GPU Accelerated 3D Face Registration / Recognition

Andrea Francesco Abate, Michele Nappi, Stefano Ricciardi, and Gabriele Sabatino

Dipartimento di Matematica e Informatica, Università degli Studi di Salerno,
20186, Fisciano (SA), Italy
{abate,mnappi,sricciardi,gsabatino}@unisa.it

Abstract. This paper proposes a novel approach to both registration and recognition of face in three dimensions. The presented method is based on normal map metric to perform either the alignment of captured face to a reference template or the comparison between any two faces in a gallery. As the metric involved is highly suited to be computed via vector processor, we propose an implementation of the whole framework on last generation graphics boards, to exploit the potential of GPUs applied to large scale biometric identification applications. This work shows how the use of affordable consumer grade hardware could allow ultra rapid comparison between face descriptors through their highly specialized architecture. The approach also addresses facial expression changes by means of a subject specific weighting masks. We include preliminary results of experiments conducted on a proprietary gallery and on a subset of FRGC database.

1 Introduction

Three dimensional face representation is object of growing interest from the biometrics research community, as witnessed by the large number of approaches to recognition proposed in the last years, whose main focus has been accuracy and robustness, often considering the computing time required a minor issue. This fact is easily understandable considering the serious challenges related to face recognition which sometimes push researchers to exploit metrics involving time intensive computing. According to literature, 3D based methods can exploit a plurality of metrics [1], some of which, like Eigenface [2], Hausdorff distance [3] and Principal Component Analysis (PCA) [4], have been originally proposed for 2D recognition and then extended to range images. Other approaches instead, have been developed specifically to operate on 3D shapes [5], like those exploiting Extended Gaussian Image [6], the Iterative Closest Point (ICP) method [7], canonical image [8] or normal map [9]. One line of work is represented by multi-modal approaches, which typically combine 2D (intensity or colour) and 3D (range images or geometry) facial data and in some case different metrics, to improve recognition accuracy and/or robustness over conventional techniques [10-12]. However, as the diffusion of this biometric increases, the need for one-to-many comparison on large galleries becomes more frequent and crucial to many applications. Unfortunately, a matching time in the range of seconds (or even minutes) is not rare in 3D face recognition, so, as claimed by Bowyer et al. in their 2006 survey "one attractive line of research involves methods to speed up the 3D

S.-W. Lee and S.Z. Li (Eds.): ICB 2007, LNCS 4642, pp. 938–947, 2007.

matching" [1]. To this regard the launch of multi-core CPUs on the market could be appealing to biometric systems developers, but the reality is that not many of the most established face recognition algorithm can take advantage of multithreading processing and, even in case this is possible, the overall theoretical speedup is by a factor 2 or 4 (for PC and workstation class machines), while database size could possibly grow by even orders of magnitude.

It is worth to note that one of the most stable technological trend in the last years for PCs and workstations has been the leap in both computing power and flexibility of specialized processors on which graphics board are based on: the Graphical Processing Units (GPUs). Indeed, GPUs arguably represent today most powerful and affordable computational hardware, and they are advancing at an incredible rate compared to CPUs, with performances growing approximately from 1.7 to 2.3 times/year versus a maximum of 1.4 times/year for CPUs. As an example, the recent G80 GPU core from Nvidia Corp. (one of the market leaders together with ATI Corp.) features approximately 681 millions of transistors resulting in a highly parallel architecture based on 128 programmable processors, 768 MB of VRAM with 86 GB/sec of transfer rate over a 384 bit wide bus.

The advantages in using these specialized processors for general purpose applications, a task referred as General Purpose computation on GPU or GP-GPU, have been marginal until high level languages for GPU programming have emerged. Nevertheless, as GPUs are inherently vector processors and not real general-purpose processing units, not every algorithm or data structure is suited to fully exploit this potential. In this paper, we present a method to register and recognize a face in 3D by means of the same normal map metric. As this metric represents geometry in terms of coloured pixels, it is particularly suited to take full advantage of vector processors, so we propose a GPU implementation aimed to maximize the comparison speed for large scale identification applications. This paper is organized as follows. In section 2. the proposed methodology is presented in detail. In section 3. experimental results are shown and briefly discussed. The paper concludes in section 4.

