Embedded System for 3D Shape Reconstruction

Raja Giryes[1,2], Alexander M. Bronstein[2], Yair Moshe[1], Michael M. Bronstein[2]
[1]Signal and Image Processing Lab (SIPL), Department of Electrical Engineering
[2]Geometric Image Processing Lab (GIP), Department of Computer Science,
Technion – Israel Institute of Technology
Technion City, Haifa 32000, Israel
http://sipl.technion.ac.il, http://gip.cs.technion.ac.il

Email: raja@cs.technion.ac.il, bron@cs.technion.ac.il,
yair@sipl.technion.ac.il, mbron@cs.technion.ac.il

Abstract

Many applications that use three-dimensional scanning require a low cost, accurate and fast solution. This paper presents a fixed-point implementation of a real time active stereo three-dimensional acquisition system on a Texas Instruments DM6446 EVM board which meets these requirements. A time-multiplexed structured light reconstruction technique is described and a fixed point algorithm for its implementation is proposed. This technique uses a standard camera and a standard projector. The fixed point reconstruction algorithm runs on the DSP core while the ARM controls the DSP and is responsible for communication with the camera and projector. The ARM uses the projector to project coded light and the camera to capture a series of images. The captured data is sent to the DSP. The DSP, in turn, performs the 3D reconstruction and returns the results to the ARM for storing. The inter-core communication is performed using the xDM interface and VISA API. Performance evaluation of a fully working prototype proves the feasibility of a fixed-point embedded implementation of a real time three-dimensional scanner, and the suitability of the DM6446 chip for such a system.

1. Introduction

Acquisition of the three-dimensional geometry of a shape arises in a wide variety of applications, including biometrics, robot navigation, movie and video game production, industrial design and modeling, to name a few. In many of these applications, the geometric data is required to be captured in real time at frame rates comparable with normal video capture. Despite the availability of different acquisition methods, a low-cost high-speed real time 3D scanner still remains a non-trivial technological challenge. In particular, one of the main purposes of the 3D scanner presented in this paper is a biometric system based on 3D face recognition [1]. This application requires a fast, accurate and inexpensive solution.

In standard 2D image acquisition, light reflected from objects in a three-dimensional scene is captured by a camera, which introduces depth ambiguity, as all objects lying on rays originating from the camera focal point are projected to the same point in the image plane. This ambiguity can be resolved by acquiring the same scene by another camera with a different optical axis. While in each of the obtained images, the depth of a pixel is ambiguous up to a ray, the intersection of two corresponding rays in the two cameras uniquely defines a single point in the three-dimensional space. Computation of such an intersection is usually referred to as triangulation. It is assumed that the location and orientation of the cameras relative to some world system of coordinates is known; in practice, it is obtained through a calibration process performed as a part of the system setup.

