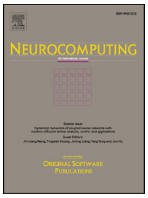
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Robust labeling of human motion markers in the presence of occlusions

J.L. Jiménez Bascones, Manuel Graña*, J.M. Lopez-Guede

Computational Intelligence Group, University of the Basque Country (UPV/EHU), San Sebastian, Spain

ARTICLE INFO

Article history:
Received 21 December 2017
Revised 21 April 2018
Accepted 28 May 2018
Available online 19 March 2019

Keywords:
Adaboost
Solver ensemble
Point cloud labeling
Marker tracking
Optical motion capture

ABSTRACT

Human motion capture by optical sensors produces snapshots of the motion of a cloud of points that need to be labeled in order to carry out ensuing motion analysis for medical or other purposes. We generate the labeling of instantaneous captures of the cloud of points, discarding temporal correlations, in the presence of occlusions. Our approach proposes an ensemble of weak classifiers defined over geometrical features extracted from small subsets of the cloud of points. We apply an Adaboost strategy to select a minimal ensemble of weak classifiers achieving a target correct labeling detection accuracy. Furthermore, we use these features to generate the labeling of the points in the cloud even in the presence of occlusions.To deal with the occlusions of markers we search for ensembles of partial labeling solvers which can provide partial consistent labelings which cover the unoccluded markers. We test two greedy search approaches and a genetic algorithm in the search for the optimal ensemble of partial solvers We demonstrate the approach on a real dataset obtained from the measurement of gait motion of persons, with available ground truth labeling. Results are encouraging, achieving high accuracy label generation at a reduced computational cost.

1. Introduction

Motion capture (MoCap) is the process of object motion quantification in order to build computer models that allow further fine analysis. Human body Mocap is widely employed in many scientific and industrial applications like entertainment, clinical analysis and rehabilitation, i.e. gait analysis [5,7], as well as in the improvement of sport performance (golf, soccer, cycling) [31] where the biomechanics of the movement plays a crucial role [19]. There are two main categories of human MoCap: (a) optical, (b) wearable. Wearable systems attach inertial, magnetic or mechanical tracking devices to the body [5,7,31]. Optical systems use a constellation of calibrated, synchronized cameras deployed around the *scene*, recording images at a constant frame rate. For each recorded frame, a set of 2D points called *markers* are extracted and 3D coordinates of a cloud of points are computed by means of photogrammetric techniques [6,9,32]. Point labeling is the unique matching of these points against a body model, allowing further biomechanical calculations. Fig. 1 illustrates the point labeling problem: given a collection of points $C = \left\{ c_i \in \mathbb{R}^3 \right\}_{i=1}^{n_c}$ and a model of the localization of these points in the body $M = \left\{ m_j \in \mathbb{R}^3 \right\}_{j=1}^{n_m}$, we look for the injective map from C into M. In our approach, we do not use color codes, surrounding image, or fiducial schemes, or tracking of the points along time. Our research question is posed as: is it possible to solve the point labeling problem at each frame independently, yet consistently, using only geometrical information even in the presence of occlusions?. Occlusions happen when a model marker point is not present in the cloud of points, i.e. the number of observed points is lower than the number of model marker points. They happen for a variety of reasons, such as the interposition of opaque objects between the marker and the cameras or when the marker goes out of the camera field of view (FOV). We propose a robust approach which recovers from occlusions, labeling all the candidate points at each frame.

Contributions of the paper. This paper improves over a previous system for point cloud labeling using geometric features as weak classifiers that we developed in [10]. Here we solve the issues raised by marker point occlusions, i.e. when the number of observed points is lower than the number of model marker points. In essence, solving the occlusions implies the combinatorial exploration of all possible labeling solutions over subsets of the model marker points, hence needing to run multiple instances of the labeling algorithm. We explore several ways to find the optimal

* Corresponding author.
E-mail addresses: jljbascones@gmail.com (J.L. Jiménez Bascones), manuel.grana@ehu.eus (M. Graña), m.lopez@ehu.es (J.M. Lopez-Guede).

https://doi.org/10.1016/j.neucom.2018.05.132

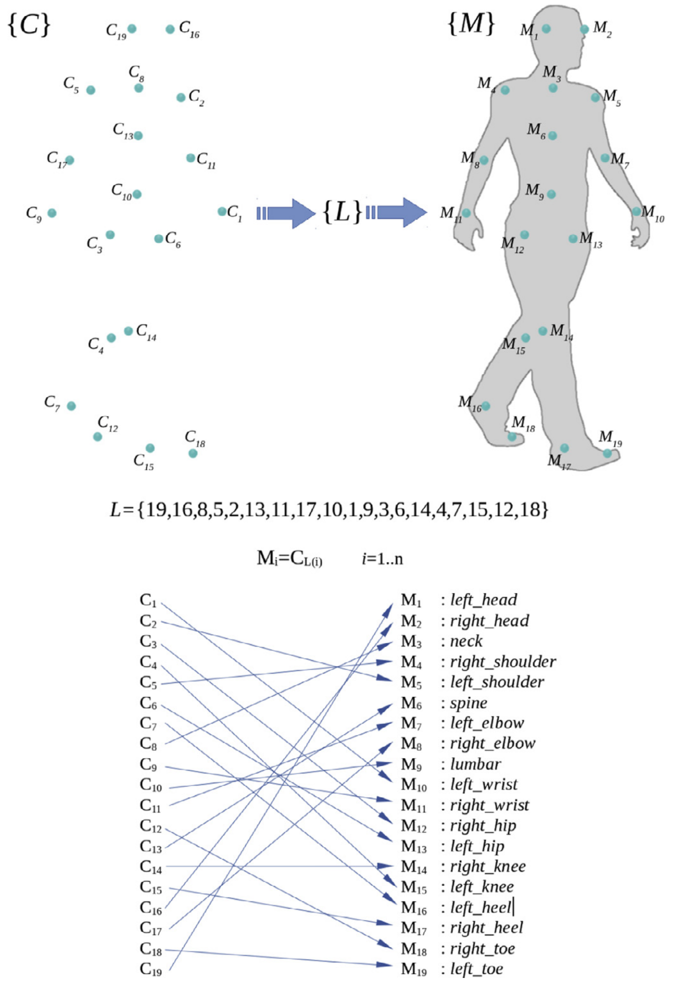


Fig. 1. Example of a humanoid model labeling L assigning a unique model point M_j to each observed point $\{C_i\}$.

matching subset: (a) a genetic algorithm, (b) a tree-search approach that computes the minimal set of weak classifiers achieving the maximal labeling accuracy, which is the greatest subset of model marker points labeled with precision high enough. We achieve an optimal balance between the number of times that the overall algorithm succeeds to establish the right labelings, called *hit rate,* and the number of times the process declares a marker as occluded when it isn't, the *false occlusions rate.*

Paper contents. Section 2 describes the related works in the literature. Section 3 describes the previous works on the labeling efforts, giving basic definitions and algorithms. Section 4 describes the actual contributions in this paper to achieve robust labeling in the presence of occlusions. Section 5 describes the experimental data used for validation computational experiments. Section 6 reports the results of computational validation experiments. Finally, Section 7 gives our conclussions and some lines for future work.

