Available online at www.sciencedirect.com

SciVerse ScienceDirect

AASRI Procedia 3 (2012) 589 – 594

AASRI Procedia

www.elsevier.com/locate/procedia

2012 AASRI Conference on Modelling, Identification and Control

Design and DSP Optimization of Real-time Multi-Camera Tracking

Xiao Yan[a,]*, Junchuan Yang[b], Bo Yao[c]

[a]*School of Information Science and Engineering, Yunnan University, No.2, Chestwood North Road, Kunming, Yunnan, China, 650091*
[b]*Department of Computer Science, University College London Gower Street, London,United Kingdom,WC1E 6BT*
[c]*School of Computer Science and Electronic Engineering, University of Essex Wivenhoe Park, Colchester, United Kingdom,CO4 3SQ*

Abstract

In recent years there is a dominanttrend towards the deploymentof advanced visual analysisalgorithms on embedded platforms. However, this deployment is very challenging as embedded systems usually afford limited resources such as calculating performance,memory and power. Therefore, to address this problem, In this paper we introduces anoptimized multi-camera tracking system based on multiple TI DSP TMS320DM6446for trackingobjects across multiple embedded smart cameras which embedsintelligent processors. Firstly, the overlapping areas and homography projection relations between adjacent cameras' field of view is calculated. Based on the relations of cameras view obtained and tracked data of each single camera,a homography based target handover procedure is done for long-term multi-camera tracking.After that, we fully implemented the tracking system on the embedded smart cameras developed by our group.Finally, to combat the huge computational complexity, a novel hierarchical optimization method is proposed. Experimental results demonstrate the robustness and real-time efficiency in dynamic real-life environments and the computationalburden is significantly optimized by 98.84% which is low enough for further biometrics tasks such as recognition.

*Xiao Yan. Tel.: +0-86-18082958972.
*E-mail address:*xxiaoyen@gmail.com.

doi:10.1016/j.aasri.2012.11.093

*Keywords:*Embedded systems, distributed smart cameras, visual tracking

1. Introduction

Together with the increasing population density in metropolitan cities, the demand for maintainingsecurity of citizens is increasing. In alignment with this demand, the smart Closed Circuit Television system(CCTV) surveillance[1] isbecoming pervasive. Hence, automated surveillance has become an emerging technology since the lastfew years.The task of automated surveillance for public locations, which are usually crowded and wide-area, such as public transport stations usingindependent smart cameras [2] almost impossible due to the limitation of cameras' field of view and the heavytarget occlusion [3]. Hence, scene surveillance using a cooperative multi-camera network [4] isbecoming the preferred solution for surveillance camera users, as will not require major hardwareupgrade. Therefore, a multi-camera tracking method is introduced in this paper, which has been completely implemented on our embedded smart cameras and testedon traffic surveillancenear our campus. Nevertheless, due to the computing limitation of the processing unit in each smart camera, advanced video processing algorithms which are usually of high computational complexity, cannot be performed without optimization. Consequently, in this paper, anovel hierarchicaloptimizationparadigm with several practical techniques for commonoptimization is presented. According to our experiment results, the overall performance is remarkably boosted using our proposed optimization methods.

The rest of this paper is organized as follows. In section 2, we provide the proposedmulti-camera tracking system. Section 3 presents the proposed hierarchicaloptimization methodology. Section 4 presents the experiments and results and finally the conclusions are presented in section 5.

2. The proposedmulti-camera tracking system

To cooperatively track and monitor movingtargets in large-scale area, multiple overlapping cameras are utilized to observerwide surveillance scenes fromdifferent views.Firstly, in the initial stage of our system, multi-camera calibration is done to gain the image plane correspondence relationshipbetween the adjacent cameras byhomography transformation [5] for target hand-off. After that, multi-objects tracking is performed continuously in each single smart camera which is shown in Fig. 1 (a) to obtain the surveillance information of the moving targets based on which object hand-off is carried out for multi-camera tracking shown in Fig. 1 (b).

