

3D Shape Acquisition using a combined SSD and Least Squares Correlation Approach

Pedram Azad, Tilo Gockel and Rüdiger Dillmann

Abstract— In this paper we want to present a new approach to 3D shape acquisition. This implementation enables the user to manually move the scan unit over the object without the scan unit being tracked. Registration from scan to scan is done on-the-fly to allow user interaction with the system. Since the user can watch the scene assembling gradually during the scan process he can immediately respond to occurring shadings due to shadows and undercuts. The chosen structured light scan approach uses an initially unknown white noise pattern that can easily be projected with any fixed pattern projection system. Object points are acquired on the basis of finding correspondences in the pattern in the camera image of the current scene. This is achieved by using a combination of a SSD and a least squares correlation algorithm. In the current test setup we are using a standard video beamer and a standard digital camera. Our approach reduces hardware complexity to a minimum using only one camera and a fixed pattern projector which both can be miniaturized. In this paper we want to present the whole system also including the calibration procedure. The focus will be on the correlation method.

Index Terms—SSD, Least Squares Correlation, Pattern Projector, Single Shot, Structured Light, 3D Scanning, 3D Shape Acquisition.

I. INTRODUCTION

3D-SCANNING of large and complex objects using a standard laser- or pattern projector-based scan system is a complex iterative process. For each single scan the user has to fixate the object, not knowing whether the new position will yield a better result for the missing regions, which occur naturally due to shadows and undercuts. Furthermore after each single scan the user has to do an offline registration with the previous scans. This step-by-step approach is a very time-consuming and inefficient process.

A scanning device which is capable of performing the registration on-the-fly and instantly showing the result during the scan process would allow the user to react interactively and therefore would be much more efficient. This type of system can be realized using a fix-pattern projection-based

Manuscript received November 21, 2003 The authors are with the University of Karlsruhe, Dept. of Computer Science, Institute for Industrial Applications in Informatics and Microsystems (IAIM), Haid-und-Neu-Strasse 12, c/o Technologiefabrik, D-76131 Karlsruhe, Germany. Head: Prof. Dr.-Ing. Rüdiger Dillmann (e-mail: {dillmann, gockel, azad}@ira.uka.de).

scan mechanism, a fast online registration and a fast visualization of the resulting 3D data.

Perhaps the most famous results of such an approach were published by Rusinkiewicz et al [4]. These researchers used a multi-pattern projection system with a complex image-to-image stripe tracking mechanism. Due to the need of projecting various patterns over time this system lacks the possibility of using any fixed pattern projection system and therefore cannot be miniaturized. Approaches of other researchers using single shot methods based on a specific pattern depend on a very precisely manufactured pattern slide [7-9]. To avoid these drawbacks we have developed a single shot / single pattern projection system with no demands for accuracy of the slide pattern.

Our system is based on the principle of finding correspondences of the pattern in the camera image of the current scene. Each correspondence provides one object point by triangulating between the projector and the camera.

The test setup uses a standard video beamer projecting only one white noise pattern. This projector can be easily substituted by a simpler device like a low-cost gobo projector known from multimedia applications. The scene is observed by a standard digital camera.

Normally projector calibration is done using a well-known pattern [1,7]. Our aim was to calibrate the pattern projector with the same pattern which is used during the scan process, without the need to be known to the system initially.

Section II of this paper explains the calibration procedure we have developed.

Section III introduces all algorithms implemented in the system, particularly giving a detailed description of the correlation method we have developed.

Section IV shows the testbed hardware and the software environment. In Section V the results of the system are summarized, ending with a conclusion and future work.

II. SYSTEM CALIBRATION

The scan system is calibrated in two steps. In the first step the camera is calibrated, which is then used for calibrating the projector in the second step.

For the calibration of the camera we use an extended DLT (*Direct Linear Transformation*) approach, which also deals with radial lens distortions [1]. This algorithm uses an

iterative procedure in order to calculate the optimal set of parameters. In each iteration the DLT parameters are calculated first, incorporating the distortion parameters of the last iteration for deskewing the camera image. After this the parameters modeling the radial lens distortion are calculated using a least squares approach.

