Topolet: From Atomic Hand Posture Structures to a Comprehensive Gesture Set

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

Keywords: Generative **Poselet**, **Topolet**, Topological-based Temporal Model, Hidden Markov Model, Gestures' Comprehensive Set, **Context-Free** Gesture Recognition, Description-based (Data) Specification.

Abstract: We propose a type of time-series model for hierarchical hand posture database which can be viewed as a Markovian temporal structure. The model employs the topology of the points' cloud, existing in each layer of the database, and exploits a novel type of atomic structure, we refer to as ***Topolet***. Moreover, our temporal structure utilizes a modified version of another atomic gesture structure, known as ***Poselet***. That modification considers ***Poselet*s** from the vector-based *generative* perspective (instead of the pixel-based *discriminative* one). The results suggest a considerable improvement in the accuracy and time-complexity. Furthermore, in contrast to other approaches, our ***Topolet*** is capable of considering random gestures, thus introduces a comprehensive set of gestures (suitable for context-free application domain) within the shape-based approach. We prove that the ***Topolet*** could be enhanced to different resolutions of gestures' set which provide the system with the potential to be adapted to different application requirements.

1 INTRODUCTION

Hand gesture recognition systems would benefit from a number of atomic structures within postures which could capture the temporal information necessary to relate a large set of postures to different gestures efficiently. These atomic structures would systematically construct interrelationships between different postures, and thus relate different gestures more effectively. Such structures could eradicate the necessity of repetitive recording of similar postures and distinct modeling of their temporal relation in different gestures. Therefore, they exhibit potentials to resolve the issue of high time-space complexities when a comprehensive/random gestures' set is considered.

In the context of human body pose/action recognition the novel idea of *Grouplet*s was proposed by (Yao and Fei-Fei, 2010). From that *Orderlet* (Yu et al., 2015) and *Gesturelet* (Meshry et al., 2016) for *body pose*, and ***Poselet*** (Bourdev and Malik, 2009) for *hand gesture* recognition were derived. These *Grouplet*s provide tools to formulate atomic structures, however, they have a number of limitations: First, they are action-dependent and thus, cannot consider a random and comprehensive set of actions/gestures. Second, they lack an effective temporal relationship, as the number of features are too large. Finally, all these features are defined as the *discriminative* features on pixel-level data. That type of data are highly dependent on camera view-point. Therefore, any possible smaller structures on pixel-level hardly follow a unique arrangement. Additionally, pixel-level features may be invisible for a considerable time interval. Therefore, robust sequential relation could not be derived.

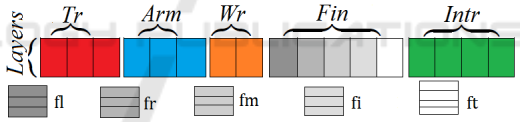


Figure 1: The data structure and the degree-of-freedom for each (sub)layer: global translation (Tr, layer 1) with 3DoF, global rotation (Arm, layer 2) with 3DoF, wrist rotation (Wr, layer 3) with 2DoF, little fin states (fl, 3DoF), middle fin states (fm, 3DoF), index fin states (fi, 3DoF), thumb fin states (ft, 4DoF), inter-finger states (Intr, 4DoF).

To address the last issue, we employ the pose-level vector of *hand*. The *hand*'s pose-vector introduces a type of data which can robustly be trackable over time. The high degree-of-freedom (28), and thus the presence of enormous number of states, is resolved by employing the hierarchical database proposed by (Dadgar and Brunnett, 2018). There, each posture is restructured by a set of sub-vectors and layers (Figure 1). Furthermore, for a set of postures that the database

Dadgar, A. and Brunnett, G.
Topolet: From Atomic Hand Posture Structures to a Comprehensive Gesture Set.
DOI: 10.5220/0007244701570164
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 157-164
ISBN: 978-989-758-354-4

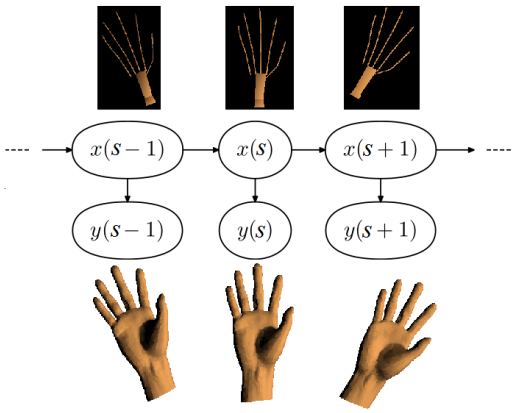


Figure 2: An example of allowable transition for the ***Topolet*** of layer 2 (global rotation) to the neighboring ***Poselets*** in that space. The skeletons indicate the hidden states (in 3D pose configuration space), and hand meshes indicate observation states (in 2D pixel space).

contains, each layer's space consists of an equal number of points. We consider these points as one type of our atomic structures (the *generative* ***Poselet***).

