Action MACH
A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition

Mikel D. Rodriguez*, Javed Ahmed†, Mubarak Shah*
*Computer Vision Lab, University of Central Florida. Orlando, FL.
†Military College of Signals. Rawalpindi, Pakistan.

Abstract

In this paper we introduce a template-based method for recognizing human actions called Action MACH. Our approach is based on a Maximum Average Correlation Height (MACH) filter. A common limitation of template-based methods is their inability to generate a single template using a collection of examples. MACH is capable of capturing intra-class variability by synthesizing a single Action MACH filter for a given action class. We generalize the traditional MACH filter to video (3D spatiotemporal volume), and vector valued data. By analyzing the response of the filter in the frequency domain, we avoid the high computational cost commonly incurred in template-based approaches. Vector valued data is analyzed using the Clifford Fourier transform, a generalization of the Fourier transform intended for both scalar and vector-valued data. Finally, we perform an extensive set of experiments and compare our method with some of the most recent approaches in the field by using publicly available datasets, and two new annotated human action datasets which include actions performed in classic feature films and sports broadcast television.

Figure 1. Our framework is capable of recognizing a wide range of human actions under different conditions. Depicted on the left are a set of publicly available datasets which include dancing, sport activities, and typical human actions such as walking, jumping, and running. Depicted on the right column are examples of two action classes (kissing and slapping) from a series of feature films.

1. Introduction

Action recognition constitutes one of the most challenging problems in computer vision, yet effective solutions capable of recognizing motion patterns in uncontrolled environments could lend themselves to a host of important application domains, such as video indexing, surveillance, human-computer interface design, analysis of sports videos, and the development of intelligent environments.

Temporal template matching emerged as an early solution to the problem of action recognition, and a gamut of approaches which fall under this general denomination has been proposed over the years. Early advocates for approaches based on temporal matching, such as Polana and Nelson [16], developed methods for recognizing human motions by obtaining spatio-temporal templates of motion and periodicity features from a set of optical flow frames. These templates were then used to match the test samples with the reference motion templates of known activities. Essa and Pentland [8] generated spatio-temporal templates based on optical flow energy functions to recognize facial action units. Bobick et al [4] computed Hu moments of motion energy images and motion-history images to create action templates based on a set of training examples which were represented by the mean and covariance matrix of the moments. Recognition was performed using the Mahalanobis distance between the moment description of the input and each of the known actions.

Efros et al. [7] proposed an approach to recognizing human actions at low resolutions which consisted of a motion descriptor based on smoothed and aggregated optical flow

measurements over a spatio-temporal volume centered on a moving figure. This spatial arrangement of blurred channels of optical flow vectors is treated as a template to be matched via a spatio-temporal cross correlation against a database of labeled example actions.

In order to avoid explicit computation of optical flow, a number of template-based methods attempt to capture the underlying motion similarity amongst instances of a given action class in a non-explicit manner. Shechtman and Irani [18] avoid explicit flow computations by employing a rank-based constraint directly on the intensity information of spatio-temporal cuboids to enforce consistency between a template and a target. Given one example of an action, spatio-temporal patches are correlated against a testing video sequence. Detections are considered to be those locations in space-time which produce the most motion-consistent alignments.

Given a collection of labeled action sequences, a disadvantage of these methods is their inability to generalize from a collection of examples and create a *single* template which captures the intra-class variability of an action. Effective solutions need to be able to capture the variability associated with different execution rates and the anthropometric characteristics associated with individual actors. Recent popular methods which employ machine learning techniques such as SVMs and AdaBoost, provide one possibility for incorporating the information contained in a set of training examples.

In this paper, we introduce the Action MACH filter, a template-based method for action recognition which is capable of capturing intra-class variability by synthesizing a single Action MACH filter for a given action class. We generalize the traditional MACH filter to video (3D spatiotemporal volume), and vector value data. By analyzing the response of the filter in the frequency domain, we avoid the high computational cost commonly incurred in template-based approaches, thereby reducing detection to a matter of seconds. Vector-valued data is analyzed using the Clifford Fourier transform, which is a generalization of the traditional scalar-valued Fourier transform. In order to asses the effectiveness of the proposed approach we perform an extensive set of experiments on both publicly available datasets such as the KTH action dataset, the Weizmann action dataset, the Cohn-Kanade facial expression database, and on two new homegrown datasets. These include a collection of sports-related actions as featured on broadcast television channels, and a pool of actions found in feature films.

