

# Automatic Parameter Configuration: A Case Study on a Rehabilitation Oriented Human Limb Tracking Algorithm

Oscar L. VERA-PÉREZ<sup>1</sup>, Alejandro MESEJO-CHIONG<sup>1</sup>,  
Antoni JAUME-I-CAPO<sup>2</sup>, Manuel GONZÁLEZ-HIDALGO<sup>2</sup>

<sup>1</sup> Facultad de Matemática y Computación, Universidad de la Habana, Cuba

<sup>2</sup> Unitat de Gràfics,

Visió i Intel·ligència Artificial. Departament de Ciències Matemàtiques i Informàtica,

Universitat de les Illes Balears, Spain

antoni.jaume@uib.es

**Abstract:** The choice of the parameters has strong influence on the quality of the results obtained by the application of algorithms. Most researchers tend to select the values of their parameters in long and tedious trial and error approaches. Although, some methods have been developed for automatic parameter selection, they have not been widely used in the computer vision area. This paper presents the design of a general purpose framework for automatic parameter selection through a case study: a human limb tracking algorithm developed for applications that will be used in rehabilitation scenarios with low cost equipment. The tracking algorithm first detects the limb by using a skin segmentation approach, then the position of an idealized limb model is updated using Simulated Annealing. The framework for automatic parameter selection treats each parameter from the tracking algorithm according to its domain and uses a modified version of Harmony Search Optimization algorithm that includes a dominance criterion. The obtained results are presented as well and show that selected parameters behave well for the case of study.

**Keywords:** Automatic parameter selection; Harmony Search Optimization; Simulated Annealing; Human limb tracking.

## 1. Introduction

The choice of the parameters has strong influence on the behavior of algorithms. In most cases the stability of parametrized algorithms and the quality of obtained results are heavily affected by parameters. Finding the right parameter values, when solving a specific kind of problem with certain algorithm, is a very difficult task and even harder in the presence of a non-deterministic algorithm such as metaheuristics.

Some authors state that most of the time spent in the test and design of some metaheuristic algorithm is devoted to find good parameter configurations that lead to acceptable results. Many researchers tend to fine-tune the parameters of their algorithms using some long and tedious trial and error approaches or they simply take previously reported parameter values without any analysis on whether those parameters are suitable or not for their specific cases. Just a few make correct use of experimental designs for the parameter selection matter [1].

A recently developed algorithm for human limb tracking [22] based on the Simulated Annealing metaheuristic [14], immersed of course in the computer vision discipline, made use of a general purpose framework for automated parameter selection with good results. We

designed this algorithm to be used in rehabilitation sessions of patients with physical disabilities, in particular persons that have little mobility in their shoulders. In these sessions, the patients should perform repetitive movements with their limbs in an indoor environment with controlled illumination and background. Also, the equipment for practical implementation is expected to be low cost, probably only a web cam and a regular PC. The tracking algorithm can facilitate progress analysis of the patients' condition and the implementation of new rehabilitation systems [12], such as *serious games*.

As it is very unlikely that it can be found a parameter configuration that works well for every possible scenario of application, the used general purpose framework obtains good parameters in a supervised learning way using a representative set of problem instances that resemble the conditions in which the tracking algorithm should work. According to a dominance definition, non-dominated parameter vectors are selected using a modified version of Harmony Search Optimization algorithm [26] that includes those dominance notions and takes advantage of the domain of definition for every parameter. The complete selection process may be viewed as an automatic experimentation phase that aids to perform deeper analysis of the algorithms being configured, in this case, the proposed tracking

algorithm optimizing the relationship between the image processing, the capture device, and the rehabilitation motions.

In the present paper, we expanded previous work [22] by presenting in detail the specific tracking problem definition, the designed tracking algorithm and the automated parameter selection strategy used in order to get good results on the scenario of application. First, we give a suitable tracking problem definition. Next, a review of related work is presented. After that, the tracking algorithm is explained and the last sections are devoted to relate the details of automatic parameter selection and the analysis of the obtained experimental results. Finally some conclusions of the presented work are discussed.

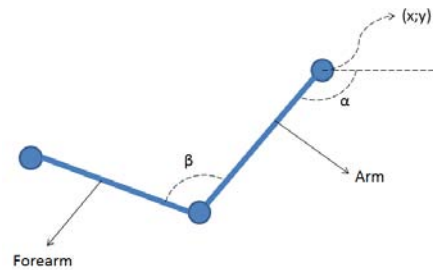
## 2. Tracking Definition Problem

As stated before, the goal of the tracking algorithm is to aid rehabilitation sessions and to provide useful information for patients' progress analysis. Therefore is designed for real-time and offline human limb tracking.