Hidden Markov Model (HMM) (Rabiner, 1989) has been frequently employed to acquire the temporal information of hand gesture between successive frames and describe that gesture by a sequence of states (Yang et al., 1994). Our novel method of acquiring temporal information extends the specifications of HMM to the topology of the layers in our hierarchical database. Our divide-and-conquer approach considers each of the mentioned *generative* ***Poselet*** as one hidden state of our HMM-like method. At each point in time only a few ***Poselets*** (features) are relevant to construct a posture as the gesture proceeds, thus the second issue would be addressed.

To overcome the first issue, our procedure reduces the number of required temporal models to one-model-for-one-*layer*. This approach is in the contrary of the conventional one-model-for-one-*gesture* method. Therefore, for any gestures one requires to have only a limited (but fixed) number of HMM-like structure. One can use our structure to construct a comprehensive gestures' set or enable a context-free (gesture-independent) application domain recognition. Since our temporal information is acquired based on the topology of each layer we refer to it as the ***Topolet*** (topology + ***Poselet***, Section 3.1).

In our results (Section 4) we prove the flexibility of the model in dealing with different resolutions of the postures' set based on shape feature. Moreover, we demonstrate the potentials of our ***Topolet*** to perform the gesture recognition task in real-time.

2 RELATED WORK

Hand gesture is inherently stochastic (Yang et al., 1994). That is, if a person repeats (or if a group of individuals performs) a certain gesture, the measurements of that gesture will be different. That implies, there are hidden specifications which are common in the different recordings for one certain gesture (Yang et al., 1994). In the field of hand gesture recognition, (Yang et al., 1994) employed nine HMMs for recognizing nine 2D gestures in drawing the digits from 1 to 9 with a mouse. These gestures were single-path (palm point) ones and no details about finger poses were estimated. To unfold the hidden patterns of gestures, HMM needs an index of the previous state (s) of the system together with the approximate dynamics (a_{ij}) of the system which would be represented in a transitional matrix (A). Additionally, it is necessary to store the *same-class* variations of the observations in an emission matrix (B) together with the initialization variable (π). To calculate and employ these variables three standard problems must be solved (Rabiner, 1989): evaluation, decoding, and training. Standard solutions to these problems, which relate them to each other under Bayesian framework (Rabiner, 1989). Bayesian semantics enable the HMM to cope with uncertainties in both human performances and sensing processes (Yang et al., 1994). The standard solutions to these problems exhibit inefficiencies. In order to more efficiently employ HMM for the purpose of hand gesture recognition, the approach of collecting the training data is an important decision.

There are two ways of collecting gestures' data: *example-based* and *description-based* (Yang et al., 1994). Most gesture recognition systems employ the example-based specification because it shows more flexibility on small variations of one gesture (Yang et al., 1994). However, its main drawback is the diffi- culty of considering a large number of gestures (Yang et al., 1994). The description-based specification was applied to the field of sign language (Starner and Pentland, 1995) and was capable of considering greater number of gestures. However, because of lacking a meaningful relation between the syntaxes and the hidden states, the description-based specification exhibits two main problems: First, the number of learned HMMs has to be still large (Yang et al., 1994). Second, slight variations of the same gestures can be easily misclassified. In order to acquire a comprehensive temporal information, description-based specification is a preferable method of data collection, however its two drawbacks must be addressed first. To address those issues, we extend the HMM's specifications to a topology-based scheme (Section 3.1).

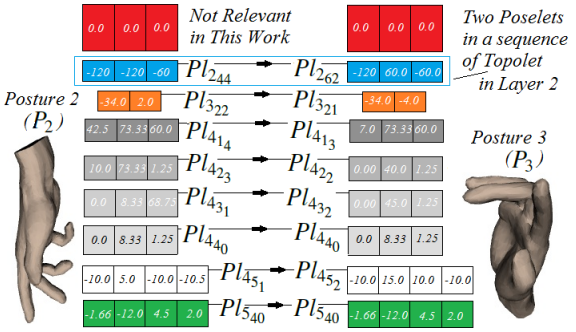


Figure 3: Next posture (P_3) in a gesture sequence modeled with eight parallel HMMs (Pl means ***Poselet***). Bigger, lower and equal indexes of next ***Poselets*** indicate forward, backward, and no transitions, respectively.

3 METHODOLOGY

One main difficulty with HMM is to determine an appropriate number of hidden states (Sangjun et al., 2015). That especially is a problem when HMM is employed to model random/comprehensive gestures' set. That is, when a large number of gestures with varied lengths is to be recognized, the number of states also must be altered. In order to model all gestures the underlying structure of the acquired temporal information should be flexible to considerable modifications. That obliges the system to repetitively perform that difficult task of state-number estimation. We effectively address that issue by employing the *generative* ***Poselets*** as the states in each (sub)layer. Since the number of ***Poselets*** in each (sub)layer is fixed, therefore, the number of states and thus the structure of our HMM-like ***Topolet*** also remains unchanged.