2. Related work

Human motion capture is often needed for clinical purposes, such as gait analysis [28], but there are emerging applications such as human-robot collaboration [14], where human motion prediction is critical for the safe interaction of humans and robots in the workplace. Human motion capture has been an active research area in last decades, many efforts have been addressed to computer vision based techniques [19], but other sensors that need to be in direct contact with the body (wearables) have been used to measure body motion parameters, such as inertial sensors [5,11], sensors attached to a tailored costume [27], wireless markers [25], or stretching, bending, torsion sensors directly applied to the skin to measure for accurate local motion measurement [31]. The main advantage of remote sensors, such as conventional computer vision, Leap Motion Controller [26]. or the kinect 2D+ sensor [21,22], is that they are (almost) not invasive so that they do not interfere on the subject motion. Techniques using body markers can be considered somehow invasive, but always less than the wearable sensors. The computer vision approaches relying only on optical information captured by cameras often require large constellations of cameras [2] (in our work we used a six camera deployment) that need careful calibration processes in order to have accurate estimations of measurement error [20]. If specific infrared light sensors are used, then careful capture conditions are required, such as warm up time for cameras or blocking any external sun infrared interference.

There are three basic approaches to extract pose information from the optical information: skeleton, silhouette, and pointwise features [19,21,22]. The skeleton and silhouette based approaches first extract the human shape region of the image and then they proceed by either computing its medial axis aka skeleton or extracting the boundaries of the shape. The pointwise feature approach usually proceeds by attaching highly reflectant markers to the body in specific places, often in the infrared band spectra in order to simplify image processing to the maximum. Image segmentation and extraction of the points is then quite easy and robust. In our work we follow this approach, which is the most frequent in industrial/commercial settings. Some authors try to extract pointwise feature from shape analysis, but such attempts have shown little reliability. It must be noted that all approaches suffer from occlusions that may lose part of the skeleton, confuse the silhouette, or miss some pointwise features.

From the computational point of view, most approaches try to use time information in order to increase robustness. For instance [31] uses a Hidden Markov Model approach on human shapes, while [8,15] apply extended Kalman filters to predict the motion of point features. Other authors minimize energy function measuring the difference between the predicted trajectory and the observed point features [13,23]. Other works [16,18,24] address the question trying to minimize the mean distance between the candidates and the predicted marker positions by a tailored implementation of the Hungarian method [12]. In general, these approaches are quite sensitive to occlusions and have difficulties to recover from small errors, often leading to divergent behaviors. For this reason, we have focus our efforts here and in [10] on the construction of a marker point labeling that does not use past time information, working on each time cloud independently. On a different note, some authors [16,18] try to use the biomechanical information of the human skeleton structure, so that they can benefit from the kinematic models of the skeleton in order to predict future locations of the marker points [4,17,30]. However, these approaches also suffer great divergences in the case of occlusions. Our approach is completely original in that we learn the relevant geometric relations between marker points from the data, selecting by an Adaboost approach the optimal collection of weak classifiers, aggregated into partial solvers that can produce the marker labeling instantaneously.

The conventional task is to recognize the human action or activity after it has been completed [22], but some works try to recognize the activity while it is happening [3]. These approaches would have the advantage of adding recognition robustness to the prediction of the human motion that can be required in some environments, such as human-robot interaction. In clinical/sport applications the task often consists in the measurement of some specific biometric parameters of the motion under observation, such as gait parameter measurement [1,2,5,7], or exploratory behavior [25]. This is specifically the relevant task for our works. The robust marker labeling is the key requirement to allow the robust measurement of motion parameters of clinical significance.

There are a number of commercial solutions for human body marker tracking, such as *Cortex* developed by Motion Analysis,[1] *Track Manager* from Qualisys,[2] or *Clima* from STT Systems,[3] often integrated in a more complete and generic hardware-software solutions. However, little has been published about the details of the core tracking mechanism they implement due to the proprietary nature of these packages. It is possible to guess from their descriptions that some of them use heuristic methods, but otherwise those systems are unrelated to the present work which presents an unpublished approach till the date. Hence it is not possible to make detailed comparisons with their performances.

3. Labeling without occlusions

In this section, we review the cloud of points labeling system in the case when there are no occlusions as introduced in [10]. This system is the baseline needed to understand the contributions of this paper. The algorithm we presented in [10] follows a classification approach that does not use any prior structural or time series information, such as the kinematic joint model of the tracked object. Instead, a set of weak classifiers defined over geometric features are trained from the available ground truth data. An optimal ensemble of weak classifiers are then chosen and assembled into a strong classifier by means of a tailored Adaboost machine learning meta-algorithm.

3.1. Some definitions

We provide here the basic definitions of the labeling algorithm elements.

- A marker model $M = \left\{m_1, m_2, \ldots, m_{n_m}; m_j \in \mathbb{R}^3\right\}$ is a set of cartesian points called *markers* to be identified in the observed-unlabeled point cloud.
- An observation at time t consists of a cloud of unlabeled XYZ points $C(t) = \left\{c_1, c_2, \ldots, c_{n_c}; c_i \in \mathbb{R}^3\right\}$ provided by the optical data capture hardware/software, that we call the *set of candidate points*.
- A labeling $L(t)$ of the observation at time t is the mapping of markers of the given model M into the candidate points $C(t)$. A labeling is encoded as a set of integers $L(t) = \{l_1, l_2, \ldots, l_{n_m}\}$, where l_i denotes the map $m_i \rightarrow c_{l_i}$, i.e. l_i is the index in the set $C(t)$ of the candidate points to be assigned to the ith marker m_i. The occlusions (markers not assigned to any candidate) are encoded as an assignment to a virtual null candidate '0', so that $l_i = 0$ means that the ith marker is considered as occluded. The labeling $L(t)$ does not have non-zero repeated values (meaning that the same candidate point cannot be labeled twice).
- Let us denote a subset of markers as $M_s \subset M$. A partial labeling $L_{M_s}(t) \subset L(t)$ is the mapping of the marker subset M_s into the candidate points at time t.
- We consider weak classifiers $h(M_s, C(t), L_{M_s}(t)) = T \in \{0, 1\}$, which are decision functions whose output is whether the partial labeling $L_{M_s}(t)$ is correct (1) or not (0).
- We build up a strong classifier as an ensemble of weak classifiers $\phi = \{h_1, h_2, \ldots\}$. We look for the minimal set of weak classifiers able to decide if a (partial or complete) labeling is correct (*true*) or not (*false*): $\phi(M, C, L) = T \in \{0, 1\}$.
- The set of weak classifiers is trained over a set of labeled samples extracted from a large number of frames whose labeling relative to a given marker set has been manually generated.
- The training method is a very specifically tailored version of Adaboost [10].
- As we are treating each data capture frame independently, we can drop the time parameter.
- The solver $S(C, M, \phi)$ is the algorithm that finds the set of feasible labeling maps $\mathcal{L} = \left\{L^1, L^2, \ldots\right\}$, such that $\phi(M, C, L^i) = true$. Solver $S(C, M, \phi)$ makes use of the strong classifier ϕ and an efficient tree exploration method to find *all* the feasible marker labelings of the candidate points. Despite its efficiency in terms of computation time, its main drawback is that it cannot handle null labels. Hence, for each labeling found $L^i \in \mathcal{L}$ all of its components $l_j \in L^i$ are positive $l_j > 0$. The set of labelings found by S might be the empty set $\mathcal{L} = \emptyset$, meaning that the solver $S(C, M, \phi)$ could not find any feasible solution. The algorithm of $S(C, M, \phi)$ is unable to deal with occlusions: either it assigns a candidate to each marker or to no one.
- Once we have trained a strong classifier ϕ, we can assess by means of random samples coming from the ground truth the *hit rate* $P(S, m_i) = P_i(S) \in [0, 1]$ of the solver in the assignment of any marker m_i to its right candidate. This assessment information is computed and stored as a *metadata* of the solver S for further usage.

