Full Body Tracking Using an Agent-based Architecture

Bing Fang, Liguang Xie, Pak-Kiu Chung, Yong Cao, Francis Quek
Center for Human Computer Interaction
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24060
Email: { fangb, xie, pchung, yongcao, quek}@vt.edu

***Abstract*—We present an agent-based full body tracking and 3D animation system to generate motion data using stereo calibrated cameras. The novelty of our approach is that agents are bound to body-part (bone structure) being tracked. These agents are autonomous, self-aware entities that are capable of communicating with other agents to perform tracking within agent coalitions. Each agent seeks for "evidence" for its existence both from low-level features (e.g. motion vector fields, color blobs) as well as from its peers (other agents representing body-parts with which it is compatible), and it also combines the knowledge from high-level abstraction. Multiple agents may represent different "candidates" for a body-part, and compete for a place within a coalition that constitutes the tracking of an articulated human body. The power of our approach is the flexibility by which domain information may be encoded within each agent to produce an overall tracking solution. We demonstrate the effectiveness of tracking system by testing actions (random moving and walking).**

I. Introduction

There is an increasing requirement for the applications which track the motion of human and other objects in our daily life. The tracking-based systems have become very popular in computer vision research for motion estimation and generation. Automatic motion generation techniques have a wide range of applications, ranging from video games interface, interactive character control in virtual environments to filmmaking. Motion capture is one of these techniques that has been widely adopted by the animation community. However, motion capture often requires high-cost equipments and usually takes long time to setup, and therefore not suitable for common use. As a result, there has been growing interest in research related to alternative, and low-cost motion synthesis techniques.

As an alternative, vision-based approaches provide low-cost solutions using off-the-shelf digital cameras. However, images obtained by cameras are often noisy and visual features are usually unstable, and therefore most of the vision-based motion synthesis approaches these days are data-driven, and thus limit the kinds of motions to those available in the database.

We propose an agent-based architecture to perform a multi-modal tracking system in this paper. The advantages of our system are three folds. First, our system employs an agent-based architecture, where each body part is implemented as an "agent" in our system, which is an independent and self-aware entity with embedded knowledge capable of seek out visual features and communicate with its neighboring agents in order to find out its own location. Second, our system can use low-cost off-the-shelf equipments, the digital cameras are quite affordable and have potential for home use. And, if the high-quality equipment is available, our system can also take the high-quality signals as input without any change in system design. Third, our approach is not data-driven and does not suffer from the limitations of data-driven techniques. This framework offers a more robust tracking capability compared with traditional vision-based techniques to detect body part locations even when visual features extracted from video sequences are incomplete and unstable. And the architecture allows us to think about tracking in highly abstracted concept.

We use a full body tracking as an example application to demonstrate the power of our agent-based architecture. Our prototype system uses two stereo-calibrated digital video cameras. Subject wears a black skin-tight suit, blue tapes are attached to joint locations, which act as markers. Agent-based architecture provides us the benefit of conceptualize the tracking tasks with a highly abstracted agent object, while not focusing on local feature tracking. In other words, our agent-based system does not need to know how to extract video sequences that are captured from these two cameras simultaneously. These video sequences are then processed by the system to generate high quality motion data. We evaluate our system by having the subject perform two different motion sequences. We then compare the resulting animation with the captured video. The evaluation shows that our system can accurately capture the motion data of these sequences.

The rest of this paper is organized as follow. Section II provides the background and related work in this area. Section III describes the agent-based architecture in a tracking system. Section IV provides a details description of our demonstration using the agent-based architecture. Section V presents the result of our preliminary test. Section VI summarizes the paper.

II. Background

In the following sub-sections, we discuss related work in vision-based motion tracking. We also give an overview of agent-based systems, and their advantages.

A. Vision-based Motion Tracking

Lee et al. [4] built a vision-based interface to obtain silhouette data from a single video camera. Then the noisy silhouette data are transformed to full-body motion by searching a motion graph using Hu moments computed from the input silhouettes. Ren et al. [7] combined information about the user's motion contained in silhouettes from three cameras with domain knowledge contained in a motion capture database to produce a high quality animation.