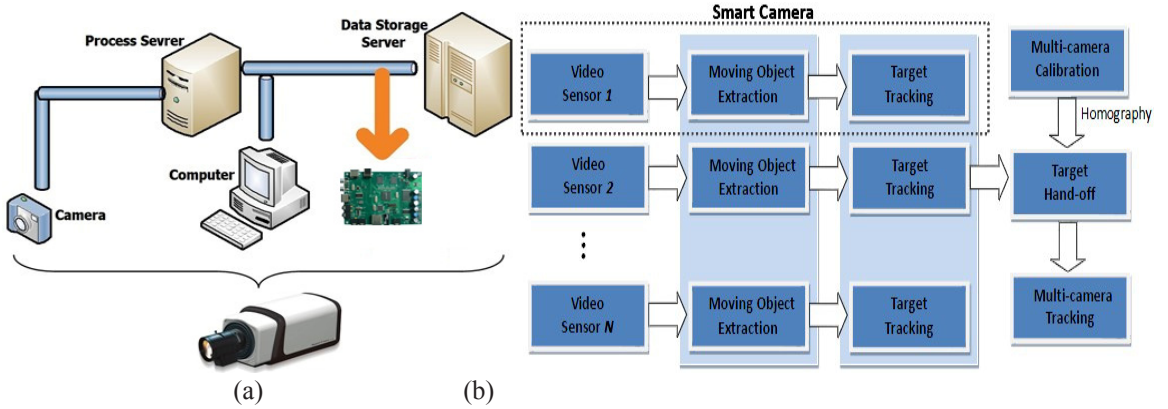


Fig. 1 Structure block diagrams (a) Integratedsmart camera; (b) Overall system structure.

The entire embedded system is divided into three components described in the block diagram above. Firstly, a CCD color sensor providing NTSC or PAL video is used for capturing raw video data. Then, the embedded video analysis agent is designed by employing a DaVinciTMS320DM6446 [8] dual-core device with an ARM9 and C64+ DSP. Fig. 2 displays the hardware architecture of embedded video agent designed in this paper. And Fig. 3 shows the highly integrated 3-level embedded system developed by our research group.

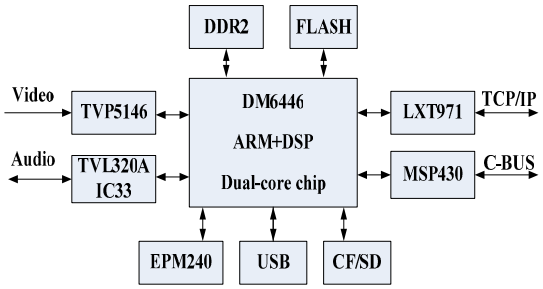


Fig. 2 Hardware ArchitectureFig. 3Our 3-level EmbeddedVideo Agent

2.1Single-camera tracking

In multi-camera network surveillance, single-camera tracking is the fundamental module to obtain the information of the moving targets such as position, motion trajectory, shape, etc. Therefore, in our system, we utilized the tracking paradigm in [5]. Specifically, Gaussian Mixture Models (GMM) is employed to compute the background images of the surveillance scenes. Then, foreground objects (Blob) extraction is done to gain the bounding boxes and centroids of the moving targets. Finally, object tracking is performed based on Mean-shift and Kalman filter to analyze the motion history and trajectories.

2.2 Multi-camera tracking

When moving objectsenter the overlapping areabetween adjacent cameras, ground plane homographymappingis employed tocreate the viewpoint correspondence bymapping and matchingtargetcentroid positions between neighboring cameras, which is defined as follows:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (1)$$

Where$H(h_{11}\sim h_{33})$ denotes3×3homography matrix describing the projection relationship of the two cameraswhile (x_i, y_i) and (x_i', y_i') represent the correspondingcentroids of the moving targets in eachcamera.

To calculate the homography matrix, in the initial stage of our system, we extract four best pairs of feature points from background images ofthe two surveillance scenes by using ASIFT [] which is robust in dynamic real-life environments. Finally, based on the feature points, L-M(Levenberg-Marquardt) [] is performed to compute the homographywith Projection Error (PE) minimization equation defined as follows:

$$PE = \sum_i \sqrt{\left(\frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + 1} - x_i'\right)^2 + \left(\frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + 1} - y_i'\right)^2} \quad (2)$$

Then, based on the homography, target hand-off is done by examining *PE* between the centroid of two target candidates. If $PE \leq D^T$, where D^T is a parameter which can be dynamically set,then the two candidatesarecorrespondingtargets and marked the bounding box and trajectory in a unique color, shown in Fig. 4.

(a) (b) (c) (d)

Fig. 4 Target hand-off for multiple camera tracking, (a)(b) before corresponding; (c)(d) after corresponding.

