Dorsal Stream: From Algorithm To Neuroscience

by

Hueihan Jhuang

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author . . . .
Department of Electrical Engineering and Computer Science
March 15, 2011

Certified by . . . .
Tomaso Poggio
Professor
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Dorsal Stream: From Algorithm To Neuroscience

by

Hueihan Jhuang

Submitted to the Department of Electrical Engineering and Computer Science
on March 15, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The dorsal stream in the primate visual cortex is involved in the perception of motion and the recognition of actions. The two topics, motion processing in the brain, and action recognition in videos, have been developed independently in the field of neuroscience and computer vision. We present a dorsal stream model that can be used for the recognition of actions as well as explaining neurophysiology in the dorsal stream.

The model consists of a spatio-temporal feature detectors of increasing complexity: an input image sequence is first analyzed by an array of motion sensitive units which, through a hierarchy of processing stages, lead to position and scale invariant representation of motion in a video sequence. The model outperforms or on par with the state-of-the-art computer vision algorithms on a range of human action datasets.

We then describe the extension of the model into a high-throughput system for the recognition of mouse behaviors in their homecage. We provide software and a very large manually annotated video database used for training and testing the system. Our system outperforms a commercial software and performs on par with human scoring, as measured from the ground-truth manual annotations of more than 10 hours of videos of freely behaving mice.

We complete the neurobiological side of the model by showing it could explain the motion processing as well as action selectivity in the dorsal stream, based on comparisons between model outputs and the neuronal responses in the dorsal stream. Specifically, the model could explain pattern and component sensitivity and distribution [161], local motion integration [97], and speed-tuning [144] of MT cells. The model, when combining with the ventral stream model [173], could also explain the action and actor selectivity in the STP area.

There exists only a few models for the motion processing in the dorsal stream, and these models were not be applied to the real-world computer vision tasks. Our model is one that agrees with (or processes) data at different levels: from computer vision algorithm, practical software, to neuroscience.

Thesis Supervisor: Tomaso Poggio
Title: Professor

Acknowledgments

Since my advisor Prof. Tomaso Poggio firstly told me ”you are ready to graduate” in 2009, I have been imagining about writing acknowledgments, but in reality I have only one hour before the due time of my thesis to actually write it.

Living in a foreign country, speaking a foreign language for almost six years is definitely not something I have been planning since my childhood. It was a bit scary in the beginning, but thanks to all the CBCL members and my friends here, it has tuned out to be a very pleasant experience and has became an important part of my life.

Tomaso Poggio, my supervisor, Lior Wolf and Thomas Serre, postdocs (now professors) whom I've been working with, have been leading me and my research with their patience, intelligence and creativity. Especially Tommy and Thomas, they have opened the door to neuroscience for me, a student with EE background, and their research styles and ways of thinking have influenced me very deeply.

Sharat Chikkerur has been a very supportive friend. We joined CBCL at the same day, took many courses together, and stayed overnight to do final projects. Gadi Geiger is like a grandpa and Jim Mutch is like a brother that I never had, listening to my secrets and troubles in life. I also want to thank all my current and former labmates, Kathleen Sullivan, Tony Ezzat, Stanley Bileschi, Ethan Meyers, Jake Bouvrie, Cheston Tan, Lorenzo Rosasco, Joel Leibo, Charlie Fronger and Nicholas Edelman. Although we never hang out for some big trip to something like Cape Cod like other labs, although I never play foosball/pool with everybody, I always enjoy their company at CBCL. All of them together make the lab an interesting and warm place where I'm willing to spend 24 hours a day and 7 days a week, like how Tommy introduced me at my defense.

Friends outside CBCL also play an important role in my life. Jiankang Wang, who specifically asked to be acknowledged, is my roommate and a very good friend for years. Our brains oscillate at very similar frequency (hers is a bit higher than mine) which makes us laugh at the same implicit jokes, even when I was in the most difficult situation. Tobias Denninger is a very dear and extremely smart friend of mine. He has been giving me a lot of mental support as well as useful feedbacks for my research. He also shows me some aspects of life which I would never find out by myself. I want to thank Chia-Kai Liang, Ankur Sinha, Andrew Steele (also a collaborator), ROCSA friends, my friends in Taiwan for their care.

My parents have served as my night line as well as day line; they are always ready to listen to all the details of my everyday life. They have been giving me, for the past 27 years, endless love and strong mental support. I simply cannot imagine life without them.

This thesis is dedicated to my parents and my bb.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The problem

The dorsal stream in the primate visual cortex is involved in the perception of motion and the recognition of actions. The two topics are closely related and have form an important research area crossing the boundaries between several scientific disciplines from computer vision to computational neuroscience and neuropsychology.

Recognizing human actions in videos has also drawn attention in computer vision due to its potential applications in video surveillance, video retrieval/ archival/ compression, and human-computer interaction (Here the term 'action' refers to a meaningful short sequence of motions, such as 'walking', 'running', 'hand-waving', etc). For example, the growing number of images, videos on the internet and movie archives rely on automatic indexing and categorization. In robotics, action recognition is a key to allow the interaction between human and computers and between robots and the environment. In video surveillance, tremendous amount of work of one human observing all the cameras simultaneously can be automated by an action recognition system.

