Multi-Forest Classification and Layered Exhaustive Search using a Fully Hierarchical Hand Posture/Gesture Database

Amin Dadgar and Guido Brunnett
Computer Science, Chemnitz University of Technology, Straße der Nationen 62, 09111, Chemnitz, Germany

Keywords: Fully-Hierarchical Hand Posture Database, Multi-Forest Classification, Layered-Exhaustive Search.

Abstract: In this paper, we propose a systematic approach to building an entirely hierarchical hand posture database. The hierarchy provides the possibility of considering a large number of hand poses while requires a low time-space complexity for construction. Furthermore, two algorithms (random decision forest and exhaustive search) are chosen and tested on this database. We show that by utilizing such a database one will achieve better performances on classifiers' training and search strategies (two main categories of the algorithms in the field of machine learning) compared with conventional (all-in-one-layer) databases.

1 INTRODUCTION

To achieve an accurate, robust and real-time hand gesture recognition system, within the "analysis by synthesis" approach (Yuille and Kersten, 2006), there are many technical challenges which need to be tackled. One of these challenges is the high computational complexity of training and searching of the posture/gesture space. That is due to a large number of degree-of-freedom (DoF) that the hand, as an articulated object, owns. Therefore, the system is obliged to recognize a small set of gestures (Schlenzig et al., 1994). Moreover, it requires a rich database to include different viewpoints of the hand (Sharp et al., 2015). Thus, to address the issue effectively, the choice of the database of the postures and their relations (gestures) seems to be an important decision.

Due to the versatile usage of the hand in applications and its vast possibilities of different motions (Heimonen et al., 2013; Jacob et al., 2011; Starner and Pentland, 1995) it should be challenging to create a reasonably extensive and application-independent database. Therefore, the field of hand gesture recognition is dominated by domain-specific databases some of which are synthetic ones. In such databases, a small set of application-specific gestures is considered which the training is mainly accomplished based on its pixel-level information (Sharp et al., 2015). As a result, this pixel-based training seems to be one of the reasons that the training of the system should be accomplished on many different positions and viewpoints. An alternative approach is to construct a database with the focus on pose-vector information instead of pixel-level (pixel-vector).

However, to our knowledge, creating a comprehensive (application-independent) synthetic database based on pose-vector seems to be difficult. That is mainly because of the high dimension (66 DoF) and time complexity of such a process ($\approx 10^{164}$ poses for 1° step changes). Using the kinematic constraint of the hand will help to reduce the hand's DoF to 28 similar to (Zhao et al., 2012). However, the number of poses is still too big ($\approx 10^{45}$) for many systems to handle its construction, training, and optimization.

In this paper, a novel construction of a fully hierarchical database of the hand postures/gestures is explained. That hierarchy provides a possibility of considering a large number of poses (hence an application-independent database) while requires a low time-space complexity for construction. Furthermore, two algorithms (random decision forest) for training and (exhaustive search) for searching the pose-space of this database are considered. Our findings show that the hierarchical definition of the hand postures will improve their performance up to a great extent.

The hierarchy proposed in this paper is inspired by the idea of 'finger-spelling' suggested by (Mo and Neumann, 2006). In their work, each finger can demonstrate a set of seven finger-states (Figure 1.a) and five inter-finger-states (Figure 1.b). Those states create a 12-dimensional spelling vector (five fingers plus seven combinations) that introduces a grammar to the finger's motion. We use this grammar to decrease the

Dadgar, A. and Brunnett, G.
Multi-Forest Classification and Layered Exhaustive Search using a Fully Hierarchical Hand Posture/Gesture Database.
DOI: 10.5220/0006591601210128
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 4: VISAPP*, pages 121-128
ISBN: 978-989-758-290-5

(a) (b)

Figure 1: a) Seven Finger-Spelling-States: Up, Forward, Side (only thumb), Half-Bend, Bend, Half-Close/Close. b) Five Inter Finger Spelling States: Group, Separate, Cross1 (j on i), Cross2 (i on j), Loop (for Thumb with all other fingers) (Mo and Neumann, 2006).

large number of DoF in the fingers when constructing a hand-pose database within a fully hierarchical manner. At the same time, we enrich the database by containing a higher number of poses (Section 3). In that context, we separate the fingers' motion from the inter-fingers' motion alongside from the wrist rotation. These separations (hierarchies) extend the existing hierarchy in the work of (Keskin et al., 2012) that only separates the global rotation-translation from the rest of the hand-pose. The formulation of the database can provide the possibility to consider and train lots of different gestures of a hand in an efficient manner. That can help to introduce an application-independent database. We hope, that feature finally leads to an accepted benchmark within the research community. A milestone which can facilitate the comparison and the verification of different methodologies more systematic. Moreover, this database can introduce a new vision to various machine learning algorithms and advance their performances. That can be accomplished by enhancing the architecture of these algorithms and their mathematical descriptions, using the provided fully layered data structure.

