Exploring Discriminative Pose Sub-Patterns for Effective Action Classification*

Xu Zhao*† zhaoxu@sjtu.edu.cn
Yuncai Liu* whomliu@sjtu.edu.cn
Yun Fu† yunfu@ece.neu.edu

*Department of Automation
Shanghai Jiao Tong University, Shanghai, 200240

†Department of ECE
Northeastern University, Boston, MA 02115

ABSTRACT

Articulated configuration of human body parts is an essential representation of human motion, therefore is well suited for classifying human actions. In this work, we propose a novel approach to exploring the discriminative pose sub-patterns for effective action classification. These pose sub-patterns are extracted from a predefined set of 3D poses represented by hierarchical motion angles. The basic idea is motivated by the two observations: (1) There exist representative sub-patterns in each action class, from which the action class can be easily differentiated. (2) These sub-patterns frequently appear in the action class. By constructing a connection between frequent sub-patterns and the discriminative measure, we develop the SSPI, namely, the Support Sub-Pattern Induced learning algorithm for simultaneous feature selection and feature learning. Based on the algorithm, discriminative pose sub-patterns can be identified and used as a series of "magnetic centers" on the surface of normalized super-sphere for feature transform. The "attractive forces" from the sub-patterns determine the direction and step-length of the transform. This transformation makes a feature more discriminative while maintaining dimensionality invariance. Comprehensive experimental studies conducted on a large scale motion capture dataset demonstrate the effectiveness of the proposed approach for action classification and the superior performance over the state-of-the-art techniques.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications; H.2.8 [**Database Applications**]: Data Mining

Keywords

Action classification; sub-pattern mining; feature transform

*This work is supported in part by the NSF CNS award 1135660 and 1314484, Office of Naval Research award N00014-12-1-0125 and N00014-12-1-1028, Air Force Office of Scientific Research award FA9550-12-1-0201, U.S. Army Research Office grant W911NF-13-1-0160, National Basic Research Program (973) under No. 2011CB302206 and NSFC grant under No. 61273285.

MM'13, October 21–25, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502094.

1. INTRODUCTION

Human Motion Analysis (HMA) has been an interdisciplinary research topic for decades and is attracting continuing interests from multimedia, computer vision, computer graphics and medical research communities, due to the potential applications in human-computer interaction, clinical studies, content-based video indexing, intelligent surveillance, assisted living and so on [15, 24]. Action classification is an important research thread of HMA and constructs the basis for many of the above mentioned applications.

Most of the action classification works focus on recognizing actions using visual appearance features extracted from videos [14, 27, 23] or still images [32, 22, 34]. Only a small number of early works [31, 16] and recent works [22, 12, 32] introduce pose as a cue, not straightforwardly, for action classification. Although extracting low level appearance features is much easier than obtaining high-level pose parameters, we argue that human body pose, the articulated configuration of human body parts, is essential to explain the inherent nature of human motion, and therefore contains more discriminative information for action classification.

The critical role of pose information for action recognition had been validated by the Johansson experiment [20], which shows that using 10-12 bright spots, describing the motions of the main joints of a living body, can evoke a compelling impression of human walking, running, dancing, etc. In comparison, appearance features are noisy and not robust to the change in external imaging condition. Most importantly, appearance is not an essential description of human motion. Furthermore, recent advances in human pose estimation [29, 11, 33, 5, 4] facilitate the use of pose as input for action recognition in a practical vision system. Motivated by the above facts, in this work, we focus on using human pose as a feature and exploring its potential effectiveness for the action classification task.

Different from visual features of which each component carries few explicit semantical meaning about human motion, pose feature has several noticeable properties: (1) *Physicality*. Each component of the pose feature has corresponding physical significance. In this work, hierarchical motion angles are introduced to represent body pose (see Section 5.1). (2) *Clarity.* Pose feature is concise and noise free; therefore it is well suited for revealing the implicit relationship between pose components and action classes. (3) *Visibility.* The intermediate and final results can be visualized to show the correlation between pose patterns and action classes through qualitative analysis. Such properties enlighten us about the different ways that pose features can be utilized. The idea is inspired by how human beings perceive actions.

As the common human experience, action class information can be easily captured from partial observations, for example, with some parts of the body occluded. If a whole body pose is viewed as

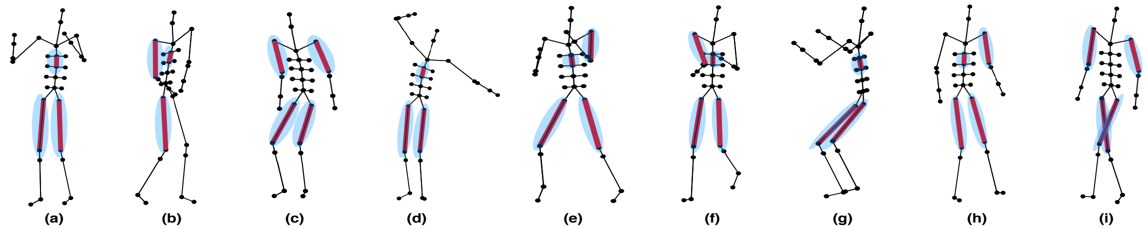


Figure 1: Body pose randomly selected from 9 action classes: chicken dance, golf swing, jump, cartwheel, boxing, run, motorcycle, pick up and walk respectively from (a) to (i). The ellipse occluded red parts are the representative pose sub-patterns. We aim at exploring the discriminative power of the sub-patterns and employing them for efficient action classification.

a full pattern, the partial observation corresponds to a sub-pattern, which is a physical configuration of some local body parts. Then, there exist representative sub-patterns in each action class, which can be used to differentiate it from other classes. As can be seen from Figure 1, one can recognize the actions very likely by just seeing the ellipse occluded body parts. To effectively utilize pose sub-pattern for action classification, however the following critical questions need to be further addressed.

- How to find the representative pose sub-patterns for a specific action class?
- How the pose sub-patterns can help improve the classification performance?
- How to evaluate the discriminative power of pose sub-patterns within the context of action classification?

In this paper, we answer the above questions systematically and provide solutions using a novel feature transform framework.

Extracting sub-patterns from full poses is an exponential complexity problem. Fortunately, benefiting from the physicality property of pose features, existing frequent pattern mining algorithm can be used to generate Support Sub-Patterns (SSP). This is motivated by the fact that the representative sub-patterns of a particular action class appear more frequently than other sub-patterns in that action class. For example, for action "Boxing", the typical sub-pattern "arms held up to defend" appears most frequently therefore it is recognized as SSP. Seeing just this sub-pattern can provide lots of valuable information about the entire action "Boxing" (see Figure 9). Thus, by mining frequent patterns, the scalability issue for classification in large pose databases could be solved.

Employing the obtained sub-patterns to make classification is the heart of this work. We propose a Support Sub-Pattern Induced (SSPI) feature transform framework. As a completely new approach, SSPI differs from two typical feature transform frameworks, namely, feature combination and feature reduction. These frameworks map the features to a higher and lower dimensional space respectively to increase the discriminative power of the transformed features. In contrast, SSPI does not change the feature dimensionality and the transform occurs in the bits correlated with the corresponding SSP. Namely, if a full pose pattern contains a SSP, this SSP will dominate a change to the original full pattern. In other words, SSPs perform like a series of fixed "magnetic centers" on the surface of a normalized feature super-sphere. Each center leads to a moving of the full pattern along the surface and the resultant force from all the SSPs forms the final transformed feature. As far as we know, this is the first work conducting feature transform in this way.

