

MoCap-based Camera Tracking for VFX

Jong-Sung Kim, Hye-Mi Kim, Jae-Hean Kim

Visual Contents Research Department, ETRI, South Korea

js.kim@etri.re.kr, miya0404@etri.re.kr, gokjh@etri.re.kr

Abstract— In this paper, we present a motion capture-based camera tracking method to create visual effects by composing motion capture-driven CG animation with real on-set images for TV and film production. Our method can compute all of the motion parameters of camera directly from motion capture data with sub-pixel accuracy. In addition, it can also calibrate the lens parameters of camera without using additional calibration tools. The effectiveness of our method has been verified through several real experiments.

Keywords— Camera tracking, motion capture, visual effects, image composition

I. INTRODUCTION

CG/real image composition is a key ingredient for creating visual effects (VFX) such as digital special make-up effects in TV and film production. Although it has been studied for a long time in computer graphics and computer vision, it still remains as a difficult problem in practice because it requires a reliable solution to the camera tracking problem, one of fundamental problems, especially in computer vision.

Traditional camera tracking techniques are categorized into the sensor-based approach [1, 2] and the vision-based approach [3, 4]. In the sensor-based approach, the changes of rotational and/or translational motion parameters of camera are tracked by using inertial motion sensors such as gyroscope and the lens parameters of camera are recorded with zoom/focus encoders. Although the sensor-based approach can run in time, it suffers from the accumulation errors in inertial sensors. In the vision-based approach, the values of camera motion parameters are tracked by applying some vision algorithms to visual data such as images and the lens parameters of camera are calibrated by using some calibration tools [5]. Although the vision-based approach does not rely on inertial sensors, it still has many problems such as visual ambiguity [6].

Motion capture (MoCap) has been originally invented to record the motion of human or props with sensors, including cameras, and then digitally map the recorded motion data, called MoCap data, to computer-generated (CG) creatures. At now, a variety of MoCap techniques, including the optical (vision-based) motion capture system [7], are used for VFX in TV and film production [8].

In this paper, we employ an optical MoCap technique to solve the camera tracking problem in TV and film production. For this aim, we present a MoCap-based camera tracking

method to obtain all of the camera parameters from MoCap and video data. More specifically, the 3D rotation and translation parameters of camera are directly computed from synchronized marker-based MoCap and video capture data with sub-pixel accuracy, which allows us to define the 3D motion of camera in 3D MoCap coordinate system. The lens parameters of camera, including focal length, principal point, distortion coefficients, are also computed by using our method without using calibration tools.

The organization of this paper is as follows. Section 2 briefly introduces our CG/real image composition system. Section 3 explains our MoCap-based camera tracking method in details. Experimental results are given in Section 4 and our conclusions are drawn in Section 5.

II. CG/REAL IMAGE COMPOSITION

CG/real image composition is widely used to create VFX in TV and film production. Especially, the composition of CG animation and real on-set images, so-called matchmation, is very popular in film production.

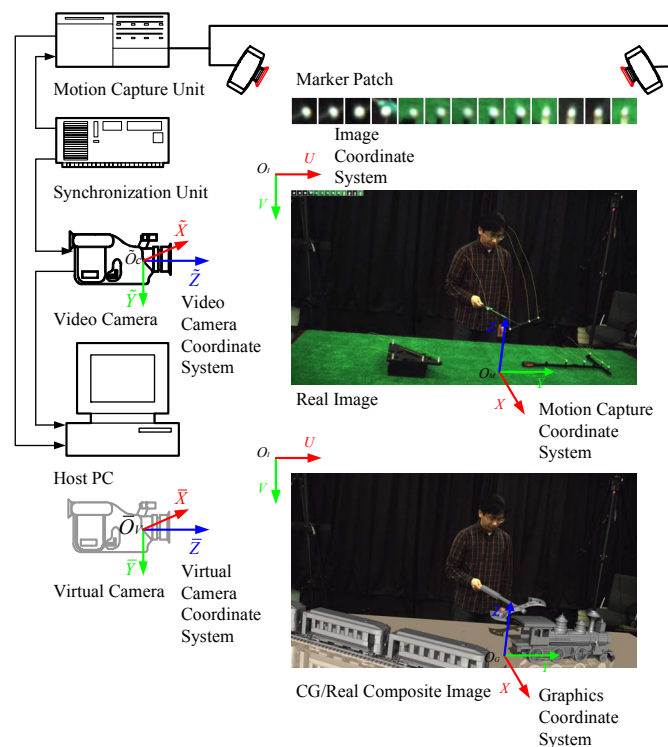


Figure 1. Configuration of our CG/real image composition system

Figure 1 shows the configuration for our CG/real image composition system. As shown in the figure, we use MoCap data and video data obtained by an optical MoCap unit and a video camera, respectively. The MoCap unit and the video camera are synchronized with a synchronization unit.