The organization of the paper is as follows. In the next section we introduce the Action MACH filter. In Section 2.2, we generalize our approach to include vector-valued fields in addition to scalar fields. In section 2.3 we employ spatio-temporal regularity flow as a means for training the MACH filter on 3D vector fields. Finally, in section 3 we describe the experiments that we performed on publicly available datasets and a homegrown collection of actions found in films and television sports shows.

2. The action MACH filter

Traditionally, MACH filters have been employed in object classification, palm print identification [9], and aided target recognition problems [19]. Given a series of instances of a class, a MACH filter combines the training images into a single composite template by optimizing four performance metrics: the Average Correlation Height (ACH), the Average Correlation Energy (ACE), the Average Similarity Measure (ASM), and the Output Noise Variance (ONV). This procedure results in a two dimensional template that may express the general shape or appearance of an object. Templates are then correlated with testing sequences in the frequency domain via a FFT transform, resulting in a surface in which the highest peak corresponds to the most likely location of the object in the frame.

The notion of a traditional MACH filter could be generalized to encompass human actions in a number of ways. A fairly straightforward approach would be to recognize an action class by a succession of two dimensional MACH filters at each frame. However, in order to fully leverage the information contained in a video sequence, the approach we propose in this work consists of generalizing the MACH filter by synthesizing a template estimated from the spatio-temporal volumes of action sequences. Such filters could be synthesized using raw pixel values, edges, temporal derivative, or optical-flow in the spatiotemporal volume. When each pixel in this volume contains multiple values it is not possible to synthesize a MACH filter using traditional Fourier transform. Solutions to this problem could include employing motion magnitude or direction alone (scalar values), instead of complete vector data. In order to deal with this problem, we propose to employ the Clifford transform, which is a generalization of the standard Fourier transform for vector valued functions.

2.1. Action MACH Filter for Scalar Data

In this subsection we describe the process of synthesizing an Action MACH filter to to recognize various actions based on scalar data. A typical example of a set of actions which we attempt to recognize is depicted in Figure 1. These consist of a set publicly available datasets, as well as a collection of actions featured on sports networks and in feature films.

We begin the process of training the Action MACH filter with the creation of a series of spatio-temporal volumes from the testing action sequences by concatenating the frames of a *single* complete cycle of an action. Subse-

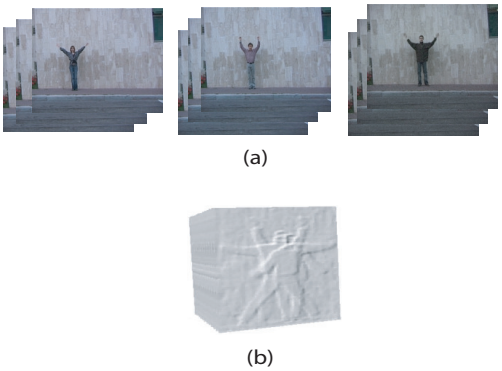

Figure 2. The 3D Action MACH filter (b) synthesized for the "Jumping Jacks" action (a) in the Weizmann action dataset using temporal derivatives.

quently, we compute the temporal derivative of each pixel resulting in a volume for each training sequence. Following the construction of the spatio-temporal volumes for each action in the training set, we proceed to represent each volume in the frequency domain by performing a 3-D FFT operation, which is given by:

$$F(u,v,w) = \sum_{t=0}^{N-1}\sum_{y=0}^{M-1}\sum_{x=0}^{L-1} f(x,y,t) \exp(-j2\pi(\frac{uv}{L}+\frac{vy}{M}+\frac{wt}{N})), \quad (1)$$

where $f(x,y,t)$ is the volume corresponding to the temporal derivative of the input sequence, and $F(u,v,w)$ is the resulting volume in the frequency-domain. L is the number of columns, M the number of rows, and N the number of frames in the example of the action. In an effort to increase the efficiency of this step, we exploit the separability property of the Fourier transform, and compute the multi-dimensional transform by performing one-dimensional transforms in x (horizontal axis), y (vertical axis), and finally t (time axis). Having obtained the resulting volumes in the frequency-domain we proceed to convert the resulting 3-D FFT matrix into a column vector by concatenating all the columns of the 3-D matrix. Let the resulting single column-vector be denoted by x_i (of dimension $d = L*M*N$), where $i = 0, 1, 2..., N_e$, where N_e is the total number of examples of the action in the training dataset. Once the column vectors are obtained for all the examples of the action, the Action MACH filter (which minimizes average correlation energy, average similarity measure, output noise variance; and maximizes average correlation height) can be synthesized in the frequency domain, as follows [19]:

$$h = (\alpha C + \beta D_x + \gamma S_x)^{-1} m_x, \quad (2)$$

where m_x is the mean of all the x_i vectors, and h is the filter in vector form in the frequency domain. C is the diagonal noise covariance matrix of size $d \times d$, where d is the total number of elements in x_i vector. If the noise model is not available, we can set $C = \sigma^2 I$, where σ is the standard deviation parameter and I is a $d \times d$ identity matrix. D_x is also a $d \times d$ diagonal matrix representing the average power spectral density of the training videos and is defined as:

$$D_x = \frac{1}{N_e}\sum_{i=1}^{N_e} X_i^* X_i, \quad (3)$$

where X_i is a $d \times d$ diagonal matrix in which the diagonal elements are the same as the elements of the x_i vector, and $*$ represents the conjugate operation. S_x is the diagonal average similarity matrix defined as:

$$S_x = \frac{1}{N_e}\sum_{i=1}^{N_e} (X_i - M_x)^*(X_i - M_x), \quad (4)$$

where M_x is a diagonal matrix whose elements are the same as those in m_x. Finally, α, β, and γ are the parameters that can be set to obtain the trade-off among the performance measures.

After designing the 1-D filter h, we assemble a complete filter by applying the reverse of the operation that was used to convert the 3D volume of the action example into a column vector. Subsequently, we perform the 3D inverse Fourier transform. The resulting matrix constitutes the Action MACH filter, H, for the particular action (Figure 2).

2.1.1 Action Classification

Once an Action MACH filter has been synthesized, we can proceed to detect similar actions in a testing video sequence by applying the action MACH filter H to the video:

$$c(l,m,n) = \sum_{t=0}^{N-1}\sum_{y=0}^{M-1}\sum_{x=0}^{L-1} s(l+x, m+y, n+t) H(x,y,t), \quad (5)$$

where s is the spatio-temporal volume of the test video, H is the spatio-temporal MACH Filter (h is its Fourier transform). P, Q, and R are the dimensions of the of the spatio-temporal volumes.

As a result of this operation, we obtain a response, c, of size $(P-L+1)\times(Q-M+1)\times(R-N+1)$ (Figure 3). We denote this location by (l^*, m^*, n^*). Due to varying illumination conditions and noise in the scene, we optimize the response of the filter by normalizing our correlation space:

$$c'(l,m,n) = \frac{c(l,m,n)}{\sqrt{E_H E_S(l,m,n)}}, \quad (6)$$

where $c(l,m,n)$ is given by equation 5. E_H is a scalar value which represents the energy of the filter, and $E_s(l,m,n)$

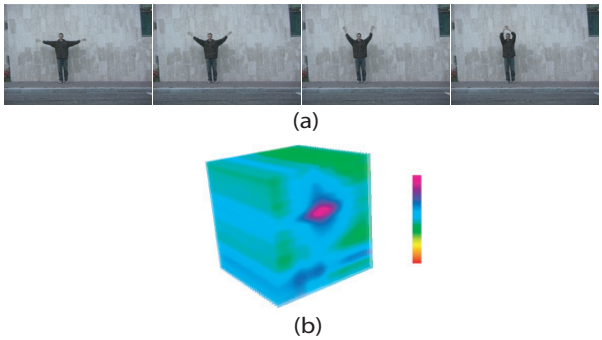

Figure 3. (a) Frames from a testing sequence containing the "wave2" action from the Weizmann action dataset. (b) The normalized correlation response for the testing sequence depicted in (a) correlated against the "Wave2" Action MACH filter depicted in Figure 2.

corresponds to the energy of the test volume at location (l, m, n), given by:

$$E_H = \sum_{t=0}^{N-1} \sum_{y=0}^{M-1} \sum_{x=0}^{L-1} H^2(x, y, t), \tag{7}$$

$$E_S(l, m, n) = \sum_{t=0}^{N-1} \sum_{y=0}^{M-1} \sum_{x=0}^{L-1} s^2(l + x, m + y, n + t). \tag{8}$$

Each element in the response of the normalized correlation lies within 0 and 1, a fact that can be used as a level of confidence in a pseudo-probabilistic manner. The peak value in the response of the filter is compared with a threshold (τ). If it is greater than the threshold, we infer that the action is occurring at the corresponding location in the test video. Thresholds for action classes are computed during training as $\tau = \xi * min(p_1, p_2, p_3, ..., p_{N_e})$, where p_i is the peak value obtained from the correlation response when ith training volume was correlated with the 3D MACH filter, ξ is a constant parameter, and N_e is the number of all the training volumes.