The limb, in this case an arm, shall be modelled as a kinematic chain with two degrees of freedom as can be seen in Figure 1. One corresponds to the shoulder articulation and the other representing the elbow. The amplitude of the latter shall be constrained with respect to the upper arm to match only possible positions. Each configuration of  $K$  shall be characterized by a tuple in which the first two components are the position of the starting point of the chain, that is, the shoulder, and the rest are the amplitudes of the degrees in the articulations. It is assumed that all movement occur in the same plane due to the simple nature of the exercise to be performed by the patients, so the kinematic chain is represented in a two-dimensional space. The length of the segments that conform  $K$  may be personalized for each patient or some standard measures may be assumed.

Having a video sequence composed of images  $S = \{I_i: 0 \leq i \leq N\}$   $S = \{I_i: 0 \leq i < N\}$  ( $N$  not necessarily known, because the tracking could be performed in real-time applications) and let be the configuration of  $K$  on the image  $I_{i-1}$ , the problem of limb tracking is to obtain the right configuration  $S_i = (x_i, y_i, \alpha_i, \beta_i)$  for the image  $I_i$ . First, it shall be assumed that the configuration for is known and the origin of the kinematic chain never changes in the whole

sequence. This second assumption is related to the nature of the exercises the patients have to do in therapy sessions, where a movement of the shoulder is considered invalid and shall be detected by other means. All sessions take place in an indoor environment with controlled illumination and background making limb recognition easier.



**Figure 1.** Graphical representation of the kinematic chain used as limb model.

The limb-tracking problem, as described, may be stated as an optimization problem considering, for each frame, the maximization of the spatial coincidence between the model and the limb. This coincidence will be computed using two binary images: one obtained from the model configuration representing an approximated solution, and another with the silhouette of the tracked limb obtained by the detection of skin regions. This spatial coincidence is modeled using a function of the kinematic chain angles, see Figure 1, in order to maximize the overlap between the projection of kinematic chain into the image of the limb. A detailed account of this can be found in [13] and in section 4 for this particular case. As there is no useful analytic expression for the objective function, an implementation of a metaheuristic seems appropriate. In this case Simulated Annealing is employed, having the fact that it is well known, easy to implement in practice and its local search strategy is suitable for the pose updating process through the complete sequence by using the position obtained in the previous frame as initial solution for the search in the next image.

## 3. Related Work

Automated parameter tuning or selection, also referred as algorithm configuration, has received some attention in the last decades but not as much as it deserves, being an important topic since the natural influence that parameters have on the final results obtained by the application of developed algorithms. One of the

first works on the subject is the design of MULTI-TAC [17], a platform that generates tuned heuristic procedures. The use of experimental designs has been exploited in [6] combined with gradient descent techniques and in [1] by developing CALIBRA, a procedure that employs Taguchi's fractional experimental design in order to reduce parameter search space. The most promising approaches to the algorithm configuration problem are presented under the name of F-Race [3] and Paramils [11], the first make use of Friedman test and the later performs iterated local search in parameter vector space. Also hyperheuristics [5, 24] and reactive algorithms [2] have been proposed as an alternative to algorithm configuration. In the field of computer vision, little attention has been given to automate parameter tuning; the few efforts reported are more focused on operator or function selection than parameter selection [4, 15, 23]. In our concern, there is no approach to the subject of parameter configuration applied to any tracking algorithm.

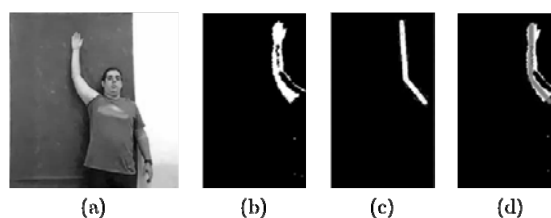
Also none of the previously cited works on automatic parameter configuration make any distinction according to parameter domain when generating new values; they treat all parameters as discrete or continuous with consequent round errors and domain sampling. This issue is avoided by the procedure used and described in the present paper. Its implementation relies on Harmony Search Optimization algorithm that provides different generation strategies according to parameter domain.

As for human motion tracking methods, a very complete taxonomy along with an extensive paper review can be found in [18, 19]. Also in [20] several papers are cited that define the tracking problem as a global optimization problem and employ particle filtering, Monte Carlo, tree-based and gradient descent techniques to solve it. In [8] the tracking problem is conveniently defined as “*an optimization problem where one searches for the pose that minimizes a previously defined error function based on some image features*” and a new algorithm based on interactive particle filtering, similar to Simulated Annealing, is proposed. More on the link between optimization and tracking can be found in [7], where a framework for combining stochastic optimization, filtering, and local optimization is discussed. The use of human

pose tracking has been widely used in rehabilitation applications, as can be seen in [12, 16, 27].