The MoCap unit records the MoCap data of human by computing the 3D coordinates of optical markers in the MoCap coordinate system O_M . At the same time, the video camera records the on-set video data of those markers on human. At the host pc, the 2D coordinates of those markers are determined in the image coordinate system O_I by tracking the marker patch of each marker between consecutive video frames. From the above 3D and 2D coordinate data of markers, we compute all of the motion and lens parameters of camera, of which details will be described in the following section.

III. MoCAP-BASED CAMERA TRACKING

Our method exploits the MoCap and video data from the MoCap unit and the camera, respectively, as shown in Fig. 1. We assume that M MoCap markers are used for MoCap and T MoCap-video frames exist in the MoCap-video data. We denote the indices of MoCap markers and MoCap-video frames by m in $[1, \dots, M]$ and t in $[1, \dots, T]$, respectively. We represent the t th video frame image with I_{tR} .

The MoCap unit reconstructs the 3D position of each MoCap marker on the current MoCap frame. We represent the 3D coordinate of the m th MoCap marker on the t th MoCap frame with X_{tm} and the set of 3D coordinates of all MoCap markers on the t th MoCap frame with $X_t = \{X_{t1}, \dots, X_{tM}\}$, where X_{tm} is a 3D vector, defined by $X_{tm} = (x_{tm}, y_{tm}, z_{tm})^T$ in the 3D MoCap coordinate system O_M .

The 2D position u_{tm} of the m th MoCap marker on the t th video frame, defined by $u_{tm} = (u_{tm}, v_{tm})^T$ is determined by minimizing the following photometric error function

$$\hat{u}_{tm} = \min_{d \in W} (I_{tR}(u_{tm} + d) - J_{mR}(d))^2, \quad (1)$$

where $J_{mR}(d)$ is the marker patch of the m th MoCap marker to represent its visual properties, including shape, color, and texture, with a real sub-image (see Fig. 1), of which the length of height and width are h and w , respectively, d is the location of each pixel, defined by $d = (d_u, d_v)$ with d_u in $[-w, \dots, w]$ and d_v in $[-h, \dots, h]$ on the marker patch, and W is the domain of d , of which size is $(2h+1) \times (2w+1)$. On occasion, it happens that we cannot detect some MoCap markers on the current video frame due to occlusion. To discriminate undetected markers from detected ones, we introduce an occlusion identifier o_{tm} . If u_{tm} is detected, the value of o_{tm} is set to one. Otherwise, it is set to zero. We represent the set of 2D coordinates of all MoCap markers on the t th video frame with $u_t = \{u_{t1}, \dots, u_{tM}\}$.

The 3D motion of the video camera are modeled with six parameters, i.e., three rotation parameters $\omega_x, \omega_y, \omega_z$, and three translation parameters t_x, t_y, t_z . Using this model, we can define a 3×3 rotation matrix $\Omega_t = \Omega(\omega_x, \omega_y, \omega_z)$ and a 3×1

translation vector $t_t = (t_x, t_y, t_z)^T$ to numerically compute the motion of the camera, denoted by $\Theta_t = \{\Omega_t, t_t\}$. As in [9], the lens of the video camera can be modeled with the following eight parameters, i.e., two focal length parameters $f = \{f_u, f_v\}$, two principal point parameters $c = \{c_u, c_v\}$, four distortion coefficients $e = \{\mu_1, \mu_2, \tau_1, \tau_2\}$, where μ_1 and μ_2 parameterize the radial distortion and τ_1 and τ_2 the tangential distortion in the camera lens.

