A Biologically Inspired System for Action Recognition

by

Hueihan Jhuang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2007

Author .....
Department of Electrical Engineering and Computer Science
September 1, 2007

Certified by.....
Tomaso Poggio
Professor

Accepted by.....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Biologically Inspired System for Action Recognition

by

Hueihan Jhuang

Submitted to the Department of Electrical Engineering and Computer Science
on September 1, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

We present a biologically-motivated system for the recognition of actions from video sequences. The approach builds on recent work on object recognition based on hierarchical feedforward architectures and extends a neurobiological model of motion processing in the visual cortex. The system consists of a hierarchy of spatio-temporal feature detectors of increasing complexity: an input sequence is first analyzed by an array of motion-direction sensitive units which, through a hierarchy of processing stages, lead to position-invariant spatio-temporal feature detectors. We experiment with different types of motion-direction sensitive units as well as different system architectures. Besides, we find that sparse features in intermediate stages outperform dense ones and that using a simple feature selection approach leads to an efficient system that performs better with far fewer features. We test the approach on different publicly available action datasets, in all cases achieving the best results reported to date.

Thesis Supervisor: Tomaso Poggio
Title: Professor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Action Recognition Problem

The problem we deal with is the recognition of actions from video sequences. We are given training data, *i.e.* video sequences of several actions, then classify a test video into one of the pre-defined actions. The applications include surveillance, video retrieval and human-computer interaction.

Humans can robustly recognize actions under various conditions like moving background, clutter, co-occurrence of multiple actions, and variations of viewing angle, position, appearance and scale. Humans can also recognize a wide range of action types including human body, head, hand, and general animal actions. The existing work on action recognition solves one or more of the challenges above, depending on their applications, and is mostly restricted to the domain of human actions. In this work, we focus on video sequences with slight background variations and different foreground variations and with a single subject performing an action throughout the video sequence. The action types are general, including both human and animal actions.

1.2 Motivation

Understanding the perception of actions in both humans and animals is an important area of research crossing the boundaries between several scientific disciplines from computer science to brain science and psychology. In this work we are interested in addressing the action recognition problem by building a model that simulates some well-known human visual capacities.

1.3 The Visual Processing System

The visual cortex appears to be organized into two functionally specialized pathways: a ventral stream that is crucial for the processing of shape information and object vision, and a dorsal stream that is crucual for the processing of the spatial relationships among objects, as well as for the analysis of motion information [72, 40]. Interestingly, the organization of these two pathways is very similar. Their organization is hierarchical; aiming, in a series of processing stages, to gradually increase both the selectivity of neurons and their invariance to spatial transformations [21]. As one proceeds from one area to the next, both neuronal response latencies and average sie of the receptive field, *i.e.* the part of the visual field that if properly stimulated may elicit a response from the neuron, increase along the hierarchy, and neuronal response properties become increasingly complex.

These two pathways originate in the primary visual cortex (V1) where one can find at least two populations of cells: cells which are tuned to spatial orientations (*e.g.* a static vertical bar) and project to areas V2 and V4 of the ventral stream, and cells which are sensitive to direction of motions (*i.e.* a bar at a specific orientation moving in a direction perpendicular to its orientation) and project to area MT and MST in the dorsal stream. The neurons in MT and MST are tuned to speed and direction of motion [37, 2, 31]. The neurons in MST have also been found to have substantial position and scale invariance and [22, 20], and respond to large flow field stimuli.

In this work, we speculate that neurons in intermediate visual areas of the dorsal stream such as MT, MST and higher superior temporal polysensory areas are tuned to spatio-temporal features of intermediate complexity, which pool over afferent input units along space and time. This includes, but is not limited to, the optical flow neurons described above. We assume that such spatio-temporal sensitivity neurons might be found at different locations in the visual cortex such as STS, temporal cortex and prefrontal cortex [24, 69, 47]. Finally, in higher polysensory areas (STSa), one can find neurons that are responsive to the observation of biological motions [55].

Motivated by the recent success of biologically inspired approaches for the recognition of objects in real-world applications [62, 44, 53], we extend a neurobiological model of recognition of biological movements [21, 65]. The model has only been applied so far to simple artificial stimuli.

Our work, based on the similar organization of the ventral and dorsal streams in the visual cortex, applies computational mechanisms that have been proven to be useful for the recognition of objects to the recognition of actions. The idea of extending representations of object to that of actions has been successfully used in a recent non-biologically motivated system [13].

1.4 Previous Work

Typically, computer vision systems for the recognition of actions have fallen into two categories. One class of approaches relies on the tracking of object parts [75, 52, 5]. While these approaches have been successful for the recognition of actions from articulated objects such as humans (see [19] for a review), they are not expected to be useful in the case of less articulated objects such as rodents. The other common class of approaches is based on the processing of spatio-temporal features, either global as in the case of low-resolution videos [76, 14, 4] or local for higher resolution images [59, 13, 16, 46].

Our approach falls in the second class of approaches to action recognition. It extends an earlier neurobiological model of motion processing in the ventral/dorsal

stream of the visual cortex by Giese and Poggio [21]. While their model has been successful in explaining a host of physiological and psychophysical data, it has only been tested on simple artificial stimuli such as point-light motion stimuli [27]. In particular, it is too simple to deal with real videos due to the use of a limited dictionary of features in intermediate stages.

Chapter 2

Background

2.1 Function of the Motion Pathway

Researchers have largely explored the properties of different cortical areas and connections among them [72, 40]. It is believed that there exists at least two functionally specialized processing pathways, ventral stream and dorsal stream, each having the primary visual cortex as the source of initial inputs. Dorsal stream, or motion pathway, is dedicated to the transmission of motion information, *i.e.* visual signals our eyes received during relative motion to the world. The motion pathway starts with retina and LGN, reaching the primary visual cortex (V1), and goes through middle temporal cortex (MT, V5) to medial superior temporal cortex (MST). (See Fig. 2-1). It subsequently projects to higher cortical areas like STD, LIP, VIP, STS, where signals from different pathways are integrated and, due to their complexity, the neural properties are less known.

2.1.1 Primary Visual Cortex (V1)

Starting from the retina, where large ganglion cells called magnocellular, or M cells, are triggered when moving objects sweeps across their receptive fields. The M cells' impulses travel along the optic nerve to a relay station in the thalamus, near the middle of the brain, called the lateral geniculate nucleus (LGN). Then they go to the

middle layer of neurons in the primary visual cortex. There, by pooling together the inputs from many M cells, neurons become sensitive to the spatial orientation and direction of motion. In the primate, most V1 cells have small receptive fields, about $1^o \times 1^o$. Such direction-sensitive cells were first discovered in the mammalian visual cortex by Hubel and Wiesel, who projected moving bars of light across the receptive fields of cells in the primary visual cortex of anesthetized cats and monkeys [26].

Most V1 cells respond to oriented moving bars or edges, and they are classified into two types according to the receptive field structure. Simple cells respond linearly due to the fixed excitatory and inhibitory subregions comprising their receptive fields. Complex cells' responses are independent of the spatial position of the stimulus within the receptive field. It is widely accepted that complex cells combine multiple simple cells to gain position invariance and thus non-linearity [26]. Both simple and complex cells are sensitive to direction of motion and spatial frequency [9, 39]. In addition, complex cells were recently found to be sensitive to the speed of the moving stimulus [51].

2.1.2 Middle Temporal Area (MT)

Area MT lies along the posterior bank of the superior temporal sulcus [18]. The cells in this area inherit the direction and speed tuning properties from their direct afferent inputs, V1 complex cells [42, 3, 37, 39]. Inside the large receptive field, about $10^o \times 10^o$, of MT cells, integration of local sensed motion into the perception of a whole moving object starts occurring, as supported by the finding of pattern-sensitive neurons by Movshon *et al.*, who presented a plaid containing two gratings with different orientations and moving independently along the direction perpendicular to their orientations [41, 36]. The direction of the plaid is thus the vector sum of the direction of the two gratings. Relative to component-sensitive cells which respond when one of the grating moves along the cells' preferred direction, pattern-sensitive cells respond when the direction of the plaid matches the cells' preferred direction. Using more complex moving patterns, pattern-sensitive cells are shown to be insensitive to the exact shape of the moving stimulus [50].

2.1.3 Medial Superior Temporal Area (MST)

Area MST receives its input from the MT area [71, 67]. This area contains at least two major subdivisions: a ventral-lateral one (MSTl), and a dorsal one (MSTd). The cells in MSTl have been shown to have relatively small receptive fields, similar in size to those found in area MT at the same eccentricity and also similar in terms of the directional selectivity and their preference for moving bars. The MSTd cells have larger receptive fields and respond to flow-field stimuli. Most of the MSTd neurons respond to radial (expansion/contraction), rotation (clockwise/counterclockwise), translation, and spiral motions (the combination of radial and rotation, see Fig. 7A in [22]), presumably from the particular combination of multiple MT afferent cells [58, 23]. Since these motions are associated with the flow-field patterns projected onto the retina during observer locomotion, it has been suggested by several groups that the area MSTd has a role in processing optical flow information used in the analysis of self motion and visual guidance of movements in space. It has also been suggested that MST may be important in analyzing the complex motions of objects. Similarly to MT pattern cells, MST cells respond to the moving stimulus regardless of the form [20], but opposed to the MT cells that respond to the position of moving stimulus, MST cells are position invariant [30, 22]. This prominent position and form invariance, as well as the large receptive field size, about one fourth the visual field, establishes the role of MST area as further integrating of motion information from MT area.

2.1.4 Superior Temporal Sulcus (STS)

Several electrophysiological or fMRI studies have shown that there exist neurons in STS that respond selectively to biological motions [24, 11, 47, 55]. Neurons in the temporal cortex can learn to associate pairs of arbitrary geometrical stimuli [69]. This is a key capacity to recognize different views of an action. In addition, the integration of form (ventral) and motion (dorsal) pathways has been found in superior temporal polysensory area (STPa) in macaque, and temporal coherence between form and motion signals have been proven to subserve the recognition of biological movements

[48].

