行政院國家科學委員會專題研究計畫 成果報告

應用於視訊監控之運動物體追蹤技術研究(第 2 年)
研究成果報告(完整版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 95-2221-E-009-106-MY2
執 行 期 間 ：96 年 08 月 01 日至 97 年 07 月 31 日
執 行 單 位 ：國立交通大學電子工程學系及電子研究所

計 畫 主 持 人：王聖智

計畫參與人員：碩士班研究生-兼任助理人員：范博凱
碩士班研究生-兼任助理人員：蕭晴駿
博士班研究生-兼任助理人員：黃敬群

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 97 年 10 月 28 日

行政院國家科學委員會補助專題研究計畫 ☒ 成 果 報 告 □期中進度報告

應用於視訊監控之運動物體追蹤技術研究

計畫類別：☒ 個別型計畫 □ 整合型計畫

計畫編號：NSC 95-2221-E-009-106-MY2

執行期間： 96 年 8 月 1 日至 97 年 7 月 31 日

計畫主持人：王聖智
共同主持人：
計畫參與人員： 黃敬群、范博凱、蕭晴駿

成果報告類型(依經費核定清單規定繳交)：□精簡報告 ☒完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：交通大學電子工程系

中 華 民 國 97 年 10 月 28 日

應用於視訊監控之運動物體追蹤技術研究

計畫編號：NSC 95-2221-E-009-106-MY2

執行期限：96年8月1日至97年7月31日

主持人：王聖智 (交通大學電子工程系教授)

計畫參與人員：黃敬群　范博凱　蕭晴駿 (交通大學電子所研究生)

一. 中文摘要

本計劃中我們提出兩套應用於視訊監控的技術: (a)多攝影機協同監控與(b)人物的姿勢估測及分析。在多攝影機協同監控的技術上，我們提出一套應用於多台主動式攝影機之分工協調系統，對於空間中大約已知臉部之位置與朝向的人群，進行攝影機的分工與協調。每一台攝影機將會負責拍攝一小部分人群的臉部，並且設法調整攝影機的旋轉角度以及放大倍率，使人臉可以清晰地在畫面中呈現。在此，我們對於人臉在畫面中清晰與否的評斷標準為：人臉是否正面朝向負責拍攝的攝影機，以及人臉在影像中的解析度。透過本系統，我們可以安排各個主動式攝影機的旋轉角度與放大倍率，盡可能地拍攝場景中所有人的臉部，以獲得理想的人臉拍攝角度與解析度，便於清楚地辨識每個人。另外，在人物的姿勢估測及分析的技術上，我們提出一個在多攝影機環境下，利用人體模型估測目標人物的姿勢與行為。我們使用流形嵌入技術中的拉普拉斯特徵映射，將三維人形的幾何形狀忠實地轉移到另一個容易切割分析的高維度空間，正確地切割出三維人形的各個部位並且找出三維人形的骨骼架構，以利後續行為分析的動作。當擷取出三維人形的骨骼架構後，我們利用粒子群體最佳化在高維度空間中有效地找出最佳姿態估測結果。我們的系統由影像的擷取至姿態的估測完全自動化，並且不需要在人體上貼附感應物，即可結合肢體的運動限制和時間軸上的動作流暢限制，估測多種動作。

關鍵詞：攝影機校正、動態攝影機控制、攝影機協調機制、姿態估測、流形嵌入技術。

ABSTRACT

In this project, we proposed two algorithms for the application of video surveillance: (a) a camera coordination system for surveillance and (b) human body pose estimation method. For the camera coordination system, we coordinate multiple PTZ cameras to capture the face pictures of monitored targets. Given the positions and orientations of people's faces in the 3-D space, this system dynamically controls the panning, tilting, and zooming of all PTZ cameras, trying to acquire better shots of targets' faces. The adopted criteria include people's facing directions with respect to the cameras and the resolutions of the facial images. Unlike other approaches, we do not limit our PTZ cameras to the capture of only one target at one time. Instead, the proposed system coordinates all PTZ cameras to capture as many high resolution frontal faces as possible. With this system, the faces in the scene can be better captured and the identity of each monitored target can be well discerned. For human body pose estimation method, we propose a 3D human body pose estimation method for a multi-camera motion capture system. The reconstructed human body is transformed into a high dimensional space using our modified Laplacian Eigenmap. In this eigenspace, the body parts can be segmented more efficiently and easily. Then, the 3D skeletons of the human body are extracted to obtain the kinematic information. Finally, pose estimation is performed by fitting a prior 3D model to the extracted skeleton via particle swarm optimization (PSO). Furthermore, with our proposed human model, the motion constraints can be easily combined with the optimization process. Temporal consistency of the pose estimation results is also achieved by adding temporal constraints over PSO. Our method can deal with various kinds of motion and has robust pose estimation results.

Keywords: Camera calibration, dynamical PTZ

control, Camera coordinate, pose estimation function, modified Laplacian Eigenmap.

1. INTRODUCTION

In this project, we proposed two algorithms for the application of video surveillance: (a) a camera coordination system for surveillance and (b) human body pose estimation method.

For the camera coordination system, we develop a surveillance system that tries to simultaneously observe as many high-resolution faces as possible. In Fig.1, we illustrate the task of the proposed system. In this example, there are 9 people in total. The triangles denotes PTZ cameras, the circles indicate people's locations, and the arrows represent the orientation of people's face. The proposed system will automatically assign these four PTZ cameras to take care of different groups of people so that the multi-camera system can capture as many high-resolution facial images as possible at every moment.