Our method tracks the 3D motion of the camera by estimating the motion parameters of the camera at the current t th MoCap-video frame from the set of 3D and 2D coordinate data of MoCap markers, i.e., X_t and u_t . We compute the optimal estimate of Θ_t , i.e., Ω_t and t_t , by minimizing the following geometric error function

$$\hat{\Theta}_t = \min \sum_{m=1}^M o_{tm} \|u_{tm} - h(\Theta_t, X_{tm}, f, c, e)\|^2, \quad (2)$$

where $h(\cdot)$ is the projection function from the MoCap space to the image plane and $\|\cdot\|$ is the l^2 -norm of a vector. We applied the Quasi-Newton method [10] to obtain a minimum of the error function from an initial guess, which is computed by using the direct linear transform method [11].

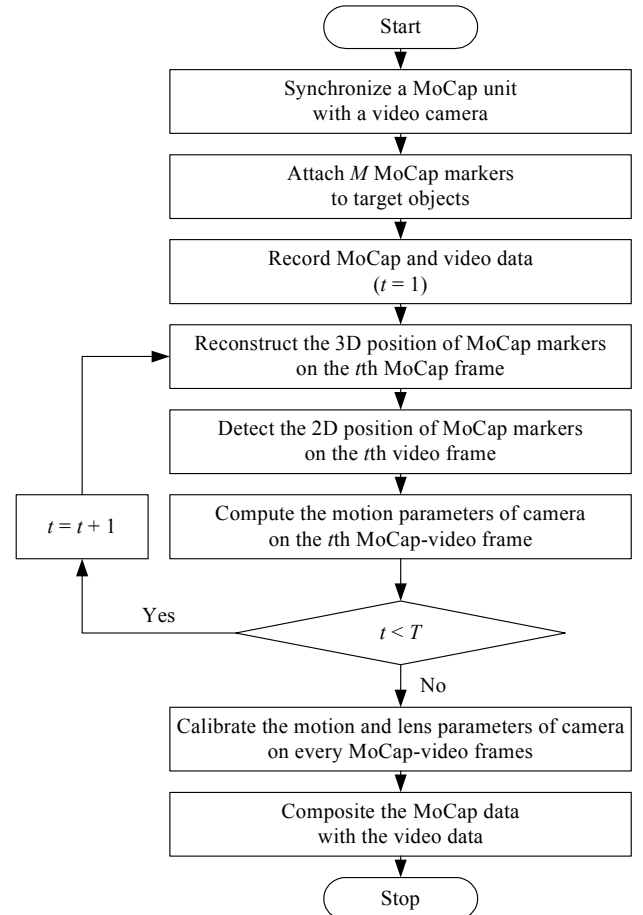


Figure 2. Flowchart of our MoCap-based camera tracking method



Figure 3. Result of *fast-motion* image composition on dataset 1

The projection function $h(\cdot)$ is defined by

$$h(\boldsymbol{\Theta}_t, \mathbf{X}_t, \mathbf{f}, \mathbf{c}, \mathbf{e}) = (1 + \mu_1 r^2 + \mu_2 r^4) \tilde{\mathbf{u}}_{tm} + \delta \tilde{\mathbf{u}}_{tm}, \quad (3)$$

where $\tilde{\mathbf{u}}_{tm}$ is the 2D coordinate of the perspective projection of $\mathbf{X}_{tm}$, which is computed with the following equation

$$\tilde{\mathbf{u}}_{tm} = \begin{pmatrix} \tilde{u}_{tm} \\ \tilde{v}_{tm} \end{pmatrix} = \begin{pmatrix} \frac{f_u \Omega_t^{[1]} \mathbf{X}_{tm} + f_u t_t^{[1]} + c_u}{\Omega_t^{[3]} \mathbf{X}_{tm} + t_t^{[3]}} \\ \frac{f_v \Omega_t^{[2]} \mathbf{X}_{tm} + f_v t_t^{[2]} + c_v}{\Omega_t^{[3]} \mathbf{X}_{tm} + t_t^{[3]}} \end{pmatrix}, \quad (4)$$

where $\Omega^{[i]}$ is the i th row vector of Ω and $t^{[i]}$ is the i th element of $\mathbf{t}$. In (3), the value of r is set to be the l^2 -norm of $\tilde{\mathbf{u}}_{tm}$ and the vector $\delta \tilde{\mathbf{u}}_{tm}$ is computed with the following equation

$$\delta \tilde{\mathbf{u}}_{tm} = \begin{pmatrix} 2\tau_1 \tilde{u}_{tm} \tilde{v}_{tm} + \tau_2 (r^2 + 2\tilde{u}_{tm}^2) \\ \tau_1 (r^2 + 2\tilde{v}_{tm}^2) + 2\tau_2 \tilde{u}_{tm} \tilde{v}_{tm} \end{pmatrix}. \quad (5)$$

