Real-time and Markerless Full-Body Human Motion Capture

Brice Michoud[1], Erwan Guillou[1] and Saïda Bouakaz[1]

forename.name@liris.cnrs.fr
Université Claude Bernard Lyon 1 - Liris
8 Boulevard Niels Bohr, 69622 Villeurbanne, France

Abstract. We propose an efficient real-time method for markerless 3D human motion capture that requires a single computer. Using input from at least three calibrated webcams, an extended "Shape from Silhouette" algorithm reconstructs the filmed person in real-time. Fasts 3D shape and 3D skin parts analysis provide a robust and real-time system for human full-body tracking. Animation skeleton and simple morphological constraints ease the motion capture process. Thanks to fast and simple algorithms, and appliance to low-cost cameras, our system is well suited for home entertainment device. Results on long video sequences with fast and complex movements, demonstrate our approach robustness.

1 Introduction

Marker-free motion capture has long been studied in computer vision as classic and fundamental problems. While commercial products are already available for real-time marker-use, robust on-line marker-free systems remains an open issue because many real-time algorithms still lack robustness. While most popular techniques run on pc cluster, our system require at least three low-cost cameras (as webcams) and a single computer. In this paper we propose a fully automated human-machine interaction device for home entertainment (see Fig. 2(a)). Because interactions constraint, our system works in real-time (at least 30 fps), without markers (active or passive) or any particular sensors.

To tackle the marker-free motion capture problem, several techniques has been proposed (according to camera numbers and feature analysis). We only review some works related to our work. Among various types of approaches many methods work with a single camera [1, 2]. When the method is based on the object silhouette, they suffers from ambiguous response in case of local minimum as different positions can yield the same silhouette. Other methods use multiple cameras. Some of them work only on a 3D human shape analysis [3, 4]. These techniques provide good results when the 3D shape topology correspond to the filmed human topology *i.e.* each body parts is clearly identifiable in the estimated 3D shape. With self-occlusion cases or large contacts between limbs and body these techniques frequently fail. Caillette *et al.* [5] method involves shape and color clues. They link colored blobs to a kinematic model to track individual body parts. This technique requires contrasted clothing between each body parts for tracking, thus adding an usability constraint. Few methods provide real-time motion capture. Most of them run only with interactive frame rate (10 fps for [5]).

We therefore propose a fully automated system for practical real-time motion capture from at least three calibrated webcams. Our method runs at 30 fps because, it is based on simple heuristics, driven by shape and skin parts topology analysis, and temporal coherence.

Figure 1(a) outlines the two main stages of our method : 3D reconstruction and analysis of the 3D information. In the first section we present our work for real-time 3D reconstruction. We explain in the second section our method for real-time full-body pose estimation Then, we discuss on the results obtained from real and complex data. Finally, we conclude about our contributions, and we present some perspectives for this work.

2 3D shape and skin parts estimation

We propose extensions for "Shape From Silhouette" (SFS) algorithms, which reconstruct in real-time 3D shape and 3D skin color parts of the person from calibrated cameras.

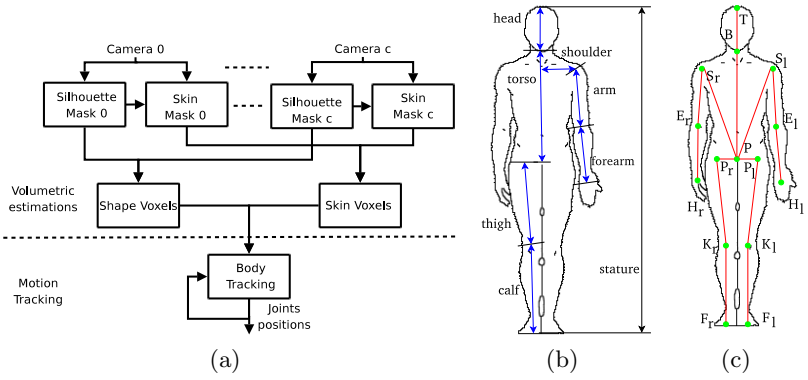


Fig. 1. (a) System overview: Reconstruction algorithms and Pose estimation algorithms. Body parts labeling (b) and joint naming (c).

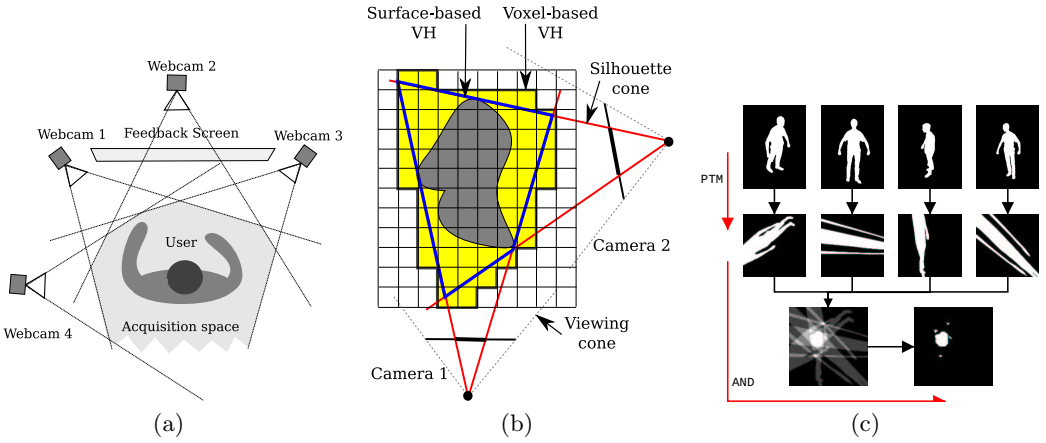


Fig. 2. (a) Interaction setup. (b) Object reconstruction by surface and volumetric approaches represented in 2D. SFS computation using "Projective Texture Mapping" method: First all silhouette masks are projected on a stack, logical AND is used to compute the projection intersection.