2.2. Action MACH Filter for Vector Fields

In the previous section we described the process of synthesizing an Action MACH filter based on scalar data. In this section we extend our approach to include vector data.

2.2.1 Spatiotemporal Regularity Flow

The estimation of motion in video sequences constitutes an integral task in numerous applications, including human action recognition. The ability to estimate motion accurately and consistently has numerous challenges associated with it, such as motion discontinuities, aperture problems, and large illumination variations. Several of these challenges lead to direct violations of the assumptions embedded in the formulation of the classical flow estimation methods, such as Horn and Schunck's optical flow [10]. Therefore, numerous flow estimation approaches have been proposed which deal with large motion discontinuities [13], complex illumination variation, iconic changes [2], and three-dimensional scene flow [20].

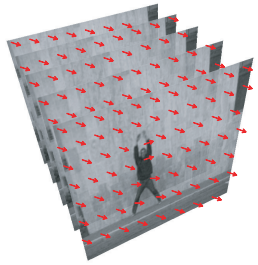

Figure 4. The directions of regularity obtained by xy-parallel SPREF for an instance of the "jumping jacks" action in the Weizmann dataset.

In this work, we capture the temporal regularity flow information of an action class using a recently proposed "Spatio-temporal Regularity Flow" (SPREF)[15]. SPREF computes the directions, along which the sum of the gradients of a video sequence is minimized:

$$E = \int\int\int \left| \frac{\partial (F * \mathcal{G})(y, x, t)}{\partial \zeta(x, y, t)} \right|^2 dxdydt, \tag{9}$$

where F is the spatiotemporal volume, and $\mathcal{G}$ is a regularizing filter (Gaussian). This formulation results in a 3-D vector field in which each location is composed of three values that represent the directions along which intensity in a spatiotemporal region is regular, i.e., the pixel intensities in the region change the least.

SPREF is designed to have three cross-sectional parallel components in order to handle the regularities that depend on the motion and the scene structure. These components are: *xy-parallel*($\mathcal{F}_t$), *xt-parallel*($\mathcal{F}_y$), and *yt-parallel*($\mathcal{F}_x$). For our experiments we employ the *xy-parallel* component of SPREF. A slice from from 3D flow field generated from the *xy-parallel* component of SPREF is depicted in Figure 4.

This approach to regularity flow estimation does not rely on edge detection, hence its success does not depend on the presence of strong edges in the scene. Instead it analyzes the entire spatio-temporal volume, and tries to find the best directions that model the overall regularity of the volume. Even when the local gradient of a pixel is not significant, the global analysis of the region assigns a well-defined direction to it. The strength of SPREF lies in treating the data not as a sequence of 2D images, but as a 3D volume, and processing all of its information simultaneously.

In order to incorporate SPREF flow vectors directly into the synthesis of Action MACH filters, the MACH framework must be generalized to incorporate vector valued data.

In the next subsection we employ an extension to Euclidean n-space based on a Clifford algebra which allows for a generalization of traditional Fourier transform functions to vector fields.

2.2.2 Clifford Embedding

Unlike Action MACH filters derived from scalar values as defined in Section 2.1, the process of synthesizing a filter based on a vector field cannot employ the traditional Fourier transform which is defined on scalar values. Both synthesis and correlation operations of MACH filters are performed in the frequency domain, therefore we require an analog to the classical Fourier transform for vector fields. For this purpose, we follow the framework proposed in [6], which consists of applying an algebraic extension to the degrees of freedom of a multi-dimensional Fourier transform by embedding the spectral domain into a domain of Clifford numbers. This class of Fourier transform is commonly referred to as the "Clifford Fourier transform." Using this embedding we preserve the full information of both magnitudes as well as directions of our vector dataset while learning the action MACH filters.

