An Integrated Environment for Stereoscopic Acquisition, Off-line 3D Elaboration, and Visual Presentation of Biological Actions

Marco Agus, Fabio Bettio, Enrico Gobbetti
CRS4, Visualization and Virtual Reality Group, Cagliari, Italy
{magus,fabio,gobbetti}@crs4.it, http://www.crs4.it/vvr

Luciano Fadiga
Institute of Human Physiology, University of Parma, Italy.
lfadiga@ipruniv.cce.unipr.it

Abstract. We present an integrated environment for stereoscopic acquisition, off-line 3D elaboration, and visual presentation of biological hand actions. The system is used in neurophysiological experiments aimed at the investigation of the parameters of the external stimuli that mirror neurons visually extract and match on their movement related activity.

1 Introduction

In spite of its fundamental role for human/animal behavior, very little is known on how individuals recognize actions performed by others. It has been often proposed that a common code should exist between the observed events and an internally produced motor activity. Neurophysiological evidence in favor of this putative common code was, however, until recently lacking. Recent experiments have shown that neurons located in a monkey premotor area (F5, [1, 2]) are very likely involved in this process [3]. These neurons discharge both when the monkey actively performs goal-directed hand/mouth actions and when it observes them performed by others. Typically, however, there is no discharge when similar actions, identical in terms of goal, are made by manipulated mechanical tools, or when the actions are mimicked without the target object [4]. Such data suggests that actions made by others are represented in those same areas of the premotor cortex where motor primitives for active execution are stored. If one admits that an individual, when making an action, may predict its outcome, it appears likely that she will recognize an action made by others because it evokes a discharge in those same neurons that fire when she makes the identical action. It remains unclear which are the visual features used to match a given seen action on the internal repertoire of motor primitives normally used for action execution. A Human Frontiers Science Program project that brings together a consortium of European, American, and Japanese researchers is investigating this subject in detail. The planned neurophysiological experiments will require the presentation of visual stimuli of different types. To simplify the creation and handling of a catalog of digitized stimuli, we are creating a specialized animation system that enables animators to create short animation sequences which closely follow a video-recorded action, and to later modify the sequences to obtain all the required variations. This paper briefly describes the current system prototype.

The rest of the paper is organized as follows. Section 2 provides a general overview of the system, section 3 concentrates on video acquisition and playback, section 4 details the

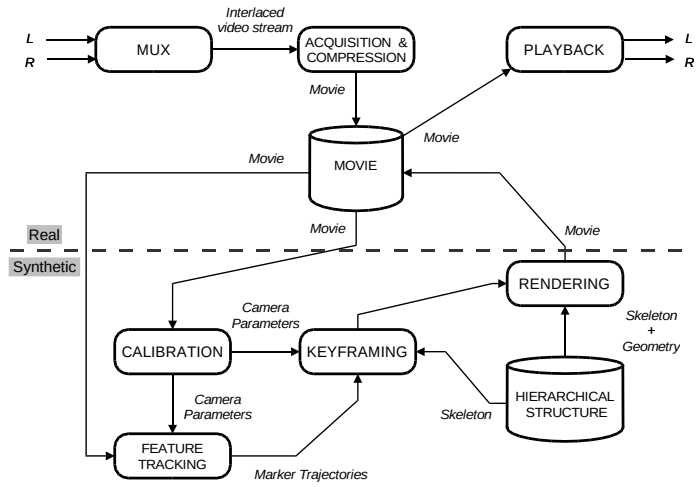

Figure 1: **System overview.** Main data processing components and flow of data among them.

video analysis and animation synthesis subsystem. The paper concludes with a discussion of the results obtained and a view of future work.

2 System Overview

In order to study the encoding of visual stimulus parameters for hand action recognition in monkeys, single neurons will be recorded from premotor area F5 and from the inferior parietal lobule of monkeys trained to fixate a spot of light on a projection screen and to detect its dimming by pressing a lever. During fixation, digitized stimuli showing goal-directed actions will be stereoscopically presented on the same screen by means of polarized light projection.