During the feature transformation, there are two critical issues. (1) What's the criterion for evaluating such a transform? (2) How much "force" from the SSPs should be acted on the full pattern? Because our final aim is to improve the accuracy of action classification, Fisher score is chosen as the criterion to construct the objective function. And, each SSP is coupled with a weight, which is used to calculate the strength of the attraction. The weights are the parameters of the Fisher score function and can be learned from pose data by maximizing the objective function. Once the weights are determined, the feature will be transformed using the Fisher score optimum. One can also evaluate the discriminative power of a SSP according to its impact on the objective function.

To summarize, the main contributions of this work are as the follows:

First, utilizing the "physicality", "clearness" and "visibility" properties of pose data , we introduce frequent pattern mining to explore Support Sub-Patterns. The SSPs are representative sub-patterns of the action classes and can be applied to action classification, even action search, pose estimation and human detection.

Secondly, we propose a SSP induced feature transform approach, through which the feature keeps the original dimensionality but is converted into a more discriminative formation for classification via SSP induced feature learning. This framework could be applied to more general feature transform applications.

Thirdly, for the SSPI framework, we construct the objective function using the Fisher score optimum. An efficient two-phases optimal algorithm for searching the optimal weights of the SSPs is developed. Experiments on large scale pose databases demonstrate the superior performance of the proposed algorithm on the action classification task.

2. RELATED WORK

Action Classification. A large number of methods have been proposed for recognizing human actions. Here we focus on methods most related to our proposed work. Readers interested in more details can refer to some recent reviews [1, 24] on this topic.

Methods that use pose as a feature for action classification usually exploit the space-time trajectories to interpret human activities. Campbell et al. [6] recognized human actions by representing them as curves in low-dimensional phase spaces. Template matching is used to find action labels. Similar methodologies can be found in [26, 35, 28, 3], where either trajectories formed by joints [28, 35, 3] or curvature patterns of trajectories formed from body parts [26] are treated as features. Slightly different from the explicit trajectory based methods, in [31], motion of body parts is treated as a set of signals to describe sequential changes of feature values. In a much earlier work [30], Siskind animated a line-drawing of a person as pose input and classified actions such as dropping, throwing, picking up, and putting down by analyzing the event logic.

In sum, a common practice of these types of work is to mainly consider the temporal behaviors of body pose but not the relationship between the spatial configuration of poses and action classes.

In this work, the proposed idea of SSPs is used for mining the internal correlation between spatial sub-patterns and action labels.

Another recent research trend to employ pose as a cue in action recognition is closely related to the concept of "poselets" [22, 12, 32]. The basic idea is that semantically similar actions or poses should belong to clusters defined by the same poselets, no matter what the appearance looks like. In doing so, pose becomes an efficient and complementary cue of appearance features for action recognition. These kind of methods are advantageous in that they combine more sources of information. However, the noisy nature of image data and the required manual annotation make it difficult to explore the relationship between pose patterns and action class labels in a meaningful way.

Pattern Based Classifcation. Pattern based classification is a focused research topic in data mining field, where the discriminative frequent patterns usually are taken as features to build high quality classifiers [21, 8, 9, 13]. Frequent sub-patterns reveal the associations between data attributes, therefore carrying rich semantic information about the data. The importance of frequent pattern in classification tasks has been thoroughly investigated in [8].

The construction of a feature representation using frequent patterns is the core of pattern based classification. Cheng et al. [8, 9] achieved this by feature combination, namely, mapping the original features into a higher dimensional space by combing the frequent patterns with the original single features. The critical part of this procedure is feature selection, by which the most discriminative frequent patterns are selected to construct the final feature representation. In [8], feature selection is separately conducted from the generation of frequent patterns. To speed up the algorithm and mine highly discriminative patterns, direct discriminative pattern mining approaches are proposed in [9, 13, 21].

Besides aforementioned methods, associative classification is another class of frequent pattern based classification approaches. It is rule based and the association between frequent patterns and class label is used for classification [19]. It's also worth mentioning that Yuan et al. [36] found the visual patterns with semantic correlations by mining frequent patterns from image databases.

Our work is different from above approaches in the following aspects: (1) Feature transform is not a simple feature combination. Changes occurr only in the bits correlated with the corresponding frequent patterns and the feature is length invariant. (2) Feature selection is implicitly integrated into the feature transform procedure by learning the transform weights of the frequent patterns. (3) The discriminative power of the frequent patterns can be explicitly evaluated according to the learned transform model.

3. PROBLEM FORMULATION

In the context of action classification, assume a pose dataset has k categorical attributes, each of which is a motion angle and has a set of values. These pose data points fall into m action classes $\mathcal{C} = \{c_1, \cdots, c_m\}$. Thus, each $(angle, value)$ pair can be mapped to a distinct item in $\mathcal{P} = \{o_1, \cdots, o_d\}$. Since pose angles are represented by continuous numerical attributes, the values should first be discretized. Here we adopt binary coding. Suppose a pair $(ang, val) \mapsto o_i$, where ang is a joint angle and val is a value. Let $\mathbf{x}$ be the pose vector of a data point s. Then $x_i = 1$ if $ang(s) = val$; $x_i = 0$ if $ang(s) \neq val$. In this way, the dataset is represented in $\mathbf{B}^d$ as $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbf{B}^d$, $y_i \in \mathcal{C}$ and $x_{ij} \in \mathbf{B} = \{0, 1\}$, $\forall i \in [1, n], j \in [1, d]$. Therefore, after discretization and binary mapping, a pose data point is mapped to a binary pattern, which is a convenient form to use for sub-pattern exploration. For the ease of analysis, in the following we define some related concepts built on the basis of pose patterns [8].

Definition 1. (*Sub-Pattern*) A sub-pattern $\alpha = \{o_{s_1}, \cdots, o_{s_k}\}$ is a subset of $\mathcal{P}$, where $s_i \in \{1, \cdots, d\}$, $\forall i \in [1, k]$. A single pattern $o_i \in \mathcal{P}$ is the special case of sub-pattern. Therefore, given a dataset $D = \{\mathbf{x}_i\}$, the set of data points that contains sub-pattern α is denoted as $D_\alpha = \{\mathbf{x}_i | x_{is_j} = 1, \forall o_{s_j} \in \alpha\}$. An equal-length binary expression of a sub-pattern α is noted as $\mathbf{x}_\alpha \in \mathbf{B}^d$ with $x_\alpha = 1$ for $\forall o \in \alpha$ and $x_\alpha = 0$ for $\forall o \notin \alpha$.

In this work, we aim to explore the discriminative pose sub-patterns for action classes. The support sub-patterns related to the action class c are defined as:

Definition 2. (*Support Sub-Pattern (SSP)*) For a dataset D_c with class label c, a sub-pattern α is supportive for this class if $\theta = \frac{|D_{\alpha,c}|}{|D_c|} \geq \theta_0$, where θ_0 is the minimum support threshold with $0 \leq \theta_0 \leq 1$, hence θ is the relative support of α. The set of support sub-patterns of class c is denoted as $\mathcal{SP}_c$.