2.1.5 Summary

Along the motion pathway, the average receptive field size and the complexity of their optimal stimuli increase steadily, suggesting that the cells receive convergent input from multiple cells in the lower cortical area. In addition, much of the neural mechanism reviewed above can be viewed as a 'bottom-up' process subserved by feedforward projections between successive pairs of areas within the motion pathway. The motion pathway can therefore be modeled as a feedforward hierarchical architecture [17, 33, 54, 21, 61].

The increasing selectivity (from moving edges to complex flow-field patterns) and invariance (position invariance of V1 complex cells, form invariance of MT cells, and from/position invariance of MST cells) observed in the dorsal stream have also been observed in the ventral stream [21, 61], indicating similar organizations of the two streams, and thus supporting the extension from model of object recognition to action recognition.

2.2 Previous Models of Specific Motion Cortical Areas

Several researchers have proposed computational models of individual or multiple motion cortical areas based on different aspects of neuronal properties. In this section we review this work.

2.2.1 V1-MT

Simoncelli and Heeger proposed a two stage model corresponding to neurons in cortical area V1 and MT [66]. Each stage computes a weighted linear sum of inputs, followed by rectification and divisive normalization. The orientation and spatial frequency selectivity of V1 simple cells are modeled by a set of three-dimensional filters

which are oriented in the space-time domain. Following the previous finding that some aspects of complex cells' responses can be obtained by combining subunits distributed over a localized spatial region [15], they computed the responses of V1 complex cells as a weighted sum of simple cells with the same space-time orientation over a local spatial region. In the second stage, MT pattern and component cells are modeled as a weighted sum of V1 complex cells. The speed and direction selectivity of MT pattern cells are constructed via an implicit implementation of IOC (intersection-of-constraints) by summing a set of V1 complex cells over a local spatial region and over orientation and spatial frequency. MT component cells sum V1 complex cells with the same space-time orientation over spatial position and spatial frequency. The rectification is imposed to simulate the positive-only responses of neurons, and the normalization accounts for nonlinear response, saturation and lateral inhibition.

2.2.2 V1-MT-MST

Grossberg *et al.* proposed a V1-MT-MST neural model to explain the flow-field pattern sensitivity of MST cells by combining well-known neural mechanisms: log polar cortical magnification, Gaussian motion-sensitive receptive fields, spatial pooling of motion-sensitive signals and subtractive extraretinal eye movement signals [23]. The mapping of visual information from retina to V1 obeys a cortical magnification, meaning the cortical resolution gradually increases from periphery to fovea [8]. The property can be modeled by transforming the visual information in a cartesian coordinate in the retina into a log-polar coordinate in V1 [60]. The mapping was calculated within a $45^o \times 45^o$ visual field, the receptive field size of MST cells. MT cells are computed as a summation of V1 cells with the same preferred direction within a Gaussian receptive field. MST cells are computed as a summation of MT cells with the same preferred direction. (see the Fig 3. in [23]) This formulation transforms the spiral motion in a cartesian coordinate into a oblique linear motion in a log-polar coordinate in the cortex, therefore MST cells' flow-field selectivity simply results from local spatial summation of MT cells with the same directional preferences, rather than from complex and specialized interactions as the template model in [58].

2.2.3 MT Speed Tuning Cells

Perrone proposed a mechanism to explain the speed tuning of MT cells by investigating their properties in the frequency domain [49]. Considering one-dimensional motion, an object moving in a constant speed has a spectrum that lies on a line in the spatio-temporal frequency domain [73]. By measuring the neuronal responses to moving sine-wave gratings of different combinations of spatial and temporal frequencies, the spectral receptive field (SRF) can be mapped out. The SRF of speed-tuned MT cells is typically oriented relative to the spatial and temporal frequency axes, similar to that of a moving edge with a fixed speed. Conversely, the typical SRF of a V1 cell is parallel to the spatial and temporal frequency axes. Therefore, the speed tuning of MT cells can be constructed by combining the non-oriented SRF of V1 cells into the oriented SRF of MT cells (a formulation is derived in Eq 1. in [49]).

2.3 Related Feedforward Hierarchical Models

Our system consists of a feedforward hierarchical architecture which has been developed by several researchers. The main connection between hierarchical stages is each unit in a stage receives inputs from multiple units in the previous stage. This idea was inspired by Hubel and Wiesel [26] and subsequently the architecture was constructed by Fukushima and applied on handwritten-digits recognition [17]. LeCun *et al.* developed the convolutional network [33], also a feedforward hierarchical architecture. With no attempt to model biology, Riesenhuber and Poggio developed the HMAX model for the ventral stream [54]. Giese and Poggio extended it to include dorsal stream and applied it to the recognition of biological motion [21]. More recently HMAX model was refined by Serre *et al.* and successfully applied to the multiple object recognition tasks in real world scenario [62, 61]. In this section, we briefly review recent work that are mostly related to our system.

2.3.1 Object Recognition with Cortex-like Mechanisms

Serre *et al.* built a computational model accounting for several well-known facts: (a) visual processing is hierarchical with increasing position and scale tolerance at each stage. (b) along the hierarchy, the receptive fields of neurons and the complexity of their preferred stimulus increase. (c)The first 100-200 ms visual information processing is feedforward (d) plasticity and learning probably occur at all stages [62, 61].

The model is hierarchical with alternating simple S units and complex C units. The S units combine their inputs with Gaussian-like tuning to increase selectivity, and the C units pool their inputs through a maximum operation to increase invariance to 2D transformations. In the first stage, S_1 units model the spatial-orientation-selective V1 simple cells by Gabor filters with a range of orientations and spatial scales. In the next complex stage, C_1 units mimic the scale and position tolerant V1 complex cells by pooling S_1 units with the same orientation over a local spatial region and over adjacent scales. In the next simple stage, S_2 units are modeled as Gaussian functions that are tuned to prototypes extracted from training examples. The S_2 units are similar to the view-tuned neurons in inferotemporal cortex (IT) , which are selective to complex shapes. The input of each S_2 unit is an image patch from the previous C_1 stage with all the orientations and at a particular scale. Therefore, S_2 maps are computed at all positions and all scales. In the next complex stage, C_2 units pool a global maximum from S_2 maps over all scales and all positions. This results in a vector representation of an input image, with each element corresponding to the best match between the image and a prototype. A support vector machine (SVM) is then trained to classify images based on these vector representations.

2.3.2 Neural Mechanisms for Biological Motion Recognition

Giese and Poggio built a model based on several experimental results relating to the recognition of biological movements [21]. The model is divided into two parallel processing streams, modeling the ventral and dorsal pathways. The model of ventral pathway is a simpler version of the HMAX model [54] (also see Sec. 2.3.1).

The model of dorsal pathway considers the tuning properties of V1, MT, and MST cells by a four-stage model. The first stage consists of motion detectors corresponding to V1 direction-selective cells and MT component cells. The second stage models cells that are sensitive to local flow-field structure. Two types of cells are considered: MT translation-flow-sensitive cells and motion-edge-sensitive (or opponent-motion sensitive) cells in MT and MST. Positional and scale invariance of MST neurons are modeled in this stage by pooling from position-specific motion-edge detectors through a maximum operation. The third stage uses Gaussian functions to model the flow-field-pattern-sensitive neurons found in STS and MST. The Gaussian functions center at flow-field patterns extracted from training sequences. The last stage achieves temporal order selectivity by adding the lateral connections between the flow-field-pattern-sensitive neurons.

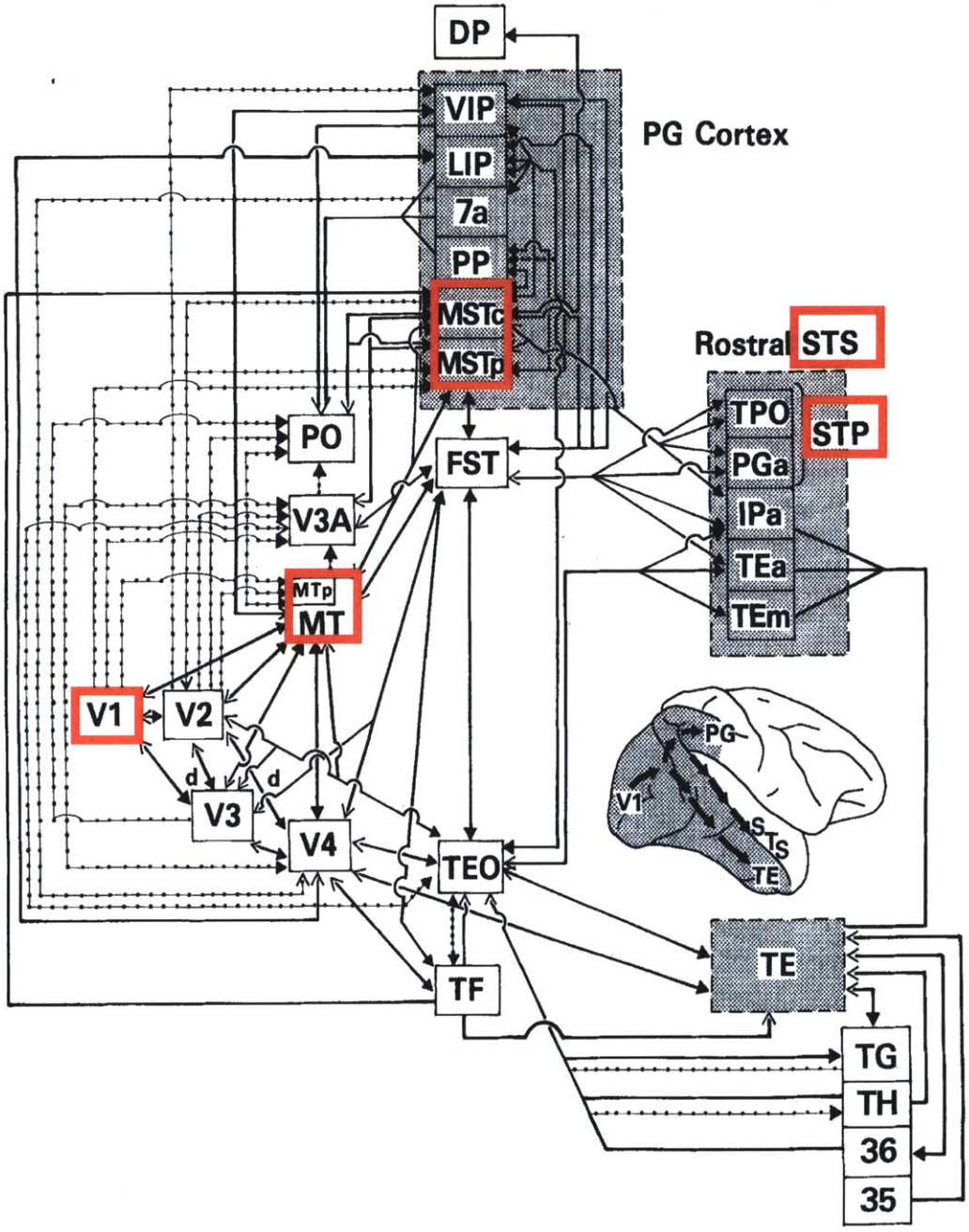

Figure 2-1: Summary diagram of the visual cortical hierarchy. Red boxes indicate the cortical areas we model along the dorsal stream. Reproduced with permission from [12].