Two different kinds of stimuli will be presented: direct reproductions of "real" sequences, and elaborations of those sequences, both in terms of modification of visual appearance and/or behavior (i.e. same action with variations in the kinematics and/or in the geometry and material of the performing object). The first situation is handled by components that support recording, storage, and playback of stereoscopic movies, while the second requires components that create artificial sequences starting from the acquired stereo movies. Since the type of operations required (variations in kinematics and/or hand shape) are not easily obtainable by image manipulation, the system needs to work directly in 3D space. In our approach, the original sequence is first reconstructed by animating the degrees of freedom (DOFs) of a virtual 3D hand model. The resulting 3D animation, interactively constructed exploiting a combination of artificial vision and standard key-framing techniques, is then refined, modified, and rendered to produce all the desired variations. Figure 1 provides a general overview of the system, depicting the main data processing components and the flow of data among them.

3 Stereoscopic Video Acquisition and Video Playback

The acquisition part of the system, built using off-the-shelf components, deals with the procedures for the acquisition of a stereoscopic video and its storage on a disk. The selected technical solution uses two video cameras placed on a tripod for stereo recording. The two

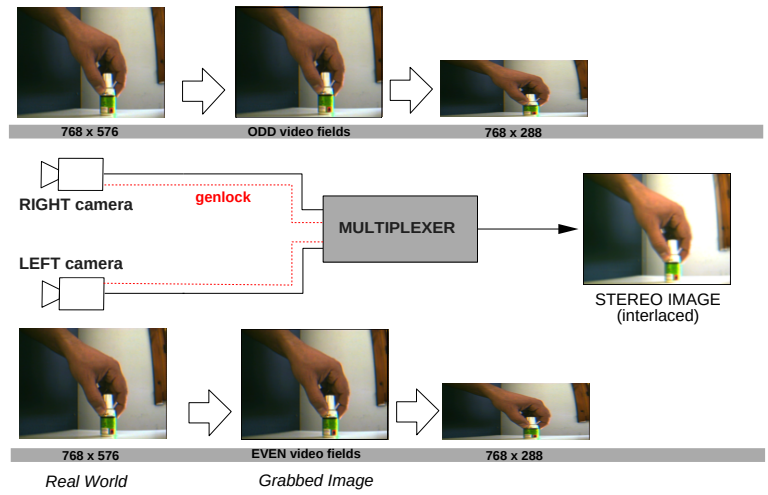

Figure 2: **Stereo acquisition.** The two inputs are synchronized and multiplexed in a single PAL or NTSC stream, which is then fed to the graphics workstation for recording or real-time display.

inputs are synchronized and multiplexed in a single PAL or NTSC stream, which is then fed to a graphics workstation for recording or real-time display (see figure 2). A digital compression card is used for real-time signal compression, enabling direct-to-disk recording. Using M-JPEG compression, a typical stereoscopic sequence of 10 s requires about 20 Mb of storage (PAL format, 50 fields per second, 768x288 pixels per eye). We have successfully run our applications on a Silicon Graphics IMPACT connected to a VREX Cam3C system and on a Silicon Graphics Octane connected to a TD003 3D Multiplexer with two CCD cameras for input.

The playback part of the system deals with the decoding of compressed stereoscopic movies and the stereo visualization of the video acquired. In order to reach high-level performances for stereo visualization, we have decided to completely load and decompress video sequences in memory prior to visualization. During visualization, left and right images are copied to the frame buffer in the format needed for stereo presentation and the graphics card is suitably configured. Shutter or polarized glasses are used for image presentation. Both stereo-in-a-window and full-screen presentations are possible.