2 Description of Proposed Methodology

In the following subsections 2.1 to 2.4. we describe in depth the proposed face recognition approach and its implementation via GPU. A preliminary face registration is required for the method to perform optimally but, as it is based on the same metric exploited for face matching, we first describe the normal map based comparison, then we expose the alignment algorithm, and the adaptation necessary to efficiently compute the metric through GPU.

2.1 Representing Face Through Normal Map and Comparing Faces Through Difference Map

Whether a subject has to be enrolled for the first time or a new query (a subject which has to be recognized) is submitted to the recognition pipeline, a preliminary face capture is performed and the resulting range image is converted in a polygonal mesh M. We intend to represent face geometry storing normals of mesh M in a bidimensional

matrix N with dimension $l \times m$. To correlate the 3D space of normals to the 2D domain of matrix N we project each vertex in M onto a 2D surface using a spherical projection (opportunely adapted to mesh size). Then we sample the mesh by means of mapping coordinates and quantize the length of the three scalar components of each normal as an RGB coded color, storing it in a bitmap N by means of the same coordinates as array indexes. More precisely, we assign to each pixel (i, j) in N, with $0 \leq i < l$ and $0 \leq j < m$, the three scalar components of the normal to the point of the mesh surface with mapping coordinates $(l/i, m/j)$. The resulting sampling resolution is $1/l$ for the s range and $1/m$ for the t range. The normal components are stored in pixel (i, j) as RGB colour components.

We refer to the resulting matrix N as the normal map of mesh M. A normal map with a standard colour depth of 24 bit allows 8 bit quantization for each normal component, this precision proved to be adequate for the recognition process. To compare the normal map N_A from input subject to another normal map N_B previously stored in the reference database, we compute the angle included between each pairs of normals represented by colours of pixels with corresponding mapping coordinates, and store it in a new *Difference Map D* with components r, g and b opportunely normalized from spatial domain to colour domain, so $0 \leq r_{N_A}, g_{N_A}, b_{N_A} \leq 1$ and $0 \leq r_{N_B}, g_{N_B}, b_{N_B} \leq 1$. The value θ, with $0 \leq \theta < \pi$, is the angular difference between the pixels with coordinates (x_{N_A}, y_{N_A}) in N_A and (x_{N_B}, y_{N_B}) in N_B and it is stored in D as a grey-scale image (see Fig. 1). To reduce the effects of residual face misalignment during acquisition and sampling phases, we calculate the angle θ using a $k \times k$ (usually 3×3 or 5×5) matrix of neighbour pixels.

Summing every grey level in D results in histogram $H(x)$ that represent the angular distance distribution between mesh M_A and M_B. On the X axis we represent the resulting angles between each pair of comparisons (sorted from 0° degree to 180° degree), while on the Y axis we represent the total number of differences found. This means that two similar faces will have an histogram $H(x)$ with very high values on little angles, while two distinct faces will have differences more distributed. We define a similarity score through a weighted sum between H and a Gaussian function G, as in (3) where σ and k change recognition sensitivity .

$$similarity\_score = \sum_{x=0}^{k} \left(H(x) \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \tag{1}$$

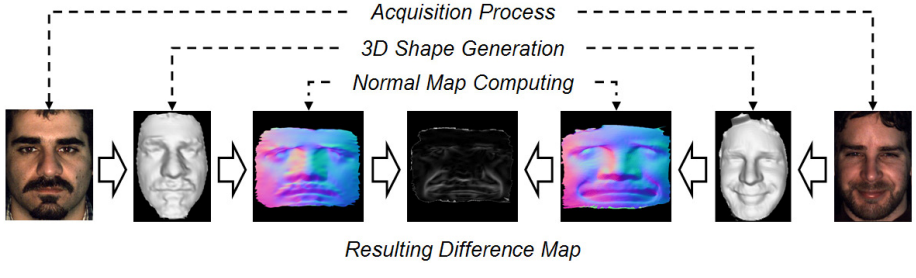

Fig. 1. Face capture, Normal Map generation and resulting Difference Map for two subjects