Chapter 3

The System

3.1 System Overview

Our approach builds on recent work on object recognition [62, 61] based on hierarchical feedforward architectures and extends a neurobiological model of motion processing in the visual cortex [21]. The system has a hierarchical structure and uses as an input a gray value video and outputs a vector-form representation. In the first stage, motion features are detected by motion-sensitive units which bear functional similarity to V1 simple cells and MT cells, and in the next stage, tolerance to spatial translation is built by a maximum-pooling mechanism which simulates V1 complex cells. In the higher stages of the hierarchy, we predict the existence of neurons that respond to spatio-temporal features and that may be similar to motion-pattern-sensitive MST neurons and temporal-order-sensitive STP neurons. Such predicted neurons are modeled by a template matching operation. By alternating the template matching (simple) and maximum-pooling (complex) operations, the extracted features gradually gain their complexity and invariance. In the last stage, features are selective to complex motion patterns and temporal orders of sequences and tolerant of local deformations in space and shifts in time. The system is illustrated in Fig. 3-1 (also see Appendix for the detailed implementation).

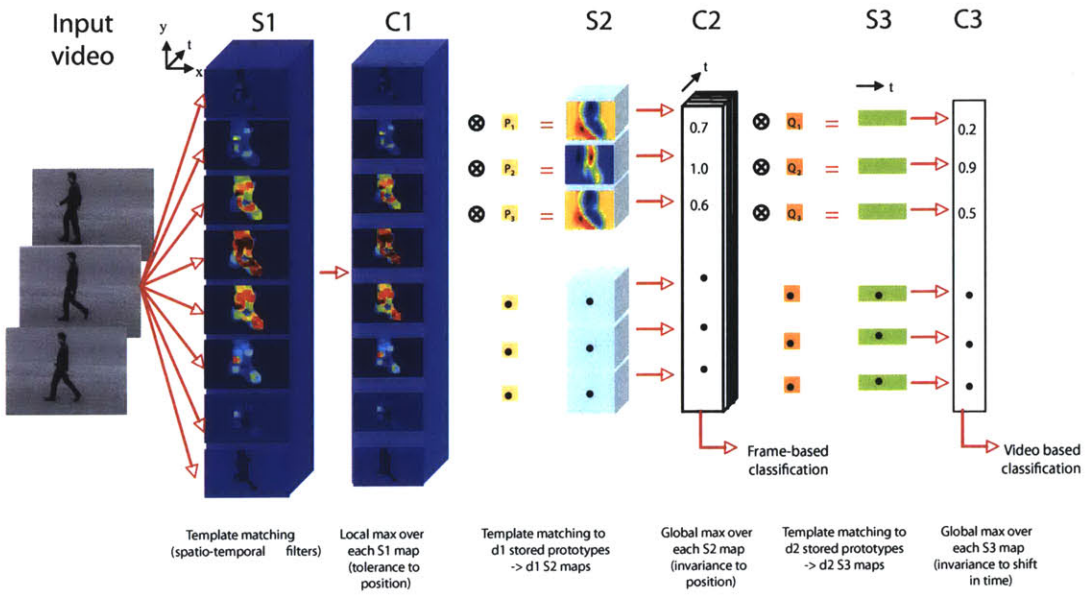

Figure 3-1: Sketch of the system (see text for details).

3.2 Representation

S_1 **units** The first stage of the system contains a set of motion-sensitive S_1 units which are similar to the V1 simple cells and MT cells in the motion pathway. Each S_1 unit extracts one attribute of motion from the input video, a 2-8 second image sequence with frame rate 25 (fps). The output of this stage is a video sequence with several layers of frames (S_1 **frames**). Each layer is the output of one S_1 unit. We will describe three kinds of S1 units and briefly review the related work that motivate our choices.

Space-time-gradient-based S_1 units: The space-time gradients are three-dimensional vectors containing gradients at each pixel along two spatial dimensions and one temporal dimension. Several studies have shown that the space-time gradients contain useful motion information. A statistical distance measurement based on normalized space-time gradients is applied to event recognition [76]. The similarity measurement between two space-time patches can be built through the statistics of space-time gradients [63]. Several optical flow algorithms based on the constant-

brightness assumption accumulate local motion information by computing space-time gradients. In this work, we use two types of S_1 units, each computing the ratio of the temporal gradient to a spatial gradient. We compute the ratios, instead of using the three gradients directly, to keep features in the same scale so as to avoid any bias of the template matching in the higher stages. Also, to make features invariant to contrast reversal between foreground and background, their absolute values are taken.

Optical-flow-based S_1 units: The S_1 units model the direction-sensitive V1 neurons and the speed-sensitive MT neurons, as motivated by the work in [21]. The directional tuning curve of V1 neurons is modeled as a circular-Gaussian-like function [6, 21, 57]. The speed-sensitive MT neurons can be classified as low-pass, speed-tuned or broad-band, based on the characteristics of responses [31]. Low-pass cells are characterized by large responses to slow speeds and a small upper cutoff speed. Broad-band cells are characterized by large responses to slow speeds and a large upper cutoff speed. Speed-tuned cells have a salient peak in the response curve, indicating the cells' preferred direction. As opposed to previous work in which broad-band cells are modeled by a band-pass function [6], we model the speed-tuned cells by an exponential function. We use eight V1 and MT neurons with preferred directions and speeds chosen to be in the range of our motion sequences. We use eight S_1 units, each combining the response of one V1 and one MT neuron in a multiplicative way.

Space-time-oriented S_1 units: Most studies focus on the spatial structure of receptive fields: V1 simple cells' receptive field profiles were modeled by two-dimensional Gabors or Gaussian-derivative functions in [28, 29]. However, the organization of the receptive field is not static: Mclean *et al.* analyzed the three-dimensional first-order properties of simple cells in cat and found two classes of cells [38]. For one class, the receptive field profiles are space-time separable, meaning the spatial and temporal profiles can be disassociated. Receptive field profiles in the other class are inseparable, meaning that the excitatory and inhibitory subregions comprising the

receptive field are tilted in space-time domain (the two classes were also reported in [10]). They found most of the simple cells with separable receptive fields are not direction-selective, and for those with inseparable receptive fields, the preferred direction can always be predicted by the oblique direction of the subregions, and the preferred speed can be derived from the slope of the tilted subregions. Motivated by their idea that the space-time tilted subregions of receptive fields underly velocity selectivity of V1 simple cells, we model S_1 units as a set of space-time-oriented three-dimensional filters.

Several studies have used three-dimensional linear filters as motion detectors. The energy model was built from two space-time separable filters whose spatial responses are 2D Gabor functions and temporal responses are based on psychophysical experimental results [1, 56]. A set of three dimensional Gabor filters were used to extract image flow [25]. MT neurons were modeled by three-dimensional Gaussian derivative filters in [66]. In this work, we use the directional (3^{rd}) derivatives of three-dimensional Gaussians as S_1 units, following the work in [66]. The size of the filters is chosen to match that of the receptive field of a typical V1 simple cell [62, 56]. The orientations of the filers in space-time depend on the preferred directions and speeds of the S_1 units.

C_1 units Tolerance to local spatial translation is achieved in this stage by pooling a maximum response from S_1 frames over local spatial positions. The pooling mechanism has been widely used to model V1 complex cells: some work computed the V1 complex cells as a linear summation of V1 simple cells [43, 66, 15], and others computed the V1 complex cells as a local maximum of V1 simple cells [21, 62]. Comparing the two operations, maximum-pooling assures that the pooled features do not lose their selectivity built by previous stages. In addition, maximum-pooling provides robustness to the background clutter.

The pooling is performed for each layer of S_1 frame, meaning the invariance is built upon each motion attribute. The resulting C_1 **frames** are smaller than S_1 frames due to the pooling, while the number of layers and the number of frames remain the

same. Note that the maximum-pooling operation is separately applied to each frame without temporal pooling.

S_2 units This stage consists of motion-prototype-sensitive S_2 units: their existence is a prediction of the model. S_2 units are similar to MST neurons in that they both respond to complex motion patterns. The difference is that S_2 units respond to class-dependent prototypes extracted from the training data, while MST neurons respond to patterns with general structures such as circular, radial, spiral or translational motion [23, 22]. The role of S_2 units in the hierarchy is to increase the feature complexity and selectivity by a template matching operation between the input features and the stored motion prototypes.

The motion prototypes are extracted at a random spatial position and across all the layers of a random training C_1 frame. See Fig. 3-2 (a) for an illustration. Taking the input as a C_1 frame with all the layers, each S_2 unit convolves the stored prototype with the input frame. This results in a S_2 **map** where each pixel represents a similarity measurement between a patch of the input C_1 frame and the stored prototype.