Chai et al. [3] implemented a vision based system that requires only two inexpensive video cameras. Using only six markers attached to a body, the system can synthesize a wide variety of human movement without a long suit-up time. The synthesized motions are very detailed because a data-driven approach is used to query a high quality motion capture database. Similarly, Liu et al. [5] applied a linear regression model to estimate human motions from a reduced marker set. However, these systems require extensive motion capture database and suffer from specific domain of synthesized motion because of the natural property of data-driven approach.

Compared with vision-based motion synthesis systems, marker-based motion capture systems can generate high-fidelity animation data with subtle details of human motions. These systems perform best in the applications that mostly play back the original motions, e.g., animated movies. However, editing the original motion capture data usually results in the non-realistic animations.

B. Agent-Based Systems for Visual Tracking

The concept of agent or multi-agent systems was originated in area of distributed artificial intelligence [10], and been applied to various fields of study in recent years. In [8] and [2], multi-agent-based systems are described as systems that designed to decompose problems that can be solved separately, and these solutions can then be synthesized in order to obtain the solution of the original problem.

Agents can be understood as autonomous, problem-solving entities that are capable of operating in complex environment. One of the key advantages of agent-based systems is abstraction [2]. Contrast to objects in object-oriented programming paradigms, agents are different in many aspects. First, agents are autonomous, and they are able to invoke actions on their own without intervention from other entities. Second, they are aware of the environment and able to act in response to environmental changes. Third, agents are able to communicate with each other using shared knowledge to accomplish their set agenda. This type of abstraction enforces well-organized design and programming, resulting in flexible, conceptually simple and extendable implementation.

Agent-based visual tracking systems are systems that employ agent-based approach to analyze, retrieve, and process images or video sequences in order effectively tracking moving objects in a motion sequences. Relative few studies have been done in this immediate field. [2] presents a hierarchical agent framework, where each agent is responsible to handle tasks of a specific layer in a hierarchical tracking framework to achieve the overall goal of identifying object trajectories.

III. Agent-Based Tracking System

As we described in previous sections, agents can be considered in a broad sense, and we are going to discuss the design of architecture in this section.

A. Architecture of an Agent-Based Systems

There are two key reasons that agent-architecture is important [6]: one is that we want to provide a methodology to build a real agent system; the other is we also want to predict and explain the behavior of the agent system based on its current state and environment. There are variety of ways to build an agent system [6].

In an agent system, each agent processes independently with its locally embedded features, algorithms, capture devices and environments, and each agent should have a representation of high level abstraction. The feature of independent processing in agent-based system makes it much natural to be implemented on the increasingly popular parallel processing system. Moreover, the highly abstracted conceptual representation of agents gives us the chance to think about the target object in its concept, while not thinking only about its detailed features.

1) Conceptual Representation and Hierarchical Architecture: Our agent-based tracking system is aimed at tracking full body motion with stereo calibrated cameras, and play the animation based on tracking results. We use a simple *joint-and-bone* model to represent the tracked skeleton of the subject. Each *joint* is conceptually implemented as an agent in our system, so that the agent can be considered as head, hand, torso, and etc. Each of these agents has embedded knowledge, such as its identity, identities of its neighbor agents and its constraints. These agents are capable of acting on their own to perform detection and tracking in order to discover their own locations. This design is quite straightforward, and the relationship between agents can be easily learned in the conceptual level. The advantage of this design is that each agent is a direct representation of a joint and so it is intuitive to train these agents by using knowledge of their real-world counterparts. These agents are also capable of communicating with their neighbors in order to re-evaluate and adjust tracking result to achieve a prediction that is consistent throughout the whole model.

As illustrated in Fig.1, we introduce a competition-surviving approach for agent-based architecture. When an agent is created, it starts to look for evidence for its existence. The evidence can come from three ways: lower level supporting, local knowledge and higher level knowledge. For example,