In the field of neuroscience, researchers have been studying how human recognize and understand each other's actions because it plays an important role in the interaction between human and the environment as well as human-human interaction. The brain mechanisms that are involved in the recognition of actions are believed to be mediated in the dorsal stream in the primate visual cortex [202, 54, 53]. Specifically, the MT neuronal responses are closely related to the perception of motion and behavioral choice [18, 20], and in area STP (superior temporal polysensory area), neurons have been found to be sensitive to whole human body movements such as walking [130], or partial body movements such as mouth-opening/ closing and hand-closing/ opening [216]. Moreover, motion processing, the process of inferring the speed and direction of stimulus based on visual inputs, is thought to be highly related to the recognition of actions. Several computational models for motion processing have been built based on neuronal responses to various types of motion [169, 177, 211, 168, 138, 135, 136, 161, 157], and the theoretic solutions have been derived to compute the velocity of an image [64]. These models were able to simulate neurons' selectivity to a range of moving patterns but they were not constructed in a system level such that the motion-selectivity could be applied to the recognition of real-world actions.

Action recognition and the motion processing in the visual cortex have been treated as independent problems. *In this work, we will bridge the gap of the two problems by building a dorsal stream model that could explain the physiological recording from neurons in the dorsal stream as well as be used for the recognition of real world actions.*

1.2 The approach

The visual information received from retina are processed in two functionally specialized pathways [202, 204]: the ventral stream ('what pathway') that is usually thought of processing shape and color information and involved in the recognition of objects and faces, and the dorsal stream ('where pathway') that is involved in the space perception, such as measuring the distance to an object or the depth of a scene, and involved in the analysis of motion signals [202, 54], such as perception of motion and recognition of actions. Both streams have the primary visual cortex (V1) as the source and consist of multiple visual areas beyond V1 (Figure 1-1). Both streams are organized hierarchically in the sense that through a series of processing stages, inputs are transformed into progressively complicated representations while remaining invariant to the change of positions and scales.

Our approach continues two lines of research for the modeling of the visual system. HMAX [152] was based on the organization of the ventral stream and has been applied to

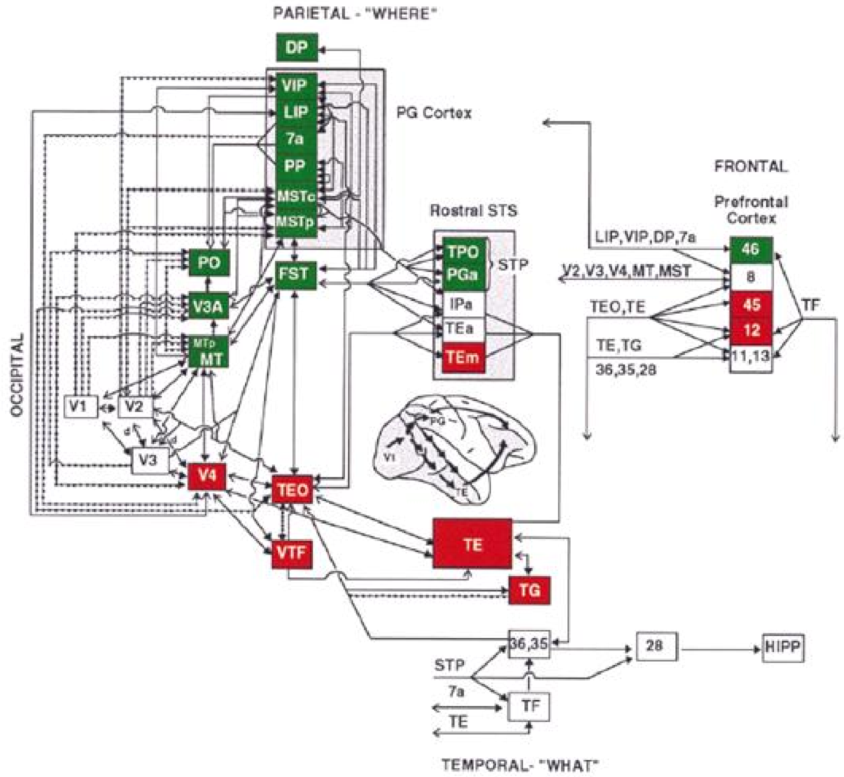


Figure 1-1: Visual processing in monkeys. Areas in the dorsal stream are shown in green, and areas in the ventral stream are shown in red. Lines connecting the areas indicate known anatomical connections, modified from [201].

the recognition of objects with simple shapes. Its was then extended by Serre *et al.* for the recognition of complex real-world objects and shown to perform on par with existing computer vision systems [171, 119]. The second line is the model developed by Giese and Poggio [52]. Their model consists of two parallel processing streams, analogous to the ventral and dorsal streams, that are specialized for the analysis of form and optical-flow information, respectively. While their model is successful in explaining physiological data, it has only been tested on simple artificial stimuli such as point-light motion.

1.3 Outline & summary of thesis chapters

The thesis is organized as follows:

Chapter 2 We give an overview of the dorsal stream model that is used throughout the whole thesis work, its physiology origin, and prior related models.