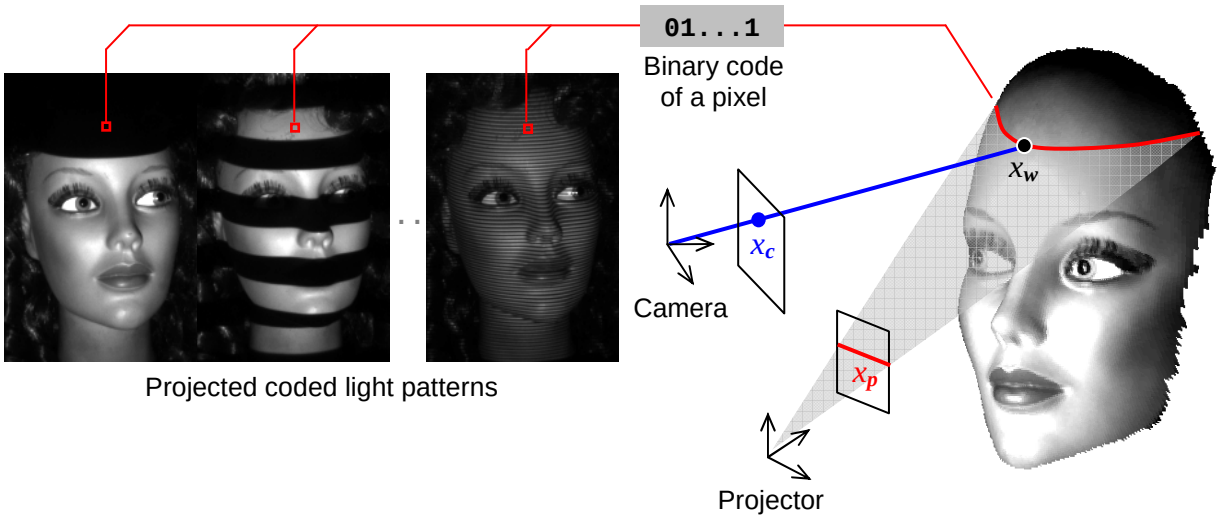

Figure 1: Scheme of three-dimensional object reconstruction in a coded light scanner system.

Geometry reconstruction from multiple views, often referred to as passive stereo, is probably the simplest and the most accessible 3D acquisition technique. However, the need to compute correspondences between the images acquired by multiple cameras raises numerous challenges. It appears that apart from being computationally intensive, passive stereo usually produces inaccurate reconstruction results, unsuitable in many applications.

A more accurate alternative to passive stereo is a family of the so-called active stereo or structured light reconstruction methods, where instead of a camera pair, one of the cameras is replaced by a projector, illuminating the scene with a one-dimensional light pattern that encodes the projection plane direction. The pattern is captured by the camera, giving the projection plane parameters at each pixel. This allows using standard triangulation techniques to reconstruct the geometry without the need to solve for correspondences.

Structured light methods differ mainly in the type of code used. Here, we adopt a time-multiplexed Grey code, consisting of projecting a sequence of black and white patterns encoding the projection plane location as the bits of a binary number. Such a choice is suitable for scenes with low to medium motion, and is dictated mainly by reconstruction simplicity.

As presented in system scheme in Figure 1, the camera and the projector can be thought of as a pair of 2D and 1D cameras with known correspondence. Formally, a point x_W in the world system of coordinates is projected to a point x_C in the camera image plane, and a point x_P in the projection plane. Assuming the pin-hole model for the camera and the projector, the projection can be described in homogeneous coordinates by