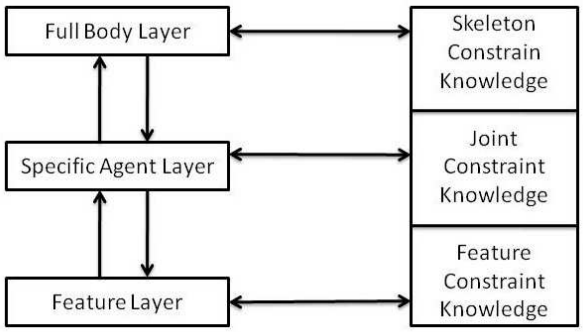

Fig. 1: *Agent Competition-Surviving Method.*

a hand agent has one of the local knowledges that a hand is skin-color blob. Then, this hand agent goes to find skin-color blobs, which is supported by some skin-color agent in lower level. At the same time, the higher level agent, such as skeleton-model agent, tells this hand agent that "I can only accept two hands in my structure". Now, all the hand agent candidates, which might be more than two, will compete with each other, and only two will survive in this case. In this processing level, the agent can be mainly considered in highly abstracted concepts, so that a hand is considered as a part of body with skin color and directly connected with lower arm and so on in this example.

2) Independent Analysis and Communication: Agent-based architecture takes the advantage of its independent processing, so we treat each agent as an independent model. First, each agent does its own processing independently. In this procedure, the agent has its local feature extraction, and local estimation. The independent model also provides agent-architecture a benefit that each agent can have its own way to extract feature and can perform its own analysis. Second, existing agents communicate with each other to extract further evidence for their existing. Communication happens between different agents, however, each agent has the knowledge which tells the communication targets. For example, a hand agent does need to communicate with lower arm agent or upper arm agent, however, it does not need to communicate with leg agent or foot agent. Because the uncorresponded agents do not affect each other in conceptual level. With this powerful property, we can perform parallel computing in our system.

B. Model-based Tracking using Agent-based Architecture

Model-based tracking approach may use stick figures, 2D contours, volumetric model and etc. The basic task is to recover the configuration of the model that corresponds to the data [2]. The main advantage is the priori modeling knowledge, but the complexity and difficulty in encoding and modifying. However, using agent-based architecture can persist the advantage of model-based tracking, and it can also make up the disadvantages.

In agent-based architecture, an agent is created with its own knowledge and relationship with other existing agents. By this definition, we can easily add an agent by creating an agent unit, and attach the knowledge information and relation information to it. This operation helps us to avoid the unnecessary work, such as recoding. We can also remove an existing agent by deleting the connections of the agent, and redirect relation relationship connections.

IV. Full Body Tracking

We use the full body tracking task as an example to demonstrate the advantage of agent-based tracking architecture. A vision-based motion tracking using our agent-based architecture system is proposed, and we perform a 3D animation to illustrate our tracking results. Our prototype system uses two stereo-calibrated digital video cameras. Subject wears a black skin-tight clothes with blue tapes attached to joint locations, which act as markers. The subject is required to be visible to both cameras. The subject is also required to face two cameras in order to maximize the exposure of the markers to the cameras.

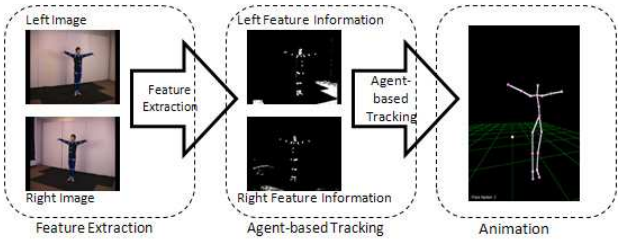

Fig. 2: *Stereo Vision-based Tracking and 3D Animation.*

In our approach shown as Fig.2, the system takes synchronized video sequences captured by two stereo calibrated cameras as input data, and processes the images in the video sequences to extract visual feature information. The agent-based tracking system is then employed to track the skeleton motion, and the skeleton information of each frame is produced for 3D animation.

A. Camera Calibration and Triangulation

Video sequences from cameras only provide 2D information. In order to convert 2D coordinates from camera frames to 3D world coordinates, we must first compute intrinsic and extrinsic parameters for the stereo cameras. Our system employs Tsai's algorithm [9] to calculate these parameters, and the cameras are set up as shown in Fig.3. Tsai's algorithm requires a minimum of 11 sets of corresponding control points to perform this calculation, but 20-60 sets are usually used in the calibration process. In our experiment, we use a calibration box with 48 control points on it, as illustrated in Fig.4. A pair of 2D coordinates (one of each camera) for each of these 48 control points are then recorded and Tsai's calibration algorithm is then applied to estimate the camera