2 LITERATURE REVIEW

The random (decision) forest (RDF) has been proposed by (Breiman, 1999). RDF quickly became popular in the field of computer vision (Sharp et al., 2015), object segmentation (Schroff et al., 2008), image classification (Bosch et al., 2007), and data mining (Verikas et al., 2011). The reason is that it works on large databases efficiently, is very fast and is robust to outliers (Breiman, 1999). Moreover, it can effectively generalize the high variations (nonlinearities) in the data (Breiman, 1999; Shotton et al., 2013). Since the output of the RGB/depth cameras can be highly noisy and can exhibit a high level of variance, the random decision forests are suitable for the classification tasks in the area of computer vision too. Therefore, many researchers, in both fields of the human body and hand pose/gesture estimation, incorporated the RDF in their frameworks (Camgoz et al., 2014; Keskin et al., 2012; Sharp et al., 2015).

In all these works, however, RDF is mainly applied within the pixel-level (shape (Keskin et al., 2012), depth (Sharp et al., 2015)) classification fashion. The pixel-level classifications using RDF have shown acceptable results. However, they require a large number of input variables (hundreds or thousands), with each one containing only a small amount of the overall image information (Keskin et al., 2012; Sharp et al., 2015). Therefore, the entire random forest does not model the whole pixel variations efficiently. As a consequence, RDF only plays a complementary role in a system for introducing a few more poses for each frame (Sharp et al., 2015). Moreover, solely a limited number of poses can be considered when the pixel-level classification is employed. In other words, most of the workload of the recognition is carried out by the computationally expensive particle swarm optimization (Kennedy and Eberhart, 1995) framework.

Additionally, one of the main goals in training the posture-space is to acquire the temporal information of the gestures from a set of postures (Miranda et al., 2012) and to capture the gesture spotting information (Camgoz et al., 2014). However, in all these works, employing the pose-vector (instead of the pixel-vector) information, to model temporal information, can be more convenient and more efficient. That together with the mentioned issues of the pixel-level classification, can affirm the necessity of a shift in considering the pose-vector data in training the random decision forests. The requirement which could be addressed more efficiently, by owning a comprehensive and a fully-hierarchical database.

Nevertheless, some attempts are accomplished to introduce a multi-layer database (Keskin et al., 2012). They proposed the two-layer pose data: The first layer for global rotation and translation (6 DoF) and the second layer for the local changes of the hand (22 DoF). However, a two layer-only database does not help the time-space complexity of a random decision forest training to be reduced effectively. That is because the highest variations of a hand postures/gestures are encoded in the local changes of that hand (fingers and wrist together possess 22 DoF out of 28). Therefore, we propose a fully-hierarchical database which formulates global rotation and translation, wrist, inter-finger and intra-finger rotations all separately. Below, a brief detail of the random forest training is presented first, to describe, how we can employ this algorithm with our database, effectively.

2.1 RDF Training

To train a forest, first, the training set should be defined as $D = \{(X_1, Y_1), ..., (X_n, Y_n)\}$. Here, $(X_1, ..., X_n)$

corresponds to the pose-vector at each layer, and $(Y_1,...,Y_n)$ represents their respective class labels. Since a forest consists of several decision tree classifiers $\{t(x,\phi_k),k=1,...\}$ (Breiman, 1999), each tree $t(x,\phi_k)$ is left to grow until the forest is constructed. Here, x is an input vector and ϕ_k is a random vector generated from a process which is named "data bagging". At each node of the tree, m features are selected randomly from the available, d dimensions ($m < d$). That process is called "feature bagging". Then from those selected features, ones which split the node with the most information gain are chosen.

Although RDF has been utilized to the mentioned fields successfully, it still has two main drawbacks hidden in its structure (Shotton et al., 2013). First, training the decision tree is an NP-hard problem. Second, deep-trees (with exponential growth of the time-space complexity) illustrate more accurate results than large ones. To address these issues, we employ the RDF classification approach in a multi-layered way (Section 4.1) using our fully-hierarchical database. Therefore, in the following section, the formulation and the construction of this database is described first.