$$x_C = C_C x_W \tag{1}$$

and

$$x_P = C_P x_W \tag{2}$$

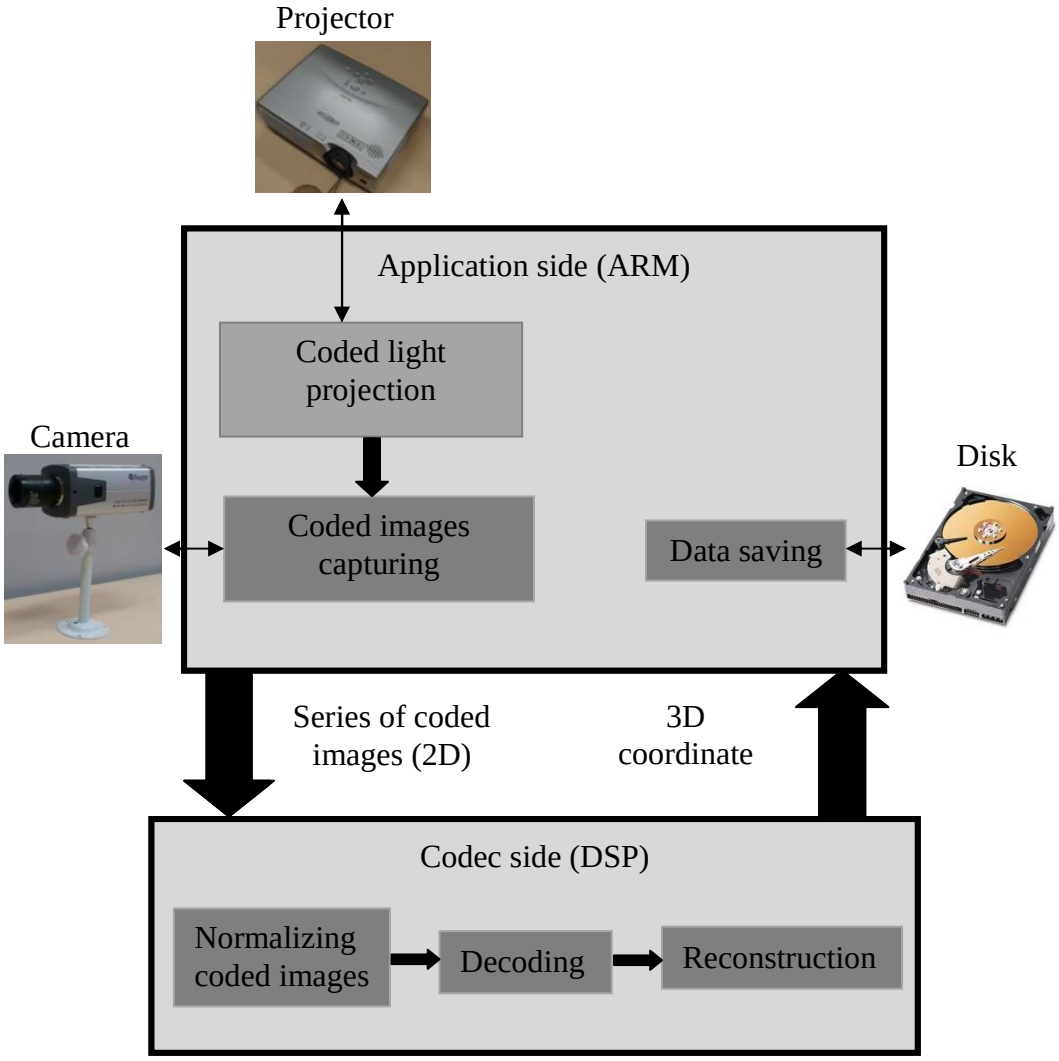

Figure 2: Division of tasks between the ARM and the DSP.

where C_C and C_P are, respectively, a 3×4 and a 2×4 perspective projection matrices (PPMs) [2]. These matrices describe the intrinsic properties as well as extrinsic properties of the camera and the projector respectively, and are obtained in a calibration process. The system acquires an image, in which x_C at each pixel is directly available from the pixel location, and x_P is computed by decoding the light code. The reconstruction process aims at finding the inverse projection $(x_C, x_P) \mapsto x_W$ as described in [3].

Since the processing of each pixel in the input images depends only on a small neighborhood of pixels, simple vectorization of the reconstruction algorithm and well-structured memory access are possible. This, in turn, allows efficient implementation on existing vector and DSP architectures. Our previous implementation of a structured light 3D scanner was based on the Intel Pentium architecture and took advantage of the SSE2 extensions, achieving real time reconstruction at about three VGA frames per second [3]. State-of-the-art graphics hardware such as the latest generation of programmable GPUs (Graphics Processing Units) also appears to be architecturally suitable for a fast implementation. However, both solutions can hardly be considered cost- or power-efficient, and hence are disadvantageous in the considered application.