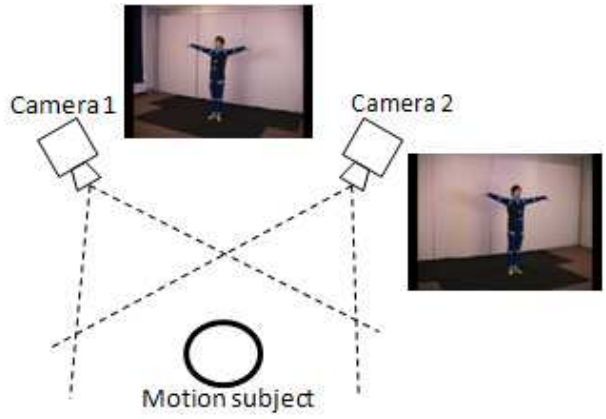

Fig. 3: *Stereo Camera Setting.*

parameters. We can then obtain the 3D world coordinates from a pair of corresponding 2D coordinates on camera frames by triangulation. [1].

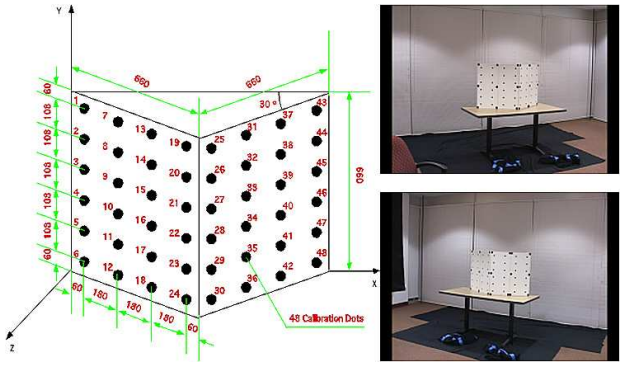

Fig. 4: *Calibration Box.*

B. Features Extraction

The low level processing of the system is feature extraction. As mentioned in previous section, blue markers are attached to the tracking clothes. Locations of these markers and the visible skin regions provide visual features needed for the agent-based system to construct the skeleton model to generate the motion data we need. This part of the system plays the role of extracting these visual features from video frames, including shapes, distribution, and center positions of the blue markers and exposed skin regions. We implement an adaptive normal distribution color model [11] in the normalized (r,g) space. And we build two different models for the blue tapes and skin color area. We then process each pixel in video frames using the adaptive color models to identify interest areas of markers and skin regions as follow:

$$P(x) = \exp[-\frac{1}{2}(x-\hat{\mu})^T \hat{\Sigma}^{-1}(x-\hat{\mu})] \quad (1)$$

where $(\hat{\mu},\hat{\Sigma})$ represents the mean and variation vector in normalized r,g space, and all pixels in each frame are separated into skin-color/non-skin-color or marker-color/non-marker-color.

After the color model filtering, we employ a region-growing algorithm to identify connected regions. These regions, including their centroid (in our approach, we describe regions by rectangles), shape, and color information, are then used in our agent-based tracking system as input features.

C. Agent-Based Tracking System Implementation

In our system, we define each joint as a specific agent, and each of these agents has embedded knowledge of things such as its identity, identities of its neighboring agents and the kind of constraints it has. The skeleton information is considered as high-level agent, and the feature extraction agent employs the Gaussian color modeling and filtering as we described in previous section. These agents are capable of acting on their own to perform detection and tracking in order to discover their own locations. This design is quite straightforward, and the relationship between agents can be easily learned during the experiment. The advantage of this design is that each agent is a direct representation of a joint and so it is intuitive to train these agents by using knowledge of their real-world counterparts. These agents are also capable of communicating with their neighbors in order to re-evaluate and adjust tracking result to achieve a prediction that is consistent throughout the whole model.