4 Video Analysis and Animation Synthesis

The video analysis and animation synthesis subsystem provides tools for creating animations of articulated 3D models that closely follow the actions of a real hand depicted in a stereoscopic movie. In order to reach this goal, we have combined in a single interactive environment the basic features of artificial vision systems, to extract meaningful information from the stereo movie, with the concept of key-framed animation. In our approach, the original sequence is reconstructed by estimating the projection and viewing matrices of the cameras, tracking the 3D trajectory of interesting hand features, and animating the DOFs of a virtual 3D hand model so as to obtain the best match with the recorded movie. The resulting 3D animation is then refined, modified, and rendered using standard editing features to produce all the desired variations. The system has been implemented on a PC platform running Windows NT and requires consumer-level graphics boards (e.g. NVIDIA GeForce based systems).

4.1 Camera Registration

The camera transformation matrices contain all the geometric information (intrinsic and extrinsic camera parameters) which is necessary to calculate 3D coordinates from correspondences between two stereo perspective images of a scene. Since a metric reconstruction of the environment is required, we have implemented a strong calibration system that estimates camera parameters by observing a calibration object whose geometry is known. We currently use a semi-automatic approach in which the user selects a set of at least seven matching points on a camera view and associates them with their known 3D coordinates. The process is applied independently for the left and right camera. Given a set of 2D points $\mathcal{U} = \{\mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_n}\}$ and a set of corresponding 3D points $\mathcal{X} = \{\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}\}$, the camera projection matrix $\mathbf{P}(\theta, \mathbf{x_s}, \mathbf{y_s}, \mathbf{x_c}, \mathbf{y_c})$ and the camera viewing matrix $\mathbf{V}(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_z}, \mathbf{t_x}, \mathbf{t_y}, \mathbf{t_z})$ are estimated by minimizing the following functional:

$$\sum_{i=1}^{n} \left\| \begin{bmatrix} \frac{\mathbf{t_1} \cdot \mathbf{x}_i}{\mathbf{t_3} \cdot \mathbf{x}_i} \\ \frac{\mathbf{t_2} \cdot \mathbf{x}_i}{\mathbf{t_3} \cdot \mathbf{x}_i} \end{bmatrix} - \mathbf{u}_i \right\|^2 \tag{1}$$

where $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ are the rows of $\mathbf{T} = \mathbf{P}(\theta, \mathbf{x_s}, \mathbf{y_s}, \mathbf{x_c}, \mathbf{y_c})\mathbf{V}(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_z}, \mathbf{t_x}, \mathbf{t_y}, \mathbf{t_z})$, θ is the horizontal field of view, x_s, y_s are the dimensions of the image in pixels, x_c, y_c are the pixel coordinates of the center of projection, r_x, r_y, r_z are the Euler angles specifying the camera orientation and t_x, t_y, t_z are the coordinates of the camera position. In our implementation, we use a conjugate gradient algorithm to search for the optimal solution. A series of different starting points for the viewing parameters are generated on the sphere containing the bounding box of the 3D data set. The minimization algorithm is applied for each starting point and the minimum error solution is chosen as the optimal one.

4.2 Feature Tracking

Once the camera transformation matrices are known, the system has all the information to calculate the 3D coordinates of a point from point matches on the left and right images. This is used to help users track interesting features (e.g. finger tips) over time and use them to guide the keyframing process. Reconstructing the 3D position of a point $\mathbf{x}$ from its two projections $\mathbf{u}^{(l)}$ and $\mathbf{u}^{(r)}$ requires the solution of an homogeneous system. Let $\mathbf{T}^{(l)}$ be the left camera transformation matrix and let $\mathbf{T}^{(r)}$ be the right camera transformation matrix. The 3D coordinates of the point $\mathbf{x}$ are the solution of the linear equation $\mathbf{Ax} = \mathbf{0}$ where $\mathbf{A}$ is the 4x4 matrix given by: $\mathbf{A} = \left[\mathbf{t_1}^{(l)} - \mathbf{u_1}^{(l)}\mathbf{t_3}^{(l)}, \mathbf{t_2}^{(l)} - \mathbf{u_2}^{(l)}\mathbf{t_3}^{(l)}, \mathbf{t_1}^{(r)} - \mathbf{u_1}^{(r)}\mathbf{t_3}^{(r)}, \mathbf{t_2}^{(r)} - \mathbf{u_2}^{(r)}\mathbf{t_3}^{(r)}\right]^{\mathbf{T}}$.