Chapter 3 We introduce the problem of action recognition and describe the use of the model in Chapter 2 as an action recognition system. The performance of the system and the comparison with state of the art computer vision systems are reported on three public action datasets. This chapter was published in 2007 [75].

Chapter 4 While much effort has been devoted to the collection and annotation of large scalable static image datasets containing thousands of image categories, human action datasets lack far behind. In this chapter we present a dataset (HMDB51) of 51 human action categories with a total of around 7,000 clips manually annotated from various sources such as YouTube, HollyWood movies, Google video. We benchmark the performance of low-level features (color and gist) on HMDB51 as well as four previous datasets to show that HMDB51 contains complex motion which can not be easily recognized using simple low-level features. We use this database to evaluate the performance of two representative computer vision systems for action recognition and explore the robustness of these methods under various conditions such as camera motion, viewpoint, video quality and occlusion. This chapter is currently under submission(Kuehne, Jhuang, Garrote, Poggio & Serre, 2011).

Chapter 5 The extensive use of mouse in biology and disease modeling has created a need for high throughput automated behavior analysis tools. In this chapter we extend the action recognition system in Chapter 3 for the recognition of mouse homecage behavior in videos recorded over a 24 hour real lab environment. In addition, two datasets (totally over 20 hours) were collected and annotated frame by frame in order to train the system and evaluate the system's performance. The system was proven to outperform a commercial software and performs on par with human scoring. A range of experiments was also conducted to demonstrate the system's performance, its robustness to the environment change, scalability to new complex actions, and its use for the characterization of mice strains. This chapter is published in [73].

Chapter 6 A substantial amount of data about the neural substrates of action recognition is accumulating in neurophysiology, psychophysics and functional imaging, but the underlying computational mechanisms remain largely unknown, and it also remains unclear how different experimental evidence is related. Quantitative models will help us organize the data and can be potentially useful for predicting the neuronal tuning for complex human movements in order to understand the representation of movements and how human recognize actions. In this chapter, we show that the model in Chapter 2 could explain neurophysiology of the dorsal stream - it not only mimics the organization of the dorsal stream, but the outputs of the model could also simulate the neuronal responses along the dorsal hierarchy. Specifically, the model account for the spatiotemporal frequency selectivity of V1 cells, pattern and component sensitivity and distribution [161], local motion integration [97], and speed-tuning [144] of MT cells. The model, when combining with the ventral stream model [173], could also explain the action and actor selectivity in the STP area, a high level cortical area receiving inputs from both the ventral and the dorsal stream. An early version of this chapter is published in [74].

1.4 Contribution

Chapter 3 Recognition of actions has drawn attention for its potential applications in computer vision and the role in social interactions that has intrigued neuroscientists. Computer vision algorithms for the recognition of actions and models for the motion processing in the dorsal stream have been developed independently. Indeed, none of the existing neurobiological models of motion processing have been used on real-world data [52, 24, 175]. As recent works in object recognition have indicated, models of cortical processing are starting to suggest new algorithms for the computer vision [171, 119, 148]. *Our main contribution for this topic is to connect the two lines of work, action recognition and motion processing in the dorsal stream, by building a biologically plausible system with the organization of the dorsal stream and apply it to the recognition of real world actions*. In order to extend the neurobiological model for object recognition [171] into a model for action recognition, we mainly modify it in the following ways:

- Propose and experiment with different types of motion-sensitive units.

- Experiment with the dense and sparse features proposed in [119].

- Experiment with the effect of the number of features on the model's performance.

- Experiment with the technique of feature selection.

- Add two stages to the model to account for the sequence-selectivity of neurons in the dorsal stream.

- Evaluate the system's performance on three publicly available datasets.

- Compare the system's performance with a state-of-the-art computer vision system.

Chapter 4 The proposed HMDB database is, to our knowledge, the largest and perhaps the most realistic available database to-date. Each clip of the database was validated by at least two human observers to ensure consistency. Additional meta-information allows a precise selection of test data as well as training and evaluation of recognition systems. The meta tags for each clip include the camera view-point, presence or absence of camera motion and occluders, and the video quality, as well as the number of actors involved in the action. This should allow for the design of more flexible experiments to test the performance of state-of-the-art computer vision databases using selected subsets of this database. *Our main contribution is the collection of the dataset HMDB51 and perform various experiments to demonstrate that it is more challanging than existing action datasets.* Our specific contribution are:

- Compare the performance of low-level features (color and gist) on HMDB51 as well as four previous datasets.

- Compare the performance of two representative systems on HMDB51: C2 features [75] and HOG/HOF features [86].

- Evaluate the robustness of two benchmark systems to various sources of image degradations.

- Discuss the relative role of shape *vs.* . motion information for action recognition.

- Using the metadata associated with each clip in the database to study the influence of variation (camera motion, position, occlusions, *etc.*) on the performance of the two benchmark systems.