Our proposed ***Topolet*** could also effectively address the first problem of description-based specification (Section 2). More specifically, we utilize one-HMM-for-one-*layer* training scheme based on the topology of the points' trajectory in each (sub)layer (nearest neighbors of each ***Poselet***). Therefore, using these atomic structures, ***Poselet*** and ***Topolet***, the number of HMM-like structures will be fixed to the number of (sub)layers (e.g. eight). Therefor, the issue of linear growth of the complexities with the number of considered gestures is effectively addressed.

We also propose an effective novel method to calculate the emission matrix. More specifically, we model the same-class variations of the ***Poselets***, and thus postures/gestures, based on the inter-topological-spaces between the neighboring ***Poselets***. That could eliminate the necessity of recording several gesture sequences which belong to an identical class and increase the flexibility of the system when slight variations in gestures are observed. Therefore, it addresses the second issue existing in the construction of description-based database.

The entire temporal information is encoded in all these layered-wise HMM-like structures in parallel (Figures 2 and 3). That is, one posture is constructed by orderly-connecting all ***Topolets*** at each point in time. A gesture also is formed by considering all these ***Topolets*** over a period of time. To acquire the ***Topolets***, however, it is necessary to appropriately modify the solutions to the standard problems of HMM.

3.1 Topolet: Temporal Information of Poselet's Topoloy

Data Structure: The ***Poselets'*** vector ($\overrightarrow{Pl}$) for each (sub)layer is as follow (See Figures 1 and 3):
$Pl_1 = \{p_x, p_y, p_z\}$: global translation (p, position),
$Pl_2 = \{a_x, a_y, a_z\}$: global (Arm) rotation (a, angels),
$Pl_3 = \{w_x, w_z, \}$: wrist rotation (angel),
$Pl_{4_1} = \{fl_{u_x}, fl_{m_x}, fl_{l_x}\}$: little finger ($fl$) states,
$Pl_{4_2} = \{fr_{u_x}, fr_{m_x}, fr_{l_x}\}$: ring finger ($fr$) states,
$Pl_{4_3} = \{fm_{u_x}, fm_{m_x}, fm_{l_x}\}$: mid finger ($fm$) states,
$Pl_{4_4} = \{fi_{u_x}, fi_{m_x}, fi_{l_x}\}$: index finger ($fi$) states
$Pl_{4_5} = \{ft_{l_z}, ft_{m_z}, ft_{l_x}, ft_{l_z}\}$: thumb fin ($ft$) states and
$Pl_5 = \{fl_z, fr_z, fm_z, fi_z\}$ inter finger states, where u, m, and l denote the upper part, middle part and the lower part of each finger, respectively.
Database Enhancement: To introduce the required modifications to HMM structure we enhance the previously proposed database (Dadgar and Brunnett, 2018) in two directions. First, we form the global (forearm) rotation layer (L_2) using the quaternions (instead of Euler angles). That assists us to eradicate the duplicated rotations introduced by Euler axis uniform quantization. Second, we loosely divide the finger-state layer (L_4) into five sub-layers (little finger L_{4_1}, ring L_{4_2}, middle L_{4_3}, index L_{4_4},and thumb L_{4_5}, Figure 1). That facilitates the system to consider one ***Topolet*** for each finger. Furthermore, these two enhancements enrich the gestures vocabulary that could be considered by our system.
Parameters of the *Topolet*: We determine the entire temporal information of all gestures as a set of parallel ***Topolets***: $\Lambda = \{Tl_i, Tl_{4_j} | i = [1,5], j = [1,5]\}$, where Tl denotes a ***Topolet***, i is the layer index and j is the finger index. The ***Topolet*** of layer i is defined as: $Tl_i = \{Pl_{i_s}, A_{L_i}, B_{L_i}, \pi_{L_i}\}$, where Pl denotes a ***Poselet***, s is index of the ***Poselet*** in the layer, A_{L_i} marks the transitional matrix, B_{L_i} defines the emission matrix, and π_{L_i} determines the initialization variable of the ***Topolet*** for that layer. In our implements, we assume that the detection and localization of hand is a solved problem. Therefore, we do not consider the layer one

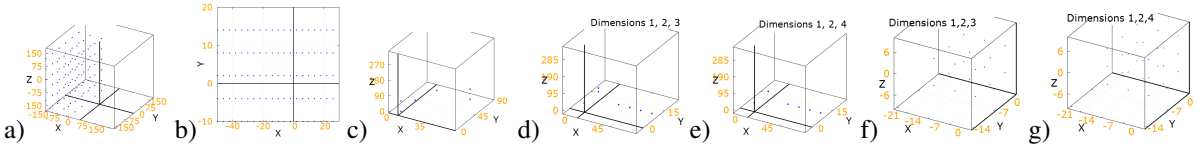


Figure 4: Topology of the layers for Arm-rotation (Figure a), Wrist-rotation (Figure b), Inter-Finger states (Figures f and g), and sub-layers Little-Fingers states (Figure c) and Thumb-Finger states (Figure d and e) in Low-resolution (LD). The little, ring, index, and middle state spaces are similar (Figure c). Therefore, only one finger is shown.

in this work.