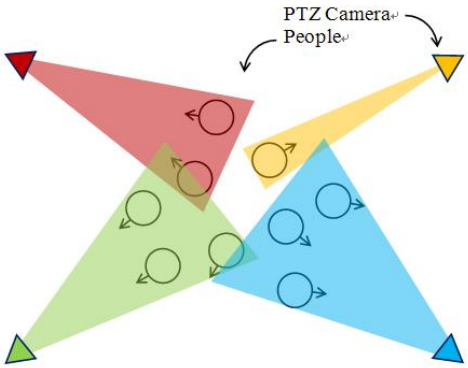


Fig. 1 Illustration of camera coordination

For human body pose estimation method, we propose a markerless motion capture system equipped with multiple cameras. First, a 3D human body represented by voxels is reconstructed from multiple video streams. A modified Laplacian Eigenmap algorithm is used *to transform the 3D voxel data into a high dimensional space. With this manifold embedding method, different body parts are mapped into discriminative branches and can be easily segmented. Unlike other approaches, this approach relieves the dependence on human model and the training database. After the segmentation of body parts, skeletons are extracted to describe the kinematic motion of the human body. Human shapes are usually deformed while skeletons can encode most of the motion information. As the skeletons are extracted from the 3D human bodies, we use the particle swarm optimization (PSO) technique to deal with the pose estimation problem. The experimental results show that our system can handle various kinds of poses and can ensure the temporal consistency and motion constraints.*

2. BACKGROUNDS

2.1 Multi-camera coordination system

Although several multi-camera surveillance systems have already been proposed, we have not found any multi-camera system that offers similar functionalities as ours. Hence, we only mention a few articles that have discussed some issues similar to ours.

In [1] and [2], Micheloni proposed a system that contains a few static cameras and PTZ cameras. The resolution of PTZ camera is higher than that of static camera. When a target appears, they estimate the 3-D location of the target and automatically control the pan angle and tilt angle of the PTZ cameras to capture the target's high-resolution images. In their approach, each PTZ camera focuses on the tracking of a single target.

In [3], the proposed system also contains multiple static cameras and PTZ cameras. The static cameras are used to estimate the 3-D positions of the detected targets. Face detection is also used to determine whether a human face exists. Once if a face exists, then they control a PTZ camera to capture a close-up of that face. In [4], the authors use pairs of static cameras to estimate the depth information. The face position of the target is estimated by combining the depth information with the face detection results. Similarly, once if a face is detected, a PTZ camera is controlled to capture a clearer facial picture of the target.

In [5] and [6], the authors proposed a camera network composed of multiple static and PTZ cameras. Similarly, they use PTZ cameras

to capture people's high-resolution images, with each PTZ camera monitoring a single person at one time. A scheduling algorithm is proposed to control the movement of all PTZ cameras so that each pedestrian will be captured at least one time before the pedestrian leaves the scene. The performance of their system is evaluated over a virtual train station scene which is synthesized by computer animation.

2.2 Human body pose estimation method

As for the human body pose estimation method, we mainly discuss markerless motion capture systems, which have drawn much attention in recent years. A markerless approach can be decomposed into several submodules: initialization module, tracking module, pose estimation module, and recognition module.

In the proposed motion capture system, we mainly focus on the initialization module and the pose estimation module. The module of initialization aims to obtain reliable prior knowledge for pose estimation and recognition. Due to error propagation, incorrect prior knowledge may lead to incorrect pose estimation. In the following paragraphs, we'll first introduce a few algorithms that are related to initialization and pose estimation. In this project, we focus on model-based pose estimation for multi-camera systems.

In [7], the reconstruction of a "visual hull" based on images from multiple cameras is introduced. In this approach, a visual hull is defined as the 3D shape formed by the intersection of visual cones projected from the 2-D silhouettes. The visual hull of an object can be thought to be a close approximation of the object based on the observations from different viewpoints.

Regard the 3-D shape human model, Mikic [8] adopted a twist framework that has been used to model the kinematic chains for robots. Sixteen rotation axes and five kinematic chains of the body joints are formulated using twists and product of exponentials. Relative to the torso-centered coordinate system, the rotation and shift of the other body parts can be easily manipulated. Pose estimation is performed by first doing template fitting and then using Bayesian network for refinement. However, the initialization based on template fitting cannot deal with self occlusion and the target person has to dress in tight clothes.

Instead of using shape models, Menier [9] adapted skeleton models to fit medial axis points extracted from visual hulls. This approach reduces the dependency on the dimension of human body, and these 3D medial axis points represent the observed skeleton data. A generic skeleton model is then fitted with the observed skeleton data based on maximum a posteriori (MAP) estimation. The pose estimation of the first frame is based on the fitting process, while non-parametric belief propagation is used to predict the pose of the following frames.

Due to the high dimensionality of the search space and the complexity of the fitness evaluation function, some researchers have adopted the particle swarm optimization (PSO) [10] method to perform pose estimation. Robertson [11] applied PSO to perform skeleton model fitting in a conference room environment, where the pose estimation is required only for the upper body. PSO is chosen for its ability to deal with nonlinear and non-convex optimization problems. Hierarchical and parallel PSO fitting is proved to be robust and computationally inexpensive.

As mentioned earlier, model based motion capture systems have the advantages of complexities reduction and robustness. However, a good initialization is required to ensure that the system commences with a good body parts labeling and initial guess. More reliable initialization approaches based on manifold learning have been proposed recently.