Floating-point Algorithm	Fixed-point Algorithm
Input: I_H (fully illuminated image), I_L (fully darkened image), $I_1, \dots, I_N$ (N coded images, I_1 is the MSB and I_N is the LSB), calibration data (for the fixed-point algorithm the data is scaled)	
Output (floating-point): X, Y, Z ($3D$ coordinates)	Output (fixed-point): X, Y, Z, W (homogeneous coordinates)
1. Normalization $I_i = \frac{I_i - I_L}{I_H - I_L}, i = 1 \dots N$ 2. Binarization $I_i = \mathrm{I_i} >$threshold, , i $= 1 \dots$ N 3. Decoding $x_P = \Sigma_{i=1}^{N} 2^{-i} I_i$ 4. $Reconst$ruction $X = \frac{f_Y(calibration\ data, x_P)}{g(calibration\ data, x_P)}$ $Y = \frac{f_Y(calibration\ data, x_P)}{g(calibration\ data, x_P)}$ $Z = \frac{f_Z(calibration\ data, x_P)}{g(calibration\ data, x_P)}$	1. Normalization For each i=1..N: $I_i = lookup\ table(I_i - I_L, I_H - I_L)$ 2. Binarization $I_i = \mathrm{I_i} >$threshold, , i $= 1 \dots$ N 5. Decoding $x_P = \Sigma_{i=1}^{N} 2^{N-i} I_i$ 3. Reconstruction $X = f_Y(calibration\ data, x_P)$ $Y = f_Y(calibration\ data, x_P)$ $Z = f_Z(calibration\ data, x_P)$ $W = f_Z(calibration\ data, x_P)$

Table 1: Pseudo-code of the fixed-point and floating-point versions of the algorithm. f_X, f_Y, f_Z, g are functions that contain only multiplications and summing of elements.

In this paper, we consider a cost- and power-efficient embedded implementation on a DSP. We present a fixed-point implementation of a real-time active stereo acquisition system on a Texas Instruments DaVinci™ system-on-chip.

The remainder of this paper is organized as follows: In Section 2, the fixed-point reconstruction algorithm and its implementation on the DM6446 are outlined. Section 3 summarizes the reconstruction accuracy and system performance results. Section 4 concludes the paper.

2. Implementation Details

Our prototype embedded system was implemented on the DM6446 EVM board [4]. The DM6446 chip includes an ARM9 core @ 297 MHz, and a Texas Instruments C64x+ DSP core @ 597 MHz. In the sequel, first fixed-point reconstruction algorithm is described. Next, its implementation on the EVM is outlined.

2.1 Fixed-point Reconstruction Algorithm

The reconstruction algorithm is divided into two main stages: pattern decoding and actual reconstruction. At the decoding stage (stages 1-3 in Table 1), a matrix X_P of the projector coordinates is computed for each pixel from the series of N input images $I_1 \dots I_N$. The i-th image I_i encodes by the absence or presence of light the i-th bit of a Grey-coded binary number, representing the projection plane location quantized to 2^N discrete levels. I_1 represents the most significant bit (MSB) and I_N represents the least significant bit (LSB).