Chapter 5 Existing sensor-based and tracking-based approaches are successfully applied to the analysis of coarse locomotion such as active vs. resting, or global behavioral states such as distance traveled by an animal or its speed. However these global measurements limit the complexity of the behaviors that can be analyzed. The limitation of sensor-based approach can be complemented by vision-based approaches. Indeed two vision-based systems have been developed for the recognition of mice behaviors [36, 218, 219]. However, these systems haven't been tested in a real-world lab setting using long uninterrupted video sequences containing potentially ambiguous behaviors or at least evaluated against human manual annotations on large databases of video sequences using different animals and different recording sessions. *Our main contribution is to successfully apply a vision-based action recognition system to the recognition of mice behaviors, to test the system on a huge dataset that includes multiple mice under different recording sessions, and to compare the performance of the system with that of human annotators and the commercial software (CleverSys, Inc).* Our specific contributions are:

- Datasets. Currently, the only public dataset for mouse behavior is limited in scope: it contains 435 clips and 6 types of actions [36]. In order to train and test our system on a real-world lab setting where mice behaviors are continuously observed and scored over hours or even days, we collect two types of datasets: *clipped database* and *full database*.

  - The *clipped database* contains $4,200$ clips with the most exemplary instances of each behavior (joint work with Andrew Steele and Estibaliz Garrote).

  - The *full database* consists of 12 videos, in which each frame is annotated (joint work with Andrew Steele and Estibaliz Garrote).

  - The *SetB*: a subset of *full database*, in which each frame has a second annotation (joint work with Andrew Steele).
  - Make above datasets available.

- Feature computation stage
  - Optimizing motion-sensitive units by experimenting with the number of tuned directions, different types of normalization of features and video resolutions.
  - Learning a dictionary of motion patterns from the *clipped dataset.*
  - Designing a set of position features that helps the recognition of context-dependent actions.
  - Implementing the computation of motion features using GPU (based on CNS written by Jim Mutch [118]).

- Classification stage
  - Experimenting with two different machine learning approaches (SVM *vs.* SVMHMM).
  - Optimizing the parameters of SVMHMM.
  - Experimenting with the number of required training examples for the system to reach a good performance.

- Evaluation
  - Comparing the accuracy of the system with a commercial software and with human scoring.
  - Demonstrating the system's robustness to partial occlusions of mice that arose from the bedding at the bottom of homecage.
  - Demonstrating the system is indeed trainable by training it to recognize the interaction of mice with a wheel.

- Large-scale phenotypic analysis

  - Building a statistical model based on the system's predictions to 28 animal of 4 strains in a home-cage environment over 24 hours, and showing that the statistical model is able to characterize the 4 strains with an accuracy of 90%.
  - Based on system's predictions, we can reproduce the results of a previous experiment that discovered the difference of grooming behaviors between 2 strains of mice.

Chapter 6 The motion processing in the dorsal stream has been studied since 80's [64, 177, 138, 161, 136, 137, 198, 197, 157, 58]. The existing models could explain a range of known neuronal properties along the dorsal hierarchy. These models are however incomplete for three reasons. First, they are not constructed to be applicable in read world tasks. Second, most of them couldn't explain the neural properties beyond the first two stages (V1 and MT) of the dorsal hierarchy. Third, they couldn't explain the recent results of neurophysiology [144, 97]. *Our main contribution is to use the model proposed for action recognition to explain the dorsal stream qualitatively and quantitatively by comparing outputs of model units with neuronal responses to stimuli with various types of complexity and motion.* Our specific contributions are:

- A detailed survey for the known neuronal properties along the dorsal stream.
- Design a population of spatiotemporal filters to match the statistics of V1 cells [47].
- Simulate the pattern and component sensitivity of MT cells [115].
- Simulate the continuous distribution of pattern and component sensitivity of MT cells [161].
- Propose the origin of continuous pattern and component sensitivity MT cells.
- Simulate the speed tuned V1 complex and MT cells [144].
- Simulate the motion opponency of MT cells [180].
- Propose a combination of dorsal and ventral stream model.

- Simulate the action/actor selectivity of STP cells [178].

Chapter 2

The Model

In this chapter, we describe the dorsal stream model that will be used in the next three chapters for various tasks from computer vision to neurophysiology.

2.1 Motivation from physiology

The receptive field (RF) of a cell in the visual system is defined as the region of retina over which one can influence the firing of that cell. In the early 1960s, David Hubel and Torsten Wiesel mapped the receptive field structures of single cells from the primary visual cortex of cat and monkey [68, 69] using bright slits and edges. They concluded that a majority of cortical cells respond to edges of a particular orientation, and cells could be grouped into ”simple” or ”complex” cells, depending on the complexity of the receptive field structures. Simple receptive field contains oriented excitatory regions in which presenting an edge stimulus excited the cell and inhibitory regions in which stimulus presentation suppressed responses. Hubel and Wiesel suggested simple cells structures could be shaped by receiving inputs from several lateral geniculate cells arranged along an oriented line, as shown in Figure 2-1.