3 HIERARCHICAL SYNTHETIC DATABASE

We propose a 31-layer hierarchical database defined by a set of primitive- and compound-layers. The primitive-layers divide the 28 DoF pose-vector such that each of these divisions corresponds to one primitive-layer (Figure 4).

We remark the following phases to define our hierarchy thoroughly. First, we determine the 28 DoF hand's pose vector:

$\overrightarrow{V} = \{v_i | i = 1,2,...28\}$, where
$\{v_1,v_2,v_3\}$ is the global translation (3 DoF),
$\{v_4,v_5,v_6\}$ is the global (Arm) rotation (3 DoF),
$\{v_7,v_8,\}$ is the wrist rotation (2 DoF),
$\{v_9,v_{10},v_{11},v_{12}\}$ is the little finger rotation,
$\{v_{13},v_{14},v_{15},v_{16}\}$ is the ring finger rotation,
$\{v_{17},v_{18},v_{19},v_{20}\}$ is the middle finger rotation,
$\{v_{21},v_{22},v_{23},v_{24}\}$ is the index finger rotation and
$\{v_{25},v_{26},v_{27},v_{28}\}$ is the thumb finger rotation.

Second, we determine the relation of above vector with first three layers of global translation ($Layer_1$ or L_1), global rotation ($Layer_2$ or L_2) and wrist rotation ($Layer_3$ or L_3). In that direction, as shown in Figure 2, the L_1 (red bits), L_2 (blue bits) and the L_3 (orange bits) can be separated from the fingers' information of the hand pose-vector. Now, we can construct the poses of these layers by defining a step-degree, which specifies a resolution for these layers. For example, we have constructed our low-resolution global-rotation-layer (L_2) by assigning each X,Y,Z component, values starting from -180 (lower bound). Then, we have assigned five step-degree increments (72) until they reached $+180$ (higher bound). Obviously, based on the specification of an application one can define different step-degree, lower and upper bounds for each of these three layers.

Third, using the inter-finger states (Figure 1.b) we extract the Z rotation components of the lowest part of the little, ring, middle, and index fingers. These four components, namely $\{v_{12},v_{16},v_{20},v_{24}\}$, are responsible for the inter-finger state changes. This 4 DoF which has been illustrated with green bits on Figure 3, forms the layer five of the database and contains information about the four fingers inter-state changes. Each of this DoF can possess four different values (shown in Figure 1.b) such as –crossed over– crossed behind–grouped–separated– states. Note that, the loop state is a state which can be defined by thumb finger-state implicitly; thus, we explain it at the end of the next paragraph. Now, we can achieve a cheap construction of the poses of this layer in a resolution we require. For example, we can determine the $Layer_5$ (L_5) to be low-resolution when in that layer each inter-state of the fingers is repeated only once.

Fourth, the remaining 16 DoF of the five fingers is reduced to 5 DoF using the finger-states represented in Figure 1.a. Here, each bit of this 5 DoF, corresponds to one finger and is shown with white on Figure 4. In this layer ($Layer_4$ or L_4), each of this 5 DoF is the combination of 4 angles which can possess 6-7 different states. Therefore, they can be used to construct finger poses in the same manner as inter-finger layer (third step). As an example, we can define a low-resolution layer, in which each of the finger states is repeated only once. Hence, the amount of the construction complexity is reduced considerably for this layer. This layer (L_4) is the most expensive layer, of the hand pose-vector, for construction in the conventional databases (16 DoF). Whereas, in our approach, instead of 16 DoF we consider 5 DoF, each of which can have 6-7 different values (states). Moreover, we can increase the resolution of this layer by considering each state twice or more. Thus, different resolutions can be constructed in a controlled and a meaningful manner. Moreover, with this formulation, all five layers have a comparable number of dimensions, $2 \leq d \leq 5$. In Section 4.1 we show that such homogeneous dimensions would be advantageous in a hierarchical training process. It is important to note that, in L_5 all inter-finger states are formulated explicitly except for the thumb inter-state. This finger's

Figure 2: Separation of the wrist, global rotation (arm) and global translation.

Figure 3: Detaching the inter-finger layer from finger layer.

interstate (loop in Figure 1.b) is an interactive state with all other four fingers. That enables us to formulate it as the combination of thumb being forward (with four different degree intervals pointing to other four fingers) and the other fingers being at half-bend state. Therefore, this inter-finger state is determined implicitly, without increasing the DoF of the L_5.