3.2. The weak classifiers

Given the candidate points, we define scalar valued geometric functions $\{g_k : D_k \rightarrow \mathbb{R}\}$, where D_k is the specific domain of the function defined by the required number of points. Examples of geometric functions are listed in Table 1, each corresponding to a geometric property of the polygon defined by the set of points. Each geometric function allows to build a collection of features from the cloud of candidate points applying it to all possible combinations of points that fit into the domain D_k definition. Thus we can compute over a given cloud of points as many features as combinations allowed by the used geometric functions. This number of features grows combinatorially with the size of the cloud of candidate points. Following an Adaboost [29] approach, we define a weak classifier from each feature as as follows: each feature falls within a range of values $[\alpha, \beta]$ when the labeling of is correct, therefore a weak classifier checks if the feature value falls within the specified interval, i.e.

$$h\left(f_k^S(M, C), \alpha_k^S, \beta_k^S\right) = \begin{cases} 1 & \text{if } \alpha_k^S < f_k^S(M, C) < \beta_k^S \\ 0 & \text{otherwise} \end{cases}, \qquad (1)$$

where f_k^S is a feature computed applying geometric function $g_k(.)$ to a subset of points $S \subset M$ selected from the candidate points C, $\left[\alpha_k^S, \beta_k^S\right]$ are interval of values where the feature falls when the labeling is correct, and class 1 denotes correct labeling of the cloud of points.

3.3. The ensemble strong classifier

The first step in building the labeling algorithm is to compute the range of values of the feature from a set of learning observations $O_i = \{C_i, L_i, b_i\}$ corresponding to a common model M. The vector b_i encodes the correctness of the mapping, so that $b_{ij} = 1$ if the label of the jth cloud point is correct. The observations in the given ground truth are error-free, but it is easy to generate incorrectly labeled observations as permutations L_i. The number of permuted elements (from 2 to n, the number of markers) is an index of the severity of the labeling error. The strong classifier consists of a collection of features whose corresponding weak classifier is

[1] http://ftp.motionanalysis.com/html/industrial/cortex.html.

[2] https://www.qualisys.com/software/qualisys-track-manager/.

[3] https://www.stt-systems.com/products/3d-optical-motion-capture/clima/.

Table 1
Examples of the basic geometric functions with different domain sizes used to build the features used by the weak classifiers.

Geometric property	g	# points	Points	Expression
Angle between consecutive angles	g_1	3	A, B, C	$\arccos\left(\frac{AB\cdot AC}{\|AB\|\cdot\|AC\|}\right)$
Distance between points	g_2	2	A, B	$\|AB\|$
Similarity ratio between segments	g_3	4	A, B, C, D	$2\frac{\|AB\|-\|CD\|}{\|AB\|+\|CD\|}$
Height difference between two points	g_4	2	A, B	$A_y - B_y$

weighted by its accuracy gain relative to the remaining weak classifiers. The output of the strong classifier is computed as:

$$\phi_J(M, C, L) = \frac{\sum_{j=1}^{J} w_j h_j(f_k^S(C), \alpha_k^S, \beta_k^S)}{\sum_{j=1}^{J} w_j}, \tag{2}$$

where the index j refers to the order of selection of the feature for inclusion in the ensemble and J is the size of the ensemble. By construction any collection of weak classifiers will have 100% true positive rate, but we need to reduce the false positive rate by ensuring that the selected weak classifiers show the least false positive rate. According to the Adaboost strategy, initially, all weights are initialized to zero and the set of selected weak classifiers is empty. Weak classifier selection is carried out by feeding all weak classifiers with incorrectly labeled observations of different error severity.

Training process after presentation of each incorrectly labeled sample is as follows: If the actual $\phi_J(M, C, L)$ does reject the incorrectly labeled sample, then no further process is done. If not, the weights of unselected (out of $\phi_J(M, C, L)$) weak classifiers that reject it are updated according to the error severity. After a number of incorrect observations is processed, the strong ensemble classifier is updated adding the unselected weak classifier having the greatest weight. The whole process eventually stops when the strong classifier reaches a preset accuracy threshold. Finally, the selected weak classifiers are stored together with the weight they got during the learning process as their score.

3.4. Generating a labeling

After constructing the strong ensemble classifier we can use it to generate the labelings when there is none. Generating the labeling of a cloud of points can be stated as looking for the value of L that maximizes the number of weak positive classifications needed to achieve $\phi_J(M, C, L) = 1$. When there are no occlusions L can be anyone of the permutations of the integers between 1 and n. Therefore, the number of possible configurations for L is $n!$. Fortunately, we can exploit the structure of the strong classifier in a tree-search strategy to explore the labeling permutations using the following properties:

- Each single weak classifier can be computed independently from a handful of points (usually from 2 to 6) which represents a subset of the vector L;
- The strong ensemble classifier ϕ can be evaluated over a *partial solution* where only a subset of elements of L has meaningful labels. Weak classifiers using unassigned labels are simply ignored;
- A single weak classifier rejecting a label permutation rules it out, so that we can stop the evaluation of the ensemble classifier over L as soon as one weak classifier gives a negative outcome.

Hence the training process is a depth first tree search process, where nodes are values of the labeling vector L, and branches are weak classifier evaluations. The number of unassigned labels decreases as the level nodes in the tree increase, when we reach the level where candidate points are labeled we are finished. The rejection of a partial label by a weak classifier terminates the search in this branch.

4. Robust labeling in the presence of occlusions

In this section, we introduce the main contributions of this paper: generating the labeling of the observed cloud points dealing with occlusions of the markers. We introduce partial solvers in order to achieve the maximal number of correct labels. Moreover, it is desirable the specification of individual target hit rates τ_i for each marker, that is, we require after training that $P_i(S(C, M, \phi)) > \tau_i$.

4.1. Partial solvers

Definition. A partial solver $S = S(C, M_s, \phi_s)$ is not forced to find corresponding candidate observed points to all markers M. Its associated strong ensemble classifier ϕ_s can be trained to generate a partial labeling $L_s \subseteq L$ for a subset of markers $M_s \subseteq M$. Obviously, the strong ensemble classifier ϕ_s only can be used to generate labelings over the markers belonging to M_s. The definition of *hit rate* per marker P_i applies also to partial solvers, provided they can be assessed against the ground truth.

Properties. We can state several interesting properties of the hit rates of the partial solver. Some of them were formulated from observations taken on the computational experiments and may be object of theoretical research in future works.

- If a marker m_i doesn't belong to the subset M_s of the partial solver, its hit rate remains undefined: if $m_i \notin M_s \Rightarrow P_i(S_s, m_i) = NaN$;
- The hit rate for a marker m_i is strictly increasing with the size of the marker subset: if $m_i \in M_A \subset M_B$, $|M_B| > |M_A| \Rightarrow P_i(S_B) \geq P_i(S_A)$;
- Because the hit rates grow with with solver size, we would expect that only big solvers may provide high hit rates. However, our empirical finding reveals that there are also small solvers showing up high hit rates.
- We consider a marker model as optimally designed if its labeling is feasible with a 100% confident rate in absence of occlusions. In other words, there is at least a solver whose hit rates are 1 for each one of the markers when working over the whole set: if $M_s \equiv M \rightarrow \exists S \setminus P_i(S_s(C, M_s, \phi_s)) = P_i(S(C, M, \phi)) = 1$;
- We found out that such solver exists for the marker set used in the experimental tests of our work, hence our set of markers was optimally designed.