2.2 Face Registration

A precise registration of captured face is required by normal map based comparison to achieve the best recognition performance. So, the obvious choice could be to use the most established 3D shape alignment method, the Iterative Closest Point (ICP), to this aim. Unfortunately ICP is a time expensive algorithm. The original method proposed by Chen and Medioni [13] and Besl and McKay [14] features a O(N2) time complexity, which has been lowered to O(Nlog(n)) by other authors [15] and further reduced by means of heuristic functions or by shape voxelization and distance pre-computing in its most recent versions [16]. Nevertheless the best performance in face recognition applications are in the range of many seconds to minutes, depending on source and target shape resolution. As the whole normal-map based approach is aimed to maximize the overall recognition speed reducing the comparison time, we considered ICP not suited to fit well into this approach.

So we introduce pyramidal-normal-map based face alignment. A pyramidal-normal-map is simply a set of normal maps relative to the same 3D surface ordered by progressively increasing size (in our experiments each map differs from the following one by a factor of 2). Our purpose is to exploits this set of local curvature descriptors to perform a fast and precise alignment between two 3D shapes, measuring the angular distance (on each axis) between an unregistered face and a reference template and reducing it to a point in which it does not significantly affect recognition precision. The template is a generic neutral face mesh whose centroid corresponds to the origin of the reference system. To achieve a complete registration the captured face has to match position and rotation of reference template.

Scale matching, indeed, is not needed as the spherical projection applied to generate the normal map is invariant to object size. The first step in the alignment procedure is therefore to compute face's centroid which allows to match reference template position offsetting all vertices by the distance from centroid to the axis origin. Similarly, rotational alignment can be obtained through a rigid transformation of all vertices once the angular distance between the two surface has been measured. As we intend to measure this distance iteratively and with progressively greater precision, we decide to rotate the reference template instead of captured face. The reason is simple: because the template used for any alignment is always the same, we can pre-compute for every discrete step of rotation the relative normal map once and offline, drastically reducing the time required for alignment. Before the procedure begins, a set-up is required to compute a pyramidal normal map for the captured face (a set of four normal maps with size ranging from 16x16 to 128x128 has proved to be adequate in our tests). At this time the variables controlling the iteration are initialised, like the initial size m of normal maps, the angular range reduction factor k, and R, the maximum angular range for the algorithm to operate, i.e. the maximum misalignment allowed between the two surfaces. We found that for biometric applications a good compromise between robustness and speed is reached setting this value to 180°, with k=4, but even R=360° can be used if required.

With the first iteration, the smallest normal map in the pyramid is compared to each of k^3 pre-computed normal maps of the same size relative to the coarsest rotation steps of template. The resulting difference maps are evaluated to find the one with the highest similarity score, which represent the better approximation to

alignment (on each axis) for that level of pyramid. Then the template is rotated according to this first estimate. The next iteration starts from this approximation comparing the next normal map in the pyramid (with size m=m*2) to every template normal map of corresponding size found within a range which has now its centre on the previous approximation and whose width has been reduced by a factor k. This scheme is repeated for i iterations until the range's width fall below a threshold value T. At this point the sum of all *i* approximations found for each axis is used to rotate the captured face, thus resulting in its alignment to the reference template (see Fig. 2).

Using the above mentioned values for initialisation, four iterations (i=4) with angular steps of 45°, 11,25°, 2.8° and 0.7° are enough to achieve an alignment adequate for recognition purpose. As the number of angular steps is constant for each level of iteration, the total number of template normal maps generated offline for i iterations is ik^3, and the same applies to the total number of comparisons. It has to be noted that the time needed for a single comparison (difference map computing), is independent by mesh resolution but it depends on normal map size instead, whereas the time needed to precompute each template's normal map depends on its polygonal resolution.