Complex receptive fields differ from the simple fields in that they respond with sustained firing over substantial regions, usually the entire receptive field, instead of over a very narrow boundary separating excitatory and inhibitory regions. Most of the complex cells also have larger receptive field size than simple cells. Hubel and Wiesel suggested

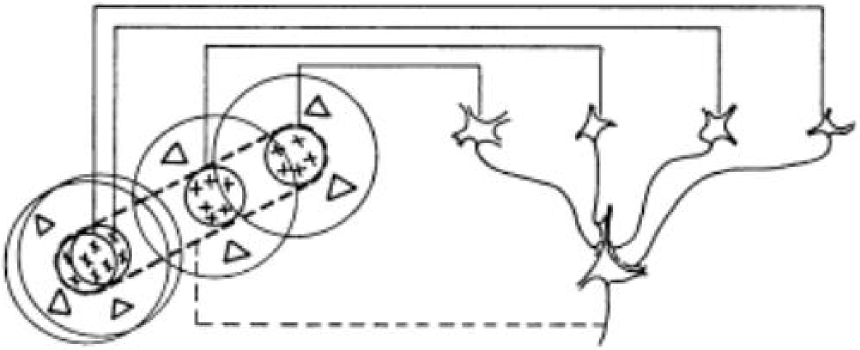


Figure 2-1: A possible scheme for explaining the elongated subfields of simple receptive field. Reprinted from [69].

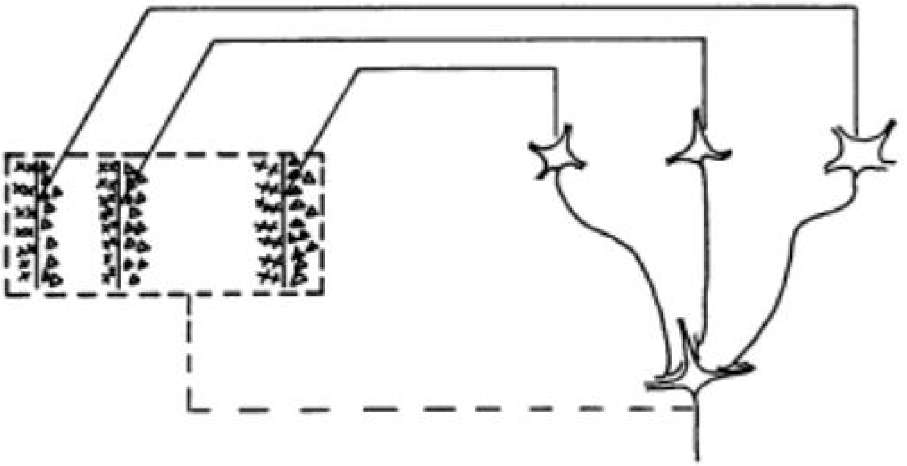


Figure 2-2: A possible scheme for explaining the organization of complex receptive fields. Reprinted from [69].

complex cells pool the response of a number of simple cells whose receptive field is located closely in space, therefore the activation of any simple cell can drive the repones of the complex cell, as shown in Figure 2-2.

Moving edges are more effective in eliciting responses of orientation selective cells than stationary edges. Some cells show similar responses to the two opposite directions perpendicular to the preferred orientation, and the rest of the cells are direction selective, meaning cells show a clear preference of moving direction. Directional selective V1 cells distribute in the upper layer 4 (4a, 4b, and $4C\alpha$) and layer 6 of the visual cortex [62]. These cells then project to area MT [203], where most of neurons are direction and speed sensitive and the receptive field is $2-3$ times larger than V1 direction selective neurons [107]. MT neurons then project to MST, where neurons are tuned to complex optical-flow patterns over a large portion of the visual field, and are invariant to the position of moving stimulus [56]. The linked pathway of visual area V1, MT, and MST is called dorsal steam (”where” pathway) and is thought to be specialized for the analysis of visual motion.

Non-direction selective V1 cells distribute in the layer 2, 3, and 4 (4Cβ) of the visual cortex. They project to cortical areas V2, to V4, and then to the inferiortemporal area (IT). IT cells respond selectively to highly complex stimuli (such as faces) and also invariantly over several degrees of visual angle. This pathway is called ventral stream (”what” pathway) and is thought to be specialized for the analysis of object shape.

It was hypothesized that the two streams form functionally distinct but parallel processing pathways for visual signals. Their computations are similar in the sense that lower level simple features are gradually transformed into higher level complex features when one goes along the visual streams [202].

2.2 Background: hierarchical models for object recognition

The recognition of objects is a fundamental, frequently performed cognition task with two fundamental requirements: selectivity and invariance. For example, we can recognize a specific face despite changes in viewpoint, scale, illumination or expression. V1 simple and complex cells seem to provide a good base for the two requirements. As a visual signal passes from LGN to V1 simple cells, its representation increases in selectivity; only patterns of oriented edges are represented. As the signal passes from V1 simple to complex cells the representation gains invariance to spatial transformation. Complex cells downstream from simple cells that respond only when their preferred feature appears in a small window of space now represent stimuli presented over a larger region.

Motivated by the finding of Hubel and Wiesel, several models have been proposed to arrange simple and complex units in a hierarchical network for the recognition of objects or digits. In these models, simple units selectively represent features from inputs, and complex units allow for the positional positional errors in the features. The series of work starts with the Neocognitron model proposed by Fukushima [49], followed by the convolutional network by Lecun [88], and then HMAX by Riesenhuber & Poggio [152].

The early version of HMAX uses a limited handcrafted dictionary of features in intermediate stages and is therefore too simple to deal with real-world objects of complex shape. In its more recent version developed by Serre *et al.* [173], a large dictionary of intermediate features are learned from natural images and the trained model is able to recognize objects from cluttered scene or from a large number of object categories. HMAX could also explain neurobiology: the computations in HMAX were shown to be biologically plausible circuits and outputs of different layers could simulate the neuronal responses in the area V1, V4, and IT [172, 80]. A sketch of HMAX is shown in Figure 2-3.