SSP actually defines the sub-patterns that most frequently appear in each action class as the supportive patterns for classification. This is consistent with our daily observation that there exist some typical poses in each action class, which can describe the action very well. Additionally, these poses often occur with high frequency while any subject performs the action. Therefore these poses, or more precisely, the pose sub-patterns, are discriminative and representative of the action class. Very often one can recognize an action class by just observing a local pose, for example, a pose of a leg or an arm when other parts of the human body are occluded. The observed sub-parts that can help people recognize actions correspond to the SSP. We will make further explorations of SSPs in the following section.

4. FEATURE TRANSFORM

In this section, we focus on the feature transform for effective classification based on SSPs. Here, feature transform is an operation performed on the original pose data points. For a data point $\mathbf{x}$, a feature transform could be any function $\{f : \mathbf{x} \mapsto \mathbf{x}', f \in \mathbb{F}\}$ from functional space $\mathbb{F}$. After the transformation, one can expect much better classification performance in the transformed feature space than the original one. We examine two kinds of feature transform methods in the following sections.

4.1 Conventional SSP Based Feature Combination

As introduced in [8], the conventional frequent patterns based feature transform is a form of non-linear feature combination over the set of single features. Under this transform, given a dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ and a set of frequent sub-patterns $\mathcal{SP}$ collected from all the classes, D is mapped to a higher dimensional feature space, by integrating the SSP and the original items. Hence the feature space $\mathbf{B}^d$ $(d = |\mathcal{P}|)$ is enlarged to $\mathbf{B}^{d'}$ with features taken in $\mathcal{P} \cup (\cup_{i=c_1}^{c_m} \mathcal{SP}_i)$. After the transform, the data is denoted as $D' = \{\mathbf{x}_i', y_i\}_{i=1}^n$, where $\mathbf{x}_i' \in \mathbf{B}^{d'}$ and

$$d' = |\mathcal{P} \cup (\cup_{i=c_1}^{c_m} \mathcal{SP}_i)|. \quad (1)$$

Then, what is the philosophy behind the conventional frequent pattern based classification? To make the picture more clear, we summarize the motivations as follows according to [8].

- The discriminative power of a SSP is much higher than the original single features because they capture more underlying semantics of the data. It's true for human action classification, as explained in Section 3.

- The dimensionality of feature space is increased to $d^{'}$ from d (see Eq. (1)). This will also likely increase the chance of including discriminative features.
- The discriminative power of low frequency sub-patterns is confined by a small value because of their limited coverage in the dataset. Symmetrically, it's true for the very high frequency sub-patterns due to their commonness in the data.

The implementation of the conventional SSP based feature combination is straightforward. The detailed description is given in Algorithm 1. One can refer to [8] for the proof of the above statements and the procedures for feature selection mentioned in the Algorithm 1.

Algorithm 1 Conventional SSP Based Feature Combination

Input: Original dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, $min\_sup : \theta_0$
Output: Transformed dataset $D^{'} = \{\mathbf{x}_i^{'}, y_i\}_{i=1}^{n}$
for $c_i = c_1$ to c_m **do**
    $\mathcal{SP}_{c_i} =$ Apriori(D_{c_i}, θ_0)
        ▷ Apriori represents the algorithm of frequent pattern mining
end for
$\mathcal{SP}_s =$ FeatureSel$(\cup_{i=c_1}^{c_m} \mathcal{SP}_i, param)$
        ▷ FeatureSel functionalized as the procedure for feature selection, $param$ is the related parameters.
for all $\mathbf{x}_i \in D$ **do**
    $\mathbf{x}_i^{'}$ = CheckSPP$(\mathbf{x}_i, \mathcal{SP}_s)$, $|\mathbf{x}_i^{'}| = |\mathbf{x}_i| + |\mathcal{SP}_s|$
        ▷ CheckSPP checks whether the SPPs appear in $\mathbf{x}_i$ or not. If appear, the corresponding additional indicator bit is set as 1; if not, set it as 0.
end for
return $D^{'} = \{\mathbf{x}_i^{'}, y_i\}_{i=1}^{n}$

4.2 SSP Induced Feature Transform

SSP based feature combination improves the classification performance by explicitly adding additional bits. It extends the features to a much higher space meanwhile loads more computational burden onto classifier. To avoid explicit feature combination and feature selection, we propose a novel SSP Induced (SSPI) feature transform algorithm, by which the feature length is kept invariant and the feature selection is implicitly incorporated into the learning process for feature transform.

4.2.1 Fisher Score Oriented Feature Learning

Fisher score is a general measure for feature selection [17]. However, in our algorithm, it is used as an orientation for learning based feature transform. Given the original pose data set $D = \{\mathbf{x}_i, y_i\}_{i=1}^{n}, \mathbf{x}_i \in \mathbf{B}^d$. The Fisher score is defined as

$$F(\mathbf{X}) = \mathrm{tr}\{(\tilde{\mathbf{S}}_b)(\tilde{\mathbf{S}}_t + \gamma \mathbf{I})^{-1}\}, \tag{2}$$

where $\mathbf{X} \in \mathbf{B}^{d \times n}$ and γ is a positive regularization parameter. $\tilde{\mathbf{S}}_b$ is called between-class scatter matrix and $\tilde{\mathbf{S}}_t$ is called total scatter matrix, which are respectively defined as follows

$$\begin{aligned} \tilde{\mathbf{S}}_b &= \sum_{k=1}^{m} n_{c_k} (\tilde{\boldsymbol{\mu}}_{c_k} - \tilde{\boldsymbol{\mu}})(\tilde{\boldsymbol{\mu}}_{c_k} - \tilde{\boldsymbol{\mu}})^T \\ \tilde{\mathbf{S}}_t &= \sum_{i=1}^{n} (\mathbf{x}_i - \tilde{\boldsymbol{\mu}})(\mathbf{x}_i - \tilde{\boldsymbol{\mu}})^T, \end{aligned} \tag{3}$$

where $\tilde{\boldsymbol{\mu}}_{c_k}$ and n_{c_k} are the mean vector and the size of the k-th class respectively. $\tilde{\boldsymbol{\mu}} = \sum_{k=1}^{m} n_{c_k} \tilde{\boldsymbol{\mu}}_{c_k}$ is the overall mean vector. To statistically improve the classification performance, we can maximize the distance between data points in different classes, while minimizing the distance between data points in the same class. This can be achieved by maximizing the Fisher score in Eq. (2). To this end, we need to find a feature transform $f(\cdot)$ for all data points such that $f : \{\mathbf{x}_i\}_{i=1}^{n} \mapsto \{\mathbf{x}_i^{'}\}_{i=1}^{n}$ and satisfy

$$\max_{f(\cdot)} F(\mathbf{X}^{'}) = \mathrm{tr}\{(\tilde{\mathbf{S}}_b)(\tilde{\mathbf{S}}_t + \gamma \mathbf{I})^{-1}\}, \tag{4}$$

where $\mathbf{x}_i, \mathbf{x}_i^{'}$ are with the same dimensionality d in our work and $\tilde{\mathbf{S}}_b, \tilde{\mathbf{S}}_t$ are constructed using the transformed data $\{\mathbf{x}_i^{'}\}_{i=1}^{n}$. Assume we collect k SSPs: $\{\mathcal{SP}_j\}_{j=1}^{k}$ from all the classes by frequent pattern mining using the same parameter min_sup. Since the SSPs contain discriminative class information, the feature transform of any data point $\mathbf{x} \in \mathbf{B}^d$ will be SSP induced and can be formulated as follows