Finally, in our database, we name the layers from one to five the 'primitive' layers. Since in these layers 'only' one of the categories at a time is under transformation (Table 1). Note that, with different combinations of these primitive layers more complex layers, such as 2-layer compound (Tables 2 and 3), 3-layer compound, or 4-layer compound (Table 4) can be constructed when it is required. For example, L_{15} is the combination of L_4 and L_5 (Table 3). In other words, in L_{15} the finger states and the inter-finger states are considered for transformation (Figure 4). Mark that for the purposes of this paper's experimentation (Section 4), only 11 of them are necessary (namely, the layers on Tables 1 and 3) and layer L_{30}. Note that, layers L_{30} and L_{31} (Table 4) contain all the information, the primitive-layers, which one can find in conventional databases. However, in those databases as a result of the high degree of freedom of the hand, it is infeasible to construct a large number of the postures/gestures in practice. Therefore, as mentioned in Section 1, they are restricted to a specific gestures' vocabulary. In our database, on the other hand, even if the number of postures in each of the primitive layers is low (100 poses), the overall number of poses could be considered implicitly will be a significant number (100^4). This implicit number of poses will increase even more when one constructs the mid-resolution primitive-layers (around 1000 poses in each layer). One can accomplish such an increase in the resolution easily and cheaply. Thus, our way of formulating the hand posture/gesture database provides a coherent approach, to potentially consider a huge number of poses. At the same time, such a formulation decreases the construction's time-complexity significantly.

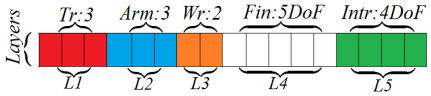

Figure 4: Creating the layers from pose vector.

Table 1: Primitive layers definition of the database.

Translation Only	Forearm Only	Wrist Only
Layer 1	Layer 2	Layer 3

Finger States Only	Inter-Finger States Only
Layer 4	Layer 5

4 EXPERIMENTS

Two algorithms, namely, RDF classification and exhaustive search, are selected to verify any possible achievements on performance gain of the training and the searching using our database. The aim is to compare the performance of these two algorithms, using two different databases: 1. the primitive layers of our database (Table 1), 2. the conventional all-in-one-layer database (L_{30} in Table 4). In that context, two classes of experiments are designed, using these two algorithms; and their results are illustrated in the Sections 4.1 and 4.2. To evaluate and analyze the performance of our proposed hierarchical database different experiments with various setups and multiple parameter values are conducted. However, only those results which indicate high accuracy or significant implication are reported.

It is important to note that, in our experimentation, we do not consider the layer 1 and all those compound layers which are defined by the layer 1 (Table 2). That is mainly because the hand's translation (layer 1) is highly dependent on the experimental setup (background colors, camera parameters). Whereas, in this paper, our two types of employed input, pose-vector which is a set of degrees (instead of pixel-vector) and synthetically generated cluttered-free images (using OpenGL), are both experimental-setup independent. Whilst, the central focus of this paper is to evaluate the performance enhancement in training and optimization phases, when our proposed database is employed. Therefore, discussion on layer 1 and all its combinations, L_{11}, are neglected in this paper. For the same reason, all-in-one-layer database counterpart, in the conventional systems, is a layer with the combination of 4 layers (L_{30}).

Table 2: Two-Layer compound definition of the database with layer 1.

L_1,L_2	L_1,L_3	L_1,L_4	L_1,L_5
L_6	L_7	L_8	L_9

Table 3: 2-Layer compound's definition of the database without layer 1.

L_2,L_3	L_2,L_4	L_2,L_5	L_3,L_4	L_3,L_5	L_4,L_5
L_{10}	L_{11}	L_{12}	L_{13}	L_{14}	L_{15}

4.1 Exp1: Multi-Forest Classifier Training of the Pose Space

A multi-forest classification framework (similar to (Keskin et al., 2012), but on the pose-vector) is designed, to address the mentioned issues exist in one 'big forest' classifier (Section 2). This multi-forest framework is employed, in our proposed fully-hierarchical database (Section 3), within the following steps. First, for each of the primitive layer, one forest (Bradski, 2000) is trained. Second, meaningful relations, based on the hierarchy of our database, are introduced between those forests. These relations (links) extend the standard random decision forest to a multi-forest random decision structure.