4.2. Partial solver ensemble

We define a partial solver ensemble as a set of partial solvers such that the union of their marker subsets covers the complete model: $\Omega = \{S_1, S_2, \ldots, S_N\}\ s.t.\ M_{s_1} \cup M_{s_2} \cup \ldots \cup M_{s_N} = M$. The aim of defining the ensemble is to overcome the limitation of an individual solver to give an answer when there is an occlusion. If there is an occlusion, the unaffected partial solvers (i.e. those defined over a subset of not occluded markers) may still provide the labeling of the unoccluded markers. More formally, let us denote M^* the set of not occluded markers, we can find a set of partial solvers $\Omega^* = \{S_1^*, S_2^*, \ldots, S_{N*}^*\} \in \Omega$ such that $M_{S_1}^* \cup M_{S_2}^* \cup \ldots \cup M_{S_N}^* \subseteq M^*$. Given that we use a deterministic learning algorithm for the construction of the strong classifiers, two partial solvers are different only if they are defined over different marker subsets: $S_A(M_A) \neq S_B(M_B) \iff M_A \neq M_B$. According to that criteria, the total number of partial solvers is the size of the markers power set $\mathscr{P}(M)$, i.e.$\sum_{i=1}^{n_m}\binom{n_m}{i}$, where $n_m = |M|$.

The problem of generating marker labeling robust to occlusion is, thus, formulated as the search for small size partial solvers with high target rates to compose a partial solver ensemble which can produce partial labelings that give the best partial labeling solution when there are occlusions. The emphasis on small size partial solvers, is because if one is affected by an occlusion it will not compromise the labeling of many additional markers. The emphasis on high target rates is because they will increase the confidence on the labeling. A brute force exhaustive search approach is, of course, unfeasible even for moderate sizes of the marker set, therefore we have explored two heuristic approaches.

Greedy search. By taking advantage of the 2nd solver property –hit rates strictly increase with dimension–, we can start with n_m solvers of dimension 1 (one solver per marker), adding one more extra marker at each step of the search. This is an incremental building process, that stops when the target hit rate is reached. This strategy avoids unnecessarily big solvers, thus saving computation time. The searching algorithm is described in Algorithm 1 and depicted in the diagram shown in Fig. 2.

Algorithm 1 Greedy bottom up partial solver search.

```
• Input data: target hit rates $\tau_i$ for each marker $m_i$of the full
  marker model $M$, $n = |M|$.
• Output data: set of partial solvers $\Phi = \{\phi_k\}_{k=1}^N$with hit rates
  higher than the target at least in one of their markers.

  1. Set up an initial set of $n$ solvers of dimension 1, $\mathit{\Omega} =$
     $\{S_1, \ldots, S_n\}$. Initialize $\mathit{\Phi} = \emptyset$;
  2. For each solver in $\Omega$, assess its hit rates, if higher thatthe
     goal, it is removed from$\Omega$ and added to $\mathit{\Phi}$;
  3. Terminate if $\Omega$ is empty, or the dimension of its elemental-
     solvers equals $n$;
  4. For each solver $S^i$ from $\Omega$, a new marker is added toit, and
     thus $n - dim(S^i)$ new solvers are generated, replacing$S^i$ in $\Omega$;
  5. Go back to 2.
```

Conversely, we can proceed in a top down way. Starting from the full marker set solver S, $dim(S) = n$, which we assume to meet the target hit rates, we can generate new partial solvers of lower dimension by progressively removing markers in a recursive manner. In this case, the process stops if the new generated solvers fall under the target hit rates (see Algorithm 2 and the diagram in Fig. 3).

Genetic algorithm search. In order to look for good approximations to global optima we have defined an *ad-hoc* genetic algorithm constructed as follows. Encoding: A partial solver acting over a subset of markers $M_s \subset M$ can be encoded as an array of n boolean values $\{b_i\}$ such that $b_i = 1$ if $m_i \in M_s$ and 0 otherwise. Such encoding is the chromosome of the genetic algorithm. The optimal ensemble of partial solvers Ω is encoded by the entire population at the end of the evolution process. The fitness function of each chromosome is the maximum of the hit rates of the corresponding partial solver.

Algorithm 2 Greedy top down partial solver mining.

```
• Input data: target hit rates $\tau_i$ for each marker $m_i$of the full
  marker model $M$, $n = |M|$.
• Output data: set of partial solvers $\Phi = \{\phi_k\}_{k=1}^N$with hit rates
  higher than the target at least in one of their markers.

  1. Set up initial solver of dimension $n$, $\mathit{\Omega} = \{S\}$ and initialize
     $\mathit{\Phi} = \emptyset$;
  2. For each solver $S^i$ in $\Omega$:
     (a) remove it from $\Omega$ ;
     (b) remove each of its marker once at a time, generating
         $dim(S^i)$ new solvers stored in $\mathit{\Omega}^i$;
     (c) for each solver of $\mathit{\Omega}^i$, its hit rates are assessed;
     (d) if no solver from $\mathit{\Omega}^i$ reaches the target rates, it isre-
         moved from $\mathit{\Omega}^i$ and joined to $\mathit{\Phi}$;
     (e) the remaining solver from $\mathit{\Omega}^i$ are added to $\Omega$;
  3. If $\Omega$ is not empty, go back to 2.
```

Starting from a randomly generated population composed by a number of partial solvers encoded as chromosomes, we apply the following genetic operators to improve the population fitness towards finding the global optimal collection of partial solvers:

- Crossover: two parent chromosomes (partial solvers) are selected randomly from the population, the crossover operator generates a new chromosome by picking randomly its genes from either one of parent chromosomes.
- Mutation: a chromosome is randomly selected and a new one is generated either by random permutation, addition or subtraction of one of the parent's genes;
  - permutation: pick a pair of genes of different values and permute them. The size of the child partial solver remains the same;
  - addition: pick a random '0' gen and reverse its value. The size of the child partial solver increases by one;
  - subtraction: pick a random '1' and reverse its value. The size of the child partial solver decreases by one. The subtraction operation is biased towards the search of small specimens;
- Selection: after application of genetic operators, the fitness of the chromosomes in the population are evaluated selecting those that meet the target hit rate, when there is equal hit rate, smaller solvers are preferred. After that, a massive die out removes the 25% worse specimens. The survivors join the ensemble of partial solvers.

We conducted several computational experiments in which we always managed to improve the initial population after a number of generations. The resulting solver ensembles proved to be good enough to meet the requirements of the labeling algorithm discussed later. In any case, the efficiency of the genetic search strongly depends on its meta-parameters: initial population size, crossover and mutation frequencies, number of operations between die outs and percentage of specimens to wipe out. Future work may be devoted to the optimal tuning of this evolutionary process.