Clifford algebra extends the Euclidean n-space to a real algebra. For a three-dimensional Euclidean vector space E^3 we obtain an eight-dimensional $\mathbb{R}$-algebra G^3 having the bases $\{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_1\mathbf{e}_2, \mathbf{e}_2\mathbf{e}_3, \mathbf{e}_3\mathbf{e}_1, \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\}$ of a real vector space. Elements belonging to this algebra are referred to as multivectors, and the structure of the algebra is given by: $1\mathbf{e}_j = \mathbf{e}_j; \mathbf{e}_j\mathbf{e}_j = 1; \mathbf{e}_j\mathbf{e}_k = -\mathbf{e}_k\mathbf{e}_j$, where $j = 1, 2, 3$. Based on this algebra, a set of of basic operators can be defined to generalize Euclidian space to encompass vector fields. These include not only the basic operations such as Clifford multiplication and integrals, but also composite operations such as Clifford Convolution and the Clifford Fourier Transform.

The Clifford Fourier transform (CFT) for multivectors-valued functions in 3D is defined as:

$$\mathcal{F}\{\mathbf{F}\}(\mathbf{u}) = \int \mathbf{F}(\mathbf{x}) \exp(-2\pi \mathbf{i}_3 \langle x, u \rangle) |d\mathbf{x}|, \qquad (10)$$

where $\mathbf{i}_3$ represents the the analog of a complex number in clifford algebra, such that $\mathbf{i}_3 = \mathbf{e}_1\mathbf{e}_3$ and ${\mathbf{i}_3}^2 = -1$. The inverse transform is given by

$$\mathcal{F}^{-1}\{\mathbf{F}\}(\mathbf{x}) = \int \mathbf{F}(\mathbf{x}) \exp(-2\pi \mathbf{i}_3 \langle x, u \rangle) |d\mathbf{x}|. \qquad (11)$$

A multivector field $\mathbf{F}$ in Clifford space corresponding to a thee-dimensional Euclidian vector field can be regarded as four complex signals which are independently transformed by a standard complex Fourier transformation. Therefore, the Clifford Fourier transform can be defined as a linear combination of several classic Fourier transforms:

$$\begin{aligned} \mathbf{F}(\mathbf{x}) \quad = \quad & [\mathbf{F}_0(\mathbf{x}) + \mathbf{F}_{123}(\mathbf{x})\mathbf{i}_3]1 + \\ & [\mathbf{F}_1(\mathbf{x}) + \mathbf{F}_{23}(\mathbf{x})\mathbf{i}_3]\mathbf{e}_1 + \\ & [\mathbf{F}_2(\mathbf{x}) + \mathbf{F}_{31}(\mathbf{x})\mathbf{i}_3]\mathbf{e}_2 + \\ & [\mathbf{F}_3(\mathbf{x}) + \mathbf{F}_{12}(\mathbf{x})\mathbf{i}_3]\mathbf{e}_3, \end{aligned} \qquad (12)$$

which can be interpreted as belonging to $\mathbb{C}^4$. Given the linearity property of the Clifford Fourier Transform, the Fourier transform for multivector is given by:

$$\begin{aligned} \mathcal{F}\{\mathbf{F}\}(\mathbf{u}) \quad = \quad & [\mathcal{F}\{\mathbf{F}_0(\mathbf{x}) + \mathbf{F}_{123}(\mathbf{x})\mathbf{i}_3\}(\mathbf{u})]1 + \\ & [\mathcal{F}\{\mathbf{F}_1(\mathbf{x}) + \mathbf{F}_{23}(\mathbf{x})\mathbf{i}_3\}(\mathbf{u})]\mathbf{e}_1 + \\ & [\mathcal{F}\{\mathbf{F}_2(\mathbf{x}) + \mathbf{F}_{31}(\mathbf{x})\mathbf{i}_3\}(\mathbf{u})]\mathbf{e}_2 + \\ & [\mathcal{F}\{\mathbf{F}_3(\mathbf{x}) + \mathbf{F}_{12}(\mathbf{x})\mathbf{i}_3\}(\mathbf{u})]\mathbf{e}_3. \end{aligned} \qquad (13)$$

Therefore, the Clifford Fourier transform of a SPREF vector field can be computed as a linear combination of four classical Fourier transforms. As a result, all of the well-known theorems that apply to the traditional Fourier transform hold for the CFT. In our experiments we map vector fields in 3D Euclidian space to $\mathbf{F}_{123}$ in the Clifford domain, and the remaining components are set to zero. For this purpose we use the publicly available GluCat library.[1] We exploit space decomposition properties described in this section to apply traditional Fast Fourier algorithms in order to accelerate the computation of the CFT, thereby reducing computation to a matter of seconds for a $320 \times 240 \times 300$ flow field.