The input of this test-field-based calibration method consists of a set of point pairs (P_{World}, P_{Image}) where P_{World} is an a priori known 3D-point and P_{Image} the corresponding 2D-point captured by the CCD-chip. We acquire this set using a stack of acrylic plates of known thickness with the top plate containing a well-known dot pattern (cp. figure 7 in chapter IV). Discrete 2D-points are acquired by determining the centers of the dots. This is done by using a region growing algorithm and by calculation of the centroid for each region that is found.

Projector calibration is done utilizing the same extended DLT algorithm which is used for camera calibration. However the input for the algorithm is acquired in an entirely different way. Initially the pattern is projected onto the plane defined by $z=0$ and captured by the calibrated camera. This image will serve as a template for the whole following process. In the second step the pattern is projected onto planes in different heights ($z=h$) using the acrylic plates.

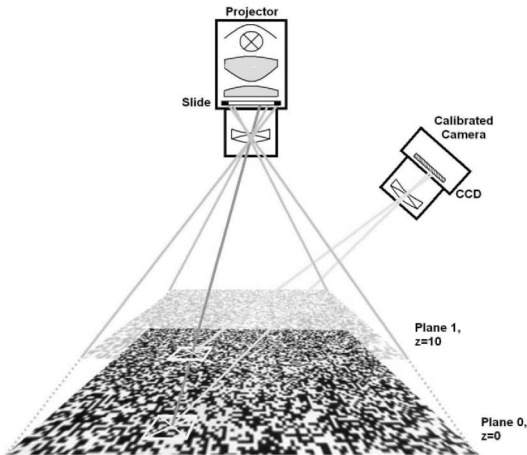


Fig. 1. Schematic sketch of the systems' geometrical layout used for Projector Calibration

Now the same correlation algorithm which will be utilized during the scan process later-on (except neglecting the epipolar constraint and searching in two dimensions) is used for finding correspondences of the template in the other takings. For each 2D-correspondence that is found the related 3D-points are calculated by using the parameters of the calibrated camera and intersecting the resulting camera rays with the well-known plane of height h . Naming the 3D-point in the template $P_{Template}(n_i, m_i, 0)$ and the corresponding 3D-point $P_{z=h}(x_i, y_i, z_i)$ the resulting point pair for the input of the calibration algorithm is defined as $(P(x_i, y_i, z_i), P'(n_i, m_i))$.

Figure 1 shows the intersections of the camera rays with the projector rays illustrating the cooperation of the camera and the pattern projector.

We want to mention that with this method not the projection of the original pattern is being calculated but the projection of the template image. Thereby, an object point is acquired during the scan process as follows: For each correspondence found, the related camera ray g_1 to the 2D-point in the camera image is calculated by using the parameters of the calibrated camera, the related camera ray g_2 to the 2D-point in the template image is calculated in the same way. Then the corresponding 3D-point $P_{Template}(n, m, 0)$ to the template match can be calculated by intersecting g_2 with the plane defined by $z=0$. Using the parameters of the calibrated projector one can now determine the projector ray g_3 easily by inserting the coordinates n and m . Finally the object point is defined by the intersection of the straight lines g_1 and g_3 . If no intersection can be found the closest point between g_1 and g_3 is used for the further calculations.

III. ALGORITHMS

A. Correlation

Given a camera image of an object with the pattern projected onto it a simple correlation algorithm such as *cross-correlation* or *SSD (Sum of Squared Differences)* would not be capable of reliably finding a sufficient amount of correspondences to the template image, due to distortions caused by a non-orthogonal projection of the pattern onto the local surface section. Since the least squares correlation algorithm described in [2,3] can cope with distortions described by an affine transformation it is capable of finding a much greater amount of correspondences. Furthermore the least squares algorithm can claim subpixel resolution, which is essential as can be seen in the figures 2 and 3. However the convergence radius of the least squares algorithm is only 2-3 pixels and therefore a search has to be performed in order to find a precise enough precondition. Since the least squares algorithm is an iterative procedure it is too slow for being performed during the search process in our system. Therefore we have developed a combined approach, using the fast SSD correlation algorithm throughout the search process and performing the least squares algorithm with the best precondition found in order to verify the validity of this initial guess and to obtain subpixel accuracy.