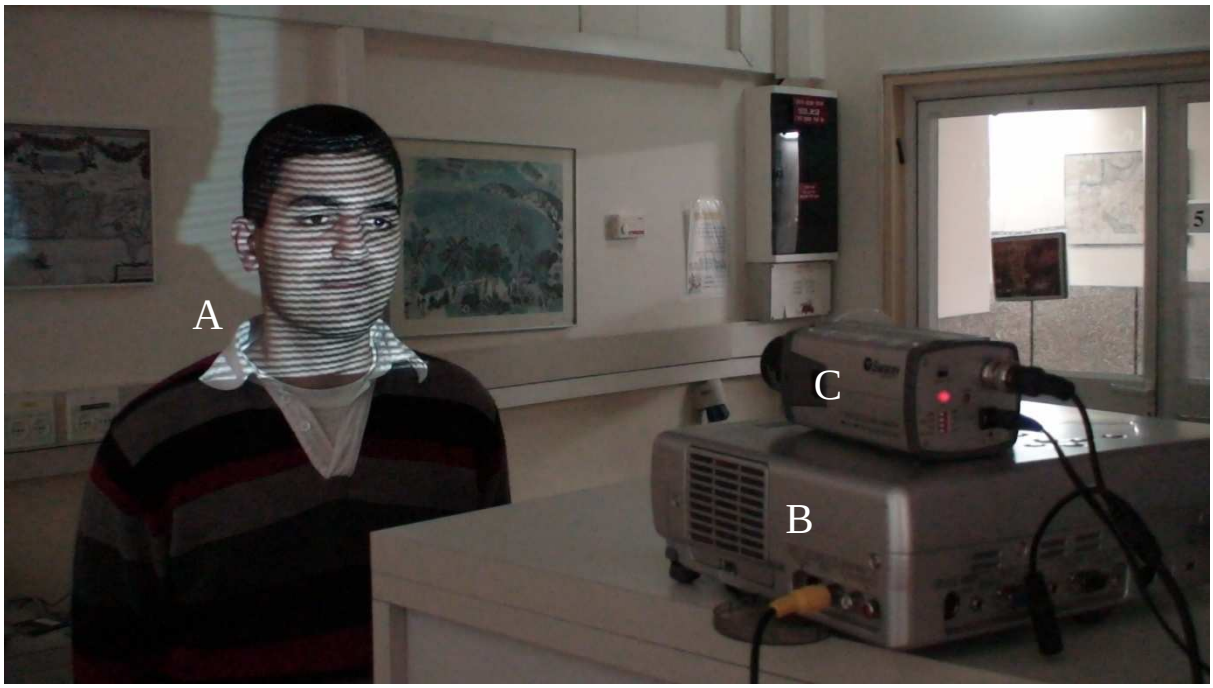

Figure 6: Capturing coded light (A) projected by a standard projector (B) using a standard camera (C).

Due to the variability of ambient illumination conditions as well as shape reflectance properties, the system also acquires a fully illuminated and a fully darkened image (denoted by I_H and I_L respectively). These images are used to normalize the binary patterns (stage 1 in Table 1). The range of values in the image is $[0,255]$, which implies that the range of the values in the normalization denominator and nominator in $\frac{I_i - I_L}{I_H - I_L}$ is also $[0,255]$ ($I_i \geq I_L$ and $I_H \geq I_L$ because I_L is a fully darkened image). Thus, the number of possible normalization values is not large and the division operation can be efficiently carried out using a lookup table. After normalization the binary number at each pixel is determined (stage 2 in Table 1) by global thresholding for the MSBs and locally adaptive thresholding for the LSBs. Local thresholding is performed based on the average pixel values in a pixel's neighborhood. In the fixed-point version neighborhood size is chosen to be a power of 2 and thus a shift operation can be used instead of division. The binary values of each coded image are summed with the corresponding weights (stage 3 in Table 1) giving the Grey code of each pixel. The corresponding binary code gives the projector coordinates x_{p} are calculated.. In the fixed-point version the weights are integers thus the range of values of x_{p} is $[0,2^{\mathrm{N}} - 1]$. Sub-pixel refinement is performed to reduce quantization effects, as described in [3].

The matrix x_P of the projector coordinates is passed to the reconstruction stage (stage 4 in Table 1), where the 3D world coordinates are computed for each pixel in the image. The camera coordinates of a pixel are simply the place of the pixel in the image. The 3D world coordinates X, Y, Z of each pixel are the ratio between multiplications and sums of elements that depends on calibration data, the projector coordinates and the camera coordinates. The denominator in each of these functions is the same thus it is possiblr to represent the output of the system in homogeneous coordinates as four matrices $X, Y, Z,$ and W. These four matrices represent the world location of every pixel (i, j) as the fractions $(x_{ij}/w_{ij}, y_{ij}/w_{ij}, z/w_{ij})$. Besides being a natural choice for many computer graphics and computer vision applications,

Figure 4: Raw depth image of a human face reconstructed using the proposed fixed-point algorithm

this representation allows for fixed-point processing for the numerators and the denominators, and thus avoids the costly floating-point division.

2.2 Application Implementation

In the described system, the reconstruction algorithm was mapped to the fixed-point DSP core (Codec side in **Error! Reference source not found.**). In order to increase performance, special attention was given to eliminating most of the branches. Tasks related to memory and peripheral management were mapped to the ARM core. Continuous memory was allocated for the use of DSP by the ARM in shared memory space (when running on a standalone DSP, static allocation can be used instead). The standard xDM (eXpressDSP Digital Media) interface [6] was used for memory management. The VISA (Video, Imaging, Speech, and Audio) API was used for inter-core communication. File I/O and camera and projector management were performed using standard Linux libraries and system calls on the ARM core.