In manifold learning, manifold embedding is a topic about how to find a transformed space for the manifold that preserves the connectivity and algebraic properties. Several approaches, such as Laplacian Eigenmap [12], have been proposed in this field. In [13] Sundaresan proposed a segmentation approach for pose estimation based on Laplacian Eigenmap. In this approach, different branches in the normal space, such as separated body parts, are transformed into distinguishable 1-D smooth curves in the

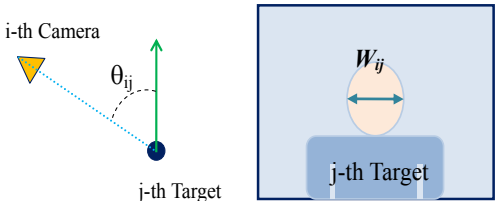
embedding space. This property makes the segmentation of 3-D human body a lot easier.

3. TECHNICAL ILLUSTRATION – CAMERA COORDINATION SYSTEM

3.1. Problem Formulation

Unlike the articles mentioned in Section 2.1, we aim to capture as many frontal high-resolution facial images as possible during the presence of the monitored targets. In the proposed algorithm, PTZ cameras are allowed to cover more than one target at each time, as long as the captured facial images are sufficiently clear. In the proposed algorithm, we design our camera coordination system based on two major criteria: frontal shoot and high-resolution shoot.

To formulate these two criteria, we define the shoot angle θ_{ij}, and the face width W_{ij}. In θ_{ij} and W_{ij}, the subscript i denotes the i-th PTZ camera, while the subscript j denotes the j-th person. As shown in Fig. 2(a), the shoot angle θ_{ij} represents the angle between the blue dotted line $\overrightarrow{cam_{ij}}$ and the green arrow $\overrightarrow{face_j}$. $\overrightarrow{cam_{ij}}$ indicates the line connecting the i-th PTZ camera and the i-th target, while $\overrightarrow{face_j}$ indicates the facing orientation of the j-th target. As the j-th target is looking toward the i-th camera, we have a smaller shoot angle. On the other hand, as shown in Fig 2(b), the shot face width W_{ij} represents the width of the j-th target's face in the image captured by the i-th camera. A larger value of W_{ij} indicates a better observation of the j-th target in the the i-th camera image.



(a) (b)

Fig. 2 Definitions of (a) θ_{ij} and (b) W_{ij}

To simplify the computation of θ_{ij} and W_{ij}, all 3-D vectors are projected onto the ground plan to form 2-D vectors. In the simplified forms, the shoot angle and the face width are defined as follows.

$$\theta_{ij} = acos\left(\frac{\overrightarrow{cam_{ij}} \cdot \overrightarrow{face_j}}{\left\|\overrightarrow{cam_{ij}}\right\| \left\|\overrightarrow{face_j}\right\|}\right) \quad (1)$$

$$W_{ij} = f_{xi} \frac{Face\ width\ in\ 3D\ space}{D_{ij}} \quad (2)$$

where $f_{xi} = \frac{Image\ width}{2 \cdot tan\left(\frac{FOV_i}{2}\right)}$. In the definition of W_{ij}, f_{xi} denotes the focal length of the i-th PTZ camera in the horizontal direction, D_{ij} is the distance between the i-th camera and the j-th target, and FOV_i is the field of view of the i-th PTZ camera.

Basically, we prefer to capture a facial image with a smaller shoot angle but a larger face width. We further apply two mapping functions $N_\theta(\ )$ and $N_w(\ )$ over θ_{ij} and W_{ij} to convert them into two normalized measures. These two mapping functions are defined as follows.

$$N_\theta(x) = \begin{cases} -\frac{r_\theta \cdot k}{th_\theta} x + \frac{(1+r_\theta)}{2} k, & 0 \le x < th_\theta \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$N_W(x) = \begin{cases} 0, & x < th_{min} \\ \frac{r_W \cdot k}{th_{max} - th_{min}} x - \frac{r_W \cdot k \cdot th_{min}}{th_{max} - th_{min}} + \frac{(1-r_W)k}{2}, & th_{min} \le x < th_{max} \\ \frac{(1+r_W)k}{2}, & x \ge th_{max} \end{cases} \quad (4)$$

In (3) and (4), k is a positive constant that controls the dynamic range of $N_\theta(\ )$ and $N_w(\ )$. r_θ and r_W are real numbers within the range [0, 1] and they control the slopes of $N_\theta(\ )$ and $N_w(\ )$. th_θ, th_{min}, and th_{max} are pre-defined thresholds. th_θ represents the worst situation that can be allowed for capturing the frontal face. th_{min} represents the minimum face width for clear observation. On the other hand, when the face width is wider than th_{max}, we think the facial image has achieved the level of perfect observation. These thresholds can be varied by the users for different applications.

With the definitions of N_θ and N_W, we then define the evaluation function Eval( ) for the face capture of the j-th target by the i-th camera.

$$Eval(AP)=\sum_{i=1}^{m}\sum_{j=1}^{n} ap_{ij}N_\theta\left(\theta_{ij}\right)N_W\left(W_{ij}\right). \quad (5)$$

In (5), m is the number of cameras and n is the number of targets. AP denotes a set of camera assignments

$$AP=\left\{ap_{ij}\right\},\ i=1,2,\ldots,m,\ j=1,2,\ldots,n, \quad (6)$$

with ap_{ij}'s representing the binary assignment parameters. ap_{ij} is equal to 1 if the i-th camera is assigned to monitor the j-th target, and ap_{ij} is equal to 0 otherwise. Hence, for a camera assignment AP, Eval(AP) represents the overall observation levels of the n targets by all m cameras. When more targets can be better observed by their corresponding cameras, with smaller shoot angles and larger face widths, we have a larger Eval(AP). Hence, the goal of the proposed camera coordination system is simply to find the optimal camera assignment that reaches the largest Eval(AP). Moreover, as these n targets keep moving within the monitored scene, we need to adaptively adjust the assignment of cameras to achieve the most preferable observation.