Since scalars and vectors are part of multivectors, both scalar and vector-valued fields can be regarded as multivector fields. Therefore, the described Clifford embedding becomes a unifying framework for scalar, vector, and multivector-values filters.

2.2.3 Filter Synthesis

Given the SPREF flow field volumes in the frequency-domain we can proceed to convert the corresponding Clifford Fourier matrix into a column vector by concatenating all the columns of the matrix. This results in a single column-vector denoted by x_i, where $i = 0, 1, 2..., N_e$ and N_e represents the number of training examples of an action class. We then proceed to synthesize the Action MACH filter in the frequency domain using the same methodology described in section 2.1. Similarly, detection of new action instances for a given class is performed as described in section 2.1.1, replacing traditional scalar convolution with Clifford convolution. We evaluate the performance of Action MACH filters synthesized on vector fields and compare it with scalar-based filters in our experimental section.

[1] glucat.sourceforge.net

Action	Walk	Jog	Run	Box	Clap	Wave
Walk	0.91	0.04	0.04	0.01	0.00	0.00
Jog	0.05	0.84	0.11	0.00	0.00	0.00
Run	0.01	0.12	0.87	0.00	0.00	0.00
Box	0.01	0.00	0.04	0.95	0.00	0.00
Clap	0.00	0.00	0.01	0.09	0.85	0.05
Wave	0.00	0.00	0.00	0.04	0.06	0.9

Table 1. Confusion matrix using our method synthesized with SPREF vectors for the KTH actions database. Mean accuracy=86.66%

2.3. Action MACH Using Spatio-Temporal Regularity Flow

In the next section we evaluate the performance of each of these approaches to synthesizing action MACH filters and compare our results with existing action recognition methods.

3. Experiments and Results

We performed an extensive set of experiments to evaluate the performance of the proposed method on a series of publicly available datasets and on a collection of actions found in feature films and broadcast television.[2] Details about the datasets and the experiments performed are given below.

3.1. KTH Dataset

The KTH human action dataset [17] contains 25 people performing six action classes, namely: walking, running, jogging, hand waving, boxing, and hand clapping. Each video sequence contains one actor repeatedly performing an action. The dataset contains a varied set of challenges including scale changes, variation in the speed of execution of an action, and indoor and outdoor illumination variations. Each sequence averages about 4 seconds in length.

Action classification is performed by cross correlation in the Clifford Fourier domain. We used the 5-fold cross-validation framework [14] to partition the dataset into K subsamples. We report the mean of the results obtained from MACH filters synthesized on spatio-temporal regularity flow vectors in Table 1. We achieve a mean accuracy of 88.66%, outperforming all other methods that rely on flow-based features alone.

A second set of experiments on the KTH dataset was geared towards evaluating the effect of using different features to train the action MACH filter. A 5-fold cross validation framework was employed to obtain mean accuracy averages for MACH filters trained using block-based optical flow vectors, temporal derivatives (scalar data), and spatio-temporal regularity flow vectors.

[2] http://www.cs.ucf.edu/~mikel/datasets.html

	Mean accuracy
Temporal derivatives	80.9%
Optical flow	87.2%
SPREF	88.66%

Table 2. Comparison of various feature sets used for the MACH filter on the KTH dataset.

Action MACH filters were synthesized for each action based on optical flow vectors by employing the 2-dimensional formulation of the Clifford transform [15]. The mean accuracy of the optical-flow based filter was 87.2%. Finally, an Action MACH filter trained on scalar data obtained from temporal derivatives yielded a mean accuracy of 80.9%.

3.2. Feature Films

We have compiled a dataset of actions performed in a range of film genres consisting of classic old movies such as "A Philadelphia Story," "The Three Stooges," and "Gone With the Wind," comedies such as "Meet the Parents," a sci-fi movie titled "Star Wars," a fantasy movie "The Lord of the Rings: The Return of the King," and romantic films such as "Connie and Carla." This dataset provided a representative pool of natural samples of action classes such as "Kissing" and "Hitting/slapping." We extracted 92 samples of the "Kissing" and 112 samples of "Hitting/Slapping." The extracted samples appeared in a wide range of scenes and view points, and were performed by different actors. Instances of action classes were annotated by manually selecting the set of frames corresponding to the start and end of the action along with the spatial extent of the action instance.