#### 4. Simulated Annealing Based Human Limb Tracking Algorithm

For each frame or image  $I_i (1 \leq i \leq N)$  in the sequence and knowing the configuration of the kinematic chain at frame, the proposed solution addresses the problem of finding the correct position of the chain at instant, that is. To achieve that goal, the silhouette of the limb is first separated from background using a skin classification/segmentation algorithm. The choice of this algorithm depends heavily on environment conditions such as background color and illumination. In a controlled scenario even the simpler and faster techniques may be used with acceptable accuracy, and so on. A good review on skin detection and classification algorithms can be seen in [25]. Several approaches are listed: from simple classification rules in RGB color space or Bayes classifiers, to neural network techniques using Self Organizing Maps and others with different degrees of accuracy. In the case that people cannot wear short sleeve shirts and skin detection is not suitable, an elastic band of singular color may be used and could be detected with some chroma keying approach. The result of this step is a binary image with limb region marked in white (see Figure 2).



**Figure 2.** a) A frame in the input video sequence. b) Binary image resulted from skin features detection. c) Binary image resulted from graphical representation of a configuration of the kinematic chain. d) Spatial coincidence between detected skin features and the kinematic chain configuration.

A configuration of can also produce a binary image by simply drawing an idealized arm with rectangles, ellipses or just lines as in Figure 2c. The two binary images (Figures 2b and 2c) can be compared according to a certain spatial coincidence measure. This measure evaluates the ratio between the intersection and union of both white and black regions in the two binary images [13]. Let  $I_L, I_K$  be the binary images of

the limb and kinematic chain respectively, and let  $I_L^W, I_K^W$  their respective white regions, the spatial coincidence is defined as:

$$\rho(S_K, I_L) = \frac{|I_L^W \cap I_K^W|}{|I_L^W \cup I_K^W|} \quad (1)$$

where  $|\cdot|$  is the number of white pixels in the image or the number of pixels in the region of interest, in this case skin pixels or those pixels belonging to the graphical representation of the limb model.

The two regions union and intersection are computed by superimposing one image on the other. The value of the measure is a real number in interval where is total coincidence and total disparity. Then, the right configuration at instant of can be found by maximizing the value of the defined measure between the silhouette of the limb at frame  $I_i$  and subsequent guesses of. So tracking process is achieved by maximizing the spatial coincidence on each frame.

The optimization problem on each frame will be solved using Simulated Annealing. This is a simple metaheuristic with broad application and good practical results reported, designed for global optimization problems and inspired in statistical mechanics. The general idea of the variation of this algorithm used in the present work is showed in Algorithm 1.

```

Algorithm 1: Pseudocode of adapted Simulated Annealing
input :
  I_i // Current image
  s^0 ← (x_{i-1}, y_{i-1}, α_{i-1}, β_{i-1}) // Previous limb position as
initial solution
  T^0 // Initial temperature
  T^f // Final temperature
  α // Cooling rate factor
  r // Repetitions at same value of temperature
  ar // Arm variation range
  fr // For arm variation range
output: Best overall solution found
begin
  t ← T^0 // Setting current temperature
  s ← s^0 // Setting current state or solution
  s^b ← s // Setting best solution found
  while t > T^f do
    for j = 1 ← to r do
      s^j ← GetNeighbor(s, ar, fr)
      // Generate neighbor solution
      if E(s^j, I_i) < E(s^b, I_i) then
        s^b ← s^j
      if E(s^j, I_i) < E(s, I_i) or with probability
      Prob(E(s^j, I_i), E(s, I_i), t) then
        s ← s^j
    t ← t * α
  return s^b

```

Starting from an initial solution or state, this algorithm explores new candidate solutions using a neighborhood relation. A new solution is accepted as the next state if it is better than the current or with certain probability if is

worse. The comparison of two solutions and probability of acceptance or worse solutions involves an *energy* function. The overall process tries to minimize this energy controlled by a global value called *temperature*. This global value is decreased as the process advances using an *annealing schedule*.

In this case, the temperature is decreased geometrically by a cooling rate factor (*cr*) and for each value of temperature reached; a certain number of iterations (*r*) is repeated. As probability of acceptance, a commonly used function will be employed:

$$Prob(E^n, E, t) = \exp\left(\frac{E-E^n}{t}\right) \quad (2)$$

The energy function  $E^n$  will be based on the spatial coincidence measure defined in Equation 1 and will be computed as

$$E(S, I) = 1 - \rho(S, I) \quad (3)$$

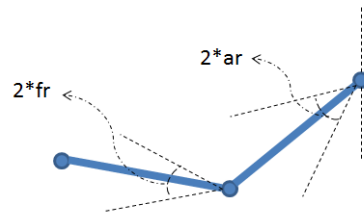
and  $E^n = E(S^n, I_i)$  where  $I_i$  is the current frame and  $S^n$  is a solution candidate.

```

Algorithm 2: Generation of neighbor solutions
function GetNeighbor((x, y, α, β), ar, fr)
  with probability 0.5
    m ← max(π/2, α - ar)
    M ← min(3 * π/2, α + ar)
    return (x, y, m + (M - m) * rand(0,1), β)
  otherwise
    return (x, y, α, β + fr * (2 * rand(0,1) - 1))

```

The *GetNeighbor* function (see Algorithm 2) will build neighbor solutions by randomly varying one of the amplitudes of the freedom degrees in the configuration. The variation will be bounded by two additional parameters, which will reduce the search space of new configurations around the initial solution by setting a velocity constraint on the movement of the kinematic chain. In Figure 3, a graphical representation of the search space is shown.