$$\mathbf{x}_j^{'} = \begin{cases} (\mathbf{x}_{\mathcal{SP}_j} \oplus \mathbf{x}) \cdot \phi_j, x_{js_i}^{'} = 1, \forall o_{s_i} \in \mathcal{SP}_j & \text{if } \mathbf{x} \in D_{\mathcal{SP}_j} \\ \mathbf{x} & \text{otherwise} \end{cases} \tag{5}$$

$$\mathbf{x}^{'} = \frac{1}{k} \sum_j \mathbf{x}_j^{'} \tag{6}$$

where $\mathbf{x}_{\mathcal{SP}_j}$ is the equal-length binary expression of sub-pattern $\mathcal{SP}_j$ and $D_{\mathcal{SP}_j}$ is the dataset containing $\mathcal{SP}_j$, as defined in Definition 1. $\mathbf{x}_j^{'}$ is the transformed feature under the function of $\mathcal{SP}_j$. $\mathbf{x} \in D_{\mathcal{SP}_j}$ ensures that feature transform is performed only when $\mathbf{x}$ contains the SSP. "$\oplus$" is the exclusive OR operation on two binary strings for checking the bit difference between $\mathbf{x}$ and $\mathbf{x}_{\mathcal{SP}_j}$. $o_{s_i} \in \mathcal{P}$ is an item of $\mathbf{x}$. After the "$\oplus$" operation, the bits of $\mathbf{x}$ with value "1" corresponding to the items of $\mathcal{SP}_j$ are unexpectedly changed to "0". We need to set them back to original "1". By doing this, change will only take place in the bits where $\mathbf{x}$ is "1" whereas $\mathbf{x}_{\mathcal{SP}_j}$ is "0". ϕ_j is the weight parameter coming up with $\mathcal{SP}_i$, which indicates the "force" of $\mathcal{SP}_i$ on the data points containing it or $\mathcal{SP}_i$'s impact on the Fisher score. The final transformed feature $\mathbf{x}^{'}$ is the resultant of the individual "forces" from all the SSPs.

Then, after Eq. (5) and (6) are plugged into Eq. (4), the Fisher score becomes a function with respect to $\Phi = [\phi_1, \phi_2, \cdots, \phi_k]$. We are going to find the optimum Φ^* to maximize the Fisher score $F(\Phi)$

$$\Phi^* = \operatorname*{argmax}_{\Phi = [\phi_1, \phi_2, \cdots, \phi_k]} F(\Phi). \tag{7}$$

Due to the complex non-linear transform operation, it's difficult to derive the derivation $\frac{\partial F}{\partial \phi}$ analytically. However, the property of the objective function is well defined and the optimum searching can be efficiently performed using non-gradient based optimization algorithms. Our optimization scheme and the implementation of feature transform procedure are introduced in the next section.

4.2.2 Algorithm Implementation

Implementation of Feature Transform. The implementation of the proposed feature transform algorithm is summarized in Algorithm 2. It's worth pointing out that after feature transform, the data points are not with binary format anymore, because the bit value has been mapped to real domain. The efficiency of the algorithm is sufficient to be utilized on large scale datasets because the computational complexity of the SSPI feature transform procedure is linear with regards to the size of dataset.

Optimization. In Algorithm 2, after going through feature transform, the new dataset $\hat{D}^{'}$ turns into the function of parameters

Algorithm 2 SSP Induced Feature Transform

Input: Training dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, $\{\mathcal{SP}_j\}_{j=1}^{k}$
Output: Transformed dataset $D^{'} = \{\mathbf{x}_i^{'}, y_i\}_{i=1}^{n}$, Φ^*
for $i = 1$ to n **do**
    for $j = 1$ to k **do**
        if $\mathbf{x}_i \in D_{\mathcal{SP}_j}$ **then**
            $\hat{\mathbf{x}}_{ij}^{'} = (\mathbf{x}_{\mathcal{SP}_j} \oplus \mathbf{x}_i) \cdot \phi_j$
            $\hat{x}_{is.}^{'} = 1, \forall o_{s.} \in \mathcal{SP}_j$
        ▷ reset the bits of $\hat{\mathbf{x}}_{ij}^{'}$ corresponding to items of $\mathcal{SP}_j$ as "1".
        else
            $\hat{\mathbf{x}}_{ij}^{'} = \mathbf{x}_i$
        end if
    end for
    $\hat{\mathbf{x}}_i^{'} = \mathrm{mean}(\hat{\mathbf{x}}_{ij}^{'})$
        ▷ mean takes the average over all j
end for
$F = F(\hat{D}^{'})$
        ▷ plug $\hat{D}^{'} = \{\hat{\mathbf{x}}_i^{'}\}_{i=1}^{n}$ into Eq. (4)
$[\Phi^*, D^{'}]$ = optimize(F)
        ▷ maximize Fisher score function w.r.t Φ

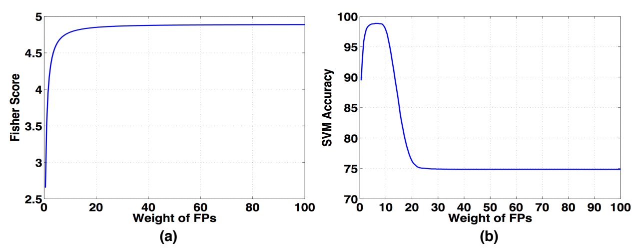


Figure 2: (a) Relationship between common SSI weight and the Fisher score. (b) Relationship between common SSI weight and the final classification accuracy.

$\Phi = [\phi_1, \phi_2, \cdots, \phi_k]$. As the weights of the SSPs, the parameters $[\phi_1, \phi_2, \cdots, \phi_k]$ determine how and what extent of the "forces" from SSPs will act on the input data points. Under the meaning of Fisher score maximization, the optimal Φ^* will lead to the best classification performance. To search the Φ^*, it's intractable to use gradient based optimal algorithm because the highly non-linear feature transform operation makes it hard to get the analytical formation of the objective function with regards to Φ. However, the interesting property of the objective function allows us to find an efficient two-phases optimal algorithm for this problem.

To check the property of the objective function which is closely related to the action classification problem, we conduct the analysis on the human pose data. We first set the weights of all the SSPs as identical[1]. By doing this, the multiple dimensional optimization turns into one-dimensional optimization, which makes it much easier to check the property of the objective function. The relationship between the common weight ϕ and Fisher score can be found in Figure 2(a). As can be seen, the functional relationship is unimodal and the objective function is convex. So conventional one dimensional optimization algorithm is sufficient to find the optimum. In this work, we use golden section search [25].

From Figure 2(b), however, it can be seen that after reaching at the top point of the objective function, in the flat domain, an abrupt decrease in classification accuracy will happen if the value of the common weight keeps increasing. Due to the monotonous increasing of the common weight, the feature vector is changed continuously; therefore it has difficulty converging to the best classification accuracy. For this reason we switch to the second phase, where the multiple constraints on the different weights can form a well-shaped convex objective function.

[1]The weights are initialized randomly within the interval [1, 10] in our problem.