This problem is solved using an iterative linear least-squares method [5]. To construct the 3D trajectory of a feature during the animation, the user determines its position in a small set of interesting frames and lets the system build an interpolating 3D spline that provides the position of the feature as a function of time (see figure 3).

4.3 Virtual Hand Positioning and Key-Framing

The key-framing animation system aims at generating synthetic animations by positioning a virtual hand model, specifying the joint values (DOFs) for some time values and interpolating for the others. The appearance and structure of the virtual hand are fixed for a single animation sequence. The model is encoded using a VRML node graph [6]. The structure of the standard hand used for creating the reference animation that matches the video recorded one is based on the one described in the MPEG4 Version 2 (PDAM1) specification [7]. The

system, however, is not restricted to this model and is able to animate objects with arbitrary structure. The model position at a key frame is specified with a 6 DOF input device (Magellan SpaceMouse), and it is also possible to specify each joint angle using both direct and inverse techniques. Our inverse specification module takes as input the marker trajectories and computes the optimal DOF values that permit to reach these goal positions, helping the animator in modeling the key-frames and obtaining more realistic animations. The approach that we used is to let animators freely manipulate the root position using the 6 DOF input device, while the system concurrently selects the optimal joint values by minimizing a cost function. This very general approach has proven very helpful in a number of animation contexts [8]. Reducing the number of DOFs under direct animator control in the generation of hand animations is also motivated by the fact that the mechanical structure of the hand introduces constraints that reduce the ability to control hand joints independently (see [9] for a task-level computer animation application). We currently use the following cost function:

$$Cost(\varphi) = \sum_{\mathbf{i}} \|\mathbf{P^i_{tip}}(\varphi) - \mathbf{P^i_{goal}}\|^{\mathbf{2}} + \sum_{\mathbf{j}} barrier(\varphi_j) + \sum_j \frac{1}{\sigma_j^2}\|\varphi_j - \hat{\varphi_j}\|^2 \qquad (2)$$

where $\varphi = \{\varphi_{\mathbf{1}}, \varphi_{\mathbf{2}}, ..., \varphi_{\mathbf{n}}\}$ is the joint value vector to determinate, P^i_{goal} is the goal position for the i-th finger tip and $P^i_{tip}(\varphi)$ is the current position of the i-th finger tip, function of DOF vector φ. The first component drives the tips of the hand fingers (or other selected points in the local reference frame of the hand) towards the associated goal positions. The second component implements joint value constraints by summing the penalty functions associated to joint limits. The limit function increases as joint angle φ_j approaches either its maximum or minimum limit and is zero otherwise. This strongly inhibits each joint from bending beyond its prescribed limits. One such function which enforces joint limits is the following:

$$barrier(\varphi_j) = \begin{cases} -\frac{\varphi_j - (\varphi_j^{\min} + \alpha)}{\alpha} + \ln\left(\frac{\delta}{\alpha}\right), & \varphi_j < \varphi_j^{\min} + \alpha \\ \ln\left(\frac{\delta}{\varphi_j - \varphi_j^{\min}}\right), & \varphi_j^{\min} + \alpha < \varphi_j < \varphi_j^{\min} + \delta \\ 0, & \varphi_j^{\min} + \delta < \varphi_j < \varphi_j^{\max} - \delta \\ \ln\left(\frac{\delta}{\varphi_j^{\max} - \varphi_j}\right), & \varphi_j^{\max} - \delta < \varphi_j < \varphi_j^{\max} - \alpha \\ \frac{\varphi_j - (\varphi_j^{\max} - \alpha)}{\alpha} + \ln\left(\frac{\delta}{\alpha}\right), & \varphi_j > \varphi_j^{\max} - \alpha \end{cases} \qquad (3)$$