In this category of experimentation, the OpenCV random forest framework (Bradski, 2000) for the purpose of training and evaluating our database, is utilized. In the scope of our paper, it is acceptable if we reuse the training set in the testing phase. The logic behind that is the focus we have to compare different types of databases (not different types of classification algorithms). In other words, we aim to show that, one training algorithm (here a classifier) can train the pose-space of a hierarchical hand database more effectively in comparison to a conventional database. Since the pose-space consists of pose-vectors (set of degrees), the classification rates indicate how well the forest can model this pose space. Moreover, we use this setup for all forests in this paper. Therefore, the comparison between the forests is reliable.

In that context, for each experiment, we set the maximum-depth of each tree and the maximum number of trees. The other parameters, such as minimum-samples required at a leaf node for it to be split, termination criteria, and sufficient accuracy, are set globally. Moreover, in Tables 5, 6, 7, and 8, the test rate, training time, and the number of nodes are calculated in the OpenCV framework. Additionally, in those tables, the sample, feature and bagged feature rows are the characteristics of a layer which is under that experiment. It is important to note that in those tables, the number of nodes and training time indicate the space and time complexities, respectively. As mentioned earlier, the test rate also indicates how well the forest(s) train (model) the pose-space. To evaluate the method, two experiments are designed.

Table 4: All-In-One-Layer equivalent to conventional DBs.

L_2,L_3,L_4,L_5	L_1,L_2,L_3,L_4,L_5
L_{30}	L_{31}

Exp 1.a) 2-Layer-Compound Training: The aim of the first experiment, in this category, is to illustrate that the multi-forest approach, compared to one 'big' random forest, improves the classification rate in the complex layers. To achieve that goal three steps are considered, as follows:

First, one forest is trained on each of the low-resolution primitive layers (L_2, L_3, L_4, L_5). The parameters in these forests, such as the max-depth (=9) and the number of trees, are set empirically to keep the classification rates the highest. At the same time, these parameters values help to retain a low space-complexity (number of nodes) and a low time-complexity (Table 5).

Table 5: One forest is trained and tested for each primitive layers. Maximum depth of all forests is 9.

Layer	2	3	4	5
Samples No	125	100	186	54
Features No	3	2	16	4
Trees	18	46	54	12
Nodes	2000	2194	6454	554
Train Time (s)	<1	<1	<1	<1
Test Rate (%)	91.2	82.00	81.72	88.89

Second, for each of the 2-layer-compound layers ($L_{10}, L_{11}, L_{12}, L_{13}, L_{14}, L_{15}$), one 'big' forest is trained. This time the max-depth is kept unchanged (=9). However, the trees' number is set up to the addition of the trees' numbers in the corresponding primitive layers. It is also clear that the number of samples in these compound layers should be ($\approx$) the product of the samples' number in the corresponding primitive layers (L_{10} contains $12500 = 125 \times 100$ samples). The motivation that only the 2-layer-compound layers are considered, in this step, is the large size of 3-/4-/5-layer compound layers (> 15000 samples). This size is the threshold value for the size of a database which above that the 32-bit system is incapable of training one big forest. Therefore, one can not test both ('big' forest and the multi-forest) frameworks on them to compare their differences. That is why also, for L_{11} and L_{13}, which their size is slightly bigger than this number, their samples number is cut off to 15000. From the results (Table 6) it can be observed that,

despite the considerable increase in the number of created nodes (bigger forests), the classification rates decrease extremely. That illustrates that one big forest, equal to the size of both smaller forests, could not model the entire space of a bigger database efficiently. Furthermore, this suggests, the decrease of classification rates could be even more severe for more complex layers (3-/4-/5-layer compound). It is important to mark that we could achieve better test rates by increasing the number of trees significantly (e.g. 2000). Such a substantial increase in trees' number helps some layers to achieve test rates as high as 79%, 91%, 86% and 75% (for layers L_{10}, L_{12}, L_{14} and L_{15}, respectively). However, this increase in the number of trees leads to a considerable increase in space (number of nodes) and time complexity (average: 800,000 nodes and 9,500 seconds) for all layers. Moreover, an effective increase in the test rates (49%, 35%), for layers L_{11}, L_{13}, respectively, is not observed. Those complexities could grow even more severely for more compound layers (e.g. L_{30}). Therefore, the substantial limitation that one 'big' forest is encountered to train a large database is suggested.