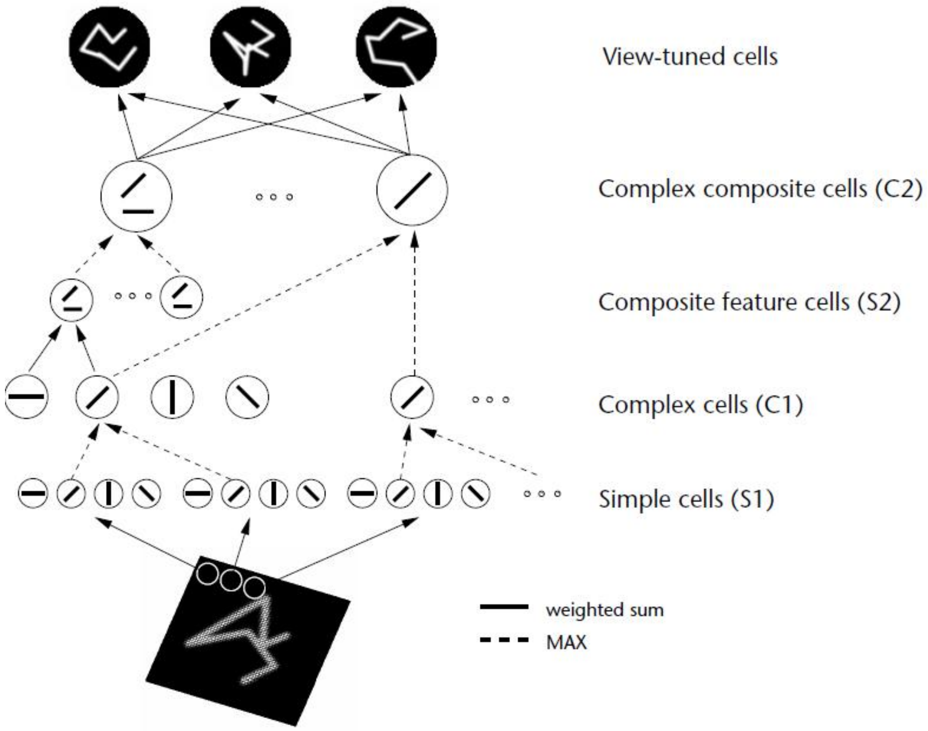


Figure 2-3: HMAX. Figure reprinted from [152]

2.3 The model

The problem of action recognition could be treated as a three-dimensional object recognition problem, in which selectivity to particular motion patterns (as a combination of direction and speed) and invariance to the visual appearance of the subjects play an important role for the recognition of particular action categories. Here we propose a model for the recognition of actions based on HMAX. Our model is also a hierarchy; where simple and complex layers are arranged repetitively to gradually gain the specificity and invariance of input features. The main difference from HMAX is, instead of representing oriented edge features from stationary stimuli, our model represents motion features (directional and speed) of stimuli. Our model is also different from HMAX in terms of detailed implementations, such as normalization of features.

Here we describe an overview of the model structure and a typical implementation. The detailed implementation will vary depending on the particular task and will be described in each of the subsequent chapters. The model's general form is a hierarchy of 4 layers $S_1 \rightarrow C_1 \rightarrow S_2 \rightarrow C_2$: 2 simple layers, S_1 and S_2, and 2 complex layers, C_1 and C_2. Features are selectively represented in the S(simple) layer using a template matching operation. Features are invariantly represented accordingly in the C(complex) layer using a max pooling operation. The model is illustrated in Figure 2-4. The first two stages (S_1, C_1) are designed to mimic the receptive field structures of V1 simple and complex cells, respectively. The latter two stages (S_2,C_2) are designed to repeat the computations in the first two stages. S_2 and C_2 units are our prediction to neurons in the higher-level cortical areas. We will verify this prediction in Chapter 5.

2.3.1 S_1

The input to the model is a gray-value video sequence that is first analyzed by an array of S_1 units at all spatial and temporal positions. A S_1 unit is a three-dimensional filter (two in space and one in time), such as Gabor filter, tuned to a combination of motion (direction and speed) in a particular spatial and temporal scale. Here scale refers to the spatial and temporal size of the filter. Let $\mathbf{I}$ denote the light intensity distribution of a stimulus, f