3. DSP performance optimization

To achieve the real time performance of the embedded DSP system, in this paper, a hierarchical optimization method is proposed based on DM6446 are used. According to the performance evaluation done by Code Composer Studio (CCS)profiling module, main performance bottlenecks are found, which areGaussian Mixture Model(GMM) for background reconstruction and moving object(Blob) extraction, respectively.Therefore, the proposed hierarchical optimization method isprimarilyconcentrated in these two modulesusing project-level optimization, algorithm-level optimization and code-level optimization.

3.1Project-level optimization

To maximize C/C++ compiler performance, the DSP code can be optimized comprehensively by using proper compiler setting [6]. Firstly, software pipelining is used to schedule instructions in a loop so that multiple instructions of the loop are executed in parallel. In C6000 compiler, we use "*-o2*" and "*-o3*" compiler options to arrange software pipelines for the codes automatically. Then, "*-pm*", "*-mt*" and "*-op3*"compiler settings are employed in our compiler to reduce theperformance costin loop iterations. Additionally,to boost the efficiency,in the proposed system, the data which is frequently visited and processed are stored in internal DSP memory and important functions and procedures are executed in CACHE which is supreme fast memory.

3.2 Algorithm-level optimization

Since the detailed information of moving target such as texture, shape, etc. is not significantly essential for GMM and blob extraction, therefore before these two procedures, input video signal can be down-sampled to a smaller resolution for computational complexity reduction. So in our system, we resized the input video from D1 (*720*×*576*) to CIF (*360*×*288*) using resizer module in Video Processing Subsystem (VPSS) which is a standalone peripheral device on DM6446 for resizing video without any computational cost in DSP. Then, the resized video is analysed by GMM and blob extraction to obtain the positions and bounding boxes of the moving objects. After that, the gainedpositions arere-mapped to D1 coordinate for the on-going procedures such as tracking, classification and recognition. Furthermore, for the purpose of better pipelining the algorithm to achieve higher performance, we divide the GMM function into three stages include backgroundmodel initialization, updatingand comparison, and performed separately. According to the profiling, the performance is greatly enhanced because the software pipeline is generated successfully.

3.3Code-level optimization

Generally, the generation of software pipeline isakey stepforcode-leveloptimization.However, there are several common situations hinder producing software pipelines, i.e., loops nesting, in-loop function calling, jump instructionsetc. Therefore, we examined and divided large loops into small loops to increases instruction-level parallelism guaranteeing the effective creation of software pipeline by using the instruction "*MUST_ITERATE*" in oursystem. After that, as the pixel value is *8-bit* length, to further improve the quality of the software pipeline, we utilized data packing techniques to pack and parallel process multiple pixels in one *32-bit* pack by executing the packing and unpacking instructions such as "*_memd8_const", "_packl4", "_hi", "_lo", "_subabs4", "_cmpgtu4", "_itoll"*, etc.

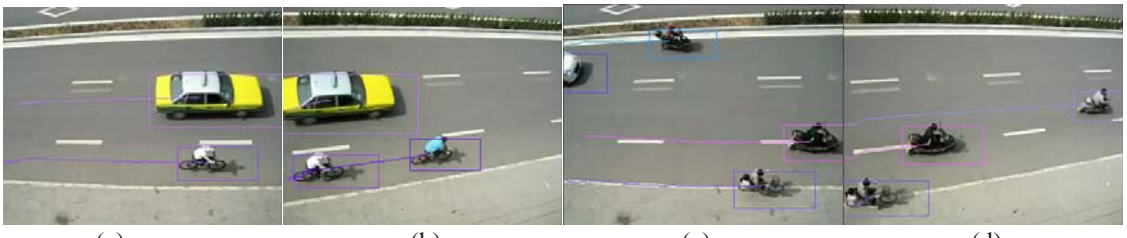
By utilizingour proposed hierarchicaloptimization method, the system performance is increased significantly as described in the Table 1.Since the performance of DSP core is at a clock rate of *810* MHz (*810* million clock cycles per second), the overall computational budget is reduced by *98.84%* and the system performance is boosted from *2.03* frames/second to *30* frames/second reaching the maximum frame rate.