Currently only SFS methods compute in real-time 3D shape estimation of an object, from its silhouette images. Silhouette images are binary masks corresponding to captured images where 0 correspond to background, or 1 stands for the (interesting) feature of the object. The formalism of SFS was introduced by A. Laurentini [6]. By definition, an object lies inside the volume generated by back-projecting its silhouette through the camera center (called silhouette's cone). With multiple views of the same object at the same time, the intersection of all the silhouette's cones build a volume called "Visual Hull", which is guaranteed to contain the real object. There are mainly two ways to compute an object Visual Hull.

Surface-based approaches Surface-based approaches compute silhouette cone surface intersections (see Fig. 2(b)). First silhouettes are converted to polygons. Each edge is back-projected to form a 3D polygon. Then each 3D polygon is projected onto each other images, and intersected with each silhouette in 2D. The resulting polygons are assembled to form polyhedral shape estimation (see [7, 8]). Resulting Surface-based shape from silhouette is underlined Fig. 2(b). These approaches are not well suited to our application because of the complexity of the underlying geometric calculations that are not real-time on a single computer. Incomplete or corrupted surface models can be created, directly depending on polyhedron sharpness and silhouette noise.

Volumetric-based approaches Volumetric-based approaches [3, 9–11] generally estimate shape by processing a set of voxels. The object acquisition area is split up into a 3D grid of voxels (volume elements). Each voxel remains part of the estimated shape if it projection in all images lies in all silhouettes (see Fig. 2(b)). This volumetric approach is adapted for real-time pose estimation, due to its fast computation and robustness to noisy silhouettes.

We propose a new framework which computes a 3D volumetric shape and skin parts estimation on a single computer. After 3D Shape from silhouette estimation on GPU, we compute voxels visibility. Then all visible voxel which project themselves on skin masks, are then classified as skin voxels. First we explain camera calibration, silhouette segmentation and skin segmentation steps, which are input data for 3D estimations. Then GPU SFS implementation, voxel visibility computation, and skin voxel computation are presented in second part.

2.1 Input Data

First, webcams are calibrated using the method proposed by Zhang *et al.* [12] which is one of the most popular calibration algorithm. A Color calibration step is added to enforce coherency between the two webcams using the method proposed by N.Joshi [13].

Second step consists in silhouette segmentation (see [14] for silhouette segmentation algorithm comparative study). Then we assume that the background is static and the subject moves. We use the method proposed by [9]. In beginning we acquire images of background (without user). The user is then detected in the pixels whose value has changed. By hypothesis only one person is in the field of view of webcams, then it is represented only by one connex component. Due to webcam noise, we can have several connex parts, but the smallest are considered as noise.

Last step before voxels computation, we extract skin parts from silhouettes and color images. Normalized Look-up Table method [15] provides fast skin color segmentation. This segmentation is applied to each images limited to silhouette mask because skin color pixels outside to the silhouette correspond to background pixels.

2.2 GPU SFS implementation

Volumetric SFS is generally based on voxel projection: a voxel remains part of the estimated shape if it projects itself into each silhouette. To better fit a GPU implementation we choose the opposite: we project each silhouette into the 3D voxel grid as proposed in [9]: if a voxel is intersected by all the silhouettes projections, then it represents the original object. The classical N^3 voxel cube can be considered as a stack of N images of resolution $N \times N$. We stack the N image in screen parallel planes. For each camera view, silhouette masks are projected on each slice using the "projective texture mapping" technique [16]. Intersection of silhouettes projections on all slices provides voxel-based 3D shape. Intersection of silhouette mask projections on a single slice is underlined Fig 2(c). To save video bus bandwidth, computations for a voxel cube are made in the same frame buffer, which is tiled by all the N slices of resolution $N \times N$.

To estimate skin voxels, we compute each voxel visibility from each camera. The voxel visibility is based on Item Buffer method used in some voxels coloring algorithms [17]. An unique identifier is associated to each voxel (like color) and voxels are rendered on raster based frame buffers, corresponding to each cameras views. For each frame buffers, colors describes visible voxels and this enable bidirectional pixel to voxel mapping. If a voxel is skin consistent (*i.e.* it is mapped to skin mask pixels in all of its viewing camera) then it is classified as skin voxel. To improve visibility computation time, only surface voxels (*i.e.* voxels which have less than 26 neighbors) are tested.

To reduce computation time for pose estimation we propose to keep the visible voxels. Let $\mathcal{V}_{\overline{\text{skin}}}$ be the the selected voxels form shape voxel set, $\mathcal{V}_{\text{skin}}$ be the skin consistent voxel set, and $\mathcal{V}_{\text{all}}$ be their union.

Our implementation provides up to 100 reconstructions per second. As webcam acquisition is done at 30 fps, it allows us to save time for motion capture calculus, hence achieving our real-time goal.