Given a template image window with the gray value function $f(x, y)$, a search image window with the function $g(x, y)$, a window size of $2W+1$ and a pixel $(i, j)^T$ in the template image, the SSD correlation algorithm searches for the displacement vector $(d_1, d_2)^T$ that minimizes the following error measure $c(d_1, d_2)$ [10]:

$$c(d_1, d_2) = \sum_{k=-W}^W \sum_{l=-W}^W (f(i+k, j+l) - g(i+k+d_1, j+l+d_2))^2$$

The least squares correlation algorithm of Gruen can be described as follows [2,3]: Given a template image window with the gray value function $f(x,y)$ and a search image window with the function $g(x,y)$ which are identical up to a difference $e(x,y)$ caused by distortion, their relationship can be described by a translation, an affine transformation and radiometric distortion:

$$f(x,y) - e(x,y) = r_0 + g(x',y')$$

$$x' = a_0 + a_1x + a_2y \quad y' = b_0 + b_1x + b_2y$$

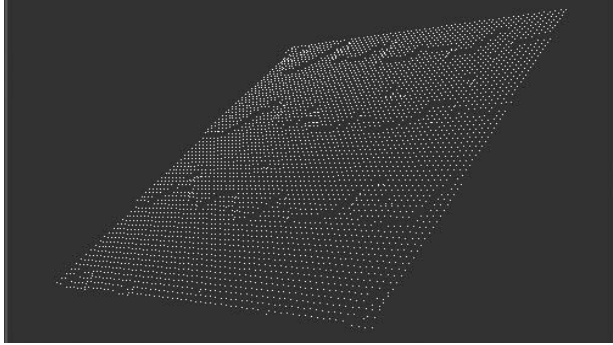


Fig. 2. Point cloud of a scanned plane with simple SSD correlation resulting in aliasing.

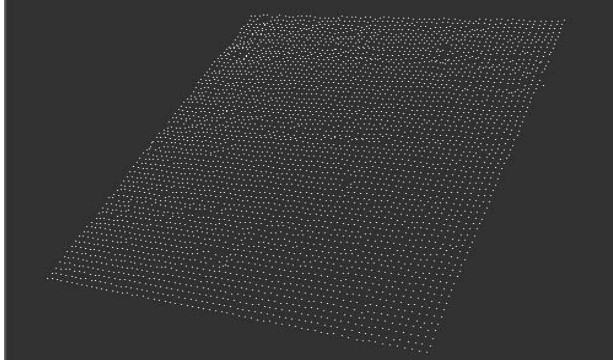


Fig. 3. Point cloud of a scanned plane with our combined SSD and least squares correlation approach.

This equation is linearized using a first order Taylor polynomial:

$$f(x,y) - e(x,y) = r_0 + g(x,y) + g_x da_0 + g_x da_1 + g_x da_2 + g_y db_0 + g_y db_1 + g_y db_2$$

with

$$g_x = \frac{\partial g(x,y)}{\partial x} \quad g_y = \frac{\partial g(x,y)}{\partial y}$$

The set of equations for all pixels of a window $(x^{(i)}, y^{(i)})$, $i \in \{1, \dots, n\}$, can then be formulated as:

$$\begin{pmatrix} g_x^{(1)} & g_x^{(1)}x^{(1)} & g_x^{(1)}y^{(1)} & g_y^{(1)} & g_y^{(1)}x^{(1)} & g_y^{(1)}y^{(1)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_x^{(n)} & g_x^{(n)}x^{(n)} & g_x^{(n)}y^{(n)} & g_y^{(n)} & g_y^{(n)}x^{(n)} & g_y^{(n)}y^{(n)} & 1 \end{pmatrix} \begin{pmatrix} da_0 \\ da_1 \\ da_2 \\ db_0 \\ db_1 \\ db_2 \\ r_0 \end{pmatrix} = \begin{pmatrix} f(x^{(1)}, y^{(1)}) - g(x^{(1)}, y^{(1)}) \\ \vdots \\ f(x^{(n)}, y^{(n)}) - g(x^{(n)}, y^{(n)}) \end{pmatrix}$$