Besides, to simplify the problem, we also add one extra constraint over (5). The constraint is

$$\sum_{i=1}^{m} ap_{ik}=1,\ k=1,2,\ldots,n. \quad (7)$$

This constraint implies that we only take into account the camera view that is assigned to the target even though that target may also appear in some other views.

3.2. Significance Weight

In theory, we can always find an optimal AP for the evaluation function at any time instant. However, people's behavior is highly versatile. It is very likely that even with the optimal camera assignment we still cannot clearly capture all people's faces at some time instants. The proposed system cannot guarantee that all people's faces are always clearly observed.

To deal with this problem, we assign each target a significance weight to represent the priority of that target. This weight will increase if the target hasn't been clearly observed in the past few moments. On the contrary, if that target has already been clearly observed for a while, we decrease its significance weight.

The adjustment of significance weight includes three different states: raise, hold, and decline. When the face of a target cannot be unclearly captured, we linearly increase its significance weight. When the weight is raised, the system will pay more attention to that target and it's more likely that the target can be better observed. Once if the system has adjusted its camera coordination to take clear facial picture of that target, the significance weight will be held at a high value for a while. This "hold" state is make sure the target's face can be clearly observed for a long enough period, but not just a short glimpse. After the hold state, the significance weight of the target is then decreased gradually to zero as long as the target's face can be clearly captured. An example of the switching of these three states is illustrated in Fig. 3.

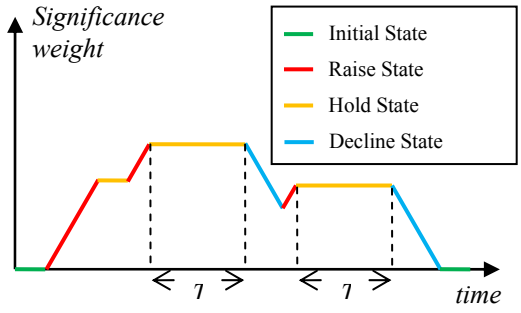


Fig. 3 Variation of significance weight

To realize the concept of importance weight, we add penalty term into the definition of Eval( ), as expressed below.

$$Eval(AP)=\sum_{i=1}^{m}\sum_{j=1}^{n} ap_{ij}\left(N_\theta\left(\theta_{ij}\right)N_W\left(W_{ij}\right)-pv_{ij}\right), \quad (8)$$

where the penalty term pv_{ij} is defined as

$$pv_{ij}=cf_{ij}\cdot sw_j\cdot k. \quad (9)$$

In (9), sw_j stands for the significance weight of the j-th target, cf_{ij} represents the clear factor of the j-th target with respect to the i-th camera, and k is a controlling scalar. The clear factor cf_{ij}

is equal to 0 if the j-th target can be clearly observed by the i-th camera. Otherwise, cf_{ij} is equal to 1. With the inclusion of the penalty term, the camera coordination system can pay more attention to these targets with larger significance weights automatically.

3.3. Modified Discrete Binary PSO

Unfortunately, Equation (8) has a nonlinear and non-differentiable form. To find the optimal AP, classical optimization algorithms, like the gradient decent algorithm, cannot be used. Instead, we adopt the particle swarm optimization algorithm [14]. In the PSO method, a set of particles is generated and each particle represents a trial solution of the problem. All particles have their own memory, and these particles communicate with each other to get the best global position. Due to the binary natural of the assignment parameters, we actually adopt the discrete binary particle swarm optimization algorithm proposed in [15]. However, since we have added one constraint in the evaluation function, we further make some modifications over the discrete binary particle swarm optimization algorithm to tackle the problem.

In the modified DBPSO, each particle represents a possible AP. In the original form of PSO, particles are randomly generated in the initial stage. However, this causes a large number of iterations. To speed up the computations, we generate a reasonable initial guess about AP. This is achieved by performing clustering over targets' positions and orientations. In the clustering process, targets with adjacent positions and similar face orientations are assigned to the same camera. Here, we define the feature vector of each target to be

$$(\lambda X', \lambda Y', IP_1, ..., IP_m), \tag{10}$$

system based on two major criteria: frontal shoot and high-resolution shoot.

$$\overrightarrow{v_i^{t+1}} = \omega \cdot \overrightarrow{v_i^t} + c_1\varphi_1 \cdot \left(\overrightarrow{P_i} - \overrightarrow{x_i^t}\right) + c_2\varphi_2 \cdot \left(\overrightarrow{P_g} - \overrightarrow{x_i^t}\right). \tag{11}$$

In (11), $\overrightarrow{v_i^t}$ is the velocity at the previous moment, ω is the inertia factor, c_1 and c_2 are scalars, and φ_1 and φ_2 are random numbers generated from the uniform distribution over [0,1]. Equation (11) is repeatedly computed until the stop criterion is reached.

In the DBPSO algorithm, the definition of velocity is different from that of the original PSO. Assume x denotes an n-bit string and xk represents its k-th bit, which can only be 0 or 1. Assume the position of the i-th particle is denoted as x_i and its d-th bit as xid. Each bit has its own velocity vid. In the original PSO algorithm, the velocity of a particle indicates the movement of that particle. However, in the DBPSO algorithm, the definition of velocity represents the tendency of being 1. The larger the velocity is, the more likely that bit will become 1. Besides the modification over velocities, the previous best position and the best global position are to be considered in the bitwise manner. Assume we denote pid as the previous best d-th bit of the i-th particle, and denote pgd as the best global d-th bit. Then the equation is modified to be

$$v_{id}^{t+1} = \omega \cdot v_{id}^t + \varphi_1 \cdot \left(p_{id} - x_{id}^t\right) + \varphi_2 \cdot \left(p_{gd} - x_{id}^t\right). \tag{12}$$

and the d-th bit of the i-th particle is updated based on the following rule:

$$\begin{aligned} &if\left(rand(0,1) < S\left(v_{id}^{t+1}\right)\right)\ then\ x_{id}^{t+1} = 1 \\ &else\ x_{id}^{t+1} = 0 \end{aligned} \tag{13}$$