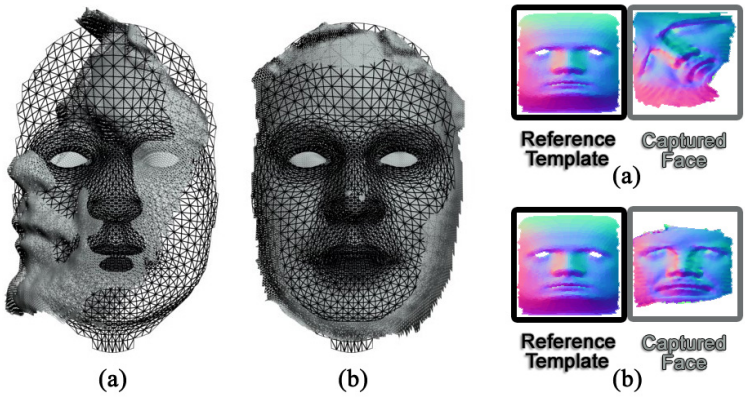

Fig. 2. Face captured (shaded) and reference template (wireframe) before (left) and after (center) alignment. Right: normal maps before (up) and after (bottom) alignment

The template does not need to have the same resolution and topology of captured face, it is sufficient it has a number of polygons at least greater than the number of pixels in the largest normal map in the pyramid and a roughly regular distribution of vertices. Finally, another advantage of proposed algorithm is that no preliminary rough alignment is needed by the method to converge if the initial face misalignment (for each axis) is within R.

2.3 Storing Facial Expressions in Alpha Channel

To improve robustness to facial expressions we introduce the expression weighting mask, a subject specific pre-calculated mask aimed to assign different relevance to different face regions. This mask, which shares the same size of normal map and difference map, contains for each pixel an 8 bit weight encoding the local face surface

rigidity based on the analysis of a set facial expressions of the same subject (see Fig. 3). In fact, for each subject enrolled, eight expressions (a neutral plus seven variations) are acquired and compared to the neutral face resulting in seven difference maps. More precisely, given a generic face with its normal map N_0 (neutral face) and the set of normal maps N_1, N_2, …, N_n (the expression variations), we first calculate the set of difference map D_1, D_2, …, D_n resulting from {`$N_0$ - $N_1$', `N_0 - N_2', …, `N_0 – N_n'}.The average of set {D_1, D_2, …, D_n} is the expression weighting mask which is multiplied by the difference map in each comparison between two faces. We can augment each 24 bit normal map with the Expression Weighting Mask normalized to 8 bit. The resulting 32 bit per pixel bitmap can be conveniently managed via various image formats like the Portable Network Graphics format (PNG) which is typically used to store for each pixel 24 bit of colour and 8 bit of alpha channel (transparency) in RGBA format. When comparing any two faces, the difference map is computed on the first 24 bit of colour info (normals) and multiplied to the alpha channel (mask).

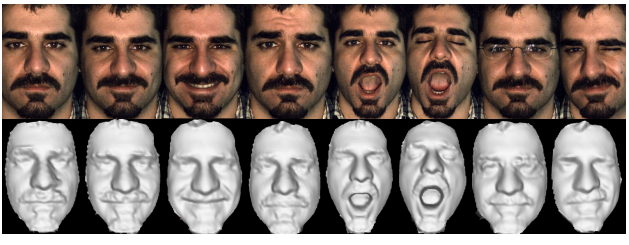

Fig. 3. Facial expressions exploited in Expression Weighting Mask

2.4 Implementing the Proposed Method Via GP-GPU

As briefly explained in the introduction to this paper, GPUs can vastly outperform CPUs for some computational topics, but two key requirements have to be satisfied: (1) the algorithm and the data types on which it operates should conform as much as possible to the computational architecture of GPU and to its specialized memory, the VRAM; (2) the data exchange with main memory and CPU should be carefully planned and minimized where possible. Because the descriptor used in our approach to face recognition is a RGBA coded bitmap the second part of first requirement is fully satisfied, but the first part is not so trivial. Indeed if the comparison stage of two normal maps requires pixel to pixel computation of a dot product, a task easily performed on multiple pixels in parallel via pixel shaders, the computation of histogram and similarity score for their algorithmic nature is not so suited to be efficiently implemented on GPU. This is mainly due to the lack of methods to access and to write in VRAM as we could easily do in RAM via CPU.