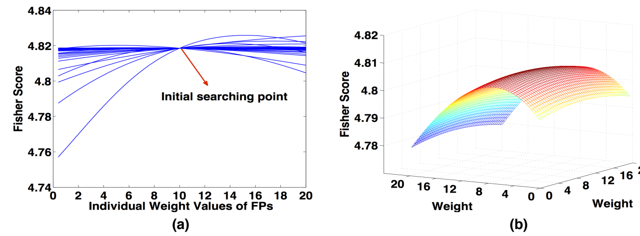


Figure 3: (a) F-W curves formed by fixing the weights of other SSPs at the common optimal weight and changing the weight of one SSP. (b) 3D visualization of the objective function.

The aim of the second phase optimization is to find the optimal individual weights for each SSP, starting from the initial searching point obtained in the first phase optimization. To this end, we analyze the property of the objective function with respect to the individual weights. In Figure 3(a), we plot the curves of Fisher score about the individual SSP weights. We call the curves as F-W curve, which is formed by fixing the weights of other SSPs at the common optimal value and changing the weight of a SSP. It's obvious that the curves are parabola-like and convex. Furthermore, for a more clear and intuitive visualization, we plot the 3D shape of the objective function in Figure 3(b). As we can see, the function is also very smooth. In view of the convexity property of the objective function, we employ the downhill simplex method [25] as the searching strategy. Benefitting from the convexity of the objective function and the given starting guess obtained from the first phase, the optimal searching converges quickly.

The two-phase optimization process is described in Algorithm 3.

Algorithm 3 Two-Phase SSP Weights Optimization Algorithm

Input: Training dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, $SSP = \{\mathcal{SP}_i\}_{i=1}^{k}$
Output: Optimized weights $\Phi^* = [\phi_1^*, \phi_2^*, \cdots, \phi_k^*]$
$\Phi = [1, 1, \cdots, 1] \cdot \phi$
        ▷ set the weights of SSPs as common weight
$\hat{D}^{'}$ = FeatureTrans(D, SSP, Φ)
        ▷ transformed dataset parameterized by ϕ by Algorithm 2
$F = F(\hat{D}^{'})$
$\phi^* = \mathrm{SectionSearch}(F)$
        ▷ Phase 1: golden section search for one dimensional optimization
$SimplexPoint = \mathrm{GenerateSimplex}(\phi^*, para)$
        ▷ generate the initial guess for simplex search, $para$ is the related parameters
repeat
    $\hat{D}^{'}$ = FeatureTrans(D, SSP, $SimplexPoint$)
    $F = F(\hat{D}^{'})$
    $SimplexPoint = \mathrm{SimplexSearch}(F)$
        ▷ Phase 2: simplex search for multiple dimensional optimization
until <converge> **return** $\Phi^* = SimplexPoint$

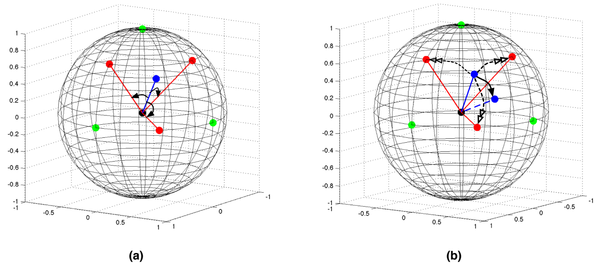


Figure 4: (a) Full pattern "111" (blue) and several sub-patterns before feature transform. (b) "111" pattern transformed (dashed blue pattern) to another place under the forces from the sub-patterns.

4.2.3 Geometric Interpretation

The proposed feature transform approach can be intuitively explained from the geometric viewpoint, as can be seen in Figure 4. For visualization, we take three-dimensional data as example.

Suppose we have a binary pattern "111" (the blue point in Figure 4(a)) and several sub-patterns "110", "101", "011" (red points) and "001", "010", "100" (green points). For simplicity, we just consider the impacts of the sub-patterns denoted by the red color. All the patterns can be easily normalized to reside on the surface of a unit 3D sphere. Then, the feature transform is actually a redistribution of the data points along the surface of the 3D sphere. One can imagine the sub-patterns as fixed magnetic centers scattered on the surface of the sphere and the data points can move freely along the surface. Since the pattern "111" contains all the sub-patterns, all the sub-patterns will have "magnetic attractive force" on it. If the force of a sub-pattern is powerful enough for pattern "111", the angle between them will be reduced and the "111" pattern will be attracted to approaching the sub-pattern. In Fig. 4, we can see that finally the pattern "111" is transformed to another point under the composition of forces from all the three sub-patterns.

For a practical problem, such as pose based action classification, the sub-patterns actually contain rich information about action classes because they frequently appear in the specific action classes. So when a sub-pattern appears in a data point, it gives partial evidence that this point contains part of the "gene" for that action class. It makes sense that we move a step further to decrease the distance between the pattern and the sub-pattern by a feature transform. Statistically speaking, this kind of "approaching" will result in the improvement of that feature's discriminative power. The geometric viewpoint is helpful to deepen the understanding of the SSPI approach theoretically and to construct a more general feature learning model. This is a direction of our future work.

5. EXPERIMENTS

In this section, we report our empirical study of the proposed approach for action classification and sub-pattern analysis of pose data. By conducting the experimental evaluation, we hope to get the answers to two main concerns: (1) Starting from the SSP of pose data, what is the performance of the proposed feature transform framework for action classification? (2) In the process of feature transform, what kind of sub-patterns are extracted and how about their discriminative power to action classification? By conducting experimental evaluations on a large scale human pose dataset, for the first concern, we examine the effectiveness of the SSPI feature transform approach and make comparison with the state-of-the-art approaches. For the second concern, we visually check the SSPs for all the action classes and evaluate their discriminative power quantitatively under the feature transform framework.

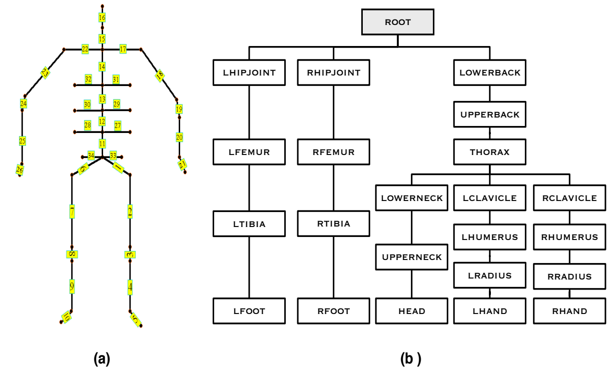


Figure 5: (a) Human body model with 34 bones. (b) Defined hierarchical structure of human body.

5.1 Experimental Setup: Data and Dataset

Human Pose Data. In this work, we work on human pose data. The human body pose is represented in the Acclaim format (ASF, AMC). With this format, human body is defined by a full body model which is consisted of 34 bones with hierarchical structures. The pose is specified by the motion angles of body bones, which describe the dynamic relationships between parent and children bones, as well as the global motion of the full body. See Figure 5 for the visualized body model and the definition of hierarchical structure. The original full body Degree of Freedoms (DOFs) are 62. However, to reduce the computational burden, we discard some unimportant joint angles, such as fingers, thumb, toes and so forth in the experiments. Finally, we select 19 body angles which cover the DoFs of humerus (left and right), radius (left and right), femur (left and right), tibia (left and right) and the upper back.

In this work, our approach is built on the basis of sub-pattern analysis, therefore the continuous pose data first needs to be quantized. To this end, we constrain the value of the motion angles into the interval $[-180^\circ, 180^\circ]$ and then perform 10 levels quantization to each angle within their respective ranges. By doing this, each pose data point is turned into a 190-bit binary pattern, which is used as the basic pose feature in the following experiments.