4.3. Integration of partial solvers for labeling generation.

After we have found an optimal ensemble of partial solvers $\Phi = \{\phi_k\}_{k=1}^N$, whose hit rates meet preset targets τ_i. Each partial solver is defined over a subset of the complete marker model, and the union of all solvers covers the complete model M. The formulation

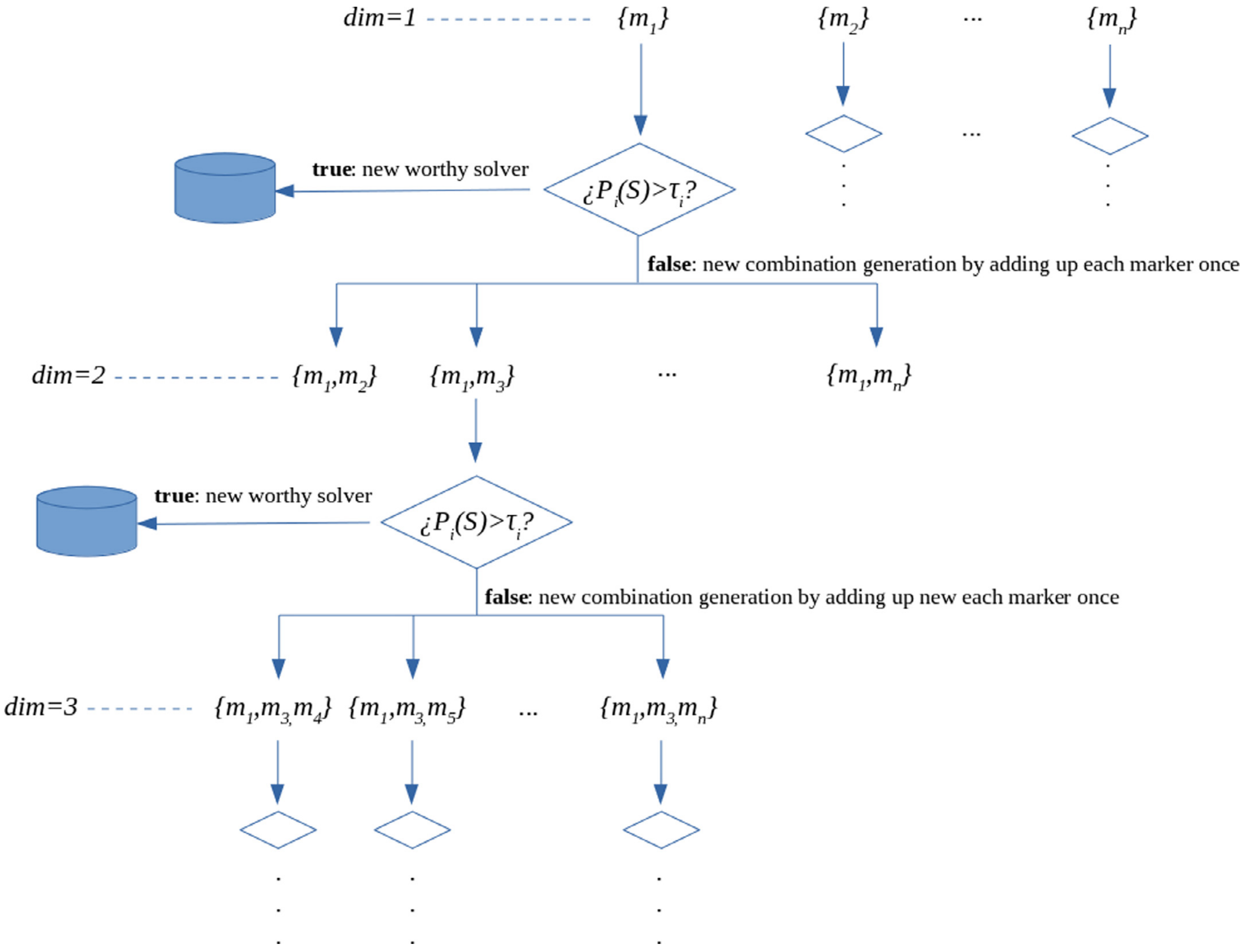


Fig. 2. Greedy bottom up search diagram representation.

of Φ is a time consuming training process to be done offline before the online execution of the complete labeling algorithm.

During the labeling process, every time a new frame is acquired, the list of candidate 3D points extracted from the multi-camera captures is built and exposed to each solver of the ensemble Φ of partial solvers. Each member of the ensemble ϕ_k output is none, one or several candidate points assigned to the markers M_k within its scope. The contribution of each solver is recorded in a *labeling matrix* that has as many rows as candidate points (n_c) and columns as model markers (n_m), so that each matrix entry (i, j) contains $\{S_s^{i,j}\}$: the set of partials solvers belonging to the ensemble Φ who suggested the ith candidate to the jth marker. This matrix will be sparse because we expect most of the times the partial solvers to agree on the mappings, as follows:

	m_1	m_2	$\cdots$	m_j	$\cdots$	m_{n_m}
c_1	$\varnothing$	$\{S_s^{1,2}\}$	$\cdots$	$\varnothing$	$\cdots$	$\varnothing$
c_2	$\varnothing$	$\varnothing$	$\cdots$	$\varnothing$	$\cdots$	$\varnothing$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
c_i	$\varnothing$	$\varnothing$	$\cdots$	$\{S_s^{i,j}\}$	$\cdots$	$\varnothing$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
c_{n_c}	$\varnothing$	$\varnothing$	$\cdots$	$\varnothing$	$\cdots$	$\{S_s^{n_c,n_m}\}$

The columns in this matrix specify the labeling of the candidate points. We may have the following situations and labeling assignments:

- A column j is empty: we interpret that it corresponds to an occluded marker, so no labeling can be given $l_j = 0$
- A column j has more than one non-null entry, which means that there is an ambiguous assignment. We also assume that this is an occluded marker, and give no labeling, $l_j = 0$.
- A row has more than on non-null entry in columns j^1, $j^{2,}$ $\ldots$, which means that a candidate point is assigned to more than one marker point. In this case, we assume that all conflicting markers are occluded giving no labeling for them, i.e. $l_{j^1} = 0$, $l_{j2} = 0$, ...
- A labeling $l_j = i$ is given when the ith row and jth column contain only one non-null entry.

5. Experimental data

In this section, we describe the data that has been used for the computational validation experiments referred below. The whole experimental setup corresponds to an industrial MoCap setting for sport performance measurement with a constellation of six synchronized cameras of 800x800 pixel with infrared filters, infrared light sources, and the actor wearing dark clothes and infrared reflecting markers and shown in Fig. 4. The experimental data has