For this reason we decided to split the face comparison step from the rank assignment step, by means of a two-staged strategy which relies on GPU to perform a huge number of comparisons in the fastest possible time, and on CPU to work on the results produced from GPU to provide rank statistics, thanks to its general purpose architecture. We addressed the second requirement by an optimised arrangement of descriptors which minimize the number of data transfers from and to the main memory (RAM) and, at the same time, allows vector units on GPU to work efficiently (see

Fig. 4). Indeed we arranged every 1024 normal maps (RGBA, 24+8 bit) in a 32x32 cluster, resulting in a single 32 bit 4096x4096 sized bitmap (assuming each descriptor is sized 128x128 pixels). This kind of bitmap reaches the maximum size a GPU can manage at the moment, allowing to reduce by a factor 1,000 the number of exchanges with VRAM . The overhead due to descriptor arrangement is negligible as this task is performed during enrolment when normal map and expression weighting mask are computed and stored in the gallery. Up to 15,000 templates could be stored within 1GB of VRAM. On a query, the system load the maximum allowed amount of clusters from main memory to available free VRAM, then the GPU code is executed in parallel on any available pixel shader unit (so computing time reduction is linear in the number of shader units) and the result is write in a specifically allocated Frame-Buffer-Object (FBO) as a 32x32 cluster of difference maps. In the next step the FBO is flushed to RAM and the CPU start to compute the similarity score of each difference map storing each score and its index. This scheme repeats until all template clusters have been sent to and processed by GPU and all returning difference map clusters have been processed by CPU. Finally the sorted score vector is outputted. We implemented this algorithm through the Open GL 2.0 library and GLSL programming language.

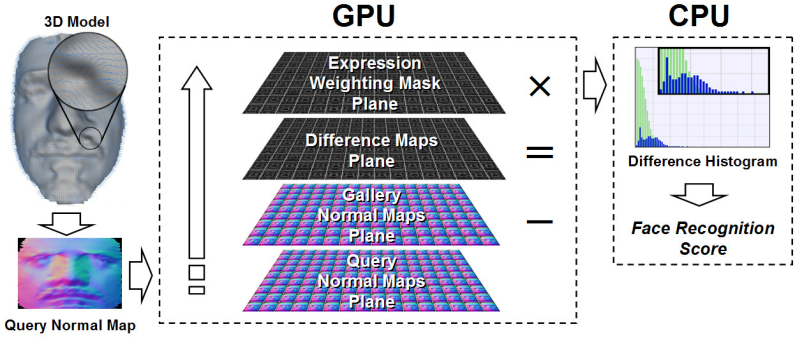

Fig. 4. Schematic representation of GPU accelerated normal map matching

3 Experiments

To test the proposed method four experiments using two different 3D face datasets have been conducted. We built the first dataset acquiring 235 different individuals (138 males and 97 females, age ranging from 19 to 40) in an indoor environment by means of a structured light scanner, the Mega Capturor II from Inspeck Corp.. For each subject eight expressions has been captured (including the neutral one) and each resulting 3D surface has an average of 60-80.000 polygons, with a minimum detail of about 1.5 millimetres. For the second dataset we used 1024 face shapes from release 2/experiment 3s of FRGC database, disregarding texture data. This dataset has undergone a pre-processing stage including mesh subsampling to one fourth or original resolution, mesh cropping to eliminate unwanted details (hair, neck, ears, etc.) and mesh filtering to reduce capture noise and artifacts. For all experiments we set σ =4.5 and k=50 for the Gaussian function and the normal map size is 128×128 pixels.

The first experiment, whose results are shown in Fig. 5-a., measures the overall recognition accuracy of proposed method through the Receiver Operating Characteristic (ROC) curve. The histogram compares the baseline algorithm (blue column, implemented exploiting the FRGC framework and applied on the preprocessed dataset described above) respectively to: proposed method on FRGC dataset using embedded alignment info (violet column), proposed method on FRGC dataset with pyramidal-normal-map based alignment (green column) and proposed method and alignment on our gallery allowing the use of expression weighting mask (orange). The result shown in the third column (green) is slightly better than the one measured on the second column (violet) as the alignment performed by proposed algorithm has proved to be more reliable than the landmarks embedded in FRGC. The best score is achieved in the fourth column (orange) as in this case we exploit both proposed alignment method and the weighting mask to better address expression variations.