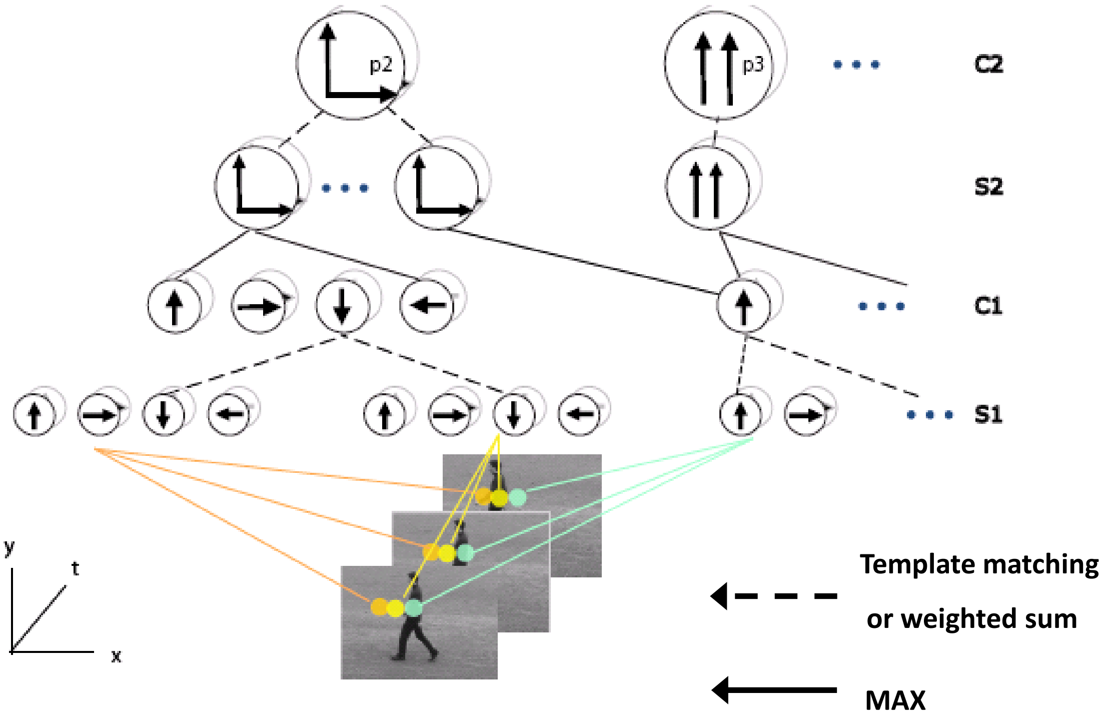


Figure 2-4: The structure of the proposed model. The model consists of a hierarchy of layers with template matching operations (S layer) and max pooling operations (C layer). The two types of operations increase the selectivity and invariance to position and scale change.

denote a receptive field of a S_1 unit. The linear response is computed as the convolution of the stimulus with the unit:

$$\frac{\mathbf{f} \cdot \mathbf{I}}{||\mathbf{f}|| \times ||\mathbf{I}||}. \tag{2.1}$$

The absolute value is then taken to make features invariant to contrast reversal. For the recognition of actions in a video with frame rate 25 fps, we typically use 8 Gabor filters tuned to 4 directions and 2 speeds. For a typical video resolution 240 × 320 (pixels), we use one single scale representation, and filter size 9 (pixels) × 9 (pixels) × 9 (frames).

2.3.2 C_1

At the next step of processing, at each point of time(frame), C_1 units pool over a set of S_1 units distributed in a local spatial region by computing the single maximum response over the outputs of S_1 units with the same selectivity (*e.g.* same preferred direction and speed). To avoid over-representation of motion feature caused by continuously pooling from adjacent spatial regions, the max-pooling is not performed at all the locations. Assume each C_1 unit pools over a spatial $n \times n$ (pixel) grid, we only use C_1 units at every $n/2$ pixel locations. If multiple scales of S_1 units are used, a C_1 unit computes the max response in both neighboring spatial positions and across scales. As a result a C1 unit will have a preferred velocity as its input S_1 units but will respond more tolerantly to small shifts in the stimulus position and scale.

2.3.3 S_2

The S_2 stage detects motion features with intermediate complexity by performing a template matching between inputs with a set of templates(prototypes) extracted during a training phase. The template matching is performed at each position and each scale of the C_1 outputs. A template is defined as a collection of responses of spatially neighboring C_1 units that are tuned to all possible selectivity at a particular scale. Each template is computed from a small spatio-temporal patch randomly sampled from training videos. One can think that a template corresponds to the weights of a S_2 unit, and the preferred feature of the S_2 unit is the template. The responses of a S_2 unit to an input video can be thought of as the similarity of the stimulus' motion (C_1 encoded) to previously seen motion patterns encoded in the same layer (C_1).

Let n_l denote the number of S_1/C_1 selectivity (*i.e.* the number of tuned directions × the number of tuned speeds) and n_c the number of spatially neighboring C_1 units converging into a S_2 unit, a template's size is n_c (pixels) $\times$ n_c (pixels) $\times$ n_l (types). A template with large spatial size (n_c) includes features from a large region and therefore has higher complexity in the feature space than a small template. In the task of action recognition, templates with many sizes are used to encode motion of a range of complexity. A set of typical values is $n_c = 4, 8, 12, 16$ (pixels).

The S_2 units compute normalized dot product (linear kernel); let $\mathbf{w}$ denote the unit's weights and $\mathbf{x}$ a C_1 patch of the same size, the response is given by:

$$\frac{\mathbf{x} \cdot \mathbf{w}}{||\mathbf{x}|| \times ||\mathbf{w}||}. \tag{2.2}$$

The S_2 response can be treated as similarity of motion, measured in the C_1 level, between the present stimulus and the stored template. The S_2 response is 1 if the motion of present stimulus is identical to the template, and close to 0 if their motion is lowly correlated.