We consider two metrics of similarity measurements: the dense Euclidean distance as used in [62] and the sparse normalized dot-product as used in [44]. The two distance measurements differ in the amount of computation. Given a prototype with size n(pixels)$\times n$(pixels)$\times l$(layers), and a patch of the same size, in the dense case, all the ln^2 values are taken. In the sparse case, based on the fact that weak features are noisy and have only minute effects on the responses, at each pixel location, only the strongest value among the l layers is taken, resulting in the size n(pixels)$\times n$(pixels) and thus only n^2 values are considered. Another difference is in terms of the form of operation. Using the Euclidean distance, the response is simply the Euclidean distance between the prototype and the input patch. Using the normalized dot-product, the patch is firstly sparsified similarly to the prototype, meaning that at each pixel location, the value of the patch is chosen from the layer used in the prototype's corresponding pixel location. The response is the dot product of the $n \times n$ prototype and the patch normalized by their norms.

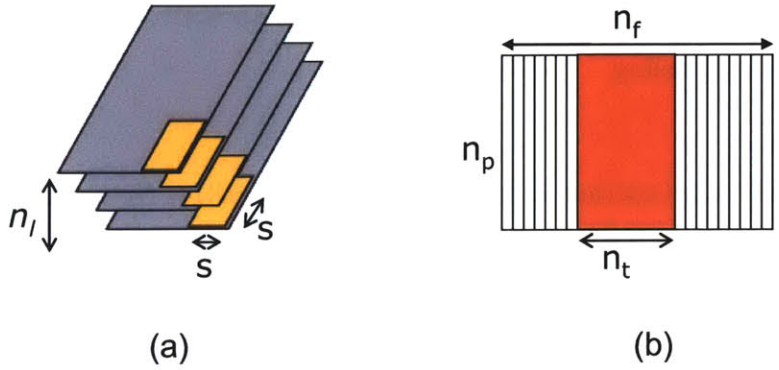

Figure 3-2: (a) The extraction of a S_2 motion prototype. (b) The extraction of a S_3 temporal prototype (see Appendix for the notations.)

C_2 units Similar to the role of the C_1 unit, the C_2 unit adds position invariance by a maximum-pooling operation. The C_2 unit pools the global maximum across all the pixel locations of an input S_2 map, resulting in a scalar representing the best match between the C_1 frame and the motion prototype. By stacking all the C_2 responses of a frame, we get a vector representation (C_2 **vector**).

S_3 units Sequence selectivity is one of the neural mechanisms involving in action recognition, meaning that neurons are tuned to a temporal order, and randomization of the temporal order of the frames doesn't trigger neurons. It was previously modeled as from asymmetric lateral connections of neurons [21]. In this work, to be consistent with the use of motion-prototype-sensitive S_2 units, we model the sequence-selective neurons by temporal-prototype-sensitive S_3 units. We firstly align the C_2 vectors of a video into columns, resulting in a C_2 **matrix**. Each temporal prototype is then extracted at a random column and across all rows of a random training C_2 matrix. See Fig. 3-2 (b) for an illustration.

Taking as an input a C_2 matrix, each S_3 unit convolves the stored temporal prototype with the input C_2 matrix. This results in a S_3 **map** where each pixel represents a similarity measurement between a patch of the input C_2 matrix and the stored temporal prototype.

C_3 units Similar to the role of the C_1 and C_2 units, the C_3 unit adds invariance to shifts in time by a maximum-pooling operation. The C_3 unit pools the global maximum across all the pixel positions of an input S_3 map, resulting in a scalar representing the best match between the C_2 matrix and the temporal prototype. By stacking all the C_3 responses of a video, we get a vector representation (C_3 **vector**).

3.3 Feature Selection

The S_2 stage is the most time-consuming part of the system because it performs template matching between each C_1 frame and each motion prototype. We perform feature selection on the C_2 features [74]. Firstly, we compute the C_2 vectors of a small subset of the training frames by matching them to all the motion prototypes. Then we apply feature selection on these C_2 vectors to identify relevant features, which come from the matching to class-dependent prototypes, and select these motion prototypes. The S_2 maps of the remaining training and test frames are then computed by matching to the selected motion prototypes. See Fig. 3-3 for an illustration.

3.4 Classification

The classification stage uses a support vector machine (SVM). Frame-based and video-based classification are both used to evaluate our system. In the frame-based case, the C_2 vectors are used to train and test an SVM. In the training phase, each frame is assigned the label of the video it belongs to. In the test phase, we obtain a predicted label for each frame of a video, and combine these predictions to get a label for the video by a majority voting scheme. In the video-based case, the C_3 vectors are used to train and test an SVM, and a single label is obtained for each test video. See Fig. 3-3 for an illustration of the two classification approaches.

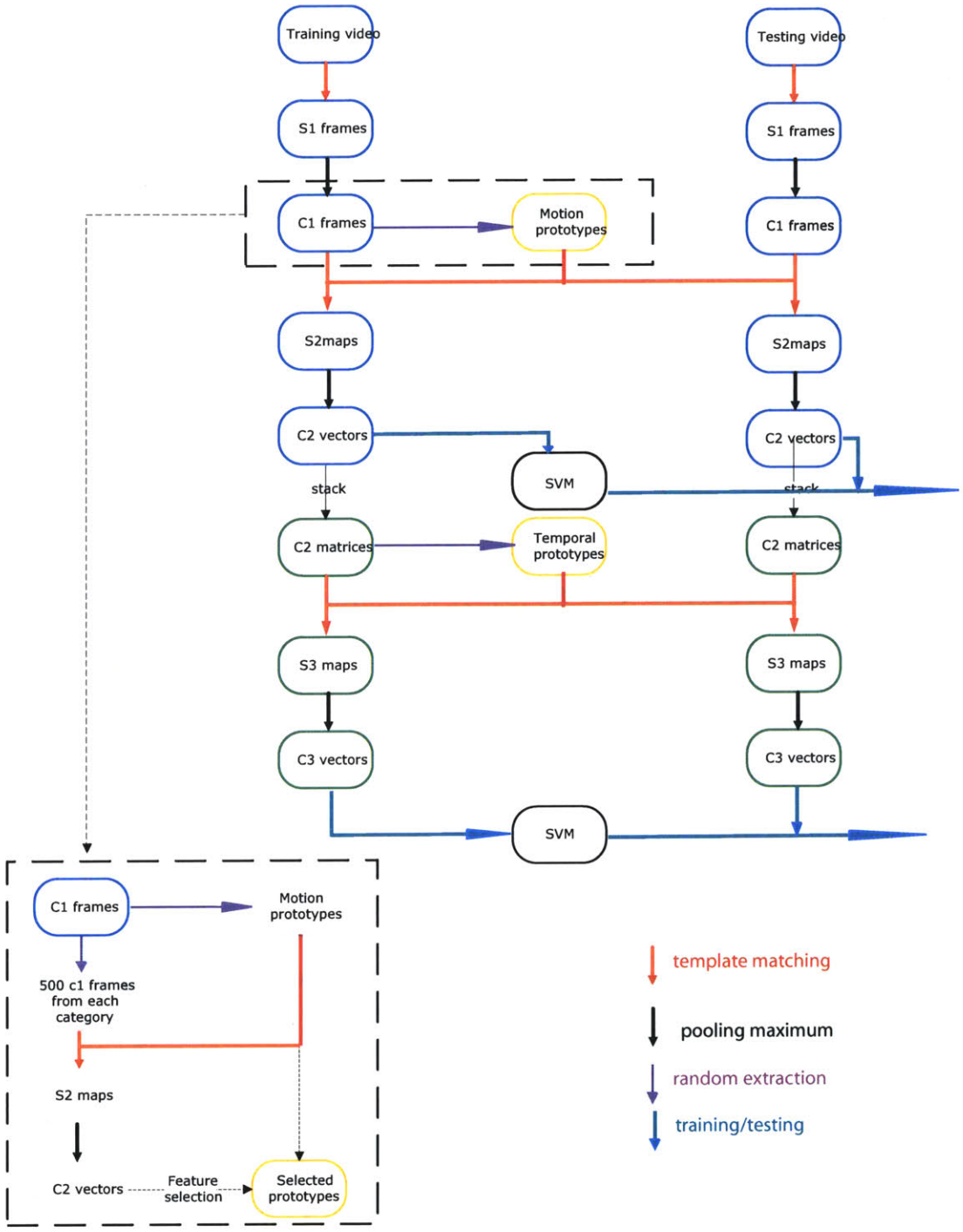

Figure 3-3: The upper part (blue) shows the building of C_2 vectors through repeated matching/pooling mechanisms in the S_1-C_2 stages, and the lower part (green) shows that, stacked C_2 vectors go through another matching/pooling mechanism to generate C_3 vectors. The upper dashed box is modified by adding feature selection, as shown in the lower left dashed box. We randomly extract 500 frames for each action category to generate C_2 vectors, and select prototypes by applying feature selection on the C_2 vectors. The remaining frames are then matched to the selected prototypes.

Chapter 4

Experiments

We have conducted an extensive set of experiments to evaluate the performance of the proposed action recognition system on three publicly available datasets: two human action datasets (KTH and Weizmann) and one mice action dataset (UCSD).

4.1 Methods

4.1.1 Datasets

KTH Human The KTH human action dataset [59] contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. These actions are performed several times by twenty-five subjects in four different conditions: outdoors ($s1$), outdoors with scale variation ($s2$), outdoors with different clothes ($s3$) and indoors with lighting variation ($s4$). The sequences are about 4 seconds in length. The sequences were down-sampled to a spatial resolution of 160×120 pixels. The dataset is shown in the Fig. 4-2.