Table 6: One big forest (equal to the size of both primitive forests) is trained and tested for two-layer compounds.

Layer	10	11	12	13	14	15
Samples No	12500	15000	6750	15000	5400	10044
Features No	5	19	7	18	6	20
Trees	64	72	30	100	58	66
Nodes	32046	35002	14020	40898	23440	32474
Train Time (s)	19	69	33	90	39	388
Test Rate (%)	15.42	10.96	17.14	10.36	26.17	16.47

Finally, the same layers with the same number of samples (as in step 2) are selected, but this time are 'tested' only within the multi-forest framework. Each forest is, first, trained on the primitive layers (Table 5). Then, according to the definition of the compound layers (Table 3), they are linked to each other in a layered manner (multi-forest). The test rates (Table 7) illustrate a significant improvement in the classification rates of the multi-forest compared to the second step (Table 6). As expected, the test rates, in this step, are close to the product of the estimated test rates in the corresponding primitive layers. That suggests the error rate in one 'big' forest is accumulative. Additionally, the time complexity and the space complexity of each multi-forest are the summation (linear growth) of the same complexities at the corresponding primitive layers. Hence, enabling the decision forest to model the data using deep-trees (implicitly) and, thus, to eliminating the exponential growth of these complexities.

Table 7: Multi-Forests are tested-only on the same combined layers. This collection of forests is trained on the primitive layers.

Layer	10	11	12	13	14	15
Samples No	12500	15000	6750	15000	5400	10044
Features No	5	19	7	18	6	20
Trees	64	72	30	100	58	56
Nodes	4274	8534	2634	8648	2748	7008
Test Rate (%)	76.36	78.19	81.77	69.92	73.78	74.07

Exp 1.b) Mid-resolution Primitive Layer Training: This experiment aims to increase the classification rates of the primitive-layers to the highest possible point (Table 8). This goal is sought to be achieved by increasing the samples' number of the primitive layers (to constructing mid-resolution primitive-layers) and training one 'big' forest for each. The motivation behind this experiment is to illustrate that, for all-in-one-layer conventional databases, the accumulative errors can be quite low (less than 5%). This resolution enhancement also provides more freedom in the creation of different possible poses in the compound layers. Note that in this experiment, the max-depth is set to 17.

Table 8: One forest is trained and tested on Mid-resolution primitive layers. That boosts the multi-forest performance (less accumulative error).

Layer	2	3	4	5
Samples No	1331	1369	9072	54
Features No	3	2	16	4
Trees	26	179	109	44
Nodes	28732	81569	456229	39346
Train Time (s)	2	8	535	3
Test Rate (%)	98.27	99.12	99.47	99.13

4.2 Exp2: Layered Exhaustive Search for Pose Estimation

Exhaustive search strategy inquires the whole database and then, estimates the best pose according to a score-penalty function. Therefore, when large databases are considered, reaching a solution will consume considerable time-resources. Hereupon, we select that naive approach, to show that incorporation of our hierarchical database converts that costly search to a feasible task.

In that context, a set of synthetic input images, in total 224, is chosen from layer 30 (L_{30}). It has been mentioned in Section 3, layer 30 is the combination of four different layers. Thus, full construction of this layer (in low resolution) could contain around 100^4 poses. That is an expensive process for our system, however, for the inputs, we need the compound po-

ses of that layer. Therefore, we construct this layer in an every other 'n' pose format. There, we aim to cover many different labels of the primitive layers. At the same time, we keep the construction complexity of the layer, affordable for our system (224 poses). Then, the 2D RGB images with a non-cluttered (white) background of these poses are created using OpenGL specification. After a simple contour extraction using OpenCV library (Bradski, 2000), we employ it as an input for the layered-exhaustive search. Then, this image contour is compared with the pose contours of the primitive layers, in a layered procedure, as follows. The exhaustive search starts from the lowest layer (L_2) which contains the global rotation of the hand. It creates the 2D RGB images of that layer, extracts the contour of the hand and compares them all with the input contour (one-by-one) using a simple penalty function (Chamfer distance (Gavrila, 2007). The optimum pose of that layer, the pose which has the lowest distance, is considered as the partial solution. Then, the higher layer poses (L_3), which contains the writs rotation of the hand, will be constructed on top of the previous partial solution. That process continues, until all primitive layers are searched. Finally, the compound pose of those estimated primitive layers is considered as the output of the system. To measure the accuracy of the layered-exhaustive search the average Euclidean distance, between the estimated joints and the ground-truth, is calculated. Note that, if each layer contains n poses, with our approach, the system searches at most $4 \times n$ (around 4×100 poses in our low-resolution database) instead of n^4 poses (100^4). In that context, five different experiments are designed, as follows:

Exp 2.a) Global Rotation Estimation: The aim of the first experiment of the second category is to examine the exhaustive search performance within the global rotation estimation (L_2) only. Therefore, the effect of the bind-pose configuration of fingers and wrist should be eliminated. To that, all fingers are cut off ($v_{12}, v_{16}, v_{20}, v_{24}, v_{26}$ are set to 180°), whereas, the wrist pose is set to the bind pose (v_7, v_8 are set to 0°). Since the arm-rotation causes the greatest variation in the hand's shape, the performance of this stage has a considerable influence on the overall success of the system. In Figures 5 and 6, the total of 33 correct estimations out of 224 inputs can be observed.

Exp 2.b) Global Rotation Estimation using Known Wrist: In this experiment, the focus is to increase the recognition rate of the global rotation. For this reason, the finger and inter-finger layers' status is kept the same as the previous experiment. However, the wrist information of each input is retrieved from the ground-truth. As it is shown in Figures 5 and 6 we could achieve higher recognition rates. That suggests that if the wrist pose is known for each input image, by utilizing our proposed database, one can achieve 180 correct estimations (80%) on the difficult task of the global rotation estimation.

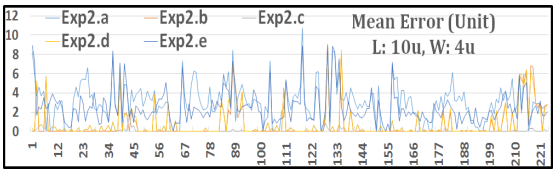

Figure 5: Mean error of all 224 returned poses.

Exp 2.c) Finger and Inter-Finger Pose Estimation using Known Arm and Wrist Poses: This experiment is designed to measure the estimation performances of the finger layer and the inter-finger layer. However, the arm and wrist pose is retrieved from the ground-truth information. Despite the known ground-truth information of the arm and the wrist, the layered-exhaustive search can be efficient on a, still, relatively, high dimensional space of 20 DoF. Moreover, in the 'hand-posture' recognition pipeline, by definition, the pose accuracy of the finger layer and the inter-finger layer are more important than the other layers. Thus, achieving high recognition rates for these layers would suggest a promising horizon for any further consideration and design using this database. In Figures 5 and 6, the recognition rate is 60% (135 correct estimations). Note that, the erroneous outputs are mostly due to 1-/2-DoF-only false recognition (additional materials) and many inputs' fingers are invisible.

Exp 2.d) Arm, Finger, Inter-Finger Pose Estimation using Known Wrist Pose: To set up a more challenging experiment, this time only the wrist pose is retrieved from the ground truth. Therefore, the system has to estimate the postures of all the other layers such as the arm, the finger and the inter-finger. Despite the hard task, the naive search strategy, simple score function, and a greater DoF (23), the estimation rate seems to be acceptable, Figures 5 and 6.

Exp 2.e) All four layers of the Arm, Wrist, Finger, Inter-Finger Estimation: Finally, the system is set to search in all four layers of this paper's interests (25 DoF). Although the number of correct recognition decreases considerably, only five correct recognition, however, most outputs have sensible (visually) relations with their inputs (additional materials).

5 CONCLUSIONS

Experimentation described in Section 4.1 illustrated the random decision forest improvement of perfor-

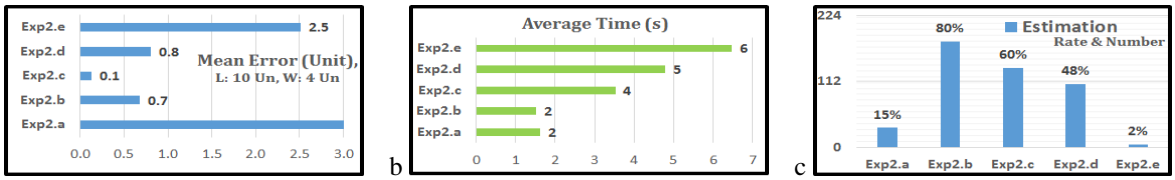

Figure 6: Results of the layered-exhaustive search: a. The average Euclidean distance (estimation error), b. Estimation time of the searches and c. Percentage of the correct estimation for each experimentation