3 Motion Capture

Motion capture is equivalent to determine the pose of the body. It may be seen as classifying each voxel to a body part. Joints labeling is presented in Figure 1(c). We propose a system based on simple and fast heuristics. Less accurate than the registration based methods, this approach nonetheless run at real-time. Robustness is increased by using a multi-modal scheme composed on both shape and skin parts analysis, temporal coherence and human anthropometric constraints.

Our system runs on two steps: initialization and tracking; both use the same algorithm with different initial conditions. The initialization step presented section 5 estimates anthropometric values, and the initial pose. Then using this information, the second step tracks joint positions (see section 4). The only assumption is that both hands and person's face are partially uncovered, that the torso is dressed, and that the clothing have a non-skin color. We present here some common notations that the reader can refer to:

L_{x} denotes the length of body part x (see Fig. 1(b)),
D_{x} its orientation and
R_{x} its radius (of sphere or cylinder).
J^n denotes the value of a quantity J (joint position, voxel set...) at frame n.
l and r indices denote respectively left and right side
$\mathcal{V}_x$ denotes a set of voxel,
$\mathcal{E}_{\mathcal{V}_x}$ its inertia ellipsoid and
$Cog(\mathcal{V}_x)$ its gravity center.
$J(i)$ denotes the J quantity value at step i when dealing with iterative algorithms.

4 Body Parts Tracking

To achieve body parts tracking we assume to be known the previous body pose and anthropometric estimations. Using 3D shape estimation and 3D skin parts we track the human body parts in real-time. The tracking process works on active voxels $\mathcal{V}_{\mathrm{act}}$. This set of voxels is initialized to all voxels $\mathcal{V}_{\mathrm{all}}$ and updated at each step by removing voxels used to estimate body parts. First we estimate head joints. Torso is connected to head, then we next track the torso. In the end we compute limb joints that are connected to torso.

4.1 Head Tracking

This step aims to find T^n and B^n, the positions of the top of the head and the connection point between head and neck at frame n.

Let $\mathcal{V}^n_{\mathrm{face}}$ be the face's voxels at the current frame. By hypothesis $\mathcal{V}^n_{\mathrm{skin}}$ contains face and hands voxels. Using Temporal coherency criteria $\mathcal{V}^n_{\mathrm{face}}$ is the nearest connex component of $\mathcal{V}^n_{\mathrm{skin}}$ from the previous set of face voxels $\mathcal{V}^{n-1}_{\mathrm{face}}$.

The center of the head C^n is computed by fitting a sphere $S(i)$ in $\mathcal{V}^n_{\mathrm{act}}$ (see figure 3). $S(i)$ is defined by its center $\mathrm{C}^n(i)$ and radius R_{head}.

head fitting algorithm $\mathrm{C}^n(0)$ is initialized as the centroid of $\mathcal{V}^n_{\mathrm{face}}$.

At step i of the algorithm, $\mathrm{C}^n(i)$ is the centroid of the set $\mathcal{V}^n_{\mathrm{head}}(i)$ of active voxels that lie into a sphere $\mathcal{S}(i-1)$ defined by its center $\mathrm{C}^n(i-1)$ and its radius R_{head} (see Fig. 3(a)).

The algorithm iterates until step k when the position of C^n stabilizes, *i.e.* the distance between $\mathrm{C}^n(k-1)$ and $\mathrm{C}^n(k)$ falls below a threshold ϵ_{head}.

head joints estimation Knowing C^n position, B^n (respectively T^n) is computed as the lower (resp. upper) intersection between $\mathcal{S}(k)$ and the principal axis of $\mathcal{E}_{\mathcal{V}^n_{\mathrm{head}}}$ (see Fig. 3(b)).

The back-to-front direction D^n_{b2f} is defined as the direction from C^n towards the centroid of $\mathcal{V}^n_{\mathrm{face}}$ (note that voxels from the back of the head are not in $\mathcal{V}_{\mathrm{skin}}$). At this point, we remove from $\mathcal{V}^n_{\mathrm{act}}$ the set of elements that belongs to $\mathcal{V}^n_{\mathrm{head}}$.

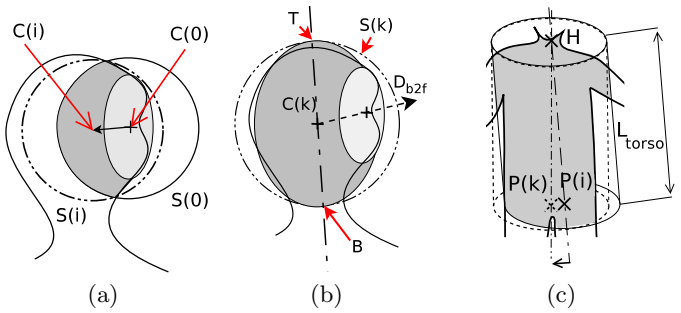


Fig. 3. (a) Sphere fitting (light gray denotes $\mathcal{V}^n_{\text{face}}$, dark gray denotes $\mathcal{V}^n_{\text{head}}(i)$), (b) joints estimation and (c) torso segmentation by cylinder fitting.

4.2 Torso Tracking

This step aims to find P^n the pelvis position, by fitting a cylinder in $\mathcal{V}^n_{\text{act}}$. Torso shape estimated by a cylinder provides simple and fast pelvis localization method. Let $\mathcal{V}^n_{\text{torso}}$ bet the set of voxels that describes the torso, they are initialized using voxels $\mathcal{V}^n_{\text{act}}$. At step i, the algorithm estimates D^n_{torso} by fitting a cylinder $\mathcal{CYL}(i-1)$ in $\mathcal{V}^n_{\text{torso}}$(i) (see Fig 3(c)). $\mathcal{CYL}(i)$ has a cap anchored at B^n, as radius R_{torso}, its length is L_{torso} and its axis is D^n_{torso}(i).