Weizmann Human The Weizmann human action dataset [4] contains eighty-one low resolution (180×144 pixels) video sequences with nine subjects performing nine actions: running, walking, jumping-jack, jumping forward on two legs, jumping in place on two legs, galloping-sideways, waving two hands, waving one hand, and bend-

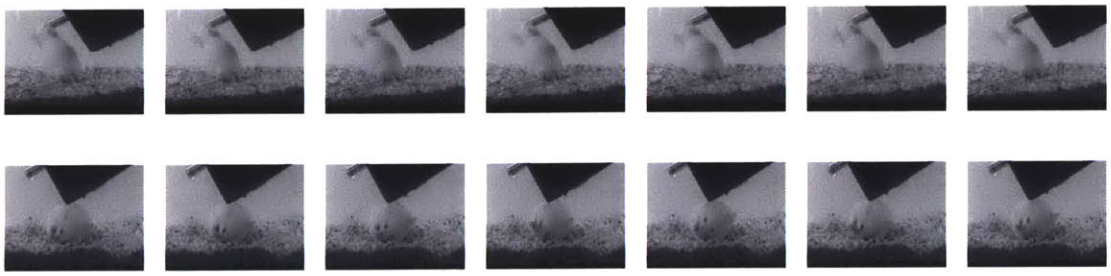

Figure 4-1: Sample videos from the mice dataset (1 out 10 frames displayed with a frame rate of 15 Hz) to illustrate the fact that the mice behavior is minute.

ing. The size of the subject in this dataset is about half the size of the subject in the KTH human action dataset. However, we run experiments on the two sets using the same parameters.

UCSD Mice The UCSD mice behavior dataset [13] contains seven subsets, each being recorded at different points in a day such that multiple occurrences of actions within each subset vary substantially. There are five actions in total: drinking, eating, exploring, grooming and sleeping. The sequences have a resolution of 240×180 pixels and a duration of about 10 seconds. This dataset presents a double challenge. First the actions of the mice are minute (see Fig. 4-1 for examples) and second the background of the video is typically noisy (due to the litter in the cage).

4.1.2 Methodology

Splits We divide each dataset into groups: each condition of KTH Human is divided into 25 groups, one per subject; Weizmann Human is divided into 9 groups, one per subject; UCSD Mice is kept 7 groups as the original setting. We report the recognition rate of our system as the average of 5 rounds. Each round, we train on randomly drawn $\frac{2}{3}$ groups and test on the rest groups. The detail is as follows: each condition of KTH Human contains 16 training groups and 9 test groups; Weizmann Human contains 6 training groups and 3 test groups; UCSD Mice contains 4 training groups and 3 test groups.

Preprocessing We preprocessed the datasets to speed up our experiments: for the KTH human and UCSD mice datasets we used the openCV GMM background subtraction technique based on [68]. In short, a mixture of Gaussians model was used to identify the foreground pixels of each frame. From the foreground mask, we extracted a bounding box (full height, half the width of the frame and centering at the mass center of the foreground pixels) for each frame. For the Weizmann Human dataset, the bounding boxes were extracted directly from the foreground masks provided with the dataset.

Performance Measurement Having represented each video as a vector, we are going to deal with a multi-class classification problem. The most common performance measure is the confusion matrix. Let the number of action categories be n. The confusion matrix is a $n \times n$ matrix, where each row represents a true label, each column represents a predicted label, and element (i, j) is the percent of label-i examples which are classified as label j. The value of the element (i, j) directly reflects the confusion between the two classes, i and j. We compute the overall recognition rate by averaging over the diagonal terms (i, i), $i = 1, .., n$.

4.1.3 Benchmark Algorithm

For benchmark we use the algorithm by Dollar *et al* [13] which has been compared favorably to several other approaches [76, 14] on the KTH human and UCSD mice dataset described earlier. Based on the assumption that a behavior(or action) can be fully described in terms of the types and locations of interest points, a space-time separable filter is applied to detect interest points:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{4.1}$$

where g is a 2D spatial Gaussian function, and h_{ev} and h_{od} are quadrature temporal Gabor functions. The local behavior is characterized by a *cuboid*, *i.e.*, a spatio-temporal window of pixel values around each point detected. A dictionary of cuboid

prototypes is built by clustering cuboids extracted from all the training sequences using K-means algorithm. In the training stage, each cuboid is assigned a type by matching it to the cuboid prototypes, and a vector representation of a sequence is obtained by computing the histogram of its cuboid-types. Each element of the vector denoted the frequency of the occurrences of each cuboid prototype. In the classification stage, a linear SVM classifier is used. The code was graciously provided by Piotr Dollar.

4.2 Results

We have studied several aspects and design alternatives for the system. First we showed that zero-norm feature selection can be applied to the C_2 units and that the number of features can be reduced from $12,000$ down to ≈ 500 without sacrificing accuracy. We then proceeded to apply feature selection for all the remaining experiments and compared different types of motion-direction sensitive input units. We also compared the performance of sparse *vs.* dense C_2 features and present initial preliminary results with the addition of a high-level C_3 stage.

Selecting C_2 features with the zero-norm SVM

The following experiment looks at feature selection and in particular how the performance of the system depends on the number of selected features. For this experiment, we used space-time oriented S_1 units and sparse C_2 features. Performance is evaluated on the four conditions of the KTH dataset.[1] In the first iteration, all the $12,000$ prototypes extracted from the C_1 frames of the training set were used to compute the C_2 features. In each of the following iteration, only features with a weight $|w_i| > 10^{-3}$ were selected.

Table 4.1 compares the performance of each round. In agreement with previous results on object recognition [44], we found that it is possible to reduce the number of C_2 features quiet dramatically (from $\sim 10^4$ down to $\sim 10^2$) with minimal loss in

[1] For computational reason the performance reported is based on a single split of the KTH dataset.

		1	5	10	15	20
$s1$	No. feat.	12000	3188	250	177	158
	accu.	91.7	91.7	89.3	88.9	90.3
$s2$	No. feat.	12000	4304	501	340	301
	accu.	86.6	86.6	85.2	87.0	85.7
$s3$	No. feat.	12000	3805	392	256	224
	accu.	90.3	90.7	89.4	88.4	88.0
$s4$	No. feat.	12000	3152	313	217	178
	accu.	96.3	96.3	96.3	95.3	95.0
Avg	accu.	91.2	91.3	90.1	90.0	89.8

Table 4.1: Selecting features: System performance for different numbers of selected C_2 features at rounds 1, 5, 10, 15 and 20 (see text for details).

	[13]	GrC_2	OfC_2	StC_2
KTH $s1$	88.2	**94.3** / 92.7	92.8 / **93.3**	89.8 / **96.0**
s.e.m. $s1$	±1.9	±1.7 / ±3.2	±2.8 / ±2.9	±3.1 / ±2.1
KTH $s2$	68.3	86.0 / **86.8**	80.7 / **83.1**	81.3 / **86.1**
s.e.m. $s2$	±2.1	±3.9 / ±3.9	±4.0 / ±3.9	±4.2 / ±4.6
KTH $s3$	78.5	85.8 / **87.5**	89.1 / **90.0**	85.0 / **88.7**
s.e.m. $s4$	±2.9	±2.7 / ±3.3	±3.8 / ±3.5	±5.3 / ±3.2
KTH $s4$	90.2	91.0 / **93.2**	92.9 / **93.5**	93.2 / **95.7**
s.e.m. $s4$	±1.8	±2.0 / ±1.9	±2.2 / ±2.3	±1.9 / ±2.1
Avg	81.3	89.3 / **90.0**	88.9 / **90.0**	87.3 /**91.6**
s.e.m. *Avg*	±2.2	±2.6 / ±3.1	±3.2 / ±3.1	±3.6 / ±3.0
UCSD	75.6	78.9 / **81.8**	**68.0** / 61.8	76.2 / **79.0**
s.e.m.	±4.4	±4.3 / ±3.5	±7.0 / ±6.9	±4.2 / ±4.1
Weiz.	86.7	91.1 / **97.0**	**86.4** / **86.4**	87.8 / **96.3**
s.e.m.	±7.7	±5.9 / ±3.0	±9.9 / ±7.9	±9.2 / ±2.5

Table 4.2: Comparison between three types of C_2 features (gradient based GrC_2, optical flow based OfC_2 and space-time oriented StC_2). In each column, the number on the left *vs.* right corresponds to the performance of dense *vs.* sparse C_2 features (see text for details). $s_1, \ldots s_4$ correspond to different conditions of the KTH database (see Section 4.1.1) and *Avg* to the mean performance across the 4 sets. Below the performance on each dataset, we indicate the standard error of the mean (s.e.m.).

accuracy. This is likely due to the fact that during learning, the S_2 prototypes were extracted at random locations from random frames. It is thus expected that most of the prototypes should belong to the background and should not carry much information about each specific action. In the following, feature selection was performed on the C_2 features for all the results reported.

	GrC_3	OfC_3	StC_3
KTH $s1$	**92.1** / 91.3	84.8 / **92.3**	89.8 / **96.0**
KTH $s2$	81.0 / **87.2**	80.1 / **82.9**	81.0 / **86.1**
KTH $s3$	89.8 / **90.3**	84.4 / **91.7**	80.6 / **89.8**
KTH $s4$	86.5 / **93.2**	84.0 / **92.0**	89.7 / **94.8**
Avg	87.3 / **90.5**	83.3 / **89.7**	85.3 / **91.7**
UCSD	73.0 / **75.0**	**62.0** / 57.8	71.2 / **74.0**
Weiz.	70.4 / **98.8**	79.2 / **90.6**	83.7 / **96.3**

Table 4.3: Comparison between three types of C_3 units (gradient based GrC_3, optical flow based OfC_3 and space-time oriented StC_3). In each column, the number to the left *vs.* the right corresponds to the performance of C_3 features computed from dense [62] *vs.* sparse [44] C_2 features. The results are based on the performance of the model on a single split of the data.

Comparing different C_2 feature-types

Table 4.2 gives a comparison between all three types of C_2 features: gradient based GrC_2, optical flow based OfC_2 and space-time oriented StC_2 features. In each column, the number on the left *vs.* the right corresponds to the performance of dense [62] *vs.* sparse [44] C_2 features (see Section 3.1 for details). $s_1, \ldots s_4$ corresponds to the different conditions of the KTH database (see Section 4.1.1).

Overall the sparse space-time oriented and the gradient-based C_2 features (GrC_2 and StC_2) perform about the same. The poor performance of the OfC_2 features on the UCSD mice dataset is likely due to the presence of the litter in the cage which introduces high-frequency noise. The superiority of sparse C_2 features over dense C_2 features is in line with the results of [44] for object recognition.