Transitional Matrix A_{L_i}: According to the above data structure and the Figures 1 and 4, the ***Poselets*** of the layers L_2, L_3, and L_5 are situated on lattices (n-orthotope). More specifically, Layers two (L_2), three (L_3), and five (L_5) are 3-orthotope (cube), 2-orthotope (rectangle), and 4-orthotope, respectively. We consider each point in the lattices as the hidden state of in our HMM-like structure. Therefore, the topological-based formation of the transitional matrix for each layer using n-nearest-neighbors scheme is as following: For any given ***Poselet*** (current-state) we find the n_s-nearest-neighbors (n_s possible next-states) in that specific lattice. These neighbors determine the possible transitions from each ***Poselet*** at each point in time to other ***Poselets***. The specific topological structure of each layer introduces a particular number of neighbors for any current state in that layer. For example, in layer L_2, which has a rectangular topology, the local number of neighbors for each state, n_s, when $n = 1$, is either 2, 3, or 4 (green and orange ***Poselets*** in Figures 9.b). Based on that local number of neighbors the transitional probability to a specific state is set as $1/n_s$. Thus, the transitional matrices for these layers ($A_{L_2}, A_{L_3}, A_{L_5}$) are constructed.

The transitional matrix of Layer 4 (L_4), which consists of five sub-layers of finger-states, is computed by a different approach. In that layer, little, ring, middle, and index fingers each contains 3DoF and thumb has 4DoF topological structure (Figure 1), but the ***Poselets*** do not form complete lattices (Figure 4.c-e). Therefore, n-nearest neighbors cannot be computed correctly. But the number of states for each finger is limited, hence the possible transitions between these states are visually determined (Figure 6). These transitional matrices for fingers are specified as $A_{L_{4_j}}$. In our implements, we set $n = 1$. However, other values (such as $n = 2$, Bakis model (Rabiner, 1989)) which suit different system/application specifications (e.g. camera frame-rate) are also possible. Furthermore, our ***Topolet*** handles the self-transition. That facilitates the system to recognize the stationary (alongside with dynamic) gestures which remain at a specific posture for an interval of time.

In addition to the mentioned advantages, our topology-based computation of the transitional matrix extends the conventional HMM in the following ways: First, it enables the system to consider the left-right *forward* and right-left *backward* transitions both at the same time and in one unified structure. Second, the major problem with conventional HMMs is that one cannot use a single observation sequence to train the parameters. That is principally due to a large number of transient states exists between two main states. Hence, to make a reliable estimation of all parameters, one has to use multiple observation sequences. In our ***Topolet***, the training is performed based on the topological positions of the trajectory points. Therefore, one could train the inter-space of two states as transient postures, and hence eradicate the necessity of several observation sequences for training.

Higher-Resolution Transitional Matrix: To provide the system with a richer set of postures, and hence smoother gestures, a collection of higher-resolution A_{L_i} is calculated. These matrices are calculated based on the layers of mid-resolution database (*MD*) which contains higher number of ***Poselets*** (Figure 5) as compared to the low-resolution database (*LD*). Thus the number of states and size of the transitional matrices are increased. Enhancing the resolution of the layers L_2, L_3, L_5 is accomplished by uniformly increasing the population of ***Poselets*** of these layers' space (Figure 5). However, the number of ***Poselets*** for each finger (sub)layer is manually increased. For example, different versions for *Bend*-state ($Bend_1$, $Bend_2$, $Bend_3$, and $Bend_4$) are visually defined (Figure 6).

The calculated higher-resolution matrices, and thus richer set of gestures, emphasizes an other important advantage of our ***Topolet***. That is, the increase in the size of the transitional matrices does not lead to higher time-complexity during the search. This is due to the fact that the local number of next ***Poselets*** (states) is determined by the value of n, which in our implementation is set to one ($= 1$). The topological properties of these layers (n_s) do not change after we increase the resolution of them. Therefore, no extra time-complexity is imposed to the system.

Initialization Variable π_{L_i}: The initial state probabilities are calculated similar to conventional HMM: $\pi_{L_{i_{Pl}}} \Longrightarrow \{0,\ Pl \neq 1\ and\ 1,\ Pl = 1\}$.