In HMAX, a Gaussian kernel (RBF kernel, as opposed to linear kernel used here) is used to compute the S_2 response:

$$exp^{-\frac{||\mathbf{x}-\mathbf{w}||^2}{\sigma^2}} \tag{2.3}$$

The parameter σ controls the sensitivity of the response to the similarity between input features and a template. A large σ value will make the response tolerate to large deviations from the template's preferred feature while a small σ value will cause a unit to respond only when the input closely resembles the stored template.

The linear kernel and Gaussian kernel are similar in the sense that they both respond maximally when the input and stored pattern are identical, and the response decreases with their dissimilarity. Indeed these two operations could be equivalent under some conditions, and linear kernel is a more biologically plausible operation [80].

2.3.4 C_2

In the next stage, C_2 units pool a maximum response over all spatial positions and scales, receiving input from all S_2 units of the same weights (template). One can think that there is exactly one C_2 unit tuned for each template but invariant to the scale and position of the present stimulus. In other words, we obtain a value of the best matching between all the input local motions and a stored motion template.

Chapter 3

A Biologically Inspired System for Action recognition

This chapter has been published as a conference paper in 2007 [75].

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. 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.

3.1 The problem

What is an action? Polana and Nelson [140] separated the class of temporal events into three groups (1) temporal textures which are of indefinite spatial and temporal extent (*e.g.* . flowing water). (2) activities which are temporally periodic but spatially restricted (*e.g.* . a person walking), and (3) motion events which are isolated events and do not repeat either in space or in time (*e.g.* . smiling). Bobick's taxonomy is from a viewpoint of possible human ”actions” [11]. He grouped the percept of motion into movements, activity, and actions. Movements are the most atomic motion primitives, requiring no contextual or sequence knowledge to be recognized. Activity refers to sequences of movements or states, where the only real knowledge required is the statistics of the sequence. Actions are larger-scale events, which typically include interaction with the environment and causal relationships. In this work, the term ’action’ refers to a meaningful short sequence of motions, such as ’walking’, ’running’, ’standing’, etc. It is an union of Polana and Nelson's group (2) and (3): spatially restricted but not necessarily temporally periodic. It is also an union of Bobick's ”activity” and ”action”.

Action recognition is one of the mostly studied computer vision problem due to its important applications such as surveillance, video retrieval and archival, and human-machine interaction. The difficulty of this task in a real world scenario comes from the large variations within action categories as well as the recording condition. For example, ”walking” can differ in speed and stride length, and the same action observed from different viewpoints can lead to very different image observations. The size and appearance difference between individuals further increase the variation. More complex actions are even involved in the interaction with the environment such as ”drinking from a cup”, or interaction wither others such as ”shaking hands” or ”hitting people”. The environment in which the action performance takes place is another important source of variation in the recording. Dynamic backgrounds increase the complexity of localizing a person in the image due to background clutter or partial occlusion of the person. Recording from a moving camera not only makes human localization difficult but also distorts the movements to be different from a static camera. A practical system for video surveillance will, for example, firstly segment all the persons from the background scene, then recognize actions, which might be each individual's action or a group's action. Most of the current action datasets were collected to test the "recognition" part, in which each clip (short video sequence) contains one single actor performing one single action, and the task is to predict an action class for the clip.

The problem of action recognition in videos could be treated as three-dimensional object recognition in a sequence of frames. Indeed, the extension of 2D shape (object) descriptors to 3D motion (action) descriptors has demonstrated its success in some previous works [85, 36, 9]. Motivated by neurophysiology experiments that studied the function and organization of the visual cortex, a series of hierarchical architecture has been proposed for the recognition of objects. A recent model HMAX, firstly developed by Riesenhuber &P oggio [152] and later on extended by Serre *et al.* , has been shown to be a promising model. On one side it is comparable to state of the art computer vision systems for the recognition of objects with complex appearances among a larger number of possible categories [171, 119]. On the other side, it could explain physiological data from various cortical areas as well as human psychophysics [172, 80]. In this work, we describe a system that extends HMAX from representing 2D objects to representing 3D actions and apply it for the recognition of actions in videos collected under real-world scenarios.

3.2 Previous work

Early progress of human action recognition has been achieved by shape-based parametric models of the human skeleton [98, 61, 14, 220, 147, 17]. These systems are based on the assumption that a moving object consists of several parts, and the time-varying relative positions of these parts characterize its action. These approaches rely on the tracking of object parts, and are suitable for recognizing actions of articulated objects such as human (see [51] for a review) but don't apply to less articulated objects such as mice [36].

More recent work shift to the paradigm that characterizes actions based on the motion patterns obtained from the space-time video volume. The representation of motion patterns can be grouped based on their scale: local or global (A complete review of action recognition algorithms is in [109, 199, 141]).

3.2.1 Global representation

Global representations encode the space-time volume of a person as a whole, as shown in Figure3-1. The volume is usually obtained through background subtraction or tracking. Common global representations are derived from silhouettes, edges or optical flow. They are sensitive to imperfect background subtraction, noise, partial occlusions and variations in viewpoint. They also have the disadvantage of not being able to distinguish the actions of less articulated objects. One of the earliest work by Bobick and Davis is to use silhouettes from a single view