**Figure 3.** Graphical representation of the search space for new neighbor generation.

As an initial solution for each run of Simulated Annealing, the best configuration found for the previous frame in the sequence will be used. The vector parameter for an execution of our Simulated Annealing Algorithm will be

denoted by  $\theta, (S^0, T^0, T^f, cr, r, ar, fr)$ . Where  $S^0$  is the previous limb position as initial solution,  $T^0$  is the initial temperature,  $T^f$  is the final temperature,  $cr$  is the cooling rate factor,  $r$  is the repetitions at the same value of temperature,  $ar$  is the arm variation range, and  $fr$  is the forearm variation range.

For a sequence  $S = \{I_i: 0 \leq i \leq N\}$  Simulated Annealing is executed  $N$  times in order to find the right configuration for  $I_i (1 \leq i \leq N)$ . The best configuration found for the previous frame in the sequence will be used as the initial solution for each run. Finally the result will be a sequence of configurations  $R = \{(x_i, y_i, \alpha_i, \beta_i): 1 \leq i \leq N\}$  being them the obtained solution for each frame in  $S$ .

## 5. Automatic Parameter Value Selection

The proposed tracking algorithm takes several parameters: initial and final temperature, cooling rate factor, number of repetitions at a same value of temperature, and the two bounds for arm and forearm amplitudes variation. All these parameters should be properly adjusted for a particular implementation in an application scenario. In this case a general-purpose strategy has been used.

This framework makes use of the particularity of the definition domain for each parameter, it being discrete or continuous. In this case there are 5 real parameters and only the repetitions at a same value of temperature are set by an integer parameter. All domains are considered bounded by upper and lower values that were roughly taken by simple inspection and are shown in Table 1. The first column shows the name of the parameters and the rest shows lower and upper values for their domain of definition. Values for angle variation bounds are given in radians.

The general procedure for automatic parameter selection treats the performance of an algorithm's execution as a function of its parameters and tries to minimize it using a Harmony Search Optimization algorithm [9, 10, 26]. This is a young metaheuristic inspired, according to its creator, in music improvisation. This algorithm maintains a matrix of candidate solutions named memory and elegantly combines strategies of solution space exploration and the exploitation of the solutions

included in this memory in order to create new candidates. It has been successfully used in many research areas [10]. In the present work the reviewed Harmony Search Optimization algorithm proposed in [26] is used with a modification to include dominance criteria.

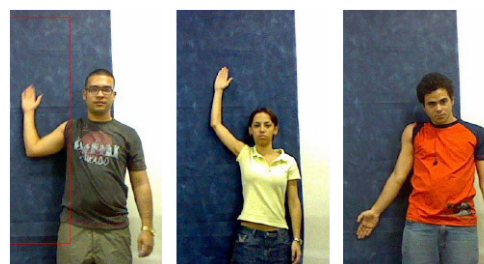
**Table 1.** Domain of definition for each parameter of the tracking algorithm

Parameter	Lower bound	Upper bound	
$T^0$	5	10	Continuous
$T^f$	0.1	1	Continuous
$cr$	0.7	1	Continuous
$r$	1	10	Discrete
$ar$	0.1	1.5	Continuous
$fr$	0.1	1.5	Continuous

The performance of the tracking algorithm will be evaluated quantitatively by the worst energy value obtained in the sequence for a frame. Let  $p(S, \theta)$  be the performance of an application of the tracking algorithm to a sequence with parameter vector obtaining a sequence of configurations, then:

$$\rho(S, \theta) = \max_R \{1 - \rho((x_i, y_i, \alpha_i, \beta_i), I_i)\} \quad (4)$$

The selection is done offline using a training set of typical problem instances. For that, three small video sequences of no more than 10 secs. are taken using subjects with different physical characteristics (see Figure 4) in the same conditions of illumination, with the same equipment and performing the same kind of movement with the right arm. To simplify skin detection phase, all limb movements are done in front of a dark background and skin areas are detected using simple rules considering pixel values in RGB space. This training set will be denoted as  $T = \{S_1, S_2, S_3\}$ .



**Figure 4.** Training sequence set for parameter selection.

Due to the non-deterministic nature of Simulated Annealing different runs of the tracking algorithm on the same sequence with same parameters will regularly produce different results. Therefore several runs are

needed for each element in the training set and each parameter vector.