We repeat the above detection and computation procedure until the frame index t reaches to T , i.e., until we finish the computation of all motion parameters, denoted by $\boldsymbol{\Theta} = (\boldsymbol{\Theta}_1, \dots, \boldsymbol{\Theta}_T)$, in the MoCap-video data. Then, we finally calibrate all of the motion and lens parameters of the camera, $\boldsymbol{\Theta}, \mathbf{f}, \mathbf{c}$, and $\mathbf{e}$. In the calibration procedure, we compute the optimal

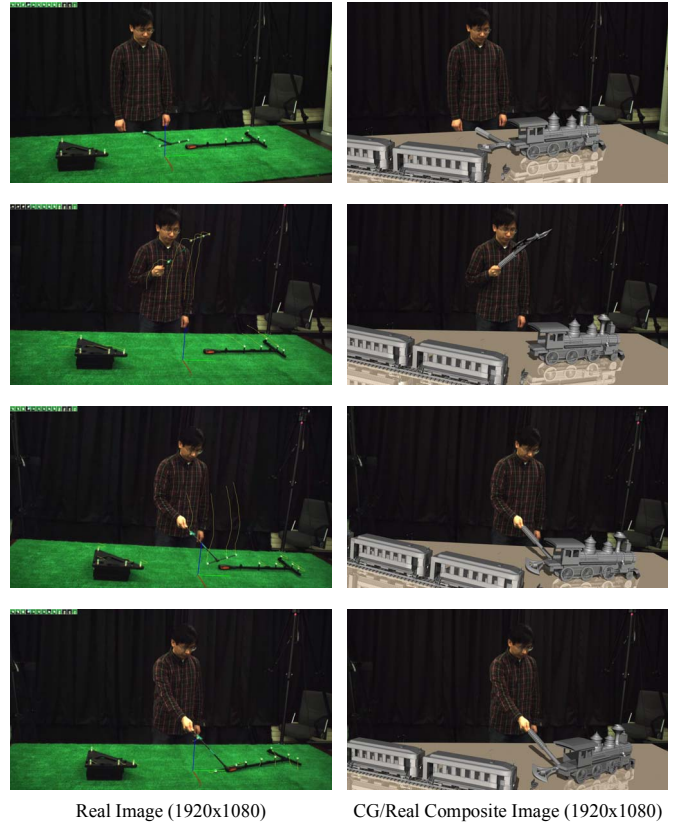


Figure 4. Result of *interaction* image composition on dataset 2

estimate of $\boldsymbol{\Theta}, \mathbf{f}, \mathbf{c}$, and $\mathbf{e}$ by minimizing the following geometric error function

$$\hat{\boldsymbol{\Theta}}, \hat{\mathbf{f}}, \hat{\mathbf{c}}, \hat{\mathbf{e}} = \min \sum_{t=1}^T \sum_{m=1}^M o_{tm} \|\mathbf{u}_{tm} - h(\boldsymbol{\Theta}_t, \mathbf{X}_t, \mathbf{f}, \mathbf{c}, \mathbf{e})\|^2. \quad (6)$$

The flowchart of our MoCap-based camera tracking method is given in Fig. 2. The estimation of all of the camera parameters with respect to the 3D MoCap coordinate system allows us to composite the video data with the MoCap data by exporting the parameter values of the video camera to the virtual camera in the graphics coordinate system of a 3D graphics tool.

IV. EXPERIMENTAL RESULTS

Experiments were conducted with real MoCap and video data to verify the effectiveness of our MoCap-based camera tracking method for CG/real image composition. For experiments, five real MoCap-video datasets of props and one of authors were actually captured in our laboratory with the MoCap unit and the video camera of our CG/real composition system as in Section 2. For the MoCap unit, we used Vicon MoCap system with four MX-F40 cameras [7]. For experiments, we set Vicon MoCap system to conduct the MoCap process in 60Hz. We used a video camera of 1920x1280 pixels, which were synchronized with Vicon MoCap system in hardware, to record high-definition video data in 30Hz. We used Autodesk Maya [12] as a 3D graphics



Figure 5. Result of special make-up image composition on dataset 3

tool which performs the CG/real image composition.