mance when our database is used for training (even if the repetition of each label in the training set is only once). Our database led the RDF to create a fewer number of trees/nodes and to achieve a higher accuracy (Exp 1.a). Moreover, the accuracy increased further if an appropriate higher resolution, of poses for the primitive-layers, is considered (Exp 1.b). These improvement were because of the following reasons. Firstly, it equipped the system to search through deep-trees –layer-by-layer– implicitly. Thus, this could lead to the increase in accuracy with almost no computational overhead. Secondly, the shift in the forest's training, from pixel-level to pose-vector, could eliminate the disadvantages of the pixel-level training using random forest mentioned in Section 2. More specifically, in our experimentation, it reduced the input vector size from hundred or thousand to 28 (DoF and less). That was because there are more significant variances (amount of information) to the pose-space than the pixel space.

Moreover, our hierarchical database, introduced the possibility, to employ the costly exhaustive search with acceptable performance. The layered-exhaustive search has difficulties, mainly, in the correct estimation of 'semi-global' (wrist) rotation. However, in the results of this experimentation (Section 4.2) if we tolerate minor errors (10%) in the accuracy, the recognition rates, in Exp 2.c and Exp 2.d, will illustrate an acceptable accuracy (98% and 79%, respectively).

REFERENCES

Bosch, a., Zisserman, A., and Muoz, X. (2007). Image Classification using Random Forests and Ferns. *IEEE, ICCV , 11th Inter Conf on Com Vis*, pages 1–8.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Breiman, L. (1999). Random Forests. *Machine Learning*, 45(5):1–35.

Camgoz, N. C., Kindirolgu, A. A., and Akarun, L. (2014). Gesture Recognition using Template Based Random Forest Classifiers. *Europeann Conf on Com Vis (ECCV) Chalearn Workshop*, pages 579–594.

Gavrila, D. M. (2007). A Bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Trans on Pattern Anal and Machine Intel*, 29(8):1408–1421.

Heimonen, T., Hakulinen, J., Turunen, M., Jokinen, J. P. P., Keskinen, T., and Raisamo, R. (2013). Designing gesture-based control for factory automation. *Lec Notes in Com Sci*, 8118 LNCS(PART 2):202–209.

Jacob, M. G., Li, Y. T., and Wachs, J. P. (2011). A gesture driven robotic scrub nurse. *IEEE Int Conf on Sys, Man and Cybrntcs*, pages 2039–2044.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. *Neural Net, Proc., , IEEE International Conf.*, 4:1942–1948.

Keskin, C., Kiraç, F., Kara, Y. E., and Akarun, L. (2012). Hand pose estimation and hand shape classification using multi-layered randomized decision forests. *Lec. Notes in Com Science*, (P6):852–863.

Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A. W., and Campos, M. F. M. (2012). Real-time gesture recognition from depth data through key poses learning and decision forests. *Brazilian Symp of Comp Grph & Im Proc*, pages 268–275.

Mo, Z. and Neumann, U. (2006). Real-time hand pose recognition using low-resolution depth images. *Proc. of the IEEE Com. Society Conf. on Com. Vis. and Pattern Recognition*, 2:1499–1505.

Schlenzig, J., Hunter, E., and Jain, R. (1994). Recursive identification of gesture inputs using hidden Markov\nmodels. *IEEE Proc on Apps of Comp Vis*, pages 187–194.

Schroff, F., Criminisi, A., and Zisserman, A. (2008). Object Class Segmentation using Random Forests. *Proc of the British Machine Vision Conf*, pages 54.1–54.10.

Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., and Izadi, S. (2015). Accurate, Robust, and Flexible Real-time Hand Tracking. *ACM Conf on Human Factors in Comp Sys (CHI)*, pages 3633—-3642.

Shotton, J., Sharp, T., and Kohli, P. (2013). Decision Jungles: Compact and Rich Models for Classification.

Starner, T. E. and Pentland, A. (1995). Visual Recognition of American Sign Language Using Hidden Markov Models. *Media*, pages 189–194.

Verikas, A., Gelzinis, A., and Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Rec.*, 44(2):330–349.

Yuille, A. and Kersten, D. (2006). Vision as Bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10(7):301–308.

Zhao, W., Chai, J., and Xu, Y.-Q. (2012). Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. *Eurographics ACM SIGGRAPH Symp on Comp Animation*.