Torso fitting algorithm $\mathcal{V}^n_{\text{torso}}(0)$ is initialized with $\mathcal{V}^n_{\text{act}}$ and the vector from B^n to P^{n-1} define $D^n_{\text{torso}}(0)$ initial value.

At step i, $\mathcal{V}^n_{\text{torso}}$(i) is computed as the set of elements from $\mathcal{V}^n_{\text{torso}}$(i-1) that lie in $\mathcal{CYL}(i-1)$. D^n_{torso}(i) is then the principal axis of $\mathcal{E}_{\mathcal{V}^n_{\text{torso}}(i)}$ (see Fig. 3(c)).

The algorithm iterates until step k when the distance between the axis of $\mathcal{CYL}(k)$ and the centroid of $\mathcal{V}^n_{\text{torso}}$(k) falls below a threshold ϵ_{torso}. P^n position is defined as the center of the lower cap of $\mathcal{CYL}(k)$

Global body orientation The top-down orientation D^n_{t2d} of the acquired subject is given by $\text{P}^n - \text{B}^n$. D_{b2f} was computed in 4.1. The left-to-right orientation D^n_{l2r} of the acquired subject is given by $D^n_{\text{l2r}} = D^n_{\text{t2d}} \times D^n_{\text{b2f}}$.

$\mathcal{V}^n_{\text{act}}$ is then updated by removing its elements that belongs to $\mathcal{V}^n_{\text{torso}}$.

4.3 Hands and forearms Tracking

We propose a simple and robust algorithm to compute the forearms joints positions. First we compute hands position from skin voxels. Helped by given anthropometric measurement of forearm length, we determine the elbows positions. Temporal coherence is used to compute their sides.

Let $\mathcal{V}^n_{\text{hand}}$ be the set of potential voxels of hands. $L_{\text{stat}}/2$ is a raising of arm length. $\mathcal{V}^n_{\text{hand}}$ is defined by the voxels of $\mathcal{V}^n_{\text{skin}} - \mathcal{V}^n_{\text{face}}$ that lie within a sphere defined by its center B^n and its radius $L_{\text{stat}}/2$. By hypothesis $\mathcal{V}^n_{\text{skin}}$ contains hands and face voxels. The different forearms configurations are:

Two distinct hands : $\mathcal{V}^n_{\text{hand}}$ contains several connex components. Let $\mathcal{V}^n_{\text{hand}0}$ and $\mathcal{V}^n_{\text{hand}1}$ be the two biggest, corresponding to the two hands with $\text{H}^n_x = Cog(\mathcal{V}^n_{\text{hand}x})$ with $x \in [0, 1]$.

Forearms have constant length L_{farm} across time. The potential voxels for forearm$_x$ are the voxels from $\mathcal{V}^n_{\text{act}}$ which lies within a sphere of radius L_{farm}, centered in H^n_x. The connex component of these voxels which contains H^n_x represents the forearm$_x$. Let $\mathcal{V}^n_{\text{farm}x}$ be this connex component; there are two possible cases to identify elbow.

If forearms did not collide *i.e.* $\mathcal{V}^n_{\text{farm}0} \cap \mathcal{V}^n_{\text{farm}1} = \emptyset$, then we use the principal axis of $\mathcal{E}_{\mathcal{V}^n_{\text{farm}x}}$ and L_{farm} to compute the elbow position E^n_x. The sides are computed using temporal coherence

criteria: the side of the forearm$_x$ is the same than the closest forearm computed at the previous frame.

Else forearms collide and $\mathcal{V}^n_{\text{farm}0} \cap \mathcal{V}^n_{\text{farm}1} \neq \emptyset$. First we identify the hand sides by the property of constant forearms length. H^n_x is right sided if

$$||d(\text{H}^n_x, \text{E}^{n-1}_r) - L_{\text{farm}}|| < ||d(\text{H}^n_x, \text{E}^{n-1}_l) - L_{\text{farm}}||$$

else H^n_x is left sided. The voxels v_i of $\mathcal{V}^n_{\text{farm}0} \cup \mathcal{V}^n_{\text{farm}1}$ are segmented in two parts $\mathcal{V}^n_{\text{farm}r}$ and $\mathcal{V}^n_{\text{farm}l}$ using point to line mapping algorithm (see 4.5). If v_i is more close to $[\text{H}^n_r \text{E}^{n-1}_r]$ than $[\text{H}^n_l \text{E}^{n-1}_l]$, v_i is added on $\mathcal{V}^n_{\text{farm}r}$. Else v_i is added on $\mathcal{V}^n_{\text{farm}l}$. Principal axis of $\mathcal{E}_{\mathcal{V}^n_{\text{farm}r}}$,$\mathcal{E}_{\mathcal{V}^n_{\text{farm}l}}$ and L_{farm} are used to compute E^n_r and E^n_l.

One hand or jointed hands : $\mathcal{V}^n_{\text{hand}}$ contains only one connex component and it corresponds to jointed hands or to only one hand (the other is not visible). We use the temporal coherence to disambiguate these two cases.