where φ_j is the angle of the current j-th joint , δ is the angular distance from the limits at which the limit function becomes non-zero, and α is the angular distance at which the logarithmic barrier is substituted with a linear barrier for numerical reasons. The third component drives each joint value φ_j towards a rest position $\hat{\varphi_j}$ and is weighted by a sensitivity parameter σ_j. This parameter approximates how much the j-th joint variation influences the correspondent tip position changes. A gradient descent algorithm is used to modify the joint angles of the model in order to drive the equation 2 to a minimum. The algorithm used is adaptive in the sense that it computes each descent steps using golden search bracketing and line search algorithms [10]. Figure 3 illustrates key-frame editing using constrained manipulation. Direct and inverse (constrained) techniques are usually iteratively applied until the animator judges the matching between the pose and the original movie sufficiently precise. Once the desired pose is found, all the joint values are stored in the key-framing system so as to participate as control points in the animation. This is repeated for a number of key-frames, until the animator is satisfied with the resulting animation.

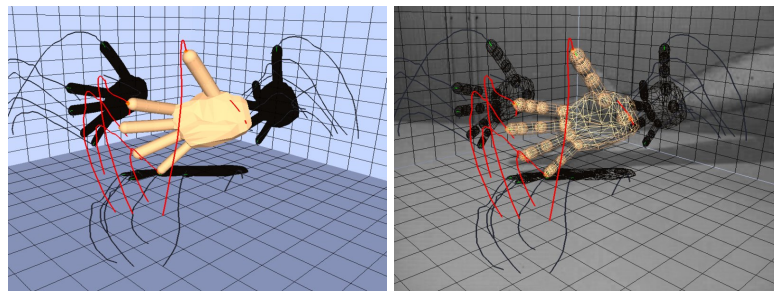

Figure 3: **Tracking features over time with constrained manipulation.** Lines represent the trajectories of selected features for the duration of the movie, while dots represent the position of the markers in this particular frame. The optimization algorithm computes optimal values for joint angles, while the user controls the hand wrist using the SpaceMouse. In the left image, orthogonal projections are used to guide interactive positioning, while in the right image the movie frame is also displayed in the background.

4.4 Animation Editing

In order to get meaningful modifications of sequences, the system makes it possible to interactively manipulate the animation. These manipulations involve geometry and appearance modifications (i.e. the system is able to apply the same animation data to various models, that differ in geometry and appearance) and time warping (i.e. the system provides tools that enable the animator to modify the animations by applying different time functions to joint angles). Currently, geometry and appearance modification is obtained by re-using the same animation tracks with different articulated structures (i.e. different VRML files) containing the same DOFs, while time warping features are limited to global scaling. More sophisticated motion retargeting techniques (e.g. those presented in [11, 12]) will be explored in the future.

5 Conclusions and Future Work

Our integrated system combines stereoscopic acquisition and presentation of biological actions in an experimental environment with the possibility to manipulate the acquired data for generating pictorial and/or kinematics modifications of the original action. The modified actions, successively presented in discrimination tasks in humans or during single neurons recording experiments in monkeys, will allow neurophysiology researchers to better understand how the brain works in understanding actions made by others. The first movies to be used as digital stimuli in the neurophysiological experiments are in the production phase. Figure 4 shows selected frames of three different versions of a grasping sequence. The future work will concentrate on improving the animation manipulation and motion warping features, and in producing higher-quality renderings.

Acknowledgments

This work is carried out within the research project “MIRRORS: Mirror Neurons and the Execution/Matching System in Humans”, sponsored by the Human Frontiers Science Program Foundation. The project partners are: Università di Parma, Italy (Coord.); CRS4, Cagliari, Italy; Helsinki University of Technology, Espoo, Finland; Kyoto University, Inuyama, Japan; Dept. of Basic Sciences, University of Crete Medical School and IACM FORTH, Heraklion, Greece; University of California School of Medicine, Los Angeles, USA. We also acknowledge the contribution of Sardinian regional authorities.