Comparing different C_3 feature-types

We have started to experiment with high-level C_3 features. Table 4.3 shows some initial results with three different types of motion-direction sensitive input units (see caption). Overall the results show a small improvement using the C_3 features *vs.* C_2 features on two of the datasets (KTH and Weiz) and a decrease in performance on the third dataset (UCSD).

Running time of the system

A typical run of the system takes a little over 2 minutes per video sequence (KTH human database, 50 frames, Xeon 3Ghz machine), most of the run-time being taken up by the $S_2 + C_2$ computations (only about 10 seconds for the $S_1 + C_1$ or the $S_3 + C_3$ computations). We have also experimented with a standard background subtraction technique [68]. This allows us to discard about 50% of each frame thus cutting down processing time by a factor of 2 while maintaining a similar level of accuracy. Finally, our system runs in Matlab but could be easily implemented using multi-threads or parallel programming as well as General Purpose GPU for which we expect a significant gain in speed.

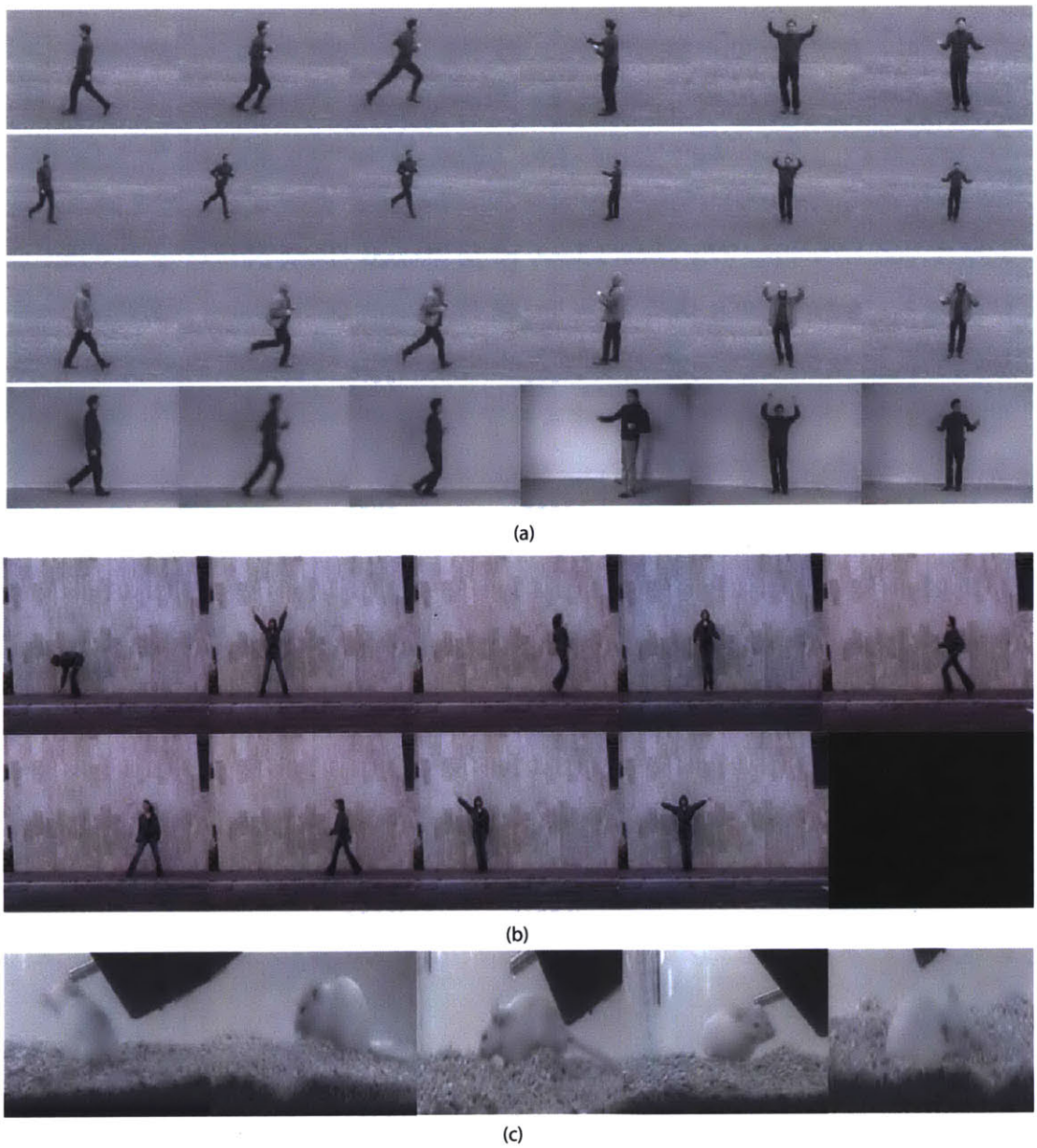

Figure 4-2: (a) KTH Human. First row: outdoor condition. Second row: outdoor with scale variance. Third row: outdoor with different clothes. Fourth row: indoor with lighting variation. Six actions from left to right: walking, running, jogging, boxing, handwaving, and handclapping. (b) Weiz. Human. Actions in the first row from left to right: bending, jumping-jack, jumping forward on two legs, jumping in place on two legs, running, galloping-sideways, walking, waving one hand, and waving two hands. (c) UCSD Mice. Five actions from left to right: drink, eat, explore, groom and sleep.

Chapter 5

Conclusion

5.1 Main Contributions

Our approach is closely related to the feedforward hierarchical architectures with alternating template matching and maximum-pooling, used for the recognition of objects in still images [61]. We list the main extensions as follows:

Using motion-direction sensitive S_1 units In the work [61], a still gray-value input image is first analyzed by an array of Gabor filters (S_1 units) at multiple orientations for all positions and scales. To extend from the system of object recognition to action recognition, we empirically searched for a suitable representation for the S_1 units. We compared three types of motion-sensitive S_1 units: a) Space-time-gradient-based units; b) Optical-flow-based units; c) Space-time-oriented units, which have been shown to be good models of motion-sensitive simple cells in the primary visual cortex [66]. Interestingly, we found that the optical flow features previously used in [21, 6, 65] lead to worse performance than the gradient-based and the space-time-oriented features.

Learning sparse spatio-temporal motion S_2 features In the work [61], a Gaussian-like function is used to compute the responses of dense S_2 features. A more recent work found that using the same Gaussian-like function, the S_2 features

can be sparsified leading to a significant gain in performance on standard object recognition databases [44]. In this work, instead of using a Gaussian-like function, we directly use the Euclidean-distance as a similarity measurement, and we compare the performance of both dense and sparse S_2 features.

Introducing feature selection to the S_2 stage As opposed to video-based processing [59, 13], our system, inherited from object recognition model, is a frame-based processing, in particular, from S_1 to C_2 stage. Using a frame-based processing system, action recognition is time consuming in that each data point is a video sequence containing up to 100 frames. Introducing feature selection can lead to an efficient system with better performance but with less features, as shown in [44]. Motivated by these findings, we experiment with the AROM feature selection technique [74] in the S_2 stage to select relevant motion prototypes, and thus facilitating the template matching. We find that a more compact S_2 feature representation can lead to significant decrease in the computation time taken by the overall system without sacrificing accuracy.

Adding new S_3 and C_3 stages Finally we experiment with an extension of the hierarchy which is specific to motion processing, *i.e.* to include time invariant S_3 and C_3 units. Preliminary experiments suggest that these units sometimes improve performance, but not significantly.

5.2 Role of the System in the Motion Pathway and Action Recognition

Rather than sorting out a "biologically realistic" model from the wealth of anatomical, physiological and biophysical evidence to provide a functional explanation and quantitative simulations of experimental data concerning cells in the dorsal stream, we built a "biologically inspired" system based on two ideas. (1) Simple features processed in low-level cortical areas are transformed into complex features in high-

level cortical areas. (2) Selectivity and invariance are key mechanisms underlying recognition. The two ideas are realized through a hierarchical system with basic simple/complex stages to achieve selectivity/invariance, and by successive use of the simple/complex stages, low-level features also gain their complexity.

The system outperforms state of the art computer vision techniques, regarding the real world action recognition problem. The work illustrates a new approach for action recognition and encourages the move towards a biologically inspired computer vision architecture.

5.3 Future work

(a) Computational complexity of our system is significant. Using the feature selection to choose a small amount of motion prototypes, the running time can be reduced to two minutes per video sequence (about 50 frames). However, in the training phase, computing the C_2 features of the pre-drawn training frames using all the motion prototypes (See Appendix for implementation details) still takes up to several hours. Moreover, for each training/test split, similar training frames and prototypes are repeatedly drawn and used to compute S_2 maps, causing a lot of redundant computation. A possible solution is to build a dictionary containing the selected motion prototypes of a variety of action categories, which are independent of the particular training/test splits. Therefore in each split, the system directly computes the C_2 features based on the prototypes stored in the dictionary, eliminating the matching to similar prototypes in multiple splits.

(b) Adding the scale-invariance by using space-time-oriented S_1 units with multiple filter sizes, as used in [61].

(c) Our system is a feedforward model which takes the segmented actions, single action with background subtraction, as an input. We can achieve visual attention and thus foreground segmentation by taking into account the backprojections known to be numerous in the cortex [70, 64].

(d) Towards a biologically-realistic system. In this work, we model cortical areas

based on the well-known neuronal properties while paying less attention on matching to data of biological experiments. For example, we don't consider scale invariant neurons, and don't explicitly model the MT pattern/component cells and MST cells. Moreover, it remains unclear the model of sequence-selective STPa neurons.

(e) The model accounts only for part of the visual system, the dorsal stream of the visual cortex, where motion-sensitive feature detectors analyze visual inputs. It has been found the integration of form and motion pathway in cortical area STS and their significance for the recognition of biological movements [55]. Giese & Poggio have combined the motion features in the ventral stream with the shape features in the dorsal stream for the recognition of biological movements. A recent work in computer vision has shown the benefit of using shape features in addition to motion features for the recognition of actions [45]. Our system will also move towards this integration.

