f) = (144, 148)$. (e) shows the found correspondence $(144, 148)$ has the minimum value $0.242188$ in the distance distribution of a partial searching range.
Fig. 4 (a) Elementary stereo geometry in canonical configuration. The synthetic stereo-pairs, (a) \((h = 35, f = 5, z = 6\), and disparity = 58.333374\), and (b) \((h = 35, f = 5, z = 14\), and disparity = 25.0\), for a planar surface with the adopted random speckle pattern shown in Fig. 1(b).
Fig. 5  (a) and (b) show the 50% “white” noise generated by a random number generator in C language are involved in the stereo-pairs of Fig. 4(b) and 4(c), respectively.
Fig. 6  (a) Three cases of planar surfaces with different distances ($z = 6, 10, 14$) as well as different noise added (Noise Ratio = 0 ~ 100%) were analyzed by Root Mean Square Error. (b)-(d) display the Root Mean Square Error plots with different cases. All these plots demonstrate that the Root Mean Square Error of using subpixel compensation will be lower than that of without using subpixel compensation support in our approach.
Fig. 7  (a) Configuration of sloping surface for measuring effective search range, $ESR$, and disparity gradients. (b) Synthetic stereo-pairs of a sloping surface having the range of disparity fields, $[17.6, 33.1]$. (c) Plot of measuring the Root Mean Square Errors with different search ranges. (d) Plot of measuring the disparity gradients along the horizontal axis.
Fig. 8 Surface reconstruction for each case given in Fig. 7.
Fig. 9 Some reconstructed 3D surfaces with different $p$ values for object 1 by fixing $s = 16$, without subpixel compensation.
Fig. 10 Some reconstructed 3D surfaces with different $p$ values for object 1 by fixing $s = 16$, with $3 \times 3$ support for subpixel compensation.
Fig. 11 Some correspondence vector maps for object 1. (a) \( s = 16 \), without subpixel compensation.  (b) \( s = 16 \), with \( 3 \times 3 \) support for subpixel compensation.  (c) \( s = 8 \), with \( 3 \times 3 \) support for subpixel compensation.  (d) \( s = 4 \), with \( 5 \times 5 \) support for subpixel compensation.
Fig. 12 Some reconstructed 3D surfaces with different $p$ values for object 1 by fixing $s = 8$, where left side is without subpixel compensation and right side is with $3 \times 3$ support for subpixel compensation.
Fig. 13 Some reconstructed 3D surfaces with different $p$ values for object 1 by fixing $s = 4$, where left side is without subpixel compensation and right side is with $5 \times 5$ support for subpixel compensation.
Fig. 14 Some reconstructed 3D surfaces with different $p$ values for object 2 (left side) and object 3 (right side) by fixing $s = 4$, with $5 \times 5$ support for subpixel compensation.
Fig. 15 Other viewing results of the reconstructed 3D surfaces for the three objects, with \( p = 65\% \), \( s = 4 \), as well as \( 5 \times 5 \) support for subpixel compensation.

<table>
<thead>
<tr>
<th>( s ) values</th>
<th>Object 1</th>
<th>Object 1</th>
<th>Object 1</th>
<th>Object 2</th>
<th>Object 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p ) values</td>
<td>( s = 16 )</td>
<td>( s = 8 )</td>
<td>( s = 4 )</td>
<td>( s = 4 )</td>
<td>( s = 4 )</td>
</tr>
<tr>
<td>( p = 15% )</td>
<td>7.6 sec</td>
<td>20.8 sec</td>
<td>73.3 sec</td>
<td>49.3 sec</td>
<td>43.2 sec</td>
</tr>
<tr>
<td>( p = 25% )</td>
<td>8.0 sec</td>
<td>21.9 sec</td>
<td>77.4 sec</td>
<td>51.9 sec</td>
<td>45.1 sec</td>
</tr>
<tr>
<td>( p = 35% )</td>
<td>8.2 sec</td>
<td>22.7 sec</td>
<td>80.5 sec</td>
<td>53.5 sec</td>
<td>46.7 sec</td>
</tr>
<tr>
<td>( p = 45% )</td>
<td>8.4 sec</td>
<td>22.9 sec</td>
<td>81.7 sec</td>
<td>54.3 sec</td>
<td>47.2 sec</td>
</tr>
<tr>
<td>( p = 55% )</td>
<td>8.5 sec</td>
<td>23.3 sec</td>
<td>82.7 sec</td>
<td>53.7 sec</td>
<td>46.8 sec</td>
</tr>
<tr>
<td>( p = 65% )</td>
<td>8.4 sec</td>
<td>22.8 sec</td>
<td>79.9 sec</td>
<td>52.0 sec</td>
<td>45.5 sec</td>
</tr>
<tr>
<td>( p = 75% )</td>
<td>8.2 sec</td>
<td>21.7 sec</td>
<td>74.8 sec</td>
<td>48.8 sec</td>
<td>43.0 sec</td>
</tr>
<tr>
<td>( p = 85% )</td>
<td>7.9 sec</td>
<td>19.7 sec</td>
<td>66.8 sec</td>
<td>43.7 sec</td>
<td>38.5 sec</td>
</tr>
</tbody>
</table>
Fig. 16  (a) Texture image of the second author of this paper.  (b)-(d) Photorealistic visualization by some manipulations on the TriD system.