In (13), rand(0,1) is a random number selected from a uniform distribution over [0, 1], and S is the sigmoid function defined by

$$S(v) = \frac{1}{1+e^{-v}}. \tag{14}$$

In our algorithm, we basically follow Equations (12)~(14), with each particle representing a possible camera assignment. Equations (12) and (13) are now rewritten as

$$v_{k,ij}^{t+1} = \omega \cdot v_{ij}^t + \varphi_1 \cdot \left(p_{k,ij} - ap_{k,ij}^t\right) + \varphi_2 \cdot \left(p_{g,ij} - ap_{k,ij}^t\right) \tag{15}$$

and

$$\begin{aligned} &if\left(rand(0,1) < S\left(v_{k,ij}^{t+1}\right)\right)\ then\ ap_{k,ij}^{t+1} = 1 \\ &else\ ap_{k,ij}^{t+1} = 0 \end{aligned} \tag{16}$$

The inertia factor is chosen to be some value within [0.8, 1].

However, due to the inclusion of the constraint (7) in the evaluation function (8), the DBPSO algorithm needs further modification. In DBPSO, the *d-th* bit of the *i-th* particle is updated based on (13). However, after the update, the new particle may violate the constraint (7). To fix this problem, we slightly modify the DBPSO algorithm based on the following concept.

Assume for the k-th target, its corresponding assignment bits are denoted as $\{ap_{1k}, ap_{2k}, ..., ap_{mk}\}$. If more than one ap_{ik} is set to 1. Then we retain the assignment bit that has the highest possibility to be 1, while set the other assignment bits to zero. In our approach, if the previous best value or the best global value of an assignment bit is 1, that assignment bit has the highest probability to be 1. If more than one assignment bits have the highest probability to be 1, then we randomly pick up one of them to be 1 and set the others to 0.

On the other hand, if none of the assignment bits are set to 1, we still apply the same strategy to correct the assignment. That is, if the previous best value or the best global value of an assignment bit is 1, that assignment bit will have the highest probability to be 1.

4. TECHNICAL ILLUSTRATION –HUMAN BODY POSE ESTIMATION

In the other way, we proposed an efficient initialization process and a robust markerless pose estimation system. The goal of pose estimation is to capture the motion of a specific person. The motion of the articulated body parts is described using some parameters of a generic human model. Using images from a set of synchronized and calibrated cameras, we can reconstruct the visual hull based on volume intersection. Then the 3D voxel data are transformed into an embedding space using our modified Laplacian Eigenmap technique. Body parts segmentation is done in the eigenspace and the skeletons of the body parts are extracted individually. Finally, we fit the human model into the skeleton data using the PSO algorithm. Pose estimation is then iteratively performed for optimization. In Fig. 1, we show the flowchart of the proposed system.

4.1. System initialization

The performance of a model-based system heavily relies on the accuracy of the initialization results. The embedding-based initialization exploits the manifold embedding methods and has the advantage of lower model dependency. Inspired by Sundaresan's method [13], we develop a modified Laplacian Eigenmap to efficiently extract the kinematic information from the visual hull. In the following sections, we'll explain more details about Sundaresan's method and our initialization method.

4.1.1 Proposed initialization method

Inspired by Sundaresan's algorithm, we develop our initialization method based on a modification of the Laplacian Eigenmap.

Given n points $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ in the p dimension, Laplacian Eigenmap aims to find its transformation $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ in the r dimension to minimize the object function:

$$\sum_{i,j} \left\| \boldsymbol{u}_i - \boldsymbol{u}_j \right\|^2 \mathrm{E}_{ij} \ , \tag{17}$$

where E is the adjacency matrix of the graph constructed from $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$. That is, if $\boldsymbol{v}_j$ is in the neighborhood of $\boldsymbol{v}_i$, then E_{ij} is equal to 1. Otherwise, E_{ij} is set to zero.

Besides (17), an extra constraint is added for the minimization of the object function. The constraint says

$$\begin{aligned} &\mathrm{U}^T \mathrm{D}\mathrm{U} = I \\ &\text{where } \mathrm{U} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_n \end{bmatrix}^T \end{aligned} \tag{18}$$

D is a diagonal matrix, whose element D_{ii} represents the degree of Node i. This constraint is to normalize the scaling factor when manifold embedding is performed. We can unroll Equation (18) to obtain the following constraints:

$$\begin{aligned} u_{11}\mathrm{D}_{11} + u_{21}\mathrm{D}_{22} + \ldots + u_{n1}\mathrm{D}_{nn} &= 1 \\ u_{12}\mathrm{D}_{11} + u_{22}\mathrm{D}_{22} + \ldots + u_{n2}\mathrm{D}_{nn} &= 1 \\ &\vdots \\ u_{1r}\mathrm{D}_{11} + u_{2r}\mathrm{D}_{22} + \ldots + u_{nr}\mathrm{D}_{nn} &= 1 \end{aligned} \quad (19)$$

In (19), we observe that nodes with more neighbors tend to converge to positions around the origin after the transformation.

As mentioned above, the segmentation of trunk is a major difficulty in Sundaresan's method. Since the nodes in the trunk tend to have bigger values of D_{ii}, this fact makes the transformed values of the trunk voxels spread around the origin of the eigenspace. Having exploited this property of Laplacian Eigenmap, we manage to assign trunk voxels with bigger values of D_{ii} so that their transformed data will shrink even closer to the origin. Once these nodes are shrunk to the origin of the 6-D eigenspace, the segmentation of the limb parts will become much easier.