If $\text{H}^{n-1}{}_r$ and $\text{H}^{n-1}{}_l$ are close to $\mathcal{V}^n_{\text{hand}}$, then the hands are jointed and $\text{H}^n{}_r = \text{H}^n{}_l = Cog(\mathcal{V}^n_{\text{hand}})$ and we compute $\mathcal{V}^n_{\text{farm}}$ as proposed previously. We segment $\mathcal{V}^n_{\text{farm}}$ in two parts $\mathcal{V}^n_{\text{farm}r}$ and $\mathcal{V}^n_{\text{farm}l}$ by the orthogonal plane to $[\text{E}^{n-1}_r \text{E}^{n-1}_l]$ containing $\text{H}^n{}_l$. Principal axis of $\mathcal{E}_{\mathcal{V}^n_{\text{farm}r}}$, $\mathcal{E}_{\mathcal{V}^n_{\text{farm}l}}$ and L_{farm} are used to compute E^n_r and E^n_l.

Else the closest hand $\text{H}^{n-1}{}_x$ to $\mathcal{V}^n_{\text{hand}}$ is used to compute the side of $\text{H}^n{}_x$ and $\text{H}^n{}_x = Cog(\mathcal{V}^n_{\text{hand}})$. We compute $\mathcal{V}^n_{\text{farm}}$ as proposed previously and its principal axis of inertia is used to compute $\text{E}^n{}_x$.

No visible hand : $\mathcal{V}^n_{\text{hand}}$ is empty, then no hand is visible. We take back the positions computed at the $n-1$ frame to the current frame.

In all case $\mathcal{V}^n_{\text{act}}$ is updated by removing its elements that belongs into forearm or hand.

4.4 Shoulders Tracking

We have estimated articulations positions of the head, the torso, the hands and the elbows. To finalize upper body tracking, we compute shoulders positions. As we argue that arms are in a sphere centered on bottom head, with a radius of $L_{\text{stat}}/2$, then voxels of $\mathcal{V}^n_{\text{act}}$ which are in this sphere, contain arms voxels and noise voxels. Let $\mathcal{V}^n_{\text{arms}}$ be the set these voxels.

Elbow is on one extremity of arm, then the second estimates shoulder. We know the current position of elbow, then we determine arm voxels. Let $\mathcal{V}^n_{\text{arm}x}$ (where x corresponds to the side) be the closest [1] connex component of $\mathcal{V}^n_{\text{arms}}$ to $\text{E}^n{}_x$. Furthermore arm length L_{arm} is constant, then current shoulder position $\text{S}^n{}_x$ for the x side is given by:

$$\text{S}^n{}_x = \text{E}^n{}_x + \frac{Cog(\mathcal{V}^n_{\text{arm}x}) - \text{E}^n{}_x}{|Cog(\mathcal{V}^n_{\text{arm}x}) - \text{E}^n{}_x|} L_{\text{arm}}$$

$\mathcal{V}^n_{\text{act}}$ is updated by removing its elements that belongs into each arm.

4.5 Legs Tracking

All body parts but the legs have been estimated, hence $\mathcal{V}^n_{\text{act}}$ contains only the legs voxels. Our leg joints extraction is inspired from "point to line mapping" process used to bind an animation skeleton on a 3D mesh [18]. The elements of $\mathcal{V}^n_{\text{act}}$ are split up into four sets $\mathcal{V}^n_{\text{thigh}_l}$, $\mathcal{V}^n_{\text{calf}_l}$, $\mathcal{V}^n_{\text{thigh}_r}$ and $\mathcal{V}^n_{\text{calf}_r}$ depending of their euclidean distance to segments $[\text{P}^{n-1}_l, \text{K}^{n-1}_l]$, $[\text{K}^{n-1}_l, \text{F}^{n-1}_l]$, $[\text{P}^{n-1}_r, \text{K}^{n-1}_r]$, and $[\text{K}^{n-1}_r, \text{F}^{n-1}_r]$ (see Fig. 4(a)). For the left/right side x, we compute the inertia ellipsoid $\mathcal{E}_{\mathcal{V}^n_{\text{calf}x}}$ (let Ex_0 and Ex_1 be its extrema points) and the inertia ellipsoid $\mathcal{E}_{\mathcal{V}^n_{\text{thigh}x}}$.

The knee is the intersection point of thigh and calf (Fig. 4(b)), hence the foot position $\text{F}^n{}_x$ is given by the farthest extrema point of $\mathcal{E}_{\mathcal{V}^n_{\text{calf}x}}$ from the inertia ellipsoid of $\mathcal{V}^n_{\text{thigh}x}$ (let say it's Ex_1). Then knee is aligned on $[Ex_0 Ex_1]$, Ex_0 sided, at a L_{calf} distance of $\text{F}^n{}_x$. Hip position $\text{P}^n{}_x$ is given by the farthest extrema point of $\mathcal{E}_{\mathcal{V}^n_{\text{thigh}x}}$ from the inertia ellipsoid of $\mathcal{V}^n_{\text{calf}x}$, corrected to be at a L_{thigh} distance of $\text{K}^n{}_x$.