Emission Matrix B_{L_i}: Conventionally HMM is defined as follows (Rabiner, 1989) (Yang et al., 1994):

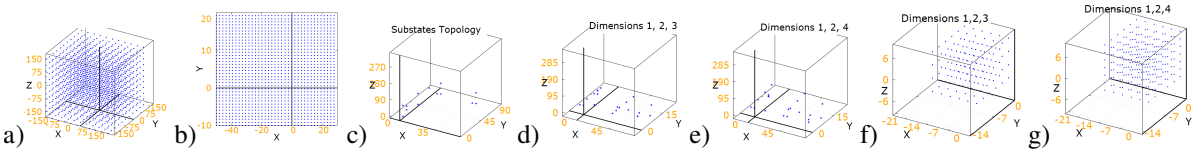
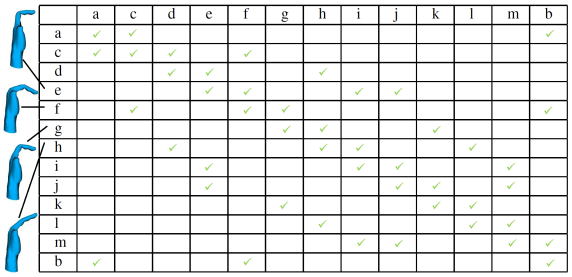
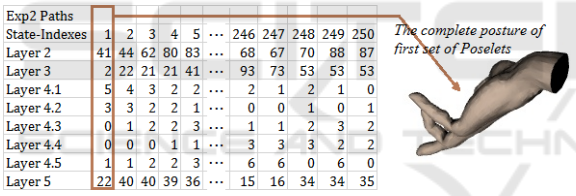


Figure 5: Topology of the layers for Arm-rotation (Figure a), Wrist-rotation (Figure b), Inter-Finger states (Figures f and g), and sub-layers Little-Fingers states (Figure c) and Thumb-Finger states (Figure d and e) in Mid-resolution (MD). The little, ring, index, and middle state spaces are similar (Figure c). Therefore, only one finger is shown.



	a	c	d	e	f	g	h	i	j	k	l	m	b
a	✓	✓											✓
c	✓	✓	✓		✓								
d			✓	✓			✓						
e				✓	✓			✓	✓				
f		✓			✓	✓							✓
g						✓	✓			✓			
h			✓				✓	✓			✓		
i				✓				✓	✓			✓	
j				✓					✓	✓		✓	
k						✓				✓	✓		
l							✓				✓	✓	
m								✓	✓			✓	✓
b	✓				✓							✓	✓

Figure 6: Visually determining the transitions between fingers' states. This figure shows the transitional matrix of the finger states for the mid-resolution database. The allowed transitions are marked with ✓: a. Up, b. Forward, c. Half-Bend$_1$, d. Half-Bend$_2$, e. Bend$_1$, f. Bend$_2$, g. Bend$_3$, h. Bend$_4$, i. Half-Close$_1$, j. Half-Close$_2$, k. Half-Close$_3$, l. Half-Close$_4$, m. Close. Notice that, in our implementation, the self-transition is allowed.



Exp2 Paths											
State-Indexes	1	2	3	4	5	⋯	246	247	248	249	250
Layer 2	41	44	62	80	83	⋯	68	67	70	88	87
Layer 3	2	22	21	21	41	⋯	93	73	53	53	53
Layer 4.1	5	4	3	2	2	⋯	2	1	2	1	0
Layer 4.2	3	3	2	2	1	⋯	0	0	1	0	1
Layer 4.3	0	1	2	2	3	⋯	1	1	2	3	2
Layer 4.4	0	0	0	1	1	⋯	3	3	3	2	2
Layer 4.5	1	1	2	2	3	⋯	6	6	0	6	0
Layer 5	22	40	40	39	36	⋯	15	16	34	34	35

Figure 7: Eight different paths of one gesture in experiment 2 which is consisted of a sequence of 250 complete (all-eight-layers) postures. At each point in time one complete posture can be constructed by using all eight indexes of ***Poselets*** in each layer.

If a multi (P)-path gesture has N ($= 10$) postures, the size of the transition matrix will be 10×10. Furthermore, if there are M ($= 5$) gestures of the same class in the observation/training data (with variations), the size of the emission matrix will be $N \times M$ (10×5). One main issue with this definition is the inaccurate and inefficient determination of the number of emissions. More specifically, we cannot systematically determine how many same-class variations of a gesture should be recorded to calculate the emission matrix efficiently. Our ***Topolet***, however, introduces a systematic approach for computing that matrix. That is, the emissions are calculated based on the distance of the inter-space points to the neighboring points. That topological information determines the same-class variations accurately and thus, reduces the required number of recordings for identical class of gestures. Furthermore, in our approach we do not need an extra signal processing technique (on pixel-level) to convert the continuous paths into P-symbols, as it is reported in (Yang et al., 1994).

To determine the emission matrix we utilize two mentioned databases (*MD* and *LD*) in K-means framework (MacQueen, 1967). That is, the ***Poselets*** within *LD* database are employed as the initialization centers of the K-means (the number of *LD* ***Poselets*** determines the number of clusters). Whereas, the ***Poselets*** in *MD* database are regarded as the points' cloud which are to be clustered (Figure 9.c). Therefore, those ***Poselets***, which are classified in one cluster, is considered as the same-class variations for each state. Within this efficient approach, the emission probabilities determine the closeness of each emission to the center of that cluster. Thus, each probability is calculated based on the ratio of two distances as: $\frac{d_e}{\sum d_e}$, where d_e marks the distance of each element from the center of cluster. Since each K-means cluster has different number of ***Poselets***, we consider the minimum number, existing in all clusters, as the number of emissions, M_{L_i}, for that layer. Therefore, $N_{L_i} \times M_{L_i}$ emission matrix is constructed.