The second experiment is meant to measure alignment accuracy, using the first dataset with 235 neutral faces for gallery and 705 (235*3) opened mouth, closed eyes and smile variations as probes. Moreover, the probes have been rotated of known angles on the three axis to stress the algorithm. The results are shown on Fig. 5-b. where after four iterations, 95.1% of probes have been re-aligned with a tolerance of less than two degree and for 73.1% of them the alignment error is below one degree.

The purpose of the third group of experiments is to measure the effect of posing variations and probe misalignment on recognition performance without the alignment step. Also in this case we used the neutral faces for gallery and opened mouth, closed eyes and smile variations, additionally rotated of known angles, as probes. The results in Fig. 5-c. show that for a misalignment within one degree the recognition rate is 98.1%, which drops to 94.6% if misalignment reaches two degrees. As the average computational cost of a single comparison (128x128 sized normal maps) is about 3 milliseconds for an Amd Opteron 2,6 GHz based PC, the total time needed to alignment is slightly more than 0.3 seconds, allowing an almost real time response. The overall memory requirement to completely store the template's precomputed normal maps is just 4 Mbytes. Finally, the fourth experiment shows in Fig. 6. how many templates can be theoretically compared to the query within 1 second if they could fit entirely in VRAM, proving how time wise the GPU based version of proposed method easily outperform any CPU based solution, whatever the processor chosen. To this aim we replicated the 1024 templates from FRGC subset to fill in all available VRAM. The system was able to compare about 85,000 templates per second (matching one-to-15,360 in 0,18 sec. on GeForce 7950 GTX/1024 with 32 pixel shaders) versus about 330 of CPU based version (AMD). In the same figure we compare the performance of different CPUs and GPUs including recently released GPU cores based on specs reported from the two main manufacturers (NVidia 8800 GTS and 8800 GTX featuring 96 and 128 programmable unified shaders respectively). Comparing these results to ICP based registration and recognition methods (typically requiring from a few seconds to tens of seconds for a single one-to-one match) clearly shows that the proposed approach is worth using it, at least timewise, regardless to the dataset dimension, as in a real biometric application the pre-processing phase (mesh

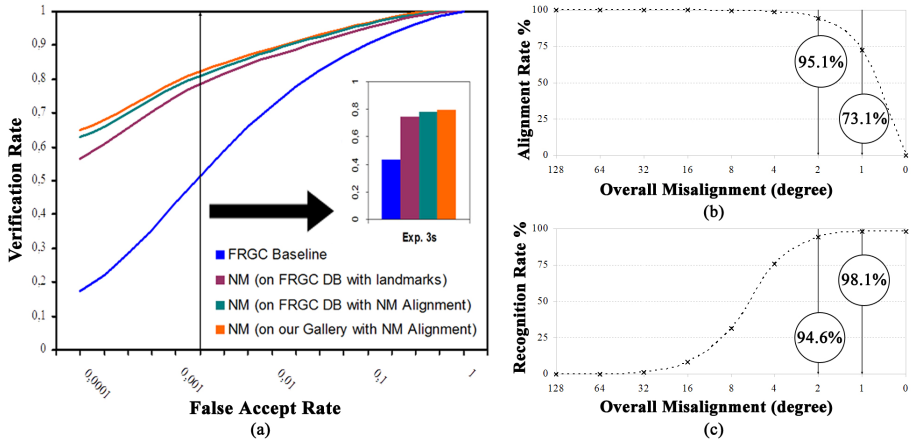

Fig. 5. ROC curve (a), alignment accuracy (b) and its relevance to recognition (c)

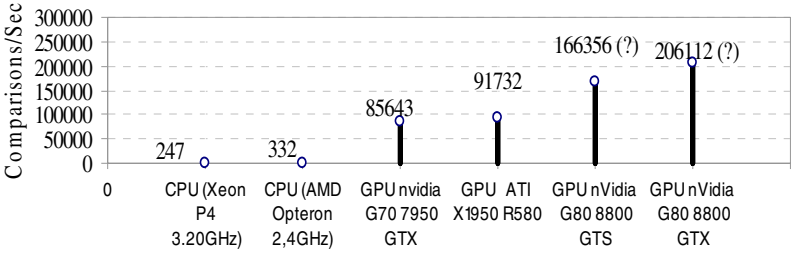

Fig. 6. Number of comparisons/sec for various computational hardware. In the graph CPU means only CPU is exploited, while GPU means that CPU (AMD Opteron 2,4 GHz) + GPU work together according to proposed scheme. (?) is just an estimate based on specs.

subsampling, filtering, cropping performed within 1 second in the tested framework) has to be performed only once at enrolment time.