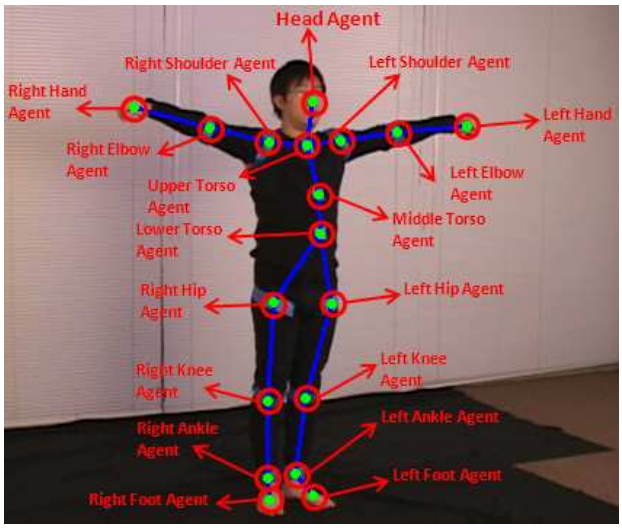

Fig. 5: *Agent definition*

The subject is asked to perform a "T-Pose" at the beginning of each action sequence for roughly 1 second (30 frames) shown as Fig.5. This is an initialization phase, agents are able to automatically map themselves to the correct locations of joints and body parts of the subject on image frames. Once the initialization phase is complete, all agents are activated.

1) Agent processing: As described in previous sections, An agent is capable of acting on its own. Fig.6 illustrates the

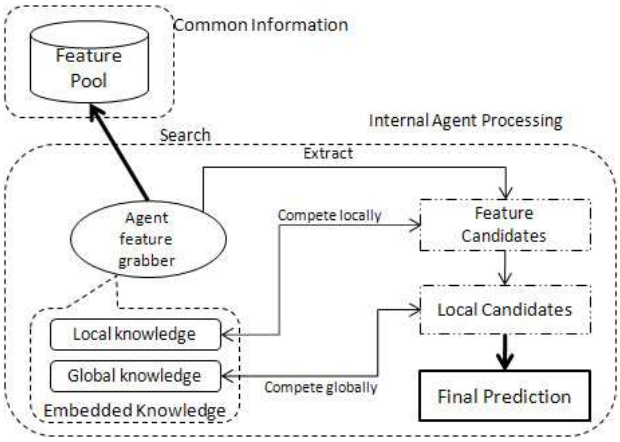

Fig. 6: *Agent Processing*

implementation of an agent. As shown, each agent comes with a feature grabber, which grabs all possible candidates from feature pool (when they are available) that may help it to determine the agent's whereabout. The agent then select the most likely candidates and label them as "local candidates" based on embedded local knowledge of each agent. Global knowledge is then applied to help select the final prediction from these "local candidates".

2) Feature Grabbing: Prior knowledge is embedded in each agent, as such the agent ignores impossible candidates when grabbing features from the feature pool. For instance, the agent knows how fast itself can move in a consecutive frame, and so it would only grab features that are within a distance from its location in the previous frame. Also, a left hand agent would only grab visual features that are identified as skin regions and ignore the marker areas.

3) Feature Selection Based on Local Knowledge: Embedded local knowledge helps agent to select the more likely candidates from all the grabbed features. One of the key local knowledge available to each agent is its motion trajectory. We employ a trajectory-based estimation technique. Suppose we have n historical positions of an agent in the previous n frames. The least-square method is used to interpolate the trajectory of this agent, and estimation should follow this trajectory. However, noise may break this trajectory. Since we used normal distribution color model and vision-based processing, incomplete visual features and noises may result in unexpected movement anytime, in such case the agent would apply anti-shake algorithm to stabilize its motion trajectory. Using the estimated positions based on the agent's motion trajectories and other embedded local knowledge, less likely candidates are discarded. The candidates left are labeled as a "local candidates".

4) Global Knowledge and Communication Between Agents: Agents are capable to act and make decision on their own, however they are also capable to communicating with neighboring agents, adjust and re-evaluate their results to achieve a prediction that is consistent both locally and globally. For instance, In some cases, the agent may have multiple, equally-likely "final prediction" based on local knowledge, or no prediction at all for the current frame (due to occlusion for example). In such case, agents communicate with each other to share knowledge. For example, the left elbow agent is connected with the left shoulder agent and the left hand agent, and the 3D distance from the left elbow agent to the other two agents should be consistent across different frames. Furthermore, it is highly unlikely for the left shoulder to move faster than the left elbow. In some other cases, agents communicate with each other to compete, which can happen in situations when only one feature candidate is available for multiple agents. Agents compete with each other to decide the who would become the "owner" of the candidate feature. When communicating with other agents, there is also the consideration of how much one agent can be "trusted", as such each agent also maintain a "confidence" value, which would give an idea to agents communicating with it about whether to trust this agent or better to trust itself. Using this global knowledge through communicating with neighbor agents, an agent is then able to make a final prediction and determine its position.