Because of the nonlinearity of $g(x,y)$ the final solution is obtained iteratively. The iteration is normally started with the first approximations:

$$a_0 = a_2 = b_0 = b_1 = r_0 = 0$$

$$a_1 = b_2 = 1$$

After each iteration the improvements $da_0, da_1, da_2, db_0, db_1, db_2$ are added to the parameters $a_0, a_1, a_2, b_0, b_1, b_2$.

r_0 is a naturally linear parameter and therefore is absolutely determined. At the end of each iteration the original image has to be resampled using the new set of parameters. For this purpose the gray values have to be interpolated, e.g. using bilinear interpolation.

In our combined SSD and least squares correlation approach we use the epipolar constraint to restrict the search space to one dimension. By doing this we obtain a problem of linear complexity. This does not only considerably speed up the search, it also reduces wrong matches in almost any random pattern to a minimum. Remaining wrong matches are filtered efficiently and effectively on point cloud level as one can see in the scan results in chapter V.

The whole correlation process is illustrated by the following pseudo-code algorithm, where (i,j) is the pixel in the template for which the correspondence is to be found for as described above:

```

FIND_CORRELATION(f: image, g: image, i: int, j: int)
begin
  min := max_ssd_error
  for each pixel (i,j) of epipolar line do
    (error, d1, d2) := SSD(f, g, i, j);
    if error < min then
      (min, a0, b0) := (error, d1, d2);
    endif
  endfor
  if error ≠ max_ssd_error then
    (error, d1, d2) := GRUEN(f, g, i, j, a0, b0);
    if error < max_gruen_error then
      ProcessCorrespondence(i, j, i+d1, j+d2);
    endif
  endif
end

```

It is substantial to pick good choices for the constants max_ssd_error and max_gruen_error . We have made good experiences with the values $\text{max_ssd_error} = 5000n$ and $\text{max_gruen_error} = 1500n$.

As a conclusion we want to mention that the strategy is to permit even bad matches as the result of the search for a precondition, since a bad match can be justified by perspective distortions at this stage. Then, the least squares algorithm will verify whether the match was a real match. This approach does not only speed up the search substantially in comparison to an implementation solely based on the least squares algorithm, we have also made the experience that it contributes to the robustness of the correlation procedure (figure 4).



Fig. 4. Two local sample patterns for demonstration of the Gruen Algorithm; each left pattern in a row is a template pattern, each middle one is the actually captured pattern (distorted by scene geometry) and each right one is the pattern after application of the Gruen Algorithm

B. Fast 3D Registration

One important requirement for interactive scanning of dynamic scenes is the ability to do a fast online registration from scan to scan. This is not the focus of this paper so we will only give a short insight: the current implementation uses an ICP (*Iterative Closest Point*) variant which utilizes the *vtkLandmarkTransform* class of the *vtk (visualization toolkit)* software library for the minimization of the error measure [5,6,11].

In order to avoid converging to (wrong) local minima a quick search for good initial guesses is performed first by using plausibility considerations. One example for these considerations would be: How fast does the user move the object respectively the scan head normally? On this basis for each of the eight basic translations one iteration of the ICP is performed. Then, the ICP is performed with the best initial guess. If no fast convergence is visible the iteration is aborted and the scan is rejected. Attention has also to be paid to the closest-point-search, which is the bottleneck of the ICP algorithm.

In the near future we want to improve this part by implementing a projection-based search algorithm allowing to perform the search in $O(1)$.

C. Online Visualization

A fast online visualization of the current overall scan result is the basis for this system to provide the described interaction. At the moment visualization is done on point cloud level using the *vtk* library but will be changed to a triangulated and rendered presentation. [11]

Single scans can be triangulated easily and also very fast for we know in which order the object points are acquired. First tests have shown that *vtk* is too slow for this purpose and therefore a visualization module has been implemented using the standard OpenGL API.

Regarding the visualization approach of Rusinkiewicz [4] using a complex *qsplat* mechanism for rendering on point clouds, our approach is to simply assemble single triangulated scans without triangulating the complete result. However this has to be done in a post process.