Experimental Setup of The Dataset. To conduct comprehensive evaluations, we perform experiments on the motion capture dataset collected at CMU [10], which is very large and includes dozens of actions performed by over one hundred subjects. From the whole database, we choose the MoCa data of 9 action classes performed by diverse number of subjects respectively. The 9 action classes include walk, run, jump, pick-up, sitting on a motorcycle, cartwheel, boxing, chicken dance and golf swing. Because the total frame number of the different action classes are different in the original dataset, different percentages are adopted when taking the data frames from the action classes. In doing so, each class will roughly have the same number of data frames. This avoids the class imbalance problem. In our experimental setup, there are in total of 169975 frames for the 9 classes, from which we randomly select 52482 frames for each experiment. Some portion of the 52482 data frames are used for training and the remainder for testing. In most of our experiments we divided this set in half.

Table 1: **Accuracy (%) of action classification on different pose features. See texts for the abbreviations.**

Action	**Pose Features**							
	RQ	BQ4	BQ10	PCA(BQ10)	LDA(BQ10)	BQFP[8, 9]	BQFP-S[8]	**SSPI**
Walk	84.00	92.03	89.27	84.09	82.92	91.02	<u>91.14</u>	**97.39**
Run	77.66	88.92	89.43	70.32	79.10	88.68	<u>93.96</u>	**99.00**
Jump	87.05	95.02	92.93	84.53	85.30	93.36	96.03	**99.02**
Pick-up	73.47	87.80	87.89	82.19	76.15	87.64	<u>93.08</u>	**98.12**
Motorcycle	99.40	99.45	99.32	99.75	98.42	99.70	99.45	**100.0**
Cartwheel	77.54	93.08	92.72	74.66	79.93	92.76	<u>94.24</u>	**99.30**
Boxing	86.08	95.13	94.73	98.18	90.62	96.16	95.96	**99.37**
Chicken-dance	91.21	95.64	96.47	84.51	90.00	96.09	96.58	**99.69**
Golf-swing	99.92	**100.0**	99.92	90.19	99.35	99.92	**100.0**	**100.0**
Average	86.26	94.12	93.63	85.38	86.87	93.92	95.60	**99.10**

5.2 Classification Performance

In this part, working on the MoCa data, we evaluate the effectiveness of the proposed feature transform algorithm for action classification. By comparing with several different methods, the performance of our approach is checked from different perspectives, which are important to have a comprehensive understanding to the SSPI feature transform framework.

Experimental Setup of the Classifier. In this work, SVM is used as the classifier. The LIBSVM [7] is chosen as the software implementation of SVM and we use RBF as the kernel of SVM. For each prediction model, before model training, a grid search is conducted to find the optimum of the parameters C, γ of RBF using 5-folds cross-validation. The model is trained also using a 5-folds cross-validation on the training dataset using the optimal C, γ. In all the classification experiments, the final accuracy is reported by averaging on 10 executions of the SVM prediction procedure, using training and test subsets randomly taken from the full dataset. One advantage of doing this is to avoid over-fitting.

Selected Approaches for Comparison. To evaluate the performance of our approach, we compare with seven methods. Among them, three coding strategies, namely, real number coding, 4-bit coding and 10-bit coding are evaluated. To work on pattern based classification, quantization of the original continuous data is required. Real-number Quantization (RQ) maps each feature dimensionality to a discrete real value which is obtained by 10-level quantization of the original single features. Using binary coding scheme, 4 bits (BQ4) and 10 bits (BQ10) coding map each feature dimensionality to a 4-bit and 10-bit binary code respectively, so the original feature vector is turned into either a 76-bit or 190-bit binary string.

We use the methods proposed in [8], which we call BQFP and BQFP-S, as the key comparison for validating our proposed SSPI feature transform algorithm. There are three reasons for selecting these algorithms: (1) Both SSPI and BQFPs work on Frequent Pattern (FP) based classification. The BQFP and its upgraded version, BDFP-S, which adds an additional step for feature selection, are the state-or-the-art techniques. (2) BQFPs use non-liner feature combination scheme, which maps the feature to a higher dimensional space. In doing so, the performance gets improved but computational burden for classifier is increased. However, the SSPI does not change the dimensionality of the features. (3) BQFP-S explicitly chooses its features, while SSPI does so implicitly using SSP weight learning. We therefore want to compare the differences mentioned in (2) and (3) quantitatively by experimental evaluations.

Furthermore, because SSPI keeps feature's dimensionality invariant whereas BQFPs increase it, it will be interesting to see if reducing the dimensionality results in better performance. So, the conventional PCA and LDA are used for this purpose. We select LDA because it is similar to SSPI, in that it also uses Fisher score as the internal criteria for dimensionality reduction. Notice that both PCA and LDA work on BQ10 coding scheme.

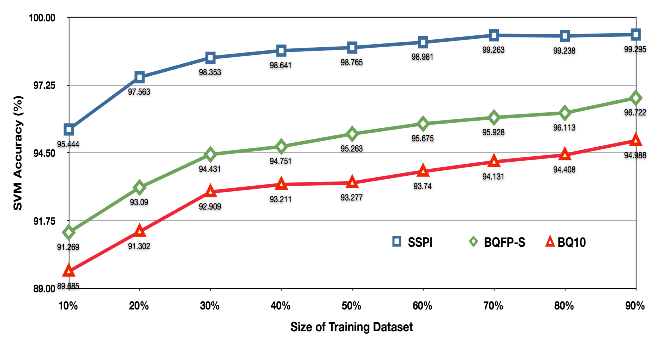


Figure 6: Classification accuracy as a function of the size of training dataset. The SSPI, BQFP-S and BQ10 are compared.

Results and Analysis. In Table 1, we report the classification accuracy of the pose features generated by different methods. For the three coding strategies, it can be seen that the performance of RQ is significantly worse than BQ4 and BQ10. This is due to its simple quantization scheme performed in real domain. BQ4 and BQ10 have close performance because they use similar binary coding scheme. However, in our problem, we prefer to use BQ10 as the basic quantization method, because each bit in BQ10 has a very clear physical meaning and the obtained SSPs are specified by the local human poses and can be easily visualized. It satisfies our requirements of "physicality" and "visibility" to pose data.