$$E_{A_i} = c_{local} \sum P_{A_i} + c_{global} \sum (P_{A_i} | P_{A_j}) \tag{2}$$

where c represents the confidence , and P represents the prediction of the agent. In the equation, the first part means the contribution from the agent itself, and the second part means the contribution from other related agents. E is the estimation result for the agent.

D. Animation Smoothing

As we described our approach above, the system has its own limitation. First, in video-based analysis, the final prediction position of an agent in one camera view may be not correspondent to the final prediction of the same agent in another camera view, although the system 'thought' they are correspondent. Secondly, even in the same camera capturing sequence, the final prediction of the agent in the current frame may be not correspondent to the prediction in the previous frame. Finally, during the estimation of the intrinsic and extrinsic parameters, we find that this calibration-triangulation approach causes around 10 mm error for each coordinates in 3D space.

So, a post-processing for the 3D animation is performed. During the post-processing step, we employ a low pass filter to smooth the synthesized motion data, and the smoothed 3D animation is produced. In this processing, we employ a k-window filter to do smoothing:

$$I_{smoothed} = \frac{1}{2} \sum_{k} I_{raw} \tag{3}$$

where I represents the 3D skeleton information.

V. Result and Discussion

In our experiment, we exam two motions with our agent-based tracking system, play the raw 3D animation obtained directly from the tracking results, and also play the smoothed 3D animation.

In the first testing motion video, the subject performed a random moving, and all the markers can be captured by both camera in each frame. From the display results, we demonstrate that the raw 3D animation is quite closed to the smoothed 3D animation result. When we looked inside the processing of each agent, the competition mechanism worked perfectly to help agents have perfect prediction.

In the second testing motion video, shown in Fig.7, the subject performed a normal walking, and not all the markers can be seen from both camera view, where occlusion happened frequently. In the figure, we can find that the left ankle was blocked by the right leg. In this situation, it is hard to allocate the position by detection, and since the occlusion persists within a certain time, the estimation becomes more difficult. However, our agent-based system will use its agent-communication function to ask its neighbor for help. In this case, the left ankle agent goes to left knee agent and left foot agent for help. It finds that both agents are confident that they do not move in this time, so the left ankle agent makes the decision that it will not move. Although there is a motion features around, which is actually the feature belongs to the right knee agent.

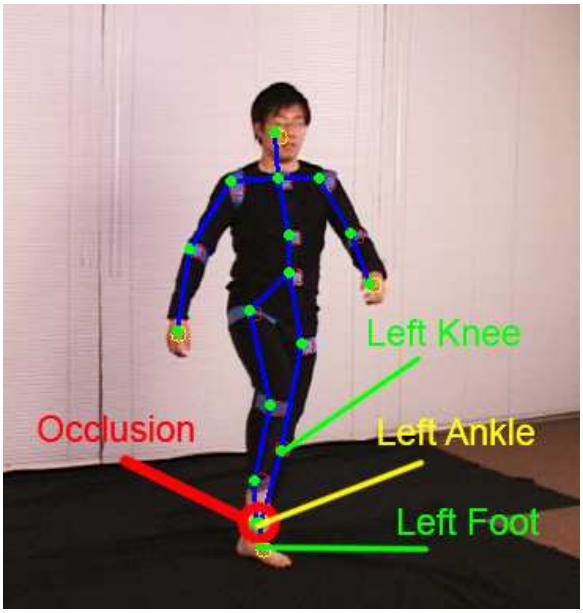

Fig. 7: *Agent communication when occlusion happens*

From the experiment, we demonstrate that the agent-based tracking system can be implemented to perform non-data-driven 3D animation shown as Fig.8. Although we use color markers for feature extraction, it gives a quite noisy feature