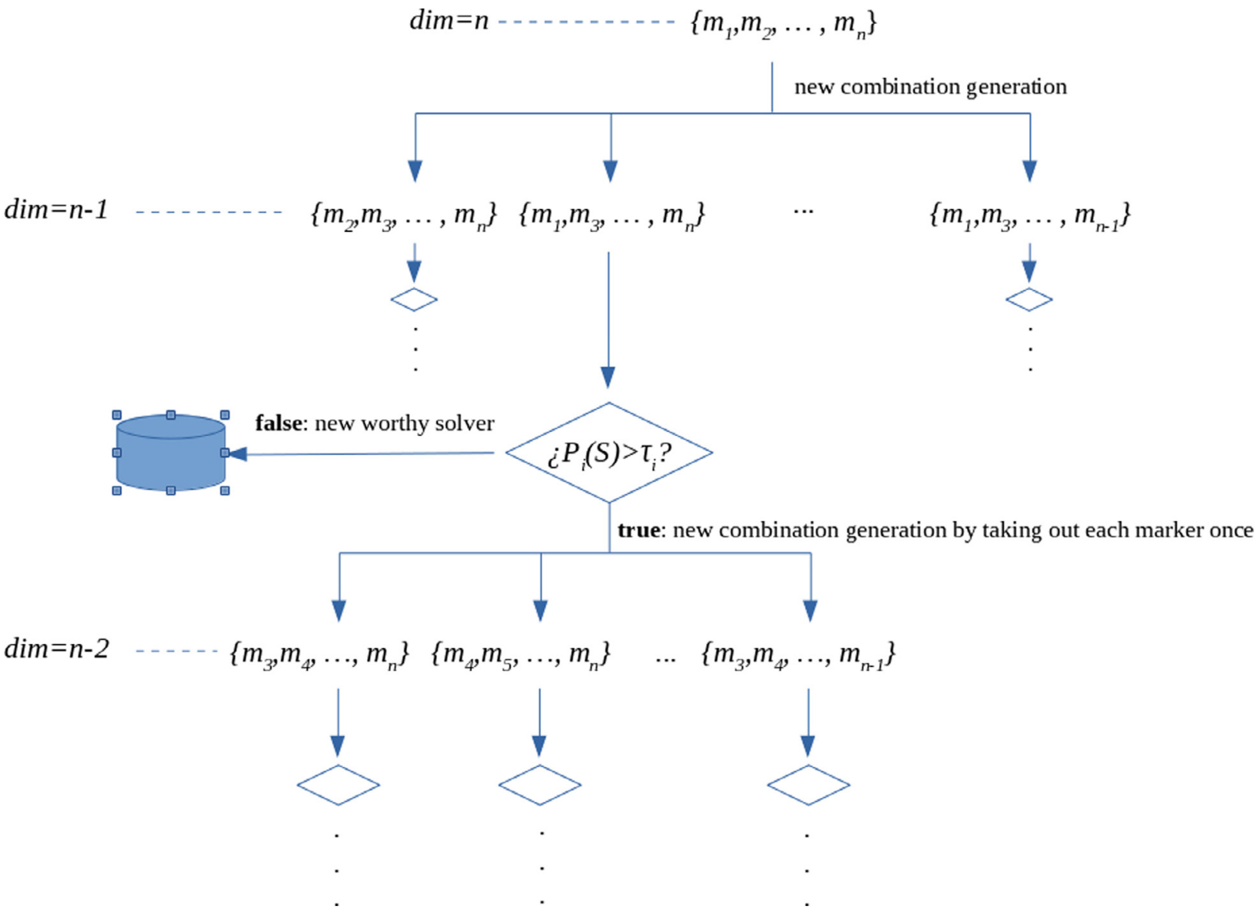


Fig. 3. Greedy top down search diagram representation.

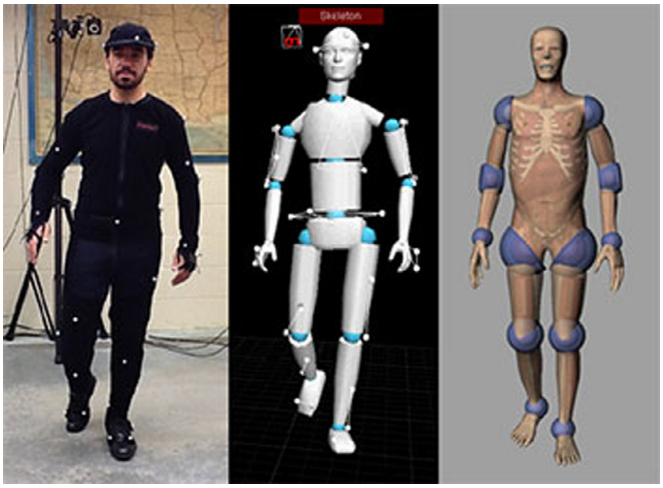


Fig. 4. Actor wearing reflective markers and corresponding digital model.

been manually labeled. It consists of 70 videosequences (100 Hz) recording 14 different people of diverse ages and body shapes walking at random speed. The average duration of the sequences is three seconds, so that we have more than 20.000 frames to extract the clouds of candidate points. Manual labeling has been carried out by the first author. Labeled clouds correspond to correct labelings categorized as class 1 for classification purposes. Point clouds with incorrect labeling corresponding to class 0 data items are generated by random permutations of the labels of correct labeling data. The point cloud sequence capture follows the Helen Hayes lower train protocol [28], widely applied in clinical gait analysis, with the use ofproprietary optical motion tracking system. The proprietary *CLIMA* software[4] automates the whole process of motion capture from camera management, camera calibration, 3D reconstruction, marker tracking and further biomechanical analysis.

6. Computational experiments results

We assess the efficiency of algorithm according to two main performance indices:

- False assignments rate (FA): number of wrong assignments of candidate points to marker vs. total number of assignments. This is the rate of incorrect labelings.
- False occlusions rate(FO) : number of wrong occlusion assignments vs. total number of occlusion assignments

It is desirable to keep both rates low: obviously we want to avoid wrong labelings, but not at the expense missing detection of many actual wrong labelings. We try to achieve a balance between both performance indices in the algorithm settings.

To validate the whole process, a large set of frames are borrowed from the ground truth. The candidates for each frame are randomly permuted to obtain wrong labelings. To simulate occlusions we remove between 1 and 5 candidate points. The labeling generated by the approach presented above is compared with the correct labeling and the validation statistics are continuously updated. Summary description of the experimental conditions is

[4] http://www.stt-systems.com/products/3d-optical-motion-capture/clima/.

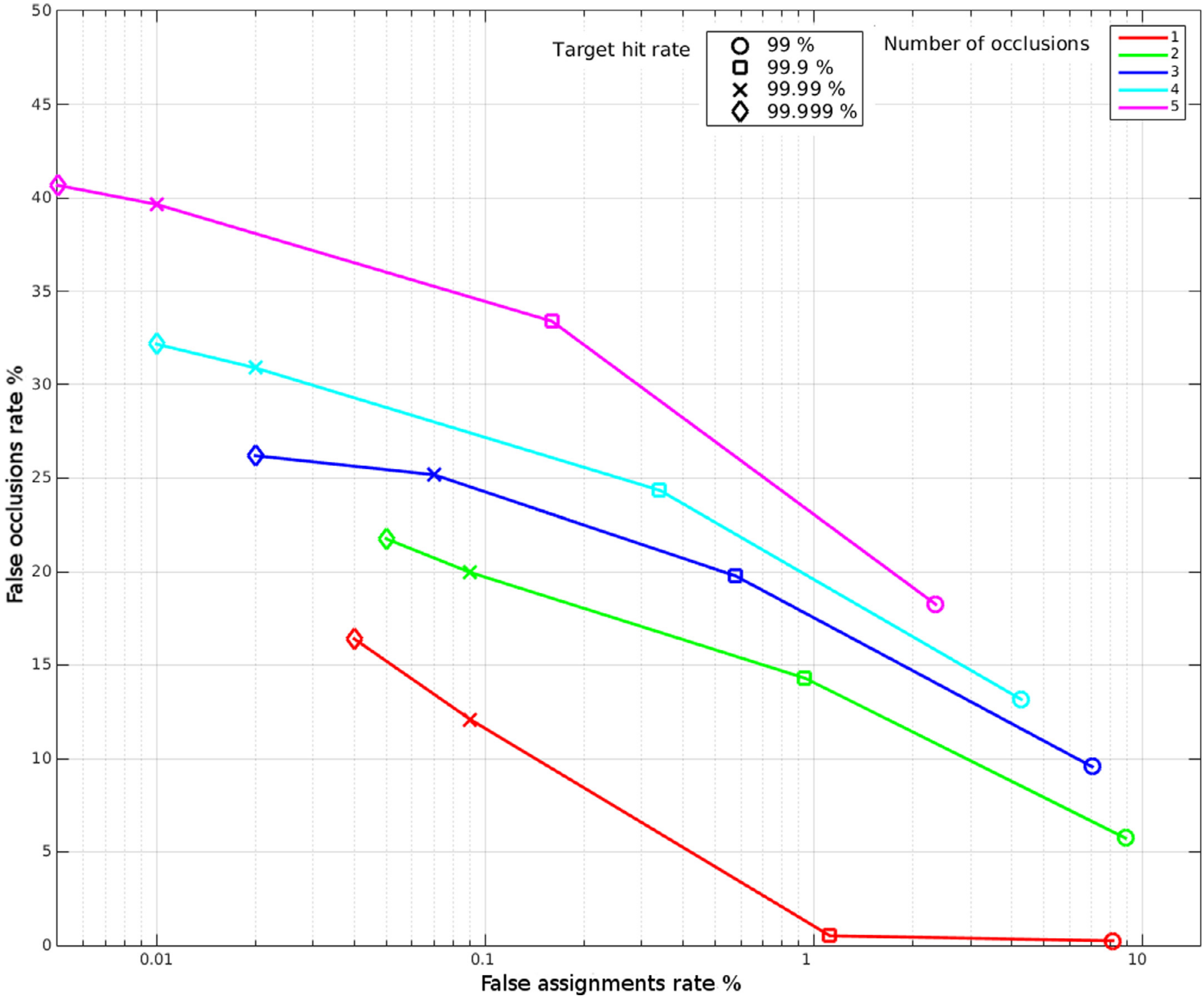


Fig. 5. Graph: false assignments and false occlusions under different test conditions.