4 EXPERIMENT

To demonstrate the efficacy of our ***Topolet***, we conduct five experiments. For these experiments, the input to the recognition system is created as following: First, a Viterbi-like algorithm is employed to return a path of Q states, based on the (maximum-a-*posterior*) probabilities of the n_s-nearest neighbors for each layer. The initial state and length of the ***Topolet***, Q, for each layer are the inputs to the Viterbi algorithm. Furthermore, the number of returned paths is equal to the number of considered layers of interest in each experiment. For example, if an experiment is aimed at evaluating the performance of complete posture sequence, eight different paths (one for each layer) is returned (Figure 7). Second, the ***Poselet***'s image corresponding to each of these state-indexes' sequence is retrieved (Figure 9.a). Similarly, if the experiment is on multiple layers (Exp 2-5), the input

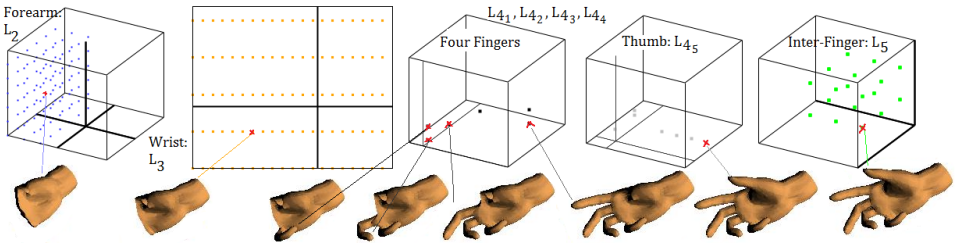


Figure 8: Process of combining eight parallel ***Poselets*** to construct a posture. The next posture is constructed from the nearest-neighbors of these ***Poselets***. The ***Poselet*** spaces' colors correspond to the colors' code of each layer in Figure 1.

will be the combination of all ***Poselets*** in the parallel format to form a complete posture image (Figure 8). Third, the OpenCV's contour extraction method (Bradski, 2000) is applied to the input frames to extract the contour points. In our experiments, the input and searched images' size is 800×600 pixels, and the 3D hand model's length and width (in 3D frontal view) are 10 and 4 units (cm), respectively.

Estimation is performed using the maximum-a-prior (MAP) criterion and the Chamfer distance as the penalty function. The criterion finds the optimal ***Poselet*** at each layer by comparing the contour points of the nearest ***Poselets*** and of the input. If the input is a sequence of ***Poselets*** in one layer, the solution to each frame is one index which minimizes the penalty function. Whereas, if the input is a sequence of complete postures, the solution will be a set of (eight) indexes. During the search, the previous state of each ***Poselet*** in the sequence is known.

The first experiment evaluates the estimation accuracy and time of the ***Topolet*** for layer-two (forearm-rotation), layer-three (wrist-rotation), and layer-five (inter-finger rotation), separately. The goal is to demonstrate, by employing our ***Topolet*** we can outperform the results (mean errors and fps) in comparison with layered-exhaustive search (Dadgar and Brunnett, 2018). For this experiment, we have created a synthetic gesture consiting of 250 ***Poselets***, and as the results show (Table 1), considerable improvement is recorded during estimation using our ***Topolet***. The estimation of the layers- three and five is relatively easier than the layer-two, because we set the forearm to bind-pose (frontal and upward-view in this case) in those two searches (Figure 9.a). On the other hand, that frontal-view of the ***Poselets*** leads to a slower fps than many other views of the forearm. That is due to the higher number of contour points existing in the frontal view of the hand. That greater number of contour points obliges more expensive contours extraction and slower pixel-wise comparison process than other views.

In experiments two and three we consider the combination of all layers (L_2, L_3, L_4, and L_5).

Table 1: Comparison of the ***Topolet*** and layered-exhaustive search (Layered-EXH) methods. This experiment is performed on layer-two (L_2), layer-three (L_3), and layer-five (L_5). The performance improvement (duration and accuracy of the searches) based on our (***Topolet***) is evident.