5.4 Summary

Our main contribution is the application of a neurobiological model of motion processing to the recognition of actions in complex video sequences and the surprising result that it can perform on par or better than existing systems on varying datasets. Indeed none of the existing neurobiological models of motion processing have been used on real-world data [21, 34, 6, 65, 32]. As recent work in object recognition has indicated, models of cortical processing are starting to suggest new algorithms for computer vision [62, 44, 53]. Conversely applying biological models to real-world scenarios should help constrain plausible algorithms.

In order to convert the neuroscience model of [21] into a real computer vision system, we alter it in two significant ways: We propose a new set of motion-sensitive units which are shown to perform significantly better and we describe new tuning functions and feature selection techniques which build on recent work on object recognition.

Appendix A

Detailed Implementation and Parameters

This section gives a quantitative description of each stage of the system.

S_1 units Given an input video with frames $\{I_i \mid i = 1, 2, ..., n_f\}$. For each frame I_i, each S_1 unit computes one layer of motion features, resulting in a three dimensional S_1 frame, denoted as $S1_i$.

Using the space-time-gradient-based S_1 units, each layer is the absolute ratio of the temporal gradient to a spatial gradient computed at each pixel position:

$$S1_i(x, y, 1) = \left| \frac{I_{i+1}(x, y) - I_i(x, y)}{I_i(x + 1, y) - I_i(x, y)} \right| \tag{A.1}$$

$$S1_i(x, y, 2) = \left| \frac{I_{i+1}(x, y) - I_i(x, y)}{I_i(x, y + 1) - I_i(x, y)} \right| \tag{A.2}$$

Using optical-flow-based S_1 units, we compute V_i and Θ_i, the magnitude and direction of motion at each pixel position using Lucas & Kanade algorithm [35]. Each layer is the response of a direction and speed-sensitive S_1 unit:

$$S1_i(x, y, l) = \{\frac{1}{2}[1 + cos(\Theta_i(x, y) - \theta_l)]\}^q \times exp(-|V_i(x, y) - \nu_l|) \tag{A.3}$$

where θ_l is the preferred direction and ν_l is the preferred speed of the i-th S_1 unit. q controls the width of the tuning curve, and is chosen as $q = 2$. $n_l = 8$ layers are computed as a combination of four preferred directions and two preferred magnitudes, which are chosen as:

$$(\theta_l, \nu_l) = \{0^o, 90^o, 180^o, 270^o\} \times \{3, 6 \text{ (pixels per frame)}\} \tag{A.4}$$

Using space-time-oriented S_1 units, we compute each layer as the response to a space-time-oriented-filter F_l:

$$S1_i(x, y, l) = [\frac{\sum_c \sum_b \sum_a F_l(a, b, c) I_{i+c}(a + x, b + y)}{\sum_c \sum_b \sum_a I_{i+c}(a + x, b + y)}]^+ \tag{A.5}$$

where $[\cdot]^+$ denotes the half-way rectification operation. We normalize the response by the average brightness intensity over space and time, and apply half-way rectification to model the positive-only cell responses. We synthesize $n_l = 8$ filters at preferred speeds and directions as Eq. A.4. (See appendix of [66] for the derivation of F_l).

C_1 units The C_1 unit pools the maximum response over a $c \times c$ grid of each S_1 frame. The pooling is done on every $\frac{c}{2}$ pixels, resulting in a C_1 frame with the same number of layers but smaller spatial dimension than the $S1$ frame:

$$C1_i(x, y, l) = \max S1_i(a, b, l) \quad \left|a - \tfrac{c}{2}x\right| \leq c, \left|b - \tfrac{c}{2}y\right| \leq c$$

We choose $c = 8$ [61].

S_2 units Let $\{P_p \mid p = 1, 2..., n_p\}$ denote the set of extracted motion prototypes. Each prototype, P_p, is obtained by extracting a $s \times s \times n_l$ patch from a random pixel position and across all n_l layers of a random training C_1 frame. Four spatial sizes are used: $s = 4$, 8, 12, 16 (pixels). n_l depends on the type the S_1 units used. 500 prototypes are drawn from each action category and for each of the four sizes, yielding the initial $n_p = 10,000$ - $18,000$ prototypes for a dataset containing 5 - 9 categories. After feature selection, the number of selected prototypes is about $n_p = 1,000$.

The S_2 map, $S2_{i,p}$, is computed by matching the i-th C_1 frame, $C1_i$, to the p-th motion prototype, P_p.

Using dense Euclidean distance, it can be expressed as:

$$\begin{aligned} S2_{i,p}(x,y) &= -\left\|C1_i^{xy} - P_p\right\|^2 \\ &= -\sum_{l=1}^{n_l}\sum_{a=1}^{s}\sum_{b=1}^{s}[C1_i(x+a,y+b,l) - P_p(a,b,l)]^2 \end{aligned}$$

where $C1_i^{xy}$ denotes a $s \times s \times n_l$ patch centering at spatial position (x,y) of the $C1_i$ frame.

Using sparse normalized dot-product

$$\begin{aligned} S2_{i,p}(x,y) &= \frac{\hat{C1_i}^{xy} \cdot \hat{P}_p}{\left\|\hat{C1_i^{xy}}\right\| \times \left\|\hat{P}_p\right\|} \\ &= \frac{\sum_{b=1}^{s}\sum_{a=1}^{s}[\hat{C1}_i(x+a,y+b) \times \hat{P}_p(a,b)]}{\sqrt{\sum_{b=1}^{s}\sum_{a=1}^{s}\hat{C1}_i(x+a,y+b)^2} \times \sqrt{\sum_{b=1}^{s}\sum_{a=1}^{s}\hat{P}_p(a,b)^2}} \end{aligned}$$

where $\hat{C1}_i$ and $\hat{P}_p$ are the sparsified $C1_i$ and P_p.

For each pixel (x,y)

$$\begin{aligned} \hat{P}_p(x,y) &= \max\nolimits_l P_p(x,y,l) \\ l^* &= \arg\max\nolimits_l P_p(x,y,l) \\ \hat{C1}_i(x,y) &= C1_i(x,y,l^*) \end{aligned}$$

C_2 units The C_2 unit pools the global maximum response from each S_2 map, $S2_{i,p}$, and the responses of the i-th frame can be stacked into a n_p-element vector:

$$C2_i(p) = \max\nolimits_{a,b} S2_{i,p}(a,b)$$

where the p-th element corresponds to the best match between $C1_i$ and the prototype P_p.

Feature selection on training C_2 vectors The feature selection algorithm we used is AROM (approximation of the zero-norm Minimization) [74]. To reduce computation, we select features based on a subset of training frames instead of the whole training set. Our method is to randomly draw 500 frames from each action category of the training set, computing their C_2 vectors, denoted by $\{C2_j\}$, and apply the following steps:

1. Train a multi-class linear SVM on $\{C2_j\}$ and get a hyperplane $\mathbf{w}$.

2. Update each C_2 vector according to the coefficients of the hyperplane.

$$C2_j \leftarrow C2_j * \mathbf{w} \quad \forall j$$

where $*$ is the element-wise multiplication.

3. Iterate the first two steps until less than 1000 coefficients of the hyperplane $\mathbf{w}$ are significant. We set the significance level as $|w_i| > 10^{-3}$.

The multi-class SVM is based on the implementation of libSVM [7]. Assume we have n action categories, using the one-against-all method, we get n hyperplanes, and we sum over the absolute value of each hyperplane to get a single $\mathbf{w}$. The selected prototypes are those who correspond to significant hyperplane coefficients. We then compute the C_2 vectors of the whole dataset based on the selected prototypes. By selecting about $1,000$ patterns, we can speed up the S_2 computation by $2n$ times. (From $2000n = 4$ (sizes) $\times$ 500 (per action category) $\times n$ (action categories) to 1000).

S_3 units Assume there are N video sequences, each having n_f frames. (Note that n_f varies from video to video.) For each video sequence, by aligning its C_2 vectors into columns, we obtain a $n_p \times n_f$ matrix, denoted as $MC2_j$.

Let $\{Q_q \mid q = 1, 2..., n_q\}$ denote the set of extracted temporal prototypes. Each prototype, Q_q, is obtained by extracting a $n_p \times n_t$ patch from a random column and across all n_p rows of a random training matrix, $MC2_j$. We choose the temporal size $n_t = 7$ because 300 (ms) (assume the frame rate is $25(fps)$) matches the response duration of a typical neuron. 50 prototypes are drawn from each action category,

yielding n_q = 250 - 450 for a dataset containing 5 - 9 categories.

The S_3 map, $S3_{j,q}$, is computed by matching the j-th training matrix, $MC2_j$, to the q-th temporal prototype, Q_q:

$$\begin{aligned} S3_{j,q}(x) &= -\left\| MC2_j^x - Q_q \right\|^2 \\ &= -\sum_{a=1}^{n_p}\sum_{b=1}^{n_t}[MC2_j(a, x+b) - Q_q(a,b)]^2 \end{aligned}$$

where $MC2_j^x$ denotes a $n_p \times n_t$ patch centering at the x-th column of the matrix $MC2_j$.

C_3 units The C_3 unit pools the global maximum response from each S_3 map, $S3_{j,q}$, and the responses of the j-th video sequence can be stacked into a n_q-element vector:

$$C3_j(q) = \max{}_a S3_{j,q}(a)$$

where the q-th element corresponds to the best match between $MC2_j$ and the prototype Q_q.

Classification We use the multi-class linear SVM implementation of libSVM [7].

Using the frame-based classification, there are totally N (videos) $\times n_f$ (frames per video) data points, which can be up to 60,000 for the largest dataset we use (KTH Human). The label of each frame is the label of the video it belongs to. We train a linear SVM on C_2 vectors of 500 training points drawn from each action category, and test on the C_2 vectors of all the testing points. Each test video is predicted as the majority predicted labels of its frames.

Using the video-based classification, there are totally N(videos) data points, which is about 600 for the largest dataset we use. We train a linear SVM on C_3 vectors of all the training points, and test on the C_3 vectors of all the testing points.