Table 1.System performance comparison by using our proposed hierarchicaloptimization method

Module Name	**Before Optimization**		**After Optimization**		**Optimization Ratio**
	clock cycles	frames/s	clock cycles	frames/s	clock cycles
GMM Background Reconstruction	379672557	2.13 f/s	338754830	f/s	99.11%
Moving Object (Blob) Extraction	19218288	30 f/s	124931230	f/s	93.49%
Overall Performance	398890845	2.03 f/s	463686030	f/s	98.84%

4.Experiments and results

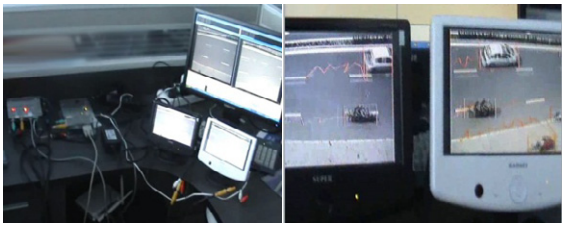
To demonstrate the robustness and effectiveness of our proposed system,we have built a test-bed environment around our campus by deploying multipledistributed cameras and performedseveral real-world experiments in various environments. As shown in Fig. 5 (a) (b) (c) (d), the images demonstrate two result set of ourmulti-camera trackingsystem in anoutdoor scene. Fig. 5 (a) (c) are the snapshots from the left camera view, whileFig. 5 (b) (d) are from the right camera view. Specifically, real-time multiple objects tracking is perform continuously on each individual camera to analyze the trajectories and bounding boxes of the moving targets, as displayed in Fig.5, each surveillance target is tracked successfully and marked with bounding box and trajectory in uniquecolor to distinguish from others. After that, once moving targets enter the overlapping area between two adjacent cameras, object hand-off procedure is carried out to compute the accordance relationships of the targets in the overlapping area for multi-camera long-term tracking. As depicted in Fig.5, targets in the overlapping area are successfully tracked and hand-off and marked in their unique tracking color.



(a) (b) (c) (d)

Fig. 5 Experiment results for dual camera-tracking in an outdoor environment around our campus

In Fig. 6, experiments results in general view are shown. As can be seen in Fig. 6 (a), raw video is captured by our 3-level embedded video analysis agent from the video input device, and then our multi-camera tracking algorithm is carried out to process the video data in real-time performance, and finally, tracking results are displayed directly intwo standalone screens as shown in the Fig. 6 (b).



(a) (b)

Fig. 6 Snapshot of our experiment devices and results in general view

Full demo video for our proposed embedded multi-camera tracking system is provided [9] on YouTube.

5.Conclusions

In this paper, we represented the multi-camera tracking algorithm and implemented on the embedded platformTI dual-core TMS320DM6446 (ARM+DSP). In our proposed system, a traditional tracking paradigm for singlecamera tracking based on GMM, Mean-shift and Kalman filter is utilized to detect and track multiple targets. Then, when the surveillance objects enter the overlapping area between adjacent cameras, a homography based target hand-off procedure is perform for long-term multi-camera tracking. However, the computation complexity for multi-camera system is huge especially for embedded processor based system. Therefore, to conquer the challenging problems in low-cost, reliable and efficient way, we designed embedded integrated smart camera by utilizing the hardware structure of TI TMS320DM6446 based on which our proposed multi-camera tracking algorithm is implemented, and optimized reaching real-time performance by our proposedhierarchical optimization method. The overall experimental results have fully verified the effectiveness and robustness of our proposed algorithm and the stability of the embedded platform.

References

[1] R. Kleihorst, B. Schueler, A. Danilin, Architecture and applications of wireless smart cameras(networks), in: Proceedings of the IEEE International Conference on Acoustics, Speech, andSignal Processing, 2007.
[2] B. Rinner, W. Wolf, Introduction to distributed smart cameras, in: Proceedings of the IEEE96 (10).
[3] W. Hu, T. Tan, L. Wang, S. Maybank, A Survey on Visual Surveillance of Object Motion andBehaviors, in: IEEE Transactions Systems, Man and Cybernetics 34 (2004) 334–352.
[4] R.T. Collins, A.J. Lipton, H. Fujiyoshi, T. Kanade, Algorithms for cooperative multi-sensorsurveillance, in: Proceedings of the IEEE, 89 (2001) 1456–1477.
[5] H. Aghajan and A. Cavallaro, Multi-camera Networks: principles andapplications. USA: Elsevier, 2009.
[6] Texas Instruments, Davinci-DM644x Evaluation Module technical reference", March, 2006.
[7] Morel, J., Yu, G.: Is SIFT scale invariant? Inverse Problems and Imaging 5(1), 115–136 (2011).
[8] M.I.A. Lourakis, A brief description of the Levenberg-Marquardt algorithm implemented bylevmar.
[9] http://youtu.be/pyKzigK30bM