After runs of the tracking algorithm over a sequence with parameter vector and same number of runs over same sequence with parameter vector  $\theta_2$ , and being  $p_i(S, \theta)$  the performance of  $i^{th}$  run over the sequence  $S$  with parameter vector  $\theta$ , then it is said that dominates when

$$\frac{1}{M} \sum_{i=1}^M p_i(S, \theta_1) < \frac{1}{M} \sum_{i=1}^M p_i(S, \theta_2) \quad (5)$$

On the contrary, when

$$\frac{1}{M} \sum_{i=1}^M p_i(S, \theta_1) > \frac{1}{M} \sum_{i=1}^M p_i(S, \theta_2) \quad (6)$$

then dominates  $\theta_1$ , otherwise they are non-dominated.

A parameter vector  $\theta_1$  dominates over the training set if it dominates the second in most sequences. If two parameter vectors dominate each other in same number of sequence then they are non-dominated over the entire set. So, the task of the parameter-adjusting algorithm is to find non-dominated parameter vectors over the training set. The selection algorithm behaves, in general lines, is shown in Algorithm 3.

Algorithm 3: Pseudocode of adapted Harmony Search Optimization algorithm for parameter selection

```

input :
    HMS // Size of memory
    M // Number of times tracking algorithm should be run on
    a sequence with a parameter vector
    NI // Number of iteration the selection algorithm should
    perform
output: Non-dominated parameter vectors in memory
begin
    Generate a random set of parameter vectors or memory of size HMS
    foreach  $\theta$  in memory and S T do
        Run tracking algorithm M times with  $\theta$  over S
    for NI times do
        Generate a new parameter vector  $\theta'$ 
        foreach S T do
            Run tracking algorithm M times with  $\theta'$  over S
            if exists a parameter vector  $\theta$  in memory worse than  $\theta'$  then
                Substitute  $\theta$  for  $\theta'$  in memory
    return non-dominated vectors in memory

```

This algorithm keeps over the entire process a set of candidate solutions named memory as Harmony Search does. In each iteration, a new vector is generated following the same criterion the original metaheuristic proposes. In order to compare the new vectors with the elements in memory, the dominance relation defined before is used, so a vector in memory is said to be worse than a new vector  $\theta'$  if is dominated by the latter and is dominated by a larger number of vector in memory than those dominating. After a parameter selection takes place, a total of vectors are analyzed and tracking algorithm is run  $M * T \vee (HMS +)$  times, where  $HMS$  is

the Harmony Search size of memory,  $M$  is the number of times tracking algorithm should be run on a sequence with a parameter vector,  $NI$  is the number of iterations.

## 6. Experimental results

As the first step in experimentation the previously shown training set of video sequences was used to learn the parameters of the tracking algorithm. Several executions of the parameter selection method were run in order to vary the number of parameter vector analyzed, number of evaluation on one video sequence and the size of the memory kept by Harmony Search.

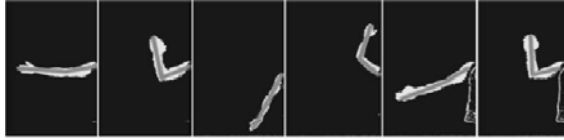
The complete selection process lasted several hours in a computer with Intel Core i5-M430 processor and 4GB RAM with graphical shared memory. A total of 44 parameter vectors were selected over 420 analyzed on 12 experimental designs, each of them executed 3 times making 36 runs of parameter selection algorithm in total. The experimental designs were different configurations on the number of iterations, size of memory in Harmony Search and the number of evaluation of the tracking algorithm over each sequence.

The different experimental designs and the mean time over their respective execution for parameter selection on the training set are shown in Table 2.

Exp.	HMS	NI	M	Mean execution time
1	3	12	3	00:21:34.1970000
2	7	8	3	00:11:28.3880000
3	7	23	3	00:40:19.1690000
4	15	15	3	00:23:41.5570000
5	15	45	3	01:13:37.6790000
6	30	30	3	00:53:24.2240000
7	3	12	6	00:37:30.3870000
8	7	8	6	00:32:01.4490000
9	7	23	6	01:08:24.7970000
10	15	15	6	00:51:20.8210000
11	15	45	6	02:06:08.2820000
12	30	30	6	01:56:08.7620000

**Table 2.** Experimental design for parameter value selection. Parameters for Harmony Search Optimization algorithm are given varying memory size (HMS), iterations (NI) and number of evaluation per problem (M). Also mean execution time among runs of a same experiment is given for each case (hours : minutes : seconds) .

Visual results of the selected parameter vectors were good in general for the training set as can be seen in Figure 5, also numerical results showed acceptable values for the mean worst energy value obtained with each vector on each sequence.



**Figure 5.** Samples of results obtained for the training set using selected parameters.

Table 3 shows the mean and standard deviation of the numerical results, related to the values of, obtained by the tracking algorithm using the selected parameters grouped by video sequence on the training set and experimental design. Table 4 summarizes the results showed in Table 3 grouped by experimental design.

**Table 3.** Numerical results of the performance of selected parameters on each experiment over training set sequences. Mean and standard deviation of  $\rho(S_K, I_L)$  among runs are given grouped by experiment.