To quantitatively evaluate the performance of our camera tracking method, we have computed the reprojection error, i.e., the Root-Mean-Squares error (RMSE) in the unit of pixel, defined by

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T \left(\frac{1}{M_{\text{eff}}} \sum_{m=1}^M o_{tm} \|u_{tm} - h(\hat{\theta}_t, X_t, \hat{f}, \hat{c}, \hat{e})\| \right)},$$

where M_{eff} is the sum of non-zero o_{tm} at the current frame.

Figure 3 shows the result of CG/real image composition results on the first MoCap-video dataset. The left column of that figure shows 2D marker detection results with marker trajectories (yellow lines). We attached six small MoCap markers to props on a desk. For experiment, we purposely shook the camera so fast to acquire fast-motion images. Nevertheless, our method successfully computed all of the parameters of the camera from the MoCap-video data of six MoCap markers. The RMSE for this dataset was 0.46 pixels.

A simple demonstration of the realistic interaction between human and CG creatures, which is a challenging VFX problem in TV and film production, was shown in Fig. 4, where a real actor touched a CG train with a CG axe. Note that the creation of interactive VFX is not a difficult problem in our CG/real image composition system. The number of markers and frames on the second dataset was 13 and 804, respectively. The RMSE for this dataset was 0.44 pixels.

We showed another demonstration of digital special make-up effects with our CG/real image composition system in Figure 5. Note that our system allowed us to use the MoCap data to animate the CG creature and to track the camera motion at the same time, which is the main feature of our MoCap-based camera tracking method. The RMSEs for all dataset were listed in Table 1.

Table 1. Result of Performance Analysis

Dataset	1	2	3	4	5
Number of frames	463	804	960	620	932
Number of markers	6	13	20	9	9
RMSE	0.46	0.44	0.43	0.28	0.35

V. CONCLUSIONS

We presented a MoCap-based camera tracking method to create VFX by composing MoCap-driven CG animation with real on-set images for TV and film production. We showed that our MoCap-based camera tracking method could compute all of the motion and lens parameters of camera from MoCap-video data with sub-pixel accuracy, which was experimentally verified with the performance analysis on several real MoCap-video datasets. In addition, we showed that our CG/real image composition system could use to create several VFX like fast-motion, interaction and digital special make-up.

ACKNOWLEDGMENT

This work was supported by the IT R&D joint program of Ministry of Knowledge Economy / Korea Evaluation Institute of Industrial Technology and Ministry of Culture, Sports and Tourism / Korea Creative Content Agency, [2007-S-051-03, Software Development for Digital Creature].

REFERENCES

- [1] Azuma, R. and Bishop, G., "Improving Static and Dynamic Registration in an Optical See-through HMD", Proc. 21th SIGGRAPH, 1994, pp. 197-204.
- [2] Azuma, R. T., "A Survey of Augmented Reality", Presence: Teleoperators and Virtual Environments, 6(4): 355-385, 1997.
- [3] Azarbayejani, A. and Pentland, A. P., "Recursive Estimation of Motion, Structure, and Focal Length", IEEE Trans. Pattern Analysis and Machine Intelligence, 17(6): 562-575, 1995.
- [4] Davison, A. J., "Real-time Simultaneous Localization and Mapping with a Single Camera", Proc. 9th IEEE Int Conf. Computer Vision, 2003, pp. 1403-1410.
- [5] Zhang, Z., "A Flexible New Technique for Camera Calibration", IEEE Trans. Pattern Analysis and Machine Intelligence, 22(11): 1130-1134, 2000.
- [6] Chiuso, A., Favaro, P., Jin, H., and Soatto, S., "Structure from Motion Causally Integrated Over Time", IEEE Trans. Pattern Analysis and Machine Intelligence, 24(4): 523-535, 2002.
- [7] Vicon Motion Capture System, <http://www.vicon.com>
- [8] Bregler, C., "Motion Capture Technology for Entertainment", IEEE Signal Processing Magazine, 24(6): 160-158, 2007.
- [9] Weng, J., Cohen, P., and Herniou, M., "Camera Calibration with Distortion Models and Accuracy Evaluation", IEEE Trans. Pattern Analysis and Machine Intelligence, 14(10): 965-980, 1992.
- [10] Antoniou, A. and Lu, W.-S., *Practical Optimization: Algorithms and Engineering Applications*, Springer, 2007.
- [11] Hartley, R., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2ed., 2004.
- [12] Autodesk Maya, <http://www.autodesk.com>