Figure 5. Detections of the kissing actions (a) and the slapping actions (b) in classic feature films.

Testing for this dataset proceeded in a leave-one-out framework. Given the significant intra-class variability present in the movie scenes, the recognition task is challenging. In our experiments using SPREF, we achieved a mean accuracy of 66.4% for the "Kissing" action, and a mean accuracy of 67.2% for the "Hitting/Slapping"action.

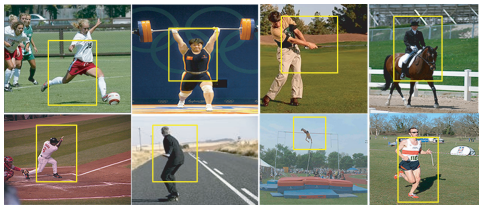

Figure 6. The collection of broadcast sports actions videos represents a set of typical network news videos featured on the BBC and ESPN.

3.3. Broadcast Television Action Dataset

We have collected a set of actions from various sports featured on broadcast television channels such as the BBC and ESPN. Actions in this dataset include diving, golf swinging, kicking, lifting, horseback riding, running, skating, swinging a baseball bat, and pole vaulting (Figure 6). The dataset contains over 200 video sequences at a resolution of 720×480. The collection represents a natural pool of actions featured in a wide range of scenes and view points. To our knowledge this dataset is the first of its kind; by releasing it we hope to encourage further research into this class of action recognition in unconstrained environments.

Testing for this dataset was performed using the leave-one-out cross-validation framework. The confusion matrix for this set of experiments is depicted in Figure 9. The overall mean accuracy for this dataset was 69.2%. Given the difficulty of the dataset, these results are rather encouraging.

3.4. Weizmann Action Dataset

We tested the proposed method on the Weizmann action dataset [3]. Data from this collection was partitioned into testing and training using 5-fold cross validation, the results are depicted in Figure 7.

The average run-time for a $144 \times 180 \times 200$ testing video from this dataset was 18.65 seconds on a Pentium 4, 3.0 GHz. Whereas [3] reports a runtime of 30 minutes on the same architecture for this dataset, our results represent a considerable increase in performance over existing template-based methods.

3.5. Cohn-Kanade Facial Expression Database

Although our main goal is to detect and locate human actions, our framework is well suited for other application domains which involve spatio-temporal matching. We adapted our algorithm to perform classification of different facial action units (AU). To test the method we used data from the commonly used subset of the Cohn-Kanade facial expression database [11]. This database consists of gray scale recordings of subjects displaying basic expressions of emotion on command.

The data included a set of upper face action units: AU1 (inner portion of the brows is raised), AU2 (outer portion of the brow is raised), AU4 (brows lowered and drawn together), AU5 (upper eyelids are raised), AU6 (cheeks are raised), and AU7 (lower eyelids are raised). The action units were partitioned into training and testing using 4-fold cross validation.

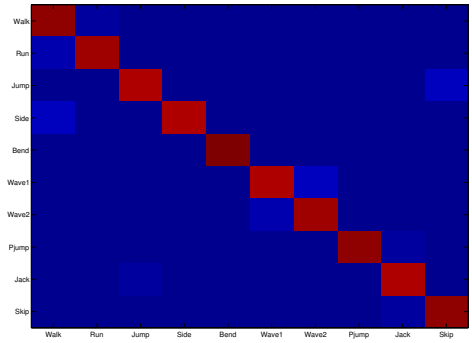

Figure 7. The confusion matrix for depicting the results of action recognition for the Weizmann dataset.

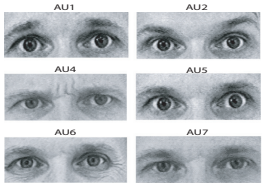

Figure 8. Example frames from the seven facial action units .

Action Unit	AU1	AU2	AU4	AU5	AU6	AU7
AU1	0.78	0.2	0.00	0.00	0.00	0.02
AU2	0.16	0.73	0.11	0.00	0.00	0.00
Au4	0.00	0.00	0.88	0.10	0.02	0.00
AU5	0.00	0.00	0.00	0.92	0.05	0.03
AU6	0.00	0.00	0.00	0.00	0.79	0.21
AU7	0.00	0.00	0.00	0.02	0.022	0.76

Table 3. Confusion matrix for 7 upper face AU. Acuracy=81.0%

Unlike a significant number of existing works [12, 5, 1], no prior facial model or feature tracking was used in training. Additionally, we do not require manual marking of feature points around face landmarks or alignment with a standard face image, yet the performance on the standard dataset (as observed in Table 3) was comparable to current state-of-the-art systems. Despite the fact that our main focus lies in recognizing human body motion patterns, these results indicate that our approach provides enough discriminating power, even when subtle motions of the face are involved.