Table 2
Experimental conditions summary.

Test conditions	
Number of markers	15
Target hit rate	99.99%
Target failure rate	0.01%
Occlusions per frame	4
Number of test frames	16, 384

Table 3
False assignments (FA) and false occlusions (FO) results. Rows correspond to model markers located over parts of the body.

Marker ID	FA #	FA % (%)	FO #	FO % (%)
r_asis	2	0.02	300	6.38
l_asis	2	0.02	254	5.55
s2	0	0.00	286	6.11
r_l_thigh	0	0.00	3564	45.03
l_l_thigh	1	0.01	220	4.85
r_knee	0	0.00	1912	30.03
l_knee	1	0.01	3532	44.58
r_calf	1	0.01	194	4.20
l_calf	3	0.03	218	4.71
r_ankle	3	0.03	2302	34.02
l_ankle	1	0.01	4348	49.30
r_heel	4	0.05	3195	42.14
l_heel	1	0.01	3579	45.06
r_toe	4	0.04	2332	34.67
l_toe	4	0.05	3627	45.13
Average	1.8	0.02	1991	31.16

given in Table 2. The frames are extracted from a gait measurement experiment, so that markers correspond to the lower limbs of the human body.

In Table 3 the rates of false assignments and false occlusions are shown for a training and validation instances where the target marker hit rate was set to 99.99% and the number of occluded points per frame was set to 4 for a model of 15 markers. While the false assignments stands around the 0.01%, the rate of unassigned markers (despite being present in the candidate list) fluctuates from 4.20% to 45.13% with an average of 31.16%. Broadly speaking, some markers are harder to *catch* with high confidence when the rate of actual occlusions reaches the 25%.

Repeating the above test with different target hit rates and different number of simulated occlusions, we see the variation of the

Table 4
False assignments sensitivity to target marker hit rate and number of occlusions.

False assignments rate					
Target marker hit rate		99.000 (%)	99.900 (%)	99.990 (%)	99.999 (%)
Number of true occlusions per frame	1	8.13	1.12	0.09	0.04
	2	8.89	0.94	0.09	0.05
	3	7.04	0.58	0.07	0.02
	4	4.28	0.34	0.02	0.01
	5	2.35	0.16	0.01	0.00

Table 5
False occlusion sensitivity to target marker hit rate and number of occlusions.

False occlusions rate					
Target marker hit rate		99.000 (%)	99.900 (%)	99.990 (%)	99.999 (%)
Number of true occlusions per frame	1	0.25	0.52	12.10	16.40
	2	5.74	14.30	19.96	21.76
	3	9.56	19.76	25.18	26.20
	4	13.16	24.35	30.90	32.17
	5	18.25	33.39	39.65	40.67

efficiency indicators. The sensitivity of the false assignments rate (see Table 4), for a constant number of simulated occlusions (the rows), when the target hit rate increases (along the columns) the algorithm reduces dramatically the number of false assignments. Likewise, the rate of false occlusions gets bigger (Table 5, right).

We plot these numbers in Fig. 5. Each line corresponds to the same number of simulated occlusions, while the dot symbol corresponds to a given target hit rate. Low false assignment rates (x axis) correspond to high false occlusions rate. On the other hand, when the number of simulated occlusions gets bigger, the rate of false occlusions increases as well.

The reason behind this behavior is the following. When it is not possible to formulate assignments due a lack of information (occlusions), the weak classifiers can't be evaluated and consequently the strong classifier loses its strength. The intuitive interpretation is that the identification of each single marker depends on the identification of the others. Indeed, the markers themselves act as a community where the identity of a member is backed up by its peers. If too many of them are missing, we just can't tell the identity of the remaining ones.

7. Conclusions and further work

Our work is directed towards the labelling of labelling of clouds of points against a model given by a collection of marker points with direct application to human body motion analysis. Though the marker points may have a priori relations due to anatomical constraints and can be tracked in time when we perform videosequence analysis, we discard using this information, aiming to achieve the observed points labelling on the basis of their geometrical relations only. Therefore, each frame in the videosequences are independently analysed. We developed a classification based approach that was able to carry out the labelling in the case of no occlusions up to a prespecified false detection accuracy. In this paper we have extended the approach to the case of occlusions, reaching also prespecified false detections rates. The empirical results on simulated occlusions show that our system is rather robust with very low false occlusion and false labelling rates. The system can be easily tuned to different human motion scenarios in offline training that may require several hours in a conventional off-the-shelf current laptop computer. Actual labelling can be carried out in realtime in the same kind of conventional computers. Another advantage of our approach is that labeling errors do not propagate in the time sequence, so that recovery from occlusions is immediate. A main line for improvement of our system is the speedup and efficiency increase of the search for the ensemble of partial solvers, for instance by fine tuning of the genetic algorithm selecting the optimal population of partial solvers. Finally, it may be interesting to use the algorithm to test the optimality of the distribution of markers over the body, as a design quality measure tool. Bad designs lead to ambiguities in the labeling, therefore the inability to achieve a labelling can be understood as a poor design of marker placements.

Acknowledgments

The work in this paper has been partially supported by FEDER funds for the MINECO project TIN2017-85827-P, and projects KK-2018/00071 and KK-2018/00082 of the Elkartek 2018 funding program of the Basque Government. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 777720.

References

[1] Y. Akiyama, H. Toda, T. Ogura, S. Okamoto, Y. Yamada, Classification and analysis of the natural corner curving motion of humans based on gait motion, Gait Posture 60 (2018) 15–21, doi:10.1016/j.gaitpost.2017.10.008.

[2] A.M. Aurand, J.S. Dufour, W.S. Marras, Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume, J. Biomech. 58 (2017) 237–240, doi:10.1016/j.jbiomech.2017.05.006.

[3] M. Barnachon, S. Bouakaz, B. Boufama, E. Guillou, Ongoing human action recognition with motion capture, Pattern Recognit. 47 (1) (2014) 238–247, doi:10.1016/j.patcog.2013.06.020.

[4] B. Chen, H. Sun, G. Xia, L. Feng, B. Li, Human motion recovery utilizing truncated schatten p-norm and kinematic constraints, Inf. Sci. 450 (2018) 89–108, doi:10.1016/j.ins.2018.02.052.

[5] A. Cutti, A. Ferrari, P. Garofalo, M. Raggi, A. Cappello, A. Ferrari, Outwalk: a protocol for clinical gait analysis based on inertial and magnetic sensors, Med. Biol. Eng. Comput. 48 (1) (2010) 17–25.