Notice that the performances of BQFP and BQ10 are very similar for all the action classes. This result is consistent with [8]. It confirms the hypothesis that for frequent pattern based classification, without feature selection, the performance of feature combination schemes can be very poor. As can be seen, after adding feature selection module to the BQFP algorithm, BQFP-S achieves much better performance. It's worth pointing out that for the pose data,

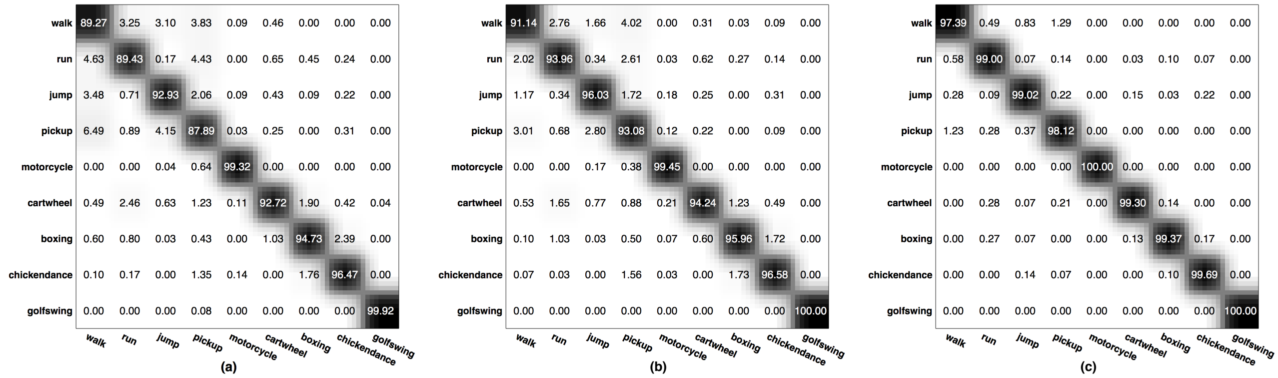


Figure 7: Confusion matrices of (a) BQ10, (b) BQFP-S and (c) SSPI for action classification, with mean classification accuracy of 93.63%, 95.6% and 99.10% respectively. Notice that many confusions are removed or relieved in the SSPI feature transform framework, especially for the action class "walk", "run", "pick-up" and "cartwheel".

even a small increase in classification accuracy is important and not easy to achieve. So the superior performance of BQFP-S over BQFP is significant. Compared with BQFPs and BQ10, the performance of PCA and LDA is discouraged. It may be caused by the information loss combined with the dimensionality reduction.

It's encouraging that SSPI significantly outperforms all the other methods in classification accuracy, for almost all the action classes except the identical 100% performance with BQ4 and BQFP-S for the action class"golf-swing". Overall, SSPI outperforms the state-of-the-art technique, BQFP-S, by 4% in classification accuracy, which is quite remarkable for pose pattern based classification. It's also worth noting that for the most confused action classes "walk", "run", "pick-up" and "cartwheel", SSPI achieves significant performance improvements on SVM accuracy compared to all the other methods (5-7% and 8-11% improvement compared to BQFP-S and BQ10 respectively), as indicated in Table 1. The four actions are prone to confuse because many "walk-like" frames are contained in these actions. From Figure 7, it can be seen that many confusions are removed or relieved in the SSPI feature transform framework, especially for the four confusing actions. The potential cause for resulting in such kind of improvement of SSPI is that under the "force" of SSPs, the bits corresponding to the "walk-like" attribute are changed so that the confusion can be mitigated accordingly.

For a more comprehensive evaluation to SSPI, we also investigate the robustness of SSPI algorithm to the sizes of training and test datasets. The results are shown in Figure 6. For clearness, we just select the BQFP-S and BQ10 for comparisons[2]. It can be seen that with the increase of training dataset size (indicated by the ratio of training data), the performance of all the three methods is improved. It also can be observed that SSPI obviously outperforms the other two methods during the change of data size.

From the experimental results, we can draw several observations. First of all, the proposed SSPI feature transform framework achieves remarkable overall better performance than the state-of-the-art techniques. Its effectiveness is demonstrated. Second, the SSPI feature transform mechanism is beneficial to remove the confusion appeared between the classes with overlapped attributes. Finally, the performance of SSPI is robust and can be applied to the datasets with different scales.

In terms of the computational expense, the mining process is quite efficient and it takes less than half a second for mining 300 SSPs from nearly $30,000$ training samples. The most time consuming part is on the weights optimization. It takes 10 minutes or so to complete the optimum searching on the training data, using a 3.1G core i5 machine with 4G memory.

5.3 Evaluation On Support Sub-Patterns

SSPI feature transform is demonstrated quite effective to action classification. The functionality of the SSP, however, is not just constrained to classification. Thanks to the good properties of human pose data, we can visualize the SSPs and intuitively build the correspondence between the SSP represented local pose and its discriminative power, which can be computed explicitly within the framework of feature transform. By doing this, the utilization of SSP can be extended to much wider scenarios, such as action search, pose estimation, event understanding and so forth. Therefore, in this part, we conduct comprehensive evaluation to the SSP.

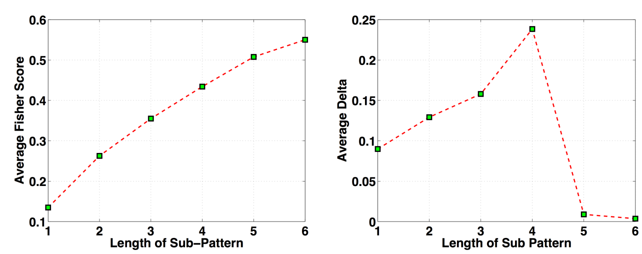


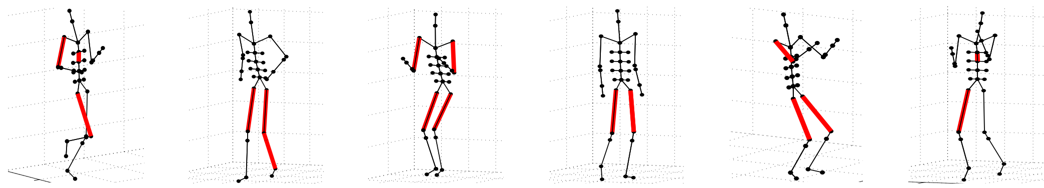
Figure 8: (a) Fisher score as a function of the pattern length. The Fisher score is averaged over all the SSPs with the same length. (b) Average discriminative power of the SSPs is compared, with regards to the pattern length.

SSP Mining. To generate SSPs, frequent pattern mining algorithm is an effective solution. In this work, we use Apriori algorithm [2, 18] to mine the frequent items from quantized human pose data, namely, the BQ10 binary string introduced in Section 5.2. There is only one parameter which needs to be specified in the Apriori algorithm, say, the minimum support threshold (min_sup) defined in Definition 1. We apply the same min_sup to all the action classes and empirically set it as 0.55 in this work.

In our experiments, for each action class, about dozens of the SSPs are generated and the length of the SSP ranges from 1 to 6. After merging the same SSP from different classes, near 300 SSPs are employed for feature transform. The same number also comes

[2]There is a slight difference between the numbers in Table 1 and Fig. 6 (50% case), as we select data randomly for each execution.