4. Conclusion

We have introduced the Action MACH filter, a method for recognizing human actions which addresses a number of drawbacks of existing template-based action recognition approaches. Specifically, we address the ability to effec-

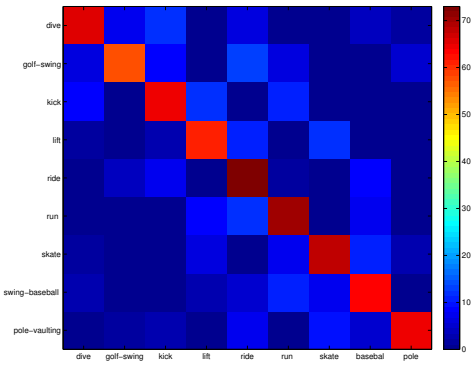

Figure 9. The confusion matrix depicting the results of action recognition for diving, golf swinging, kicking, lifting, horseback riding, running, skating, swinging a baseball bat, and pole vaulting.

tively generate a single action template which captures the general intra-class variability of an action using a collection of examples. Additionally, we have generalized the traditional MACH filter to operate on spatiotemporal volumes as well as vector valued data by embedding the spectral domain into a domain of Clifford algebras.

The results from our extensive set of experiments indicate that the proposed method is effective in discriminating a wide range of actions. These include both whole-body motions (such as jumping jacks or waiving) and subtle localized motions (such as smiling or raising eyebrows). Additionally, by analyzing the response of the Action MACH filter in the frequency domain, we avoid the high computational cost which is commonly incurred in template-based approaches.

5. Acknowledgements

The authors are thankful to Professor Xin Li from the Mathematics department at UCF for his useful discussion on Clifford algebras. The authors would also like to thank the summer interns Aakif Tanveer and Shehzad Aziz, from GIK, Pakistan who under guidance of Saad Ali collected the UCF action data set.

References

[1] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. *CVPR*, 2, 2005.

[2] M. Black, D. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, 78(1):8–31, 2000.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes. *ICCV*, 2, 2005.

[4] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267, 2001.

[5] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang. Facial expression recognition from video sequences: temporal and static modeling. *CVIU*, 91(1-2):160–187, 2003.

[6] J. Ebling and G. Scheuermann. Clifford Fourier transform on vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):469–479, 2005.

[7] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, pages 726–733, 2003.

[8] I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *PAMI*, 19(7):757–763, 1997.

[9] P. Hennings-Yeomans, B. Kumar, and M. Savvides. Palmprint Classification Using Multiple Advanced Correlation Filters and Palm-Specific Segmentation. *Information Forensics and Security, IEEE Transactions on*, 2(3 Part 2):613–622, 2007.

[10] B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[11] T. Kanade and J. Tian. Comprehensive database for facial expression analysis. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 46–53, 2000.

[12] A. Kapoor, Y. Qi, and R. Picard. Fully automatic upper facial action recognition. *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*, pages 195–202, 2003.

[13] Y. Kim, A. Martínez, and A. Kak. Robust motion estimation under varying illumination. *Image and Vision Computing*, 23(4):365–375, 2005.

[14] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 2:1137–1145, 1995.

[15] Y. O. Alatas and Shah. Spatio-Temporal Regularity Flow (SPREF): Its Estimation and Applications. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(5):584–589, 2007.

[16] R. Polana and R. Nelson. Low level recognition of human motion (or how to get your manwithout finding his body parts). *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 77–82, 1994.

[17] L. Schuldt and Caputo. Recognition of human actions. *ICPR*, 2004.

[18] E. Shechtman and M. Irani. Space-time behavior based correlation. *CVPR*, 1, 2005.

[19] S. Sims and A. Mahalanobis. Performance evaluation of quadratic correlation filters for target detection and discrimination in infrared imagery. *Optical Engineering*, 43:1705, 2004.

[20] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *PAMI*, 27(3):475–480, 2005.