Furthermore, to prevent mistakenly shrinking the other thick parts of the human body, like the head, color is used as the auxiliary information. For each 3-D voxel and its neighbors, we project them back into the image plane of each camera and record the colors at the projected positions. If the colors of two voxels, say v_i and v_j, are similar for most cameras, we increase the value of E_{ij} to enhance the connectivity. By this way, we can weaken the connection of the nodes between head and torso. When nodes belonging to the torso are shrunk to the origin, we can still preserve the distinguishable branches of the other body parts.

In summary, our body part segmentation method is briefly described as below:

A. Modified Laplacian Eigenmap

A.1 Graph Construction

Given n voxels $v_1, \ldots, v_n$ in the visual hull, we construct an adjacency graph G for these voxels. Each element E_{ij} of E records the relationship between Node i and Node j. For position information, if two nodes are 6-connection neighbors, E_{ij} gets one point. For color information, we project each voxel into the image planes and calculate the similarity between their colors. In our simulation, there are eight cameras in total in the scene. If for some camera the value of d, as defined below, is less than some threshold, one vote is recorded.

$$d \equiv \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2} \quad (20)$$

Therefore, at most 8 votes can be recorded for each 6-connection neighbor of a node. The extra bonus on E_{ij} is added based on the following rule:

$$\mathrm{E}_{ij} = \mathrm{E}_{ij} + \begin{cases} a & total\_votes = 8 \\ b & th_1 \le total\_votes < th_2 \\ c & total\_votes < th_3 \end{cases} \quad (21)$$

$$0 < b < a \le 1 \quad -1 < c < 0$$

Since the original value of E_{ij} is at most 1 for the position information, the bonus for the E_{ij} due to the color information is restricted to be no more than 1. Please note that the color information is only auxiliary. This is because different colors don't necessarily mean different body parts. Here, we simply use color information to prevent a mistaken shrinkage of the head part.

A.2. Shrinking of Nodes

After having constructed the adjacency graph, we impose more weights on those voxels that have more connections to their neighbors. For a Node i, its degree is defined as $\sum_{j, j \neq i} \mathrm{E}_{ij}$. Once we increase the weights of these nodes that have larger degrees, the transformation of these nodes will shrink toward the origin in the eigenspace.

B. Body Part Segmentation

Since the trunk voxels will be roughly mapped to the origin of the eigenspace based on the modified LE, a simple but efficient termination method can thus be developed for body part segmentation. In the following, we briefly explain the process of body part segmentation.

B.1. Spline Initialization

The process of spline initialization is the

same as Sundaresan's method.

B.2. Spline Propagation

Starting from the end of the (P+1) nodes, the nearest N points are selected. Unlike Sundaresan's method, we don't have to count the number of outliers. The site values are also fitted using a 6-D spline.

B-3. Spline Termination

The process of spline propagation continues until the distance between the end of the spline and the origin is less than a pre-defined threshold.

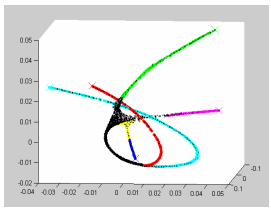
In Fig. 4, we show the comparison between the Sundaresan's method and ours. It can be seen that our method map the trunk part to the origin of the eigenspace. Hence, after spline fitting, the trunk is well detected and the limb parts of the human body, especially the left arm, can be successfully extracted.

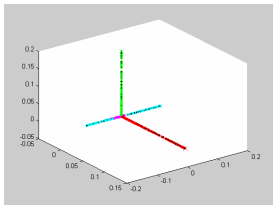
Once the segmentation of human body parts is done, we can extract the skeleton of the visual hull. Skeleton extraction has the advantage of feature reduction. Furthermore, skeletons encode the information of kinematic motion and won't deform in any pose.

In our approach, each body part is individually transformed into a 1-D eigenspace based on the LE algorithm. The smallest nonzero eigenvalue represents the most important dimension that corresponds to the trend of the body part. Spline fitting is performed and the site values that encode the geometric relation in the normal space are calculated along this dimension. The skeleton extraction is then performed by finding a 3-D spline $\boldsymbol{h}$ which minimizes (22)

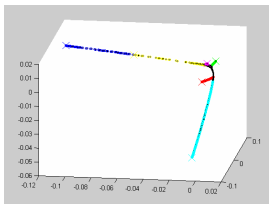
$$\sum_{v_i \in some\ body\ part} \left\| \boldsymbol{v}_i - \boldsymbol{h}(s_i) \right\|^2 \qquad (22)$$

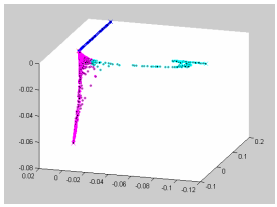
(a) (b)



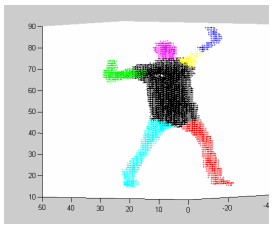


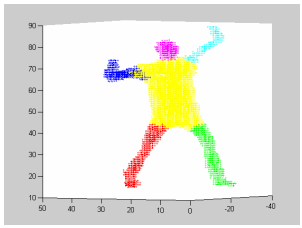
(c) (d)





(e) (f)

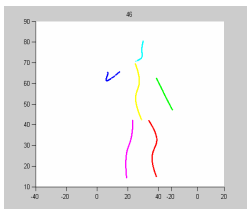




(g) (h)

Fig. 4: Comparisons between Sundaresan's method and ours (a) input image (b) the color representation for the segmentation results (c)(e) the segmentation result in the 6-D eigenspace using original LE (g) the segmentation result in the normal space using original LE (d)(f) the segmentation result based on our modified method (h) the segmentation result in the normal space using our modified method.