[6] O. Faugeras, Three-Dimensional Computer Vision: a Geometric Viewpoint, MIT Press, Boston, MA, 1993.

[7] A. Ferrari, A. Cutti, P. Garofalo, M. Raggi, M. Heijboer, A. Cappello, A. Davalli, First in vivo assessment of outwalk: a novel protocol for clinical gait analysis based on inertial and magnetic sensors, Med. Biol. Eng. Comput. 48 (1) (2010) 1–15.

[8] P.G. Guerra-Filho, Optical motion capture: theory and implementation, J. Theor. Appl. Inf. 12 (2) (2005) 61–89.

[9] J. Heikkila, O. Silven, A four-step camera calibration procedure with implicit image correction, in: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), 1997, p. 1106.

[10] J. Jimenez-Bascones, M. Graña, An ensemble of weak classifiers for pattern recognition in motion capture clouds of points, in: M. Kurzynski, M. Wozniak, R. Burduk (Eds.), Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017, Springer Nature, 2017, pp. 201–210.

[11] M. Kok, J.D. Hol, T.B. Schön, An optimization-based approach to human body motion capture using inertial sensors, IFAC Proc. Volumes 47 (3) (2014) 79–85. 19th IFAC World Congress. doi: 10.3182/20140824-6-ZA-1003.02252.

[12] H.W. Kuhn, The hungarian method for the assignment problem, Naval Res. Logist. Q. 2 (1–2) (1955) 83–97.

[13] A.-A. Liu, Y.-T. Su, W.-Z. Nie, Z.-X. Yang, Jointly learning multiple sequential dynamics for human action recognition, PLoS ONE 10 (7) (2015) e0130884.

[14] H. Liu, L. Wang, Human motion prediction for human-robot collaboration, J. Manuf. Syst. 44 (2017) 287–294. Special Issue on Latest advancements in manufacturing systems at NAMRC 45. doi: 10.1016/j.jmsy.2017.04.009.

[15] D. Lunardi Flam, D. Pacheco de Queiroz, T. Alves de Souza Ramos, A. de Albuquerque Araujo, J. Boechat Gomide, Openmocap: an open source software for optical motion capture, in: Proceedings of the VIII Brazilian Symposium on Games and Digital Entertainment, 2009, pp. 151–161.

[16] J. Maycock, T. Rohlig, M. Schroder, M. Botsch, H.J. Ritter, Fully automatic optical motion tracking using an inverse kinematics approach, in: Proceedings of the International Conference on Humanoids, 2015, pp. 461–466.

[17] M. Mehling, Implementation of a low cost marker based infrared light optical tracking system, Ph.D. thesis, Institute for Software Technology and Interactive Systems, 2006.

[18] J. Meyer, M. Kuderer, J. Muller, W. Burgard, Online marker labeling for fully automatic skeleton tracking in optical motion capture, in: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 5652–5657.

[19] T.B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vis. Image Underst. 104 (2) (2006) 90–126. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour. doi: 10.1016/j.cviu.2006.08.002.

[20] G. Nagymáté, T. Tuchband, R.M. Kiss, A novel validation and calibration method for motion capture systems based on micro-triangulation, J. Biomech. (2018), doi:10.1016/j.jbiomech.2018.04.009.

[21] A. Napoli, S. Glass, C. Ward, C. Tucker, I. Obeid, Performance analysis of a generalized motion capture system using microsoft kinect 2.0, Biomed. Signal Process. Control 38 (2017) 265–280, doi:10.1016/j.bspc.2017.06.006.

[22] F. Patrona, A. Chatzitofis, D. Zarpalas, P. Daras, Motion analysis: action detection, recognition and evaluation based on motion capture data, Pattern Recognit. 76 (2018) 612–622, doi:10.1016/j.patcog.2017.12.007.

[23] M. Qiao, J. Cheng, W. Bian, D. Tao, Biview learning for human posture segmentation from 3d points cloud, PLoS ONE 10 (7) (2014) e0130884.

[24] T. Schubert, A. Gkogkidis, T. Ball, W. Burgard, Automatic initialization for skeleton tracking in optical motion capture, in: Proceedings of the International Conference on Robotics and Autom. (ICRA), 2015, pp. 734–739.

[25] I. Siddiqui, G. Remington, P.J. Fletcher, A.N. Voineskos, J.W. Fong, S. Saperia, G. Fervaha, S.D. Silva, K.K. Zakzanis, G. Foussias, Objective assessment of exploratory behaviour in schizophrenia using wireless motion capture, Schizophr. Res. (2017), doi:10.1016/j.schres.2017.09.011.

[26] A.H. Smeragliuolo, N.J. Hill, L. Disla, D. Putrino, Validation of the leap motion controller using markered motion capture technology, J. Biomech. 49 (9) (2016) 1742–1750, doi:10.1016/j.jbiomech.2016.04.006.

[27] A. Szczęsna, P. Pruszowski, P. Skurowski, E. Lach, J. Słupik, D. Pęszor, M. Paszkuta, A. Polański, K. Wojciechowski, M. Janiak, K. Lebek, Inertial motion capture costume, Procedia Technol. 27 (2017) 139–140. Biosensors 2016. doi: 10.1016/j.protcy.2017.04.061.

[28] C. Vaughan, L. Brian, J. Connor, Dynamics of human gait, Kiboho Publishers, Cape Town, South Africa, 1999.

[29] P. Viola, M. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.

[30] Q. Yu, Q. Li, Z. Deng, Online motion capture marker labeling for multiple interacting articulated targets, Comput. Gr. Forum 26 (3) (2007) 477–483.

[31] J. Zhang, Y. Cao, M. Qiao, L. Ai, K. Sun, Q. Mi, S. Zang, Y. Zuo, X. Yuan, Q. Wang, Human motion monitoring in sports using wearable graphene-coated fiber sensors, Sensors Actuat. A: Phys. 274 (2018) 132–140, doi:10.1016/j.sna.2018.03.011.

[32] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334.

Juan L Jiménéz Bascones was born in San Sebastian, Spain, in 1975. He received the degree in industrial engineering from the University of Navarra, 1998. Final degree project on Photogrametry and Biomechanics. Completed the Ph.D. courses in the Public University of Navarra, 2003. Final dissertation on dense 3D scanning and surface reconstruction. From 1998 to 2017 held a position as software developer in the Motion Capture Unit at STT. Involved in multiple projects in charge of machine vision, camera calibration, marker tracking, 3D scanning, IMU calibration and data processing, software architecture. Since 2016 he is working towards his Ph.D degree in the Computational Intelligence Group from the Basque University (UPV/EHU). His main interests are focused in machine learning and numerical analysis algorithms.

Manuel Graña Romay received the M.Sc. and Ph.D. degrees from Universidad del Pais Vasco (UPV/EHU), Donostia, Spain, in 1982 and 1989, respectively, both in computer science. His current position is a Full Profesor (Catedrico de Universidad) with the Computer Science and Artificial Intelligence Department of the Universidad del Pais Vasco (UPV/EHU). He is the head of the Computational Intelligence Group (Grupo de Inteligencia Computational). His current research interests are in applications of computational intelligence to linked multicomponent robotic systems, medical image in the neurosciences, multimodal human computer interaction, remote sensing image processing, content based image retrieval, lattice computing, semantic modelling, data processing, classification, and data mining.

Jose Manuel Lopez-Guede, Ph.D. is an assistant professor at the Universidad del Pais Vasco (UPV/EHU). He received Ph.D. in 2012 in the UPV/EHU His research interests include robotics and reinforcement learning as well as machine learning applications to health care data.