4 Conclusions and Future Works

We presented a 3D face registration and recognition method optimized for large scale identification applications. The proposed approach showed good accuracy and robustness and proved to be highly suited to take advantage of GPU architecture, allowing to register a face and to compare it to many thousands of templates in less than a second.

As the recent release of Nvidia “Cuda” GPU based programming environments promises further advances in term of general purpose capability, we are currently working to fully implement the method on GPU, including those stages (as normal map and histogram computing) which are still CPU based in this proposal.

References

[1] Bowyer, K.W., Chang, K., Flynn, P.A.: A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. In: Computer Vision and Image Understanding, vol. 101, pp. 1–15. Elsevier, Amsterdam (2006)

[2] Zhang, J., Yan, Y., And Lades, M.: Face Recognition: Eigenface, Elastic Matching, and Neural Nets. Proc. of the IEEE 85(9), 1423–1435 (1997)

[3] Achermann, B., Bunke, H.: Classifying range images of human faces with Hausdorff distance. In: 15-th International Conference on Pattern Recognition, September 2000, pp. 809–813 (2000)

[4] Hesher, C., Srivastava, A., Erlebacher, G.: A novel technique for face recognition using range images. In: Seventh Int'l Symposium on Signal Processing and Its Applications (2003)

[5] Lu, X., Colbry, D., Jain, A.K.: Three-dimensional model based face recognition. In: 7th IEEE Workshop on Applications of Computer Vision, pp. 156–163 (2005)

[6] Tanaka, H.T., Ikeda, M., Chiaki, H.: Curvature-based face surface recognition using spherical correlation principal directions for curved object recognition. In: Third International Conference on Automated Face and Gesture Recognition, pp. 372–377 (1998)

[7] Medioni, G., Waupotitsch, R.: Face recognition and modeling in 3D. In: IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2003), October 2003, pp. 232–233 (2003)

[8] Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Expression-invariant 3D face recognition. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 62–70. Springer, Heidelberg (2003)

[9] Abate, A.F., Nappi, M., Ricciardi, S., Sabatino, G.: Fast face recognition based on normal map. In: Proceedings of ICIP 2005, IEEE International Conference on Image Processing, Genova, Italy, July 2005. IEEE Computer Society Press, Los Alamitos (2005)

[10] Tsalakanidou, F., Tzovaras, D., Strintzis, M.G.: Use of depth and color eigenfaces for face recognition. Pattern Recognition Letters 24(9-10), 1427–1435 (2003)

[11] Papatheodorou, T., Rueckert, D.: Evaluation of Automatic 4D Face Recognition Using Surface and Texture Registration. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, May 2004, pp. 321–326. IEEE Computer Society Press, Los Alamitos (2004)

[12] Gokberk, B., Salah, A.A., Akarun, L.: Rank-based decision fusion for 3D shape-based face recognition. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 1019–1028. Springer, Heidelberg (2005)

[13] Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. Image and Vision Computing 10, 145–155 (1992)

[14] Besl, P., McKay, N.: A method for registration of 3-D shapes. IEEE Transaction on Pattern Analysis and Machine Intelligence 14, 239–256 (1992)

[15] Jost, T., Hügli, H.: Multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images. In: Fourth International Conference on 3-D Digital Imaging and Modeling, October 06 - 10, 2003, pp. 427–433 (2003)

[16] Yan, P., Bowyer, K.: A Fast Algorithm for ICP-Based 3D Shape Biometrics. In: Proceedings of the ACM Workshop on Multimodal User Authentication, December 2006, pp. 25–32. ACM, New York (2006)