In Fig. 5, we show an example of the skeleton extraction process.



(a) (b)

Fig. 5: The skeleton extraction result using the proposed method (a) one of the eight input images (b) the extracted skeleton

4.2 Pose estimation

After skeleton extraction, the posture of the specific person is estimated using a prior human model. The joints of the human model have their individual degrees of freedom (DOF). In total, there are usually 20 or more parameters. Thence, the fitting of the human model to the skeleton data is an optimization problem in a very high dimensional search space. In this case, it is very challenging to simultaneously find the optimal solution for the current skeleton data and to ensure the temporal smoothness over time. In the following section, we will discuss the adopted 3-D human model and the proposed pose estimation technique.

4.2.1 3-D human model

Our human model is a 3-D skeleton model. Here, we adopt the popular twists and exponential products formulation. This mathematical framework helps us in describing the kinematic chains in the human body. The concept for kinematic chains is introduced by Murray [16] and is generalized to the application of 3D human models by Mikic [8].

In our model, a human body is a 3-D skeleton composed of 12 segments and 23 parameters. It is based on the twists formulation. The position of each point can be described using exponential products.

In Mikic's design, the rotation axis of the torso is an arbitrary unit vector ω_0 in the world coordinate [8]. However, it is not easy to control the orientation of the human model. The orientation of the human model determines which part is the right-hand side. This information is important since the motion constraints for the right side and the motion constraints for the left side are somewhat different. With the motion constraints, we can make our pose estimation more natural and reasonable. Unfortunately, Mikic's method does have some problem in defining the orientation of the model.

To determine the right side from the left side, Mikic switches the right side and left side of the human model and compares their fitting errors. For the case in Fig. 5, it happens that the smaller fitting error actually corresponds to the wrong decision. A more natural thinking is that if we can make the model self-spin, the orientation of the human body can be easily manipulated and decided. Hence, we redefine the rotation axis of the human body as the torso stick of the model. Furthermore, the neck position and the torso center control the incline of the human model. With self spin, the human model can easily spin to the correct orientation to obtain less fitting error.

4.2.2 PSO based pose estimation

PSO (Particle Swarm Optimization) has the advantages of being capable of dealing with non-concave and nonlinear cost functions. Moreover, its computational cost is usually very light. This PSO method provides a powerful tool for dealing with an optimization problem in a high-dimensional search space. Inspired by [11], we apply PSO to the fitting of the 3D skeleton model to the extracted skeleton data.

In the process of pose estimation, we fit the 3-D skeleton model defined to the extracted and labeled skeleton. The skeleton model is composed of 12 line segments while the extracted skeleton data consist of many nodes in the 3-D space. Our goal is to find the minima of the evaluation function. The evaluation function is defined as the Euclidean distance between the human model and the extracted skeleton.

In the proposed pose estimation algorithm, a swarm of particles and an evaluation function *f* are defined in the search space with the dimensionality of D. Each particle is represented as a vector $\mathbf{p}_i = [p_{i1}\ p_{i2}\ \dots\ p_{iD}]^T$ with D elements. Furthermore, every particle has its associative velocity $\mathbf{v}_i=[v_{i1}\ v_{i2}\ \dots\ v_{iD}]^T$ to guide its motion. In every iteration, the value of the evaluation function is computed and recorded for each particle. Two kinds of information are evaluated. The first kind of information is the best position so far for each particle, recorded as $\mathbf{b}_i$. This $\mathbf{b}_i$ is to keep the information of self experience. The second kind of information is the globally best position, denoted as **gb**. **gb** is evaluated by finding the minimal value of *f* so far. The new location of each particle is then updated using the information of self experience and the

globally best position. Gradually, most particles will converge to the optimal position which has the minimal value of $f(\mathbf{p})$.

Besides the use of the PSO method, we also need to ensure the temporal consistency between frames. The motion changes between the current frame and its previous frame should be smoothly changing. To ensure the temporal consistency, we propagate the values of the estimated parameters from the current frame to the next frame. In other words, we restrict the values of the parameters for the next frame to be within some range around the estimated parameters at the current frame. However, since an incorrect estimation may also propagate over time, we add a re-initialization mechanism for each frame. When the fitting error is larger than some pre-defined threshold, we will re-initialize the whole pose estimation process based on the current frame only. This can prevent the propagation of errors.

5. SIMULATION RESULTS

5.1 Multi-camera coordination system

Fig. 6 shows a few sets of images captured at different time instants. In our experiments, the test videos are synthesized by ObjectVideo Virtual Video (OVVV) [17], which is a publicly available visual surveillance simulation test bed. By using OVVV, we can easily design various kinds of scenario and camera setups. In Fig. 4, there are nine people walking around in the scene. Each person is assigned a color and we use this color to plot a bounding box for that person. The synthesized scene is captured by four static cameras and four PTZ cameras, locating at different positions. In each figure, the left four frames denotes the pictures captured by the static cameras, while the right four frames are the pictures captured by the PTZ cameras. The use of static cameras can help the reader to easily realize the relations among these nine people; while the images captured by the PTZ cameras demonstrate the results of camera coordination.