[1] In term of euclidean distance

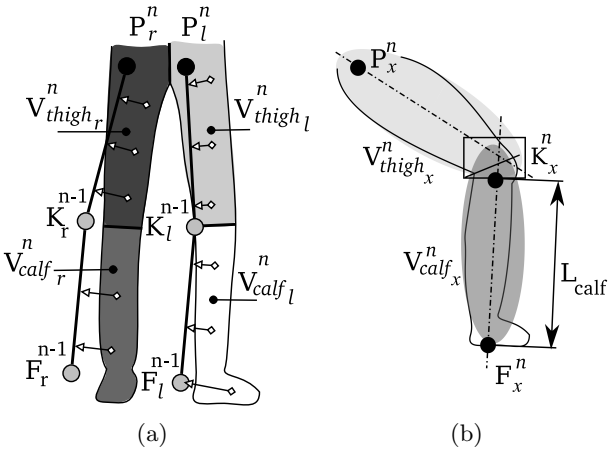


Fig. 4. (a) torso segmentation by cylinder fitting. (b) the "binding" step of legs tracking and (e) legs articulations estimation.

5 Body Parts Initialization

We present in this section our techniques to estimate the anthropometric measures and the initial body pose. The literature in connection with this step can be classified in three categories. In the first one [7], the dimensions and initial pose are manually specified. Second kind of methods need of an initialization pose like T-pose [19]. These methods are real-time. The third class is composed by fully automated methods [3] which are generally non real-time processes. Our approach is real-time and fully automated for any kind of movements as long as the filmed person is standing up, his/her hands are below the level of the head, and his/her feet are not joined. After anthropometric estimations, our method computes each body parts parameters sequentially with the tracking step ordering.

Anthropometric Measurements They correspond to lengths of each body parts[20]. We have estimated some anthropometric measures as average ratios of the human body length. Let L_{stat} be the acquired human body length, estimated as the maximum distance of foreground voxels to floor plane. Hence, knowing L_{stat}, guesses for anthropometric measures are given by these ratios:

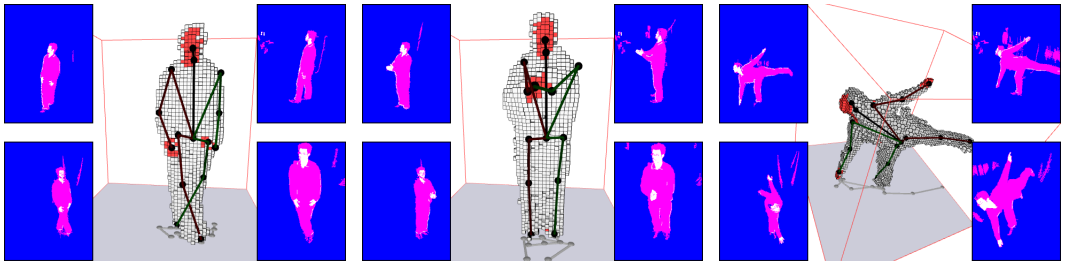
$$R_{\text{head}} \approx L_{\text{stat}}/16 \; L_{\text{torso}} \approx 3L_{\text{stat}}/8 \; L_{\text{calf}} \approx L_{\text{stat}}/4$$
$$L_{\text{farm}} \approx L_{\text{stat}}/6 \quad L_{\text{arm}} \approx L_{\text{stat}}/6 \quad L_{\text{thigh}} \approx L_{\text{stat}}/4$$

Like for tracking step, active set of voxels $\mathcal{V}_{\text{act}}$ is initialized by all voxels $\mathcal{V}_{\text{all}}$.

Head Initialization This step aims to find T^0 and B^0. From our initialization hypothesis, the face's voxels $\mathcal{V}^0_{\text{face}}$ of acquired subject are defined by the top most connex component among $\mathcal{V}^0_{\text{skin}}$. Then Head Tracking algorithm (section 4.1) is applied to compute T^0 and B^0, without estimation of the face position step. $\mathcal{V}^0_{\text{act}}$ is updated by removing elements that belongs to $\mathcal{V}^0_{\text{head}}$.

Torso Initialization The torso fitting algorithm (section 4.2) is applied using $\mathcal{V}^0_{\text{act}}$ as initial value for $\mathcal{V}^0_{\text{torso}}(0)$.$D^0_{\text{torso}}(0)$ is initialized as the vector from N^0 toward the centroid of $\mathcal{E}_{\mathcal{V}^0_{\text{act}}(0)}$. Pelvis position P^0, D^0_{t2d} and D^n_{l2r} are then computed. $\mathcal{V}^0_{\text{act}}$ is updated by removing the elements that belongs to $\mathcal{V}^0_{\text{torso}}$.

Arms Initialization We initialize hands and forearms positions using the tracking algorithm presented Section 4.3. We have no previous arms position, then we can only compute forearms positions when there is two distinct forearms. Having this criteria verified, we can compute ${\text{H}^0}_r$, ${\text{H}^0}_l$,${\text{E}^0}_l$ and ${\text{E}^0}_l$. $\mathcal{V}^0_{\text{act}}$ is updated by removing its elements that belongs into forearms. Shoulders



(a) (b) (c)

Fig. 5. (a) (b) and (c) underline results for challenging poses. The user recovered pose is presented as an animation skeleton having right sided parts in red and left sided parts in green. Shape voxels are presented in white and skin voxels in red.

positions ${S^0}_r$ and ${S^0}_l$ are initialized using directly the shoulder tracking algorithm presented part 4.4.

Legs Initialization Tracking algorithm outlined Section 4.5 need of legs previous positions. We simulate them by a coarse estimation of knees, feet and hips articulations, then we compute more precise position of the legs articulations using the legs tracking algorithm.