Table 2: **Top six SSPs ranked by their discriminative power and listed in descending order from left to right.**

SSP_i						
$\frac{\partial F}{\partial \phi_i}$	1.7185	0.8406	0.5798	0.4758	0.3918	0.2809

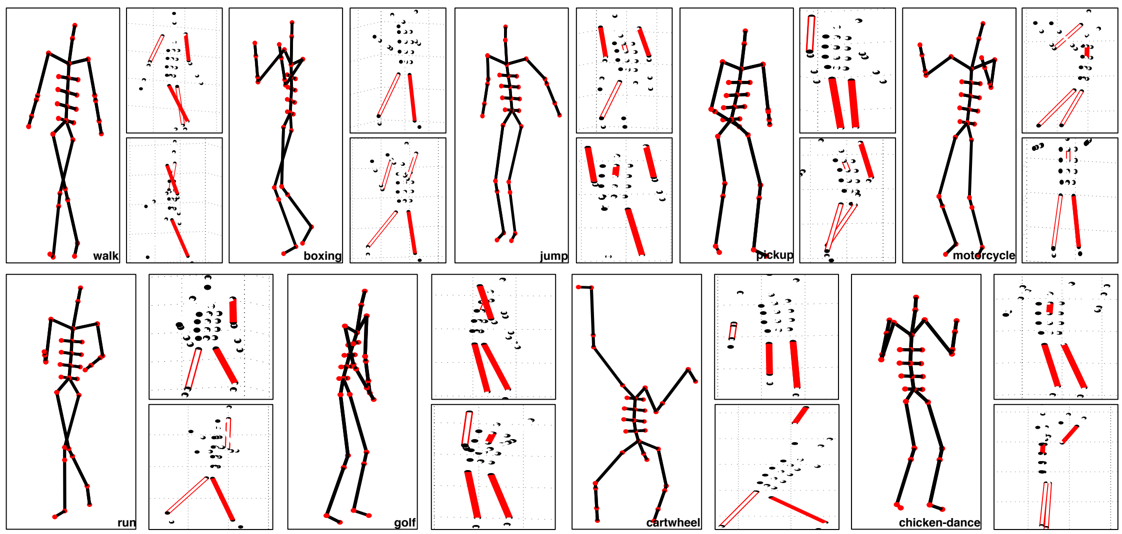


Figure 9: Sample poses and their corresponding two SSPs. Both pose and SSP are randomly selected from the 9 action classes. The SSPs are generated for each action class respectively. The actions are walk, boxing, jump, pick-up, motorcycle pose, run, golf-swing, cartwheel and chicken dance from left to right and top to bottom.

up with the dimensionality of the weight vector Φ. In Figure 9, we show the pose samples randomly selected from the 9 action classes and two corresponding SSPs in their right sides. As can be seen, the visualization of the SSP presents us an intuitive observation that the SSP is the representative local pose and can be viewed as the "signature" of the corresponding action class.

Analysis on SSP's Discriminative Power. SSP plays important role in feature transform. However, in this process, the importance of different SSPs is unequal. One critical advantage of the SSPI feature transform framework is that the "importance", namely the discriminative power of the SSPs can be quantitatively evaluated. To this end, we define the Discriminative Power (DP) of the SSP as: $\frac{\partial F}{\partial \phi_i}$, where F is the Fisher score function and ϕ_i is the weight parameter associated with the i-th SSP.

We can see that the DP actually measures the sensitivity of the Fisher score function to the change of ϕ corresponding to the SSP. In our implementation, we compute the $\frac{\partial F}{\partial \phi_i}$ in this way: when calculating the DP of the i-th SSP, we first fix the weights of other SSPs at their optimal values and then change the values of ϕ_i within a scope, which is taken as $[0, 20]$ in our experiments, to measure the change of Fisher score function ΔF. Because the objective function is convex according to the analysis in Section 4.2.2, ΔF is taken as the difference between peak value and valley value of the objective function within the scope.

As a SSP actually corresponds to a feature subset, it can be evaluated with Fisher score based feature selection method [17]. To evaluate the DP of the SSP, we first check the Fisher score of the SSP by taking the SSP as the selected feature subset. The average Fisher score with regards to different pattern length is compared in Figure 8(a). As can be seen, with the increase of pattern length, the Fisher score is monotonously increasing. It's reasonable because long patterns contain more information than the shorter ones when they are used as feature subset. However, when SSP is adapted into the SSPI feature transform framework, it is no longer treated as feature subset and it plays more like a "force" to induce the change of feature. Therefore the pattern length is not the dominant factor to determine SSP's DP. In Figure 8(b), the DP of SSPs with different pattern lengths are compared. The DP is computed by averaging on all the SSPs with the same pattern length. It can be found that the SSP with length 4 become the most discriminative sub-patterns and the length 5, 6 degrade to the worst comparing with Figure 8(a).

We also rank the SSPs according to their DPs. The result is shown in Table 2, where the figures of the top 6 SSPs and their corresponding DPs are presented. Although the SSP is plotted just on the pose of one class, actually they may appear in multiple action classes. Statistically speaking, the higher the rank of a SSP, the greater the contribution it makes for action classification. Therefore, essentially under the framework of SSPI, the SSPs are selected by an implicit way according to their DPs, which is different from the explicit manner for feature selection presented in [8].

Potentially, the results of quantitative evaluation on SSP can be used for action search, action retrieval and pose estimation. For example, the DP of a SSP can be viewed as a confidence of classifying a pose to a specific action class represented by the SSPs.

6. CONCLUSION

In this paper, we propose a new feature transform framework for pose based action classification. Utilizing the properties of pose data, we introduce frequent pattern mining to explore the Support Sub-Patterns (SSP), which are representative sub-patterns of the action class and can be applied for effective action classification. Based on the sub-patterns, the original pose feature can be converted into a more discriminative formation for classification, by SSP induced feature learning. In this framework, we construct the objective function under the meaning of Fisher score optimum. A two-phases efficient optimal algorithm for searching the optimal weights of SSPs is developed. By applying the learned optimal weights on feature transform, the discriminative feature for classification is obtained. Experimental studies demonstrate the superior performance of the proposed approach over the state-of-the-art techniques on the task of action classification. Furthermore, the discriminative power of the SSP can be explicitly evaluated within the feature transform framework. It could derive many interesting applications within the field of human motion analysis.

7. REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 2011.

[2] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *VLDB*, 1994.

[3] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *ICCV*, 2007.

[4] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

[5] A. Bissacco, M. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *CVPR*, 2007.

[6] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *ICCV*, 1995.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[8] H. Cheng, X. Yan, J. Han, and C. Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, 2007.

[9] H. Cheng, X. Yan, J. Han, and P. Yu. Direct discriminative pattern mining for effective classification. In *ICDE*, 2008.

[10] CMU. http://mocap.cs.cmu.edu.

[11] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. Andriacchi. Markerless motion capture through visual hull, articulated icp and subject specific model generation. *International Journal of Computer Vision*, 87(1):156–169, 2010.

[12] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*, 2012.

[13] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. Yu, and O. Verscheure. Direct mining of discriminative and essential frequent patterns via model-based search tree. In *KDD*, 2008.

[14] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

[15] D. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[16] D. Gavrila, L. Davis, et al. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International workshop on automatic face and gesture recognition*, 1995.

[17] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *UAI*, 2011.

[18] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[19] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[20] G. Johansson. Visual perception of biological motion and a model for its analysis. *Attention, Perception, & Psychophysics*, 14(2):201–211, 1973.

[21] H. Kim, S. Kim, T. Weninger, J. Han, and T. Abdelzaher. Ndpmine: efficiently mining discriminative numerical features for pattern-based classification. *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, 2010.

[22] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011.

[23] H. Ning, T. Han, D. Walther, M. Liu, and T. Huang. Hierarchical space-time model enabling efficient search for human actions. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(6):808–820, 2009.

[24] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010.

[25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

[26] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[27] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[28] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. In *ICCV*, 2005.

[29] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[30] J. M. Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8(5):371–391, 1994.

[31] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. In *ICCV*, 1998.

[32] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.

[33] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[34] B. Yao and L. Fei-Fei. Action recognition with exemplar based 2.5d graph matching. In *ECCV*, 2012.

[35] A. Yilma and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *ICCV*, 2005.

[36] J. Yuan, Y. Wu, and M. Yang. From frequent itemsets to semantically meaningful visual patterns. In *SIGKDD*, 2007.