Table 1 The statistical results of all test sequences

	Average Score	Unclear Rate
SEQ-1 (9)	7.5355	0.0200
SEQ-2 (9)	5.0644	0.0393
SEQ-3 (6)	7.6656	0.0056
SEQ-4 (6)	8.0526	0.0017
SEQ-5 (7)	8.3151	0.0138
Average	7.3266	0.0161

Fig. 6 Experimental results of the test sequence SEQ-1

Table 1 shows the statistical results of all experimental sequences. The “average score” is the average score for all people in a sequence. “Unclear Rate” is the average ratio of the zero-score time over the total time. The highest average score is 10 and the lowest is 0. Here, the frame size of each camera is 320×240 for all experiments. We let the value of th_{max}, th_{min}, and th_{θ} be 50, 15, and $\pi/2$, respectively for all experiments. The upper bound of the unclear period is set to 20.

5.2 Human body pose estimation

In our simulation, we use ObjectVideo Virtual Video (OVVV) [17] to simulate a synthesized environment, which contains eight virtual cameras. We test 9 sequences in total, as listed in Table 1. Some simulation results are shown in Fig. 7. In Fig. 7, we can clearly see that the proposed method can faithfully generate the corresponding skeleton models based on a set of images collected from multiple cameras.

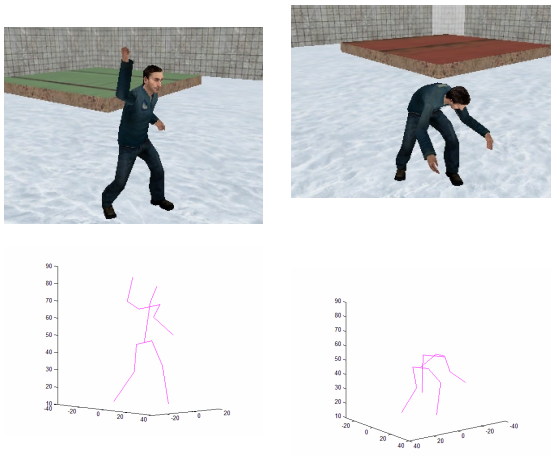


Fig. 7: The pose estimation results for 4 different sequences.

6. CONCLUSIONS

In this project, we proposed two algorithms for the application of video surveillance: (a) a camera coordination system for surveillance and (b) human body pose estimation method. For the camera coordination system, we construct a camera coordination system to control multiple PTZ cameras to capture as many frontal high-resolution facial images as possible. We formulate the camera coordination problem in terms shoot angle and the face width. By taking into account the overall scores in capturing facial images, we convert the coordination of cameras into an optimization problem. We then develop a modified algorithm over the discrete binary particle swarm optimization method to get the optimal camera assignment. For human body pose estimation method, we proposed a model-based pose estimation technique for multiple camera motion capture system. The whole process, which includes initialization and pose estimation, is automatic and markerless. For system initialization, we reconstruct the 3D visual hull from multiple foreground silhouettes. We segment the human body in the eigenspace, and then extract the skeletons to reduce the dimensions of the feature space. In the initialization stage, no prior model is needed. Furthermore, we modify the Laplacian Eigenmap to make the body parts segmentation easier than Sundaresan's method. After system initialization, a prior 3D human model is fitted to the extracted skeleton based on the PSO algorithm. Our human model allows self-spin and combines motion constraints with the pose

REFERENCES

[1] C. Micheloni, G. L. Foresti and L. Snidaro, "A cooperative multicamera system for video-surveillance of parking lots," *IEE Symposium on Intelligence Distributed Surveillance Systems*, pp. 1-5, Feb. 2003.

[2] C. Micheloni, G. L. Foresti and L. Snidaro, "A network of co-operative cameras for visual surveillance," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 152, pp. 205-212, April 2005.

[3] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Magazine*, vol. 22, pp. 38-51, Mar. 2005.

[4] A. Khiat, S. Yous, T. Ogasawara and M. Kidode, "Combining Fixed Stereo and Active Monocular Cameras into a Platform for Security Applications," *IEEE Int. Conf. on Robotics and Biomimetics*, pp. 1134-1139, Dec. 2006.

[5] Faisal Z. Qureshi and Demetri Terzopoulos, "Surveillance Camera Scheduling: A Virtual Vision Approach," *Multimedia Systems*, vol. 12, pp. 269-283, Dec. 2006.

[6] Faisal Z. Qureshi and Demetri Terzopoulos, "Surveillance in Virtual Reality: System Design and Multi-Camera Control," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, Jun. 2007.

[7] A. Laurentini, "How Many 2D Silhouettes

Does It Take to Reconstruct a 3D Object?" *Computer Vision and Image Understanding,* vol. 67, no.1, pp. 81-87, 1997.

[8] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman, "Human Body Model Acquisition and Tracking Using Voxel Data," *International Journal of Computer Vision*, vol.53, pp. 199-223, 2003.

[9] C. Menier, E. Boyer, and B. Raffin,"3D Skeleton-based Body Pose Recovery," *International Symposium on 3D Data Processing, Visualization and Transmission,* pp. 389-396, 2006.

[10] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of the IEEE International Conference on Neural Networks,* vol. 4, pp. 1942-1948, 1995.

[11] C. Robertson and E. Trucco, "Human Body Posture via Hierarchical Evolutionary Optimization," *British Machine Vision Conference,* vol. 3, pp. 999-1008, 2006.

[12] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Comput*., pp. 1373-1396, 2003.

[13] A. Sundaresan, and R. Chellappa, "Model Driven Segmentation of Articulating Humans in Laplacian Eigenspace," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[14] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[15] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, pp. 4104-4109, Oct. 1997.

[16] R. Murray, Z. Li, S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1993.

[17] G. R. Taylor, A. J. Chosak, and P. C. Brewer, "OVVV: Using Virtual Worlds to Design and Evaluate Surveillance Systems," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1-8, June 2007.