IV. IMPLEMENTATION

A. Testbed Hardware

In our testbed hardware we are using a standard video beamer for projection purposes. This enables us to easily switch different resolutions and different patterns.

The camera in the test setup is a standard digital video camera which is connected to a standard BT878-based frame grabber. The system is assembled using a profile construction to allow easy adjustability in all directions. In the left corner of the base plate one can see a bracket, which serves as a stopper for the acrylic calibration plates (figure 5).

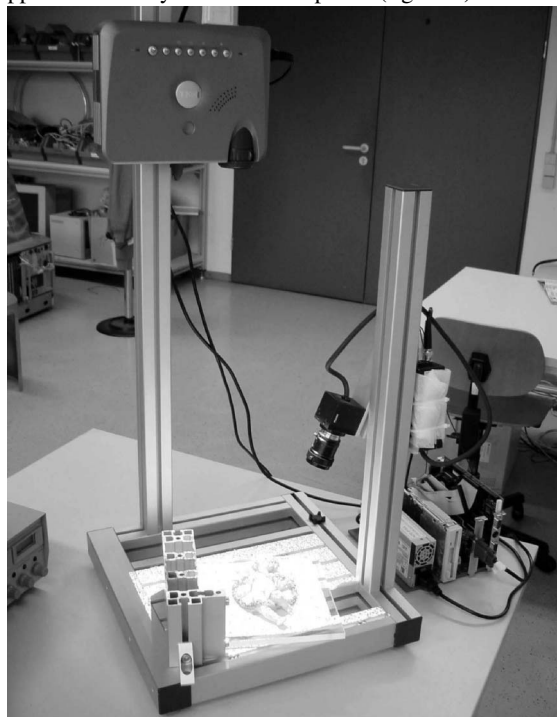


Fig. 5. Photo of the testbed hardware consisting of a standard miniature video beamer on top and camera on the right.

Since we are still experimenting with different projection patterns at the moment, a small embedded pc is necessary for this purpose, located at the right side of the testbed.

We want to mention that naturally it does not make any difference whether the object is moved or the complete scan unit is moved or even both the object and the scan unit are moved at the same time.

B. Software Environment

The algorithms as described above were implemented in C++ using the operating system Windows 2000 and the development environment Microsoft Visual Studio 6. The frame grabber was interfaced utilizing the *VfW (Video for Windows)* interface. The graphical user interface was realized with a Win32 API dialog. For online visualization a *vtk* 3D render window was used [11].

V. RESULTS

A. Examples and Screenshots

In the first test runs of the system we tried to scan a variety of different objects: a textured box (a packet of cigarettes), a small goose figurine made of plaster, pieces of chalk, a teeth plaster cast, and so on. Here we want to show screenshots of the acquired camera image and of the resulting point cloud from one single shot scan of a teeth plaster cast (fig. 6 + 7).

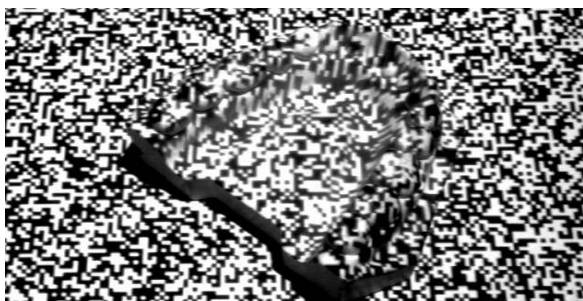


Fig. 6. Teeth plaster cast with projected speckle pattern.

The white noise pattern can be recognized easily. We have chosen a relatively rough resolution of 320x240 in order to benefit from oversampling in the correlation procedure. The resolution of the camera is 640x480.