Exp.	Sequences on training set T					
	1		2		3	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
1	0.7298	0.0049	0.6932	0.0135	0.7567	0.0353
2	0.7379	0.0145	0.6982	0.0146	0.7612	0.0389
3	0.7301	0.0042	0.6798	0.0045	0.7336	0.0015
4	0.7309	0.0071	0.6846	0.0107	0.7375	0.0096
5	0.7284	0.0029	0.6797	0.0020	0.7330	0.0011
6	0.7291	0.0039	0.6823	0.0045	0.7348	0.0032
7	0.7381	0.0142	0.6959	0.0212	0.7461	0.0211
8	0.7293	0.0036	0.6879	0.0093	0.7460	0.0247
9	0.7283	0.0037	0.6825	0.0061	0.7351	0.0051
10	0.7331	0.0082	0.6845	0.0071	0.7480	0.0272
11	0.7286	0.0032	0.6819	0.0055	0.7358	0.0126
12	0.7323	0.0079	0.6825	0.0053	0.7482	0.0241

**Table 4.** Numerical results of the performance of selected parameters grouped by sequence of the training set (Mean and standard deviation of  $\rho(S_K, I_L)$ ).

Sequence	Mean	Std. Dev.
1	0.7320	0.0093
2	0.6868	0.0134
3	0.7427	0.0217

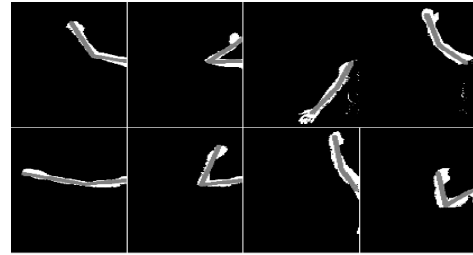
In order to check their performance, all 44 selected parameter vectors were used in six new different sequences (see Figure 6). These sequences were similar to those used as in training phase, and each vector was used 5 times on them having then 1320 runs of the tracking algorithm over this evaluation set.



**Figure 6.** Samples of frames for six sequences of the evaluation.

These 44 vectors perform very similar in the new sequences as they did in the training

sequences. Visually, the overall performance was reasonably good for the original purposes as can be seen in Figure 7. Obtained mean numerical values of worst per-frame solution on each sequence showed also acceptable levels.



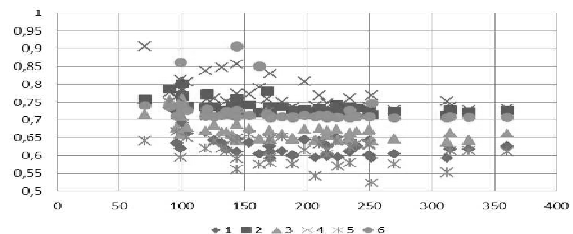
**Figure 7.** Samples of results obtained for the evaluation set using selected parameters.

Table 5 shows the mean and standard deviation of the results obtained using all selected vectors grouped by video sequence in the evaluation set.

**Table 5.** Numerical results of the performance of the tracking algorithm using the selected parameters over the sequences of evaluation set (Mean and standard deviation of  $\rho(S_K, I_L)$ ).

Sequence	Mean	Std. Dev.
1	0.6361	0.0484
2	0.7349	0.0231
3	0.6709	0.0327
4	0.7651	0.0474
5	0.6224	0.0544
6	0.7249	0.0446

At the end of the experimental phase it was noticed that the numerical results tend to decrease on those parameter vectors that provoke Simulated Annealing perform a higher number of total iterations, but they also had, as a natural consequence, larger execution times, and that make them less suitable for real time applications. This fact was observed on both training and evaluation set. In Figure 8 a plot of obtained numerical results against the number of iterations performed by Simulated Annealing on each frame by video sequence on the evaluation set is shown.



**Figure 8.** Numerical values obtained (vertical axis) versus number of iterations performed (horizontal axis) on 6 evaluation sequences. The values ( $\rho(S_K, I_L)$ ) tend to decrease when the number of iterations increases.

By modifying parameter restrictions in a way that Simulated Annealing never performs more than 25 or 30 iterations by frame (for example removing cooling rate factor from free parameters and conveniently computing its value from the given initial and final temperatures) and applying the parameter selection process, is possible to get good configurations capable of obtaining acceptable visual results at 22 or more frames by seconds and so, they can be used in real time applications [21].

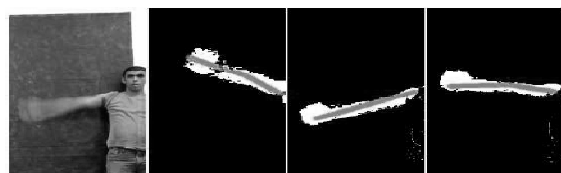
Besides the validation of selected parameter vectors, the use of evaluation sequences helped in the analysis of the main weakness of the proposed tracking algorithm. Some minor drawbacks came out from that analysis such as: bad illumination, blurred images and the impact over results of a same limb model with different subjects.