$\mathcal{V}^0_{\text{act}}$ contains the voxels that haven't been used for any other parts of the body. First we compute the set of connex components from elements of $\mathcal{V}^0_{\text{act}}$ having their height below $L_{\text{stat}}/8$. If there is less than 2 connex components, we assume that feet are joined and can't be distinguished. Otherwise we use the two major connex components $\mathcal{V}^0_{\text{foot}l}$ and $\mathcal{V}^0_{\text{foot}r}$. Left and right assignation of voxel's set is done using the left-to-right vector D_{l2r}. For the left/right side x, let v_x be the vector from P^0 to the centroid of $\mathcal{V}^0_{\text{foot}x}$. Knee and Foot joints are guesses using the following equations:

$$ {\text{K}^{-1}}_x = \text{P}^0 + v_x \frac{L_{\text{thigh}}}{|v_x|} \text{ and } {\text{F}^{-1}}_x = \text{P}^0 + v_x \frac{L_{\text{thigh}} + L_{\text{calf}}}{|v_x|} $$

We estimate hips previous positions P_l^{-1} and P_r^{-1} as P^0. Finally we compute ${\text{F}^0}_r$, ${\text{K}^0}_r$, ${\text{F}^0}_l$ and ${\text{K}^0}_l$ using the legs tracking algorithm.

6 Results

Figure 2(a) outlines the system configuration. The acquisition infrastructure is composed of four Phillips webcams (SPC900NC) connected to a single Pc (CPU: p4 3.2ghz,GPU: NVIDIA Quadro 3450). Webcams produce images of resolution 320×240 at 30fps.

Our method has been applied on different persons doing fast and challenging motions. Thanks to shape analysis and skin parts knowledge, our system is able to acquire the joint positions for a challenging pose outlined on the Figure 5(a). This pose is difficult because the 3D shape topology is not a human corresponding one. The temporal coherence is the success key for the pose presented Figure 5(b). This underlines the case of jointed hands (4.3) which is successfully recognized. A very difficult pose is underlines Fig. 5(c) which is successfully recovered by our system. Images of Figure 6 argue that our system works for large range movements acquisition.

Some results are included in the supplementary video clip [2]. This proves the robustness of our approach on long video sequences with rapid and complex movements.

Our current experimental implementation provides more than 30 poses tracking per second on a single computer, which is faster than the webcams acquisition frame rate. An optimized implementation can be usable for current generation of home entertainment computers. As our algorithm is based on 3D reconstruction, it is independent of the number of cameras used, but it depends on the voxel grid resolution. We reconstruct a voxel grid composed by 64^3 voxels in a $6m^3$ box. This resolution is sufficient for entertainment human-machine interfaces.

[2] http://liris.cnrs.fr/brice.michoud/gtas07videos/

Our motion capture system is based on a Shape-From-Silhouette algorithm. This algorithm computes an object 3D shape estimation from its silhouettes. The result directly depends on the silhouette segmentation quality, which is always an opened problem of the computer vision science. If the silhouette mask contains some noises like camera noise or object shadows, the volume reconstruction will be very noised. Thus the results of the motion capture will be worse. But our method is also based on a skin segmentation which is a more robust faced to camera noise. Then the hand and head articulations are more noise-resistant, than others articulations.

7 Conclusion

In this paper, we describe a new marker-free human motion capture system at least three webcams connected to a single computer. Fully automated and working under real-time constraint, the system is based on both a 3D shape analysis, human morphology constraints, and a 3D shape skin segmentation. Combining different 3D information, the approach is robust to self-occlusion and to coarse 3D shape approximation provided by voxel estimation sub-system. We are able to estimate the fifteen main human body joints, at more than 30 frames per second, which can be used for home entertainment applications.

The current system provides real-time motion capture for only one person. Current work aims to provide motion capture of multiple persons filmed together in the same area, even they are in contact. For home entertainment application, the major limitation is silhouette precessing, because the background cannot be guaranteed to be static at home. We work on a new segmentation algorithm based on statistical background model helped by optical flow algorithm.

References

1. Agarwal, A., Triggs, B.: Monocular human motion capture with a mixture of regressors. In: CVPR '05, IEEE Computer Society (2005) 72
2. Chen, Y., Lee, J., Parent, R., Machiraju, R.: Markerless monocular motion capture using image features and physical constraints. In: CGI '05: Proceedings of the Computer Graphics International 2005 (CGI'05), Washington, DC, USA, IEEE Computer Society (2005) 36–43
3. Mikic, I., Trivedi, M., Hunter, E., Cosman, P.: Human body model acquisition and tracking using voxel data. Int. J. Comput. Vision **53**(3) (2003) 199–223
4. Tangkuampien, T., Suter, D.: Human motion de-noising via greedy kernel principal component analysis filtering. In: ICPR (3), IEEE Computer Society (2006) 457–460
5. Caillette, F., Galata, A., Howard, T.: Real-Time 3-D Human Body Tracking using Variable Length Markov Models. In: Proceedings BMVC '05. Volume 1. (2005)
6. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Trans. Pattern Anal. Mach. Intell. **16**(2) (1994) 150–162
7. Ménier, C., Boyer, E., Raffin, B.: 3d skeleton-based body pose recovery. In: Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization and Transmission, Chapel Hill (USA). (2006)
8. Franco, J.S., Boyer, E.: Exact polyhedral visual hulls. In: Proceedings BMVC'03. (2003) 329–338 Norwich, UK.
9. Hasenfratz, J.M., Lapierre, M., Sillion, F.: A real-time system for full body interaction with virtual worlds. Eurographics Symposium on Virtual Environments (2004) 147–156
10. Cheung, K.M., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3d voxel reconstruction of human motions. In: Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00). Volume 2. (2000) 714 – 720
11. Michoud, B., Guillou, E., Bouakaz, S.: Shape from silhouette: Towards a solution for partial visibility problem. In: Eurographics 2006 Short Papers Preceedings. (2006) 13–16
12. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: ICCV. (1999) 666–673
13. Joshi, N.: Color calibration for arrays of inexpensive image sensorss. Technical report, Stanford University (2004)
14. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: ICCV (1). (1999) 255–261

15. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: Proceedings of Graphicon-2003. (2003)
16. Segal, M., Korobkin, C., van Widenfelt, R., Foran, J., Haeberli, P.: Fast shadows and lighting effects using texture mapping. In: Proceedings of SIGGRAPH. (1992)
17. Culbertson, W.B., Malzbender, T., Slabaugh, G.G.: Generalized voxel coloring. In: Workshop on Vision Algorithms. (1999) 100–115
18. Sun, W., Hilton, A., Smith, R., Illingworth, J.: Layered animation of captured data. The Visual Computer **17**(8) (2001) 457–474
19. Fua, P., Gruen, A., D'Apuzzo, N., Plankers, R.: Markerless Full Body Shape and Motion Capture from Video Sequences. In: Symposium on Close Range Imaging, International Society for Photogrammetry and Remote Sensing, Corfu, Greece. (2002)
20. Dreyfuss, H., Tilley, A.R.: The Measure of Man and Woman: Human Factors in Design. John Wiley & Sons (2001)

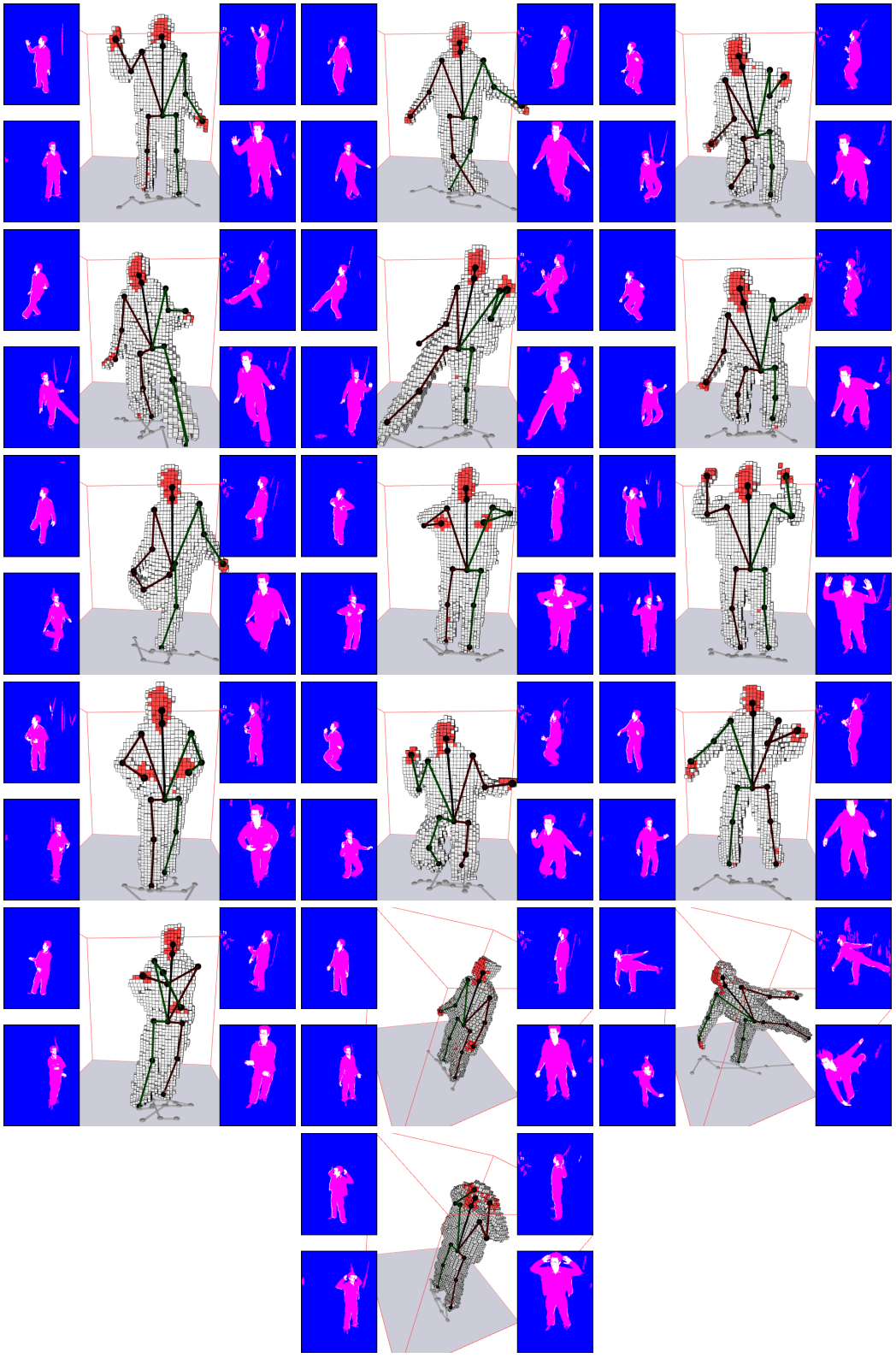


Fig. 6. Results for a wide range of movements