Exp1 *Topolet*	L_2	L_3	L_5
Mean 3D Error (cm)	0.246	0.244	0
Mean 2D Error (px)	0.7	0.7	0
Search Time (s)	0.094	0.108	0.195
Exp1 Layered-EXH	L_2	L_3	L_5
Mean 3D Error (cm)	1.503	0.557	0
Mean 2D Error (px)	11.4	16.5	0.1
Search Time (s)	1.859	2.480	1.718

Accordingly, we consider a synthetic gesture with 250 postures for experiment-two and 2500 postures for experiment-three. While the experiment-two is oriented to evaluate the performance on the low-resolution (*LD*) database, the experiment-three is targeted to examine the efficiency of the ***Topolet*** on the mid-resolution (*MD*) database. The goal of the experiment-three is to demonstrate the possibilities of decreasing the errors (compared to experiment-two) by training our ***Topolet*** on the layers with a more aggregated points-cloud. We can observe that experiment-three exhibits better accuracy while the fps remains almost the same (Table 2). The main source of the errors is due to the invisibility of the fingers in many inputs. Since the image sequences are selected in a random Viterbi-like process, encountering invisible fingers is inevitable. Depending on the application, one could constraint the hand to have reasonable rotation toward the camera by which the

Table 2: The overview of experiment-two and -three. These two experiments are aimed to find the optimum complete-postures (combination of all layers). The enhancement of the accuracy in experiment-three (which is performed on mid-resolution (*MD*) database) is evident while the search duration is not noticeably affected.

Experiments	Exp 2	Exp 3
Mean 3D Error (cm)	2.221	2.019
Mean 2D Error (px)	6.5	10
Search Time (s)	0.49	0.64

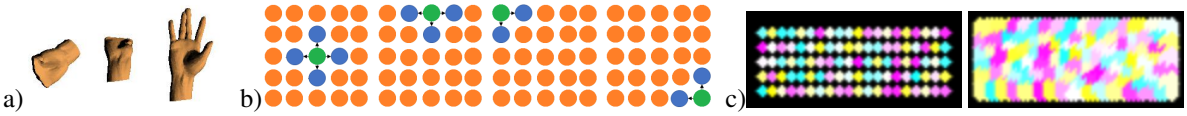


Figure 9: a) Three examples of ***Poselets*** in layer-2, layer-3, and layer-5. b) $n_i = 4$ possible $n = 1$-nearest-neighborhoods (blue) of the current state (green). If the current state is at the edge of the rectangle, the number of possible transition will be less ($n_i = 3$ or 2). In our implementation we consider the self-transition to make the iterative gestures also possible ($n_i = 5, 4$, or 3). Note that, n is different from n_i: n determines the how far the algorithm should look for the next state in the topological space. The value of n is constant for the entire process and we set it to one. Whereas, n_i determines the actual number of neighbors for each state. The value of n_i depends on the position each state is situated in the topology. c) Kmean over the LD and MD databases and the formation of emission matrix.

fingers become (at least) partially visible. That could increase the accuracy by a significant factor.

Table 3: The overview of experiment-four and -five. Experiment-four is performed on distorted contours and experiment-five is accomplished without considering the layer-five in the search. Enhancement of the search duration in both experiments (in comparison with experiment-two and -three) is evident while the accuracy remained almost unchanged.

Experiments	Exp 4	Exp 5
Mean 3D Error (cm)	2.025	2.028
Mean 2D Error (px)	10.3	10.3
Search Time (s)	0.32	0.25

The result of experiment-three shows a reasonable accuracy, however, we have conducted two more experiments (the experiments- four and five) to improve the time-complexity of the search. Therefore, similar to experiment-three, these experiments are designed and performed on the mid-resolution database (*MD*), with identical gesture sequence. However, in experiment-four the number of contour points (in contour extraction) are reduced, and in experiment-five the layer five (inter-finger rotation) is not considered and is set to bind-pose. The idea behind experiment-four is to evaluate the robustness of our ***Topolet*** when distorted contours are employed. Whereas, the idea behind experiment-five is to examine the entropy of information existing in layer-five in comparison to the time-complexity it imposes on the whole process. As it is demonstrated in Table 3, the results exhibit vivid improvements in the time-complexities of experiments- four and five (3 fps and 4 fps, respectively) in comparison to the experiment-three (while the accuracy remains almost unchanged). Additionally, the reduction of the contour points in experiment-four has no effect on improvement of the time-complexity after a certain value. That is, in every other p contour point selection process, $p = 4$ is the optimum value.

Table 4: The detailed overview of experiment-four and -five. According to the table, it is evident that contour extraction time and visualization of the ***Poselets*** using openGL is the most expensive part of the system.

Experiments	Exp 4	Exp 5
Contour Ext Time (s)	0.11	0.08
2D Distant Comp Time (s)	0.03	0.03
Visualization Time (s)	0.11	0.09

5 DISCUSSION & CONCLUSION

In the past decades many researchers strove to overcome the obstacles in designing of a reliable hand gesture recognition system. Those obstacles were especially more severe if the system is ought to perform in real-time, employ single RGB camera, and recognize a large set of gestures. Therefore, many systems relaxed one or two of these requirements. We proposed a type of temporal model for gestures based on our hierarchical hand posture database to meet all the above requirements. Our HMM-like model utilizes the topology of the points' cloud in each layer and exploits a novel type of atomic structure, we refer to as ***Topolet***. Furthermore, it benefits from an enhanced version of another atomic gesture structure, known as ***Poselet***. These two atomic structures allowed us to consider random, and thus context-free application domain, gesture set. Moreover, the system utilized a single and RGB (instead of depth (Sharp et al., 2015)) camera and showed a great potential to perform in real-time.