As mentioned before, illumination conditions may harm the results obtained by skin detection algorithms and therefore harm the positions obtained by the Simulated Annealing algorithm on each frame. In Figure 9 some spots that are part of the limb are not classified as skin points and this will affect the spatial coincidence value. This problem may be solved with connected component removal in the binary image of the limb.



**Figure 9.** Samples of the effects of skin misclassification.

Fast movements may produce blurred frames in which limb silhouette is not accurately detected (see Figure 10). This issue is mainly caused by the use of regular web cams.



**Figure 10.** Samples of the effects of fast movements and blurred images.

Also, the use of a same limb model for different subjects may lead to wrong pose estimation in some frames (see Figure 11).

The better the model fits the real limb, the better the results.



**Figure 11.** Samples of the effects of different physical characteristics.

Figure 12 shows a visual comparison between runs of the tracking algorithm using selected parameters and runs with random parameter vectors. As can be observed the limb model position is more accurate when selected parameters are used.



**Figure 12.** Visual comparison for the same frames of limb tracking using automatic parameter selection and without. Top row using the parameters obtained with the designed algorithm. Bottom row using a set of predefined parameters. As we can observe in the top row we obtain a better adjustment.

This tracking system is used in different video games for rehabilitation, where the human-computer interaction is performed through patient elbow and their wrist. Our team has designed these video games. Obviously as in these activities the real time is essential, the part of the designed algorithm to record images has been eliminated, only the tracking has been operational in the final application because the focus is to evaluate the response of the disabled. The presented algorithm allows a real time tracking with a low cost system (web camera and desktop). Through automatic parameter configuration we can adjust the system to the characteristics of the patients motion, and we can obtain better performance and real-time. The algorithm optimizes the relationship between the image processing, the capture device, and the rehabilitation motions. In Figure 13 we show the rehabilitation environment where we have used it with real patients.





Figure 13. Rehabilitation environment.

## 7. Conclusions

In this work, a parameter selection framework was presented through an algorithm for human limb movement tracking. The parameter selection process for the tracking algorithm using the mentioned framework was described. The parameters were trained using a small video set and showed acceptable results on both training and evaluation instances. The automated experimental process helped to analyze the performance of the tracking algorithm and some drawbacks were encountered and reported. The developed framework is not restricted to be used for the described tracking algorithm, with very small modifications it can be used to tune algorithms designed to solve other problems such as image segmentation, edge detection, or any problem where a measure of performance or quantitative evaluation can be defined, that is the case of the two previous problems or the case of study presented in this paper. Further work on the subject of limb tracking will explore the substitution of skin detection and similar techniques by depth images obtained with special sensors such as Microsoft Kinect as well as comparison with other limb-tracking approaches and other metaheuristic implementations. Also experimentation will be increased.

## Acknowledgments

This work is partially supported by *Departament de Ciències Matemàtiques i Informàtica* UIB, the projects 28/2011 (Balearic Island competitive groups funds) granted by the Government of the Balearic Islands, MAEC-AECID A/030033/10, MAEC-AECID A2/037538/11, and TIN2012-35427 of the Spanish Government, with FEDER support.

## REFERENCES

1. ADENSO-DIAZ, B., M. LAGUNA, **Fine-tuning of Algorithms using Fractional Experimental Designs and Local Search.** *Operations Research*, vol. 54(1), 2006, pp. 99-114.
2. BATTITI, R., M. BRUNATO, F. MASCIA, **Reactive Search and Intelligent Optimization,** *Op. Res. / Comp. Sc. Interfaces*, vol. 45, 2009.
3. BIRATTARI, M., Z. YUAN, P. BALAPRAKASH, T. STUTZLE, **F-race and Iterated f-race: An Overview.** in T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, (Eds.): *Experimental methods for the analysis of optimization algorithms*, Springer-Verlag New York Inc, 2010, pp. 311-336.
4. BRUMBY, S. P., N. R. HARVEY, S. PERKINS, R. B. PORTER, J. J. SZYMANSKI, J. THEILER, J. J. BLOCH, **A Genetic Algorithm for Combining New and Existing Image Processing Tools for Multispectral Imagery.** in: S. S. Shen and M. R. Descour, (Eds.): *Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI. Proceedings of SPIE*, Vol. 4049, 2000, pp. 480-490.
5. COTTA, C., M. SEVAUX, K. SORENSEN, **Adaptive and Multilevel Metaheuristics,** *Studies in Computational Intelligence*, vol. 136 Springer Verlag, 2008.
6. COY, S. P., B. L. GOLDEN, G. C. E. A. RUNGER, G. C. WASIL, **Using Experimental Design to Find Effective Parameter Settings for Heuristics.** *Journal of Heuristics*, vol. 7(1), 2001, pp. 77-97.
7. GALL, J., B. ROSENHAHN, T. BROX, H. P. SEIDEL, **Optimization and Filtering for Human Motion Capture.** *International Journal of Computer Vision*, vol. 87(1-2), 2010, pp. 75-92.
8. GALL, J., B. ROSENHAHN, H. P. SEIDEL, **An Introduction to Interacting Simulated Annealing.** In: R. Klette, D. Metaxas, and B. Rosenhahn, (Eds.): *Human Motion - Understanding, Modeling, Capture and Animation, Computational Imaging*, Springer, Heidelberg. vol. 36, 2008, pp. 319-345.