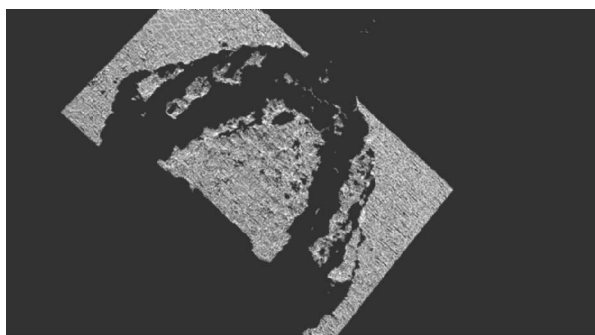


Fig. 7. Resulting sample scan (point cloud) of the plaster cast.

B. Considerations on Accuracy

Due to the fact that the implementation on hand has just been finished by the time writing this paper we have not managed to do intensive test runs yet. Anyway the calibration procedures deliver average errors of approx. 0.03mm for the camera calibration and approx. 0.08mm for the projector calibration. Of course this has to be put in relation to the chosen focal length of the camera (zoom, approx. 12mm, CCD-to-object distance approx. 250mm, 2/3" CCD).

Another preliminary test was to calculate a regression plane through a point cloud of the scan of a plane, delivering a maximal deviation of approx. 0.5mm and an average deviation of approx. 0.2mm.

VI. CONCLUSION AND FUTURE WORK

This paper has described an effective and efficient approach to 3D shape acquisition based on a combined SSD and least squares correlation approach. The algorithms described above have been implemented in a test application and their functioning has been proven. The testbed hardware consisting of a video projector and a digital camera on a frame assembly is easily scalable in geometry, resolution and focal distance. The next step will be to do a theoretical consideration of the system's errors and the achievable accuracy.

At the time the camera images are preprocessed by simply spreading the gray value ranges in tiles of a fixed size, in order to obtain a normalized form as a basis for finding correspondences on surfaces of different colors. This procedure is still to be improved, primarily by incorporating it into the resampling procedure of the least squares correlation algorithm.

Another plan for the future is to replace the video projector with a strobe-lighted gobo projector to minimize velocity distortions.

The described ICP variant is already implemented and works fine with test data sets. Two important optimizations, namely the implementation of a projection-based search algorithm, which has been mentioned above, and the replacement of the *vtkLandmarkTransform* class by a more efficient own implementation, are still to be done.

REFERENCES

- [1] H. Hoppe, S. Däuber, C. Kübler, J. Raczkowsky, H. Wörn, "A New, Accurate and Easy to Implement Camera and Video Projector Model", in *Medicine Meets Virtual Reality, Conf. 02/10 (MMVR)*, Newport Beach, January 2002, pp.204-206
- [2] A. W. Gruen, "Adaptive Least Squares Correlation: A Powerful Image Matching Technique", in *South African Journal of Photogrammetry, Remote Sensing and Cartography, Vol. 14, No. 3*, 1985, pp. 175-187
- [3] T. Luhmann, *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen*. Wichmann, Heidelberg / Germany, 2000 (german)
- [4] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, "Real-Time 3D Model Acquisition", in *Siggraph 2002*, San Antonio / Texas / USA, July 2002
- [5] S. Rusinkiewicz, M. Levoy, "Efficient Variants of the ICP Algorithm", in *Proc. of the 3. Int. Conf. on 3D Digital Imaging and Modeling (3DIM)*, Quebec / Canada, 2001
- [6] P. J. Besl, N. D. McKay, "A Method for Registration of 3-D Shapes", in *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2*, February 1992, pp. 239-255
- [7] F. Devernay, O. Bantiche, E. Coste-Maniere, "Structured light on dynamic scenes using standard stereoscopy algorithms", in *Institut National de Recherche en Informatique et en Automatique (INRIA), Research Report 4477*, June 2002
- [8] P. Vuylsteke, A. Oosterlinck, "Range Image Acquisition with a Single Binary-Encoded Light Pattern", in *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No. 2*, February 1990, pp. 148-164
- [9] E. Mouaddib, J. Battle, J. Salvi, "Recent Progress in Structured Light in order to solve the Correspondence Problem in Stereo Vision", in *Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation*, Albuquerque, New Mexico, April 1997, pp. 130-136
- [10] E. Trucco, A. Verri: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, USA, 1998
- [11] vtk, visualization toolkit, Kitware Inc., free download via <http://www.vtk.org>