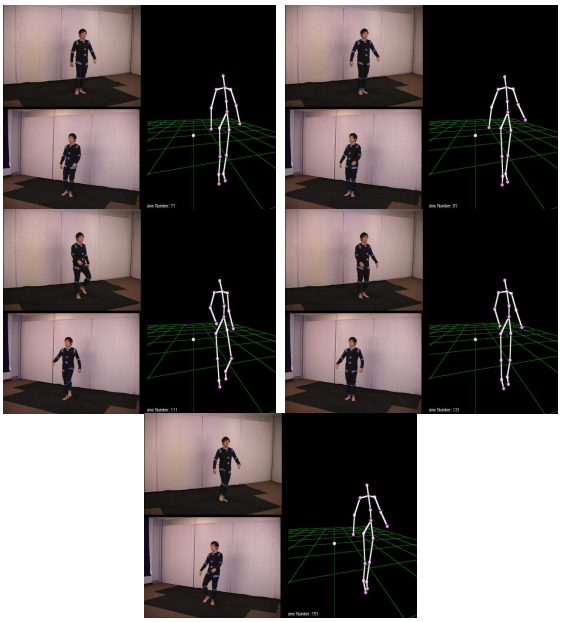

Fig. 8: *Result for a walking motion. For each subfigure, the left two pictures are images captured by the cameras, and the right picture is the animation generated by our system.*

information. And, when features 'disappear' from the camera view, the system can still handle by agent competition and communication with embedded agent knowledge. However, if the correlated feature can not be extracted for longer time, the accuracy of estimation is still acceptable, but the result might affect the final prediction result, which is a common bottle-neck in video-based analysis.

VI. Conclusion

We demonstrate the possibility to apply an agent-based tracking system to track the skeleton motion using stereo cameras. We use two simple motions to evaluate our system, and the result shows the supporting to our system.

In our system, we employ color marker model for feature extraction. During the experiment, we discover that not all these markers are essential for tracking and 3D animation. For example, the upper torso agent can be estimated to be the centroid of left shoulder agent and right shoulder agent, etc. And, if we add one more camera in our setting, we can provide three pair of calibrated feature information. The additional 2 pair of information will make all these markers visible by one pair of camera capturing at least. This can also provide a more accurate prediction result after the combination with these three pairs of information.

Another discovery for this experiment is that, although our design of agents work well for the motion, the agents defined by joint position are too sensitive in 3D space. However, the good extendability of agent-based system provide us a possibility to change the definition of agent easily. So we only

need to embedded a new knowledge to the agent, and the new agent is produced without much changing of original codes. Moreover, we can use multi-model feature inputs with our feature extraction agents, for example, we can attach some low-cost accelerometers on the torso part, which has less features and is difficult to track. The agent-based architecture also provides this flexibility to implement feature extraction agents with multi-model input devices.

ACKNOWLEDGEMENTS

This research has been partially supported by NSF grants Embodied Communication: Vivid Interaction with History and Literature, IIS-0624701, Interacting with the Embodied Mind, CRI-0551610, and Embodiment Awareness, Mathematics Discourse and the Blind, NSF-IIS- 0451843.

REFERENCES

[1] Vislab, `http://vislab.cs.vt.edu`.

[2] R. Bryll, R. Rose, and F. Quek. Agent-based gesture tracking. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 35(6):795–810, 2005.

[3] J. Chai and J. K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. Graph.*, 24(3):686–696, 2005.

[4] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of ACM Siggraph 2002*, pages 491–500, New York, NY, USA, 2002. ACM.

[5] G. Liu, J. Zhang, W. Wang, and L. McMillan. Human motion estimation from a reduced marker set. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 35–42, New York, NY, USA, 2006. ACM.

[6] R. A. Michael Luck and M. d'Inverno. *Agent-Based Software Development*. Artech House, London, 2004.

[7] L. Ren, G. Shakhnarovich, J. K. Hodgins, H. Pfister, and P. Viola. Learning silhouette features for control of human motion. *ACM Trans. Graph.*, 24(4):1303–1331, 2005.

[8] P. Stone and M. M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[9] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[10] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[11] Y. Xiong, B. Fang, and F. Quek. Extraction of hand gestures with adaptive skin color models and its application to meeting analysis. *the Eighth IEEE International Symposium on Multimedia*, Sep. 2006.