Bibliography

[1] EH Adelson and JR Bergen. Spatiotemporal energy models for the perception of motion. *Journal of Optical Society of America*, 2(2):284–299, 1985.

[2] TD Albright. Direction and orientation selectivity of neurons in visual area MT of the macaque. *Journal of Neurophysiology*, 52(6):1106–1130, Dec 1984.

[3] JC Anderson, T Binzegger, KAC Martin, and KS Rockland. The connection from cortical area V1 to V5: A light and electron microscopic study. *Journal of Neuroscience*, 18(24):10525–10540, December 1998.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *International Conference on Computer Vision*, pages 1395–1402, 2005.

[5] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.

[6] A. Casile and M.A. Giese. Critical features for the recognition of biological motion. *Journal of Vision*, 5:348–360, 2005.

[7] C.C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines.* NTU, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` edition, 2001.

[8] PM Daniel and D Whitteridge. The representation of the visual field on the cerebral cortex in monkeys. *Journal of Physiology*, 159:203–221, 1961.

[9] GC DeAngelis, I Ohzawa, and RD Freeman. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. II. linearity of temporal and spatial summation. *Journal of Neuroscience*, 69(4):1118–1135, Apr 1993.

[10] GC DeAngelis, I Ohzawa, and RD Freeman. Receptive-field dynamics in the central visual pathways. *Trends in Neurosciences*, 18(10):451–458, Oct 1995.

[11] J. Decety and J. Grèzes. Neural mechanisms subserving the perception of human actions. *Trends in Cognitive Sciences*, 3(5):172–178, 1999.

[12] C. Distle, D. Boussaoud, R. Desimone, and LG. Ungerleider. Cortical connections of inferior temporal area teo in macaque monkeys. *Journal of Comparative Neurology*, 334(1):125–150, Oct. 2004.

[13] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal feature. In *VS-PETS*, Beijing, China, 2005.

[14] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *International Conference on Computer Vision*, pages 726–733, 2003.

[15] RC Emerson, JR Bergen, and EH Adelson. Directionally selective complex cells and the computation of motion energy in cat visual cortex. *Vision Research*, 32(2):203–218, Feb 1992.

[16] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[17] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[18] R. Gattass and C.G. Gross. Visual topography of striate projection zone (MT) in posterior superior temporal sulcus of the macaque. *Journal of Neuroscience*, 46(3):621–638, Sep. 1981.

[19] D.M. Gavrilla. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[20] B. J. Geesaman and R. A. Andersen. The analysis of complex motion patterns by form/cue invariant MSTd neurons. *Journal of Neuroscience*, 16(15):4716–4732, August 1996.

[21] M. Giese and T. Poggio. Neural mechanisms for the recognition of biological movements and action. *Nat. Rev. Neurosci.*, 4:179–192, 2003.

[22] MSA Graziano, RA Andersen, and RJ Snowden. Tuning of MST neurons to spiral motions. *Journal of Neuroscience*, 14:54–67, 1994.

[23] S. Grossberg, E. Mingolla, and C. Pack. A neural model of motion processing and visual navigation by cortical area MST. *Cerebral Cortex*, 9(8):878–895, Dec 1999.

[24] E. D. Grossman and R. Blake. Brain areas active during visual perception of biological motion. *Neuron*, 35(6):1167–1175, 2002.

[25] DJ Heeger. Model for the extraction of image flow. *Journal of Optical Society of America*, 4:1455–1471, 1987.

[26] D. Hubel and T. Wiesel. *Brain and Visual Perception*, chapter 10,14. Oxford Press, 2005.

[27] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[28] J.P. Jones and L.A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987.

[29] J. J. Koenderink and A. J. van Doom. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367 – 375, March 1987.

[30] L. Lagae, H. Maes, S. Raiguel, D.K. Xiao, and G.A. Orban. Responses of macaque STS neurons to optic flow components: a comparison of areas MT and MST. *Journal of Neuroscience*, 71:1597–1626, 1994.

[31] L. Lagae, S. Raiguel, and G. A. Orban. Speed and direction selectivity of macaque middle temporal neurons. *Journal of Neurophysiology*, 69(1):19–39, 1993.

[32] J. Lange and M. Lappe. A model of biological motion perception from configural form cues. *Journal of Neuroscience*, 26(11):2894–2906, 2006.

[33] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. *The Handbook of Brain Theory and Neural Networks*, 1995.

[34] J. Lee and W. Wong. A stochastic model for the detection of coherent motion. *Biological Cybernetics*, 91:306–314, 2004.

[35] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[36] N.J. Majaj, M. Carandini, and J. A. Movshon. Motion integration by neurons in macaque MT is local, not global. *Journal of Neuroscience*, 27(2):366–370, Jan. 2007.

[37] J. H. Maunsell and D. C. Van Essen. Functional properties of neurons in middle temporal visual area of the macaque monkey. I. selectivity for stimulus direction, speed, and orientation. *Journal of Neurophysiology*, 49(5):1127–1147, 1983.

[38] J McLean, S Raab, and LA Palmer. Contribution of linear mechanisms to the specification of local motion by simple cells in areas 17 and 18 of the cat. *Visual Neuroscience*, 11(2):271–94, Mar-Apr 1994.

[39] A Mikami, WT Newsome, and RH Wurtz. Motion selectivity in macaque visual cortex. II. spatiotemporal range of directional interactions in MT and V1. *Journal of Neuroscience*, 55(6):1328–1339, 1986.

[40] M. Mishkin, L.G. Ungerleider, and K. A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 6:414–417, 1983.

[41] JA Movshon, EH Adelson, MS Gizzi, and WH Newsome. The analysis of moving visual patterns. *Pattern Recognition Mechanisms*, pages 117–151, 1985.

[42] JA Movshon and WT Newsome. Functional characteristics of striate cortical neurons projecting to MT in the macaque. *Soc. Neurosci . Abstr.*, 1984.

[43] J.A. Movshon, I.D. Thompson, and D.J. Tolhurst. Spatial summation in the receptive fields of simple cells in t he cat's striate cortex. *Journal of Physiology*, 1978.

[44] J. Mutch and D. Lowe. Multiclass object recognition using sparse, localized HMAX features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[45] JC Niebles and L Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[46] JC Niebles, H Wang, and L Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *British Machine Vision Conference*, 2006.

[47] Y. Ninokura, H. Mushiake, and J. Tanji. Integration of temporal order and object information in the monkey lateral prefrontal cortex. *Journal of Neuroscience*, 91:555–560, 2004.

[48] MW Oram and DI Perrett. Integration of form and motion in the anterior temporal polysensory area (STPa) of the macaque monkey. *Journal of Neurophysiology*, 76:109–129, 1996.

[49] JA Perrone. A single mechanism can explain the speed tuning properties of MT and V1 complex neurons. *Journal of Neuroscience*, 26(46):11987–11991, Nov. 2006.

[50] N.J. Priebe, C.R. Cassanello, and S.G. Lisberger. The neural representation of speed in macaque area MT/V5. *Journal of Neuroscience*, 23(13):5650–5661, July 2003.

[51] NJ Priebe, SG Lisberger, and JA Movshon. Tuning for spatiotemporal frequency and speed in directionally selective neurons of macaque striate cortex. *Journal of Neuroscience*, 26:2941–2950, 2006.

[52] D. Ramanan and D.A. Forsyth. Automatic annotation of everyday movements. In *Neural Information Processing Systems*, 2003.

[53] M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies, with application to object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[54] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[55] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9):601–670, 2001.

[56] J. G. Robson. Spatial and temporal contrast sensitivity function of the visual system. *Journal of Optical Society of America*, 56:1141–1142, 1966.

[57] N Rust, VM, EP Simoncelli, and JA Movshon. How MT cells analyze the motion of visual patterns. *Nature Neuroscience*, 9(11):1421–1431, Nov 2006.

[58] H Saito, M Yukie, K Tanaka, K Hikosaka, Y Fukada, and E Iwai. Integration of direction signals of image motion in the superior temporal sulcus of the macaque monkey. *Journal of Neuroscience*, 6:145–157, 1986.

[59] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *International Conference on Pattern Recognition*, 2004.

[60] EL Schwartz. Afferent geometry in the primate visual cortex and the generation of neuronal trigger features. *Biological Cybernetics*, 28(1):1–14, March 1977.

[61] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Object recognition with cortex-like mechanisms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 411–426, 2007.

[62] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex cortex. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[63] E. Shechtman and M. Irani. Space-time behavior based correlation. In *In IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[64] M. Shiffrar and J.J. Freyd. Apparent motion of the human body. *Psychological Science*, 1:257–264, 1990.

[65] R. Sigala, T. Serre, T. Poggio, and M. Giese. Learning features of intermediate complexity for the recognition of biological motion. In *ICANN*, 2005.

[66] EP Simoncelli and DJ Heeger. A model of neural responses in visual area MT. *Vision Research*, 38:743–761, 1998.

[67] A.T. Smith. *Visual Detection of Motion.* Academic Press, Dec. 1994.

[68] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[69] M.P. Stryker. Temporal associations. *Nature*, 354:108–109, 1991.

[70] J.K. Tsotsos, Y. Liu, J.C. Martinez-Trujillo, M. Pomplun, E. Simine, and K. Zhou. Attending to visual motion. *Computer Vision and Image Understanding*, 100:3–40, Oct. 2005.

[71] L.G. Ungerleider and R. Desimone. Cortical connections of visual area MT in the macaque. *Journal of Comparative Neurology*, 248(2):190–222, Oct 2004.

[72] L.G. Ungerleider and M. Mishkin. *Analysis of Visual Behavior*, chapter Two cortical visual systems, pages 549–586. MIT Press, Cambridge, MA, 1982.

[73] A.B. Watson and Jr. A.J. Ahumada. A look at motion in the frequency domain. Technical Memorandum 85355, Washington, DC: NASA, 1983.

[74] J. Weston, A. Elisseeff, B. Sch'olkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *JMLR special Issue on Variable and Feature Selection*, 3:1439–1461, 2002.

[75] Y. Yacoob and M.J. Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73(2):232–247, 73.

[76] L. Zelnik-Manor and M. Irani. Event-based video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.