9. GEEM, Z. W., **Music Inspired Harmony Search Algorithm: Theory and Applications**, Studies in Computational Intelligence. Springer Verlag, vol. 191, 2009.
10. GEEM, Z. W., **Recent Advances in Harmony Search Algorithm**, Studies in Computational Intelligence. Springer Verlag, vol. 270, 2010.
11. HUTTER, F., H. H. HOOS, K. LEYTON-BROWN, T. STUTZLE, **Paramils: an Automatic Algorithm Configuration Framework**. Journal of Artificial Intelligence Research, vol. 36(1), 2009, pp. 267-306.
12. JAUME-I-CAPÓ, A., P. MARTINEZ-BUESO, B. MOYA-ALCOVER, J. VARONA, **Interactive Rehabilitation System for Improvement of Balance Therapies in People With Cerebral Palsy**. Neural Systems and Rehabilitation Engineering, IEEE Transactions on. DOI: 10.1109/TNSRE.2013.2279155.
13. JAUME-I-CAPÓ, A., J. VARONA, M. GONZALEZ-HIDALGO, F. J. PERALES, **Adding Image Constraints to Inverse Kinematics for Human Motion Capture**. EURASIP J. Adv. Signal Process, Vol. 4, 2010, Art. 4. doi:10.1155/2010/142354.
14. KIRKPATRICK, S., JR. C. D. GELATT, M. P. VECCHI, **Optimization by Simulated Annealing**. Science, Vol. 220, No. 4598, 1983, pp. 671-680.
15. LEE, G., V. BULITKO, I. LEVNER, **Automated Selection of Vision Operator Libraries with Evolutionary Algorithms**. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC 2004), IEEE Press, vol. 1, 2004, pp. 1127-1134.
16. LUO, Y., H. YANG, Z. HU, **Human Limb Motion Real-time Tracking based on Camshift for Intelligent Rehabilitation System**. IEEE International Conference on Robotics and Biomimetics (ROBIO), 2009, pp. 343-348.
17. MINTON, S., **An Analytic Learning System for Specializing Heuristics**. Proceedings of the 13th International Joint Conference on Artificial intelligence, IJCAI'93, vol. 2, 1993, p. 922.
18. MOESLUND, T. B., E. GRANUM, **A Survey of Computer Vision-based Human Motion Capture**. Computer Vision and Image Understanding, vol. 81(3), 2001, pp. 231-268.
19. MOESLUND, T. B., A. HILTON, V. KRUGER, **A Survey of Advances in Vision-based Human Motion Capture and Analysis**. Computer Vision and Image Understanding, vol. 104(2), 2006, pp. 90-126.
20. POPPE, R., **Vision-based Human Motion Analysis: An Overview**. Computer Vision and Image Understanding, vol. 108(1-2), 2007, pp. 4-18.
21. VARONA, J., A. JAUME-I-CAPÓ, J. GONZALEZ, F. J. PERALES, **Toward Natural Interaction through Visual Recognition of Body Gestures in Real-time**. Interacting with Computers, vol. 21(1-2), 2009, pp. 3-10.
22. VERA-PÉREZ, O. L., A. MESEJO-CHIONG, A. JAUME-I-CAPÓ, M. GONZÁLEZ-HIDALGO, **Human Limb Tracking using Simulated Annealing and Skin Features**. Actas del VIII congreso español sobre metaheurísticas, algoritmos evolutivos y bioinspirados (in Spanish).
23. OSORIO, R., S. GARCÍA, M. PEÑA, I. LOPEZ-JUAREZ, G. LEFRANC, **Movement and Color Detection of a Dynamic Object An Application to a Mobile Robot**. Studies in Informatics and Control, 2012, vol. 21, no 1, p. 34.
24. TANGOUR, F., P. BORNE, **Presentation of Some Metaheuristics for the Optimization of Complex Systems**. Studies in Informatics and Control, 2008, vol. 17, no 2, p. 169.
25. VEZHNEVETS, V., V. SAZONOV, A. ANDREEVA, **A Survey on Pixel-based Skin Color Detection Techniques**. Proceedings of the GraphiCon 2003, Moscow, Russia, vol. 3, 2003, pp. 85-92.
26. WANG, C. M., Y. F. HUANG, **Self-adaptive Harmony Search Algorithm for Optimization**. Expert Systems with Applications, vol. 37(4), 2010, pp. 2826-2837.
27. ZHOU, H., H. HU, **Human Motion Tracking for Rehabilitation—A Survey**. Biomedical Signal Processing and Control, vol. 3(1), 2008, pp. 1-18.