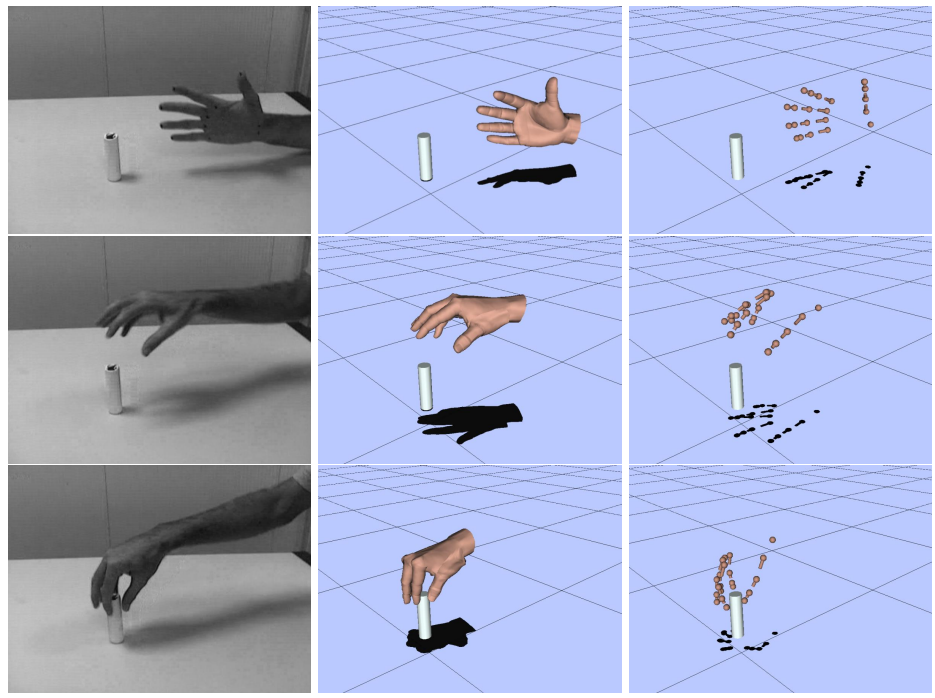

Figure 4: **Animation results.** The left images show selected frames of a video-taped grasping sequence. The middle images shows a synthesized animation replicating the same motion. The right images shows the same animation applied to a different geometric structure.

References

[1] M.Matelli, G. Rizzolatti, and G.Luppino. Patterns of cytochrome oxidase activity in the frontal agranular cortex of the macaque monkey. *Behav. Brain Res.*, 18:125–136, 1985.

[2] G. Di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Exp. Brain Res.*, 91:176–180, 1992.

[3] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Behav. Brain Res.*, 3:131–141, 1996.

[4] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119:593–609, 1996.

[5] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis Cedex, France, July 1996.

[6] Rikk Carey and Gavin Bell. *The VRML 2.0 Annotated Reference Manual*. Addison-Wesley, Reading, MA, USA, January 1997. Includes CD-ROM.

[7] ISO/IEC 14496-1: MPEG-4 PDAM1. Available on the web at the address http://www.cslet.stet.it/mpeg.

[8] David E. Breen. Cost minimization for animated geometric models in computer graphics. *The Journal of Visualization and Computer Animation*, 8(4):201–220, 1997. ISSN 1049-9807.

[9] Hans Rijpkema and Michael Girard. Computer animation of knowledge-based human grasping. *Computer Graphics*, 25(4):339–348, July 1991.

[10] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, second edition, 1992.

[11] Andrew Witkin and Zoran Popović. Motion warping. *Computer Graphics*, 29(Annual Conference Series):105–108, November 1995.

[12] Kwangjin Choi and Hyeongseok Ko. On-line motion retargetting. In *Proceedings of the International Pacific Graphics '99*, 1999.