Our ***Topolet*** successfully addressed many traditional issues exist in conventional HMM. For example, one difficulty with HMM is to determine an appropriate number of hidden states even if the context of application domain is known (Sangjun et al., 2015). The database we utilized contained a fix number of ***Poselets*** at each layer which was uniformly distributed within the lower (than 28) dimensional space. Since our proposed ***Topolet*** considered these ***Poselets*** as the transitional states, the number of states in the temporal model was remained fixed, thus the issue was resolved. An other difficulty with conventional HMM is the linear space-complexity as the number

of considered gestures increases. Our ***Topolet*** specified the sequence of possible ***Poselets*** (and not postures) within a gesture. One ***Topolet*** was trained for each layer and a concrete relation between different ***Poselets*** of that layer is introduced. That led to construction of *semi-gestures* (or ***Topolets***). The ***Topolets*** of all layers together captured the temporal information between different postures. Therefore, the entire temporal information of any gesture consisted of several but limited (eight) ***Topolets*** in a parallel mode. This limited number of ***Topolets*** remained the same for any configurations and for any number of gestures, and thus the issue of linear growth was resolved. Additionally, our ***Topolet*** enhanced the structure of the training, decoding, and evaluation algorithms. For example, conventional HMM theoretically requires infinite number of examples for one gesture to reliably train the parameters using the complex Baum-Welch algorithm. Our method investigated the topological relation of the ***Poselet*** points and, therefore, reliable relations between the postures were established using one example. After all, within a vision-based approach (RGB Camera), and in dealing with (potentially) infinite number of (randomly created) gestures, we achieved a notable 3fps in Exp4 and 4fps in Exp5. We used a naive contour extraction method which was an expensive part of our process (Table 4). By employing a real-time contour extraction algorithm (deep net (Bertasius and Torresani, 2015)) we could achieve higher fps on this time consuming process.

In addition to those proven advantages, our proposed method could introduce a number of other potential benefits. The atomic structures could be used to construct a comprehensive grammar between the ***Poselets***, ***Topolets***, postures, and gestures. That is, if we consider each gesture as a sentence and each posture as a word, one ***Poselet*** could be viewed as a word syllable (with each DoF of the ***Poselet*** as a letter). Then each of the trained ***Topolet*** specifies how each of those ***Poselets*** could evolve as the gesture proceeds in time. Therefore, the system will be capable of forming a flexible description-based specification for hand gesture database (Wang et al., 2012). Furthermore, in the process of gestures transition, there might be some transient (false) gestures depending on the context of the application, where the system should perform no specific action. With our comprehensive gesture's temporal model, one could deactivate some of the points' cloud in each layer and acquire the ***Topolets*** on a smaller space. That feature could be used to concretely adopt the proposed ***Topolet*** to different application domains. The evaluation of these statements will be the topics of future works.

ACKNOWLEDGMENT

I would like to thank the Europäischer Sozialfonds für Deutschland (ESF) scholarship and the Professorship of Graphische Datenverarbeitung und Visualisierung at TU-Chemniz who made this research possible.

REFERENCES

Bertasius, G. and Torresani, L. (2015). DeepEdge : A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection. In *CVPR*.

Bourdev, L. and Malik, J. (2009). Poselets: Body Part Detectors Trained using 3D Human Pose Annotations. *IEEE Int Conf on Com Vis*, pages 1365–1372.

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

Dadgar, A. and Brunnett, G. (2018). Multi-Forest Classification and Layered Exhaustive Search Using a Fully Hierarchical Hand Posture / Gesture Database. In *VISAPP*, Funchal.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Berkeley Symp on Maths, Statis, Prob*, 1(233):281–297.

Meshry, M., Hussein, M. E., and Torki, M. (2016). Linear-Time Online Action Detection from 3D Skeletal Data using Bags of Gesturelets. *IEEE WACV*.

Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.

Sangjun, O., Mallipeddi, R., and Lee, M. (2015). Real Time Hand Gesture Recognition Using Random Forest and Linear Discriminant Analysis. *Proc of the Inter Conf on Human-Agent Interaction*, (October):279–282.

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.

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

Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012). Mining Actionlet Ensemble for Action Recognition with Depth Cameras. pages 1290–1297.

Yang, J., Xu, Y., and Chen, C. S. (1994). Gesture Interface: Modeling and Learning. *IEEE Proc Int Conf on Robotics and Automation*, pages 1747–52 vol.2.

Yao, B. and Fei-Fei, L. (2010). Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions. *IEEE Proc CVPR*, pages 9–16.

Yu, G., Liu, Z., and Yuan, J. (2015). Discriminative Orderlet Mining for Real-Time Recognition of Human-Object Interaction. *Lec Notes in Comp Sci (AI & Bioinf)*, 9007:50–65.