3D scanning and reconstruction is performed as illustrated in Figure 2. First, coded light is projected on the scanned object. A series of images with the projected light are captured using the camera. The projection is performed using a standard LCD projector that is operated using the frame buffer driver [7]. Images are captured using a standard video camera, operated using the video for Linux Two interface (v4l2) [8]. The Camera is connected to the EVM board by Composite connection and the projector is connected using RCA connectors. Using the VISA interface, 3D coordinates reconstruction from the sequence of coded images, is performed by the DSP. Reconstruction results ares passed back to the ARM that, in turn, stores the results to disk. Projection of coded pattern and image capturing is demonstrated in Figure 3.

3. Results

Low RMSE (root mean square error) and almost the same visual reconstruction results were obtained with the fixed-point code compared to the floating-point code. In figure 4 a reconstruction result from the fixed-point algorithm is presented. The corresponding result

from the floating-point version of the algorithm looks almost the same so it is not presented. Currently, using VGA resolution and 10 bits of the Grey code, the running time of a non-optimized reconstruction code on the DM6446 is 2.5 seconds compared to 300 ms with SSE2-optimized code on PC. Profiling shows that 77% of the DSP cycles are stalls, caused by many memory accesses to relatively slow external memory. From this and from some other simulations performed, we deduce that running time could be reduced to a running time comparable to the one measured on a PC. If running time is more important than accuracy, a lower number of Grey code bits can be used. For example, the use of 6 bits instead of 10 decreases run time by half, as can be seen in Table 2.

Decoding bits	RMS [mm]
10	1.55
9	1.75
8	2.34
7	3.9
6	7.3
5	14

Table 2: RMS for different number of decoding bits.

4. Conclusions

In this paper, an embedded implementation of a structured light 3D scanner on a Texas Instruments DaVinci™ system-on-chip is described. For this system, a fixed-point version of the time-multiplexed structured light reconstruction algorithm is used. The proposed fixed-point algorithm is generic and is not tailored for on any specific hardware architecture. On the DM6446 the reconstruction algorithm is performed on the DSP core which is efficient is performing such calculations. Data management and communication with a camera a projector is performed by the ARM. The ARM is running Linux thus providing great flexibility in handling such tasks. The fixed-point algorithm is almost as accurate as the floating-point version. This enables an accurate 3D scanner using a low-cost and power-efficient fixed-point DSP. In many applications, such as a face recognition system, considered as the main motivation here, efficiency of implementation plays a crucial role, thus making the DSP platform an advisable one.

5. Acknowledgment

The authors are grateful to Yaron Honen, Yevgeni Litvin, Prof. David Malah, Prof. Ron Kimmel, Nimrod Peleg, Avi Rosen, and Alon Salzman for their help which allowed us to carry out this project.

This research was partly supported by Elias Fund for Medical Research and by Horowitz fund.

References

[1] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Three-dimensional Face Recognition," International Journal of Computer Vision (IJCV), vol. 64 (1), pp. 5-30, August 2005.

[2] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2000.

[3] A. M. Bronstein, M. M. Bronstein, E. Gordon, and R. Kimmel, "High-resolution Structured Light Range Scanner with Automatic Calibration," Technical Report CIS-2003-06, Dept. of Computer Science, Technion, Israel, August 2003.

[4] Spectrum Digital, "DaVinci-DM644x Evaluation Module Technical Reference (Rev.E)," March 2007.

[5] Texas Instruments, "TMS320DM6446 Digital Media System-on-Chip (Rev. E)," sprs283e, March 2007.

[6] Texas Instruments, "xDAIS-DM (Digital Media) User Guide (Rev. B)," spruec8b, Jan. 2007.

[7] FBDev – Development of Frame Buffer Drivers, http://linux-fbdev.sourceforge.net/, 2007.

[8] Video for Linux Two (v4l2), http://www.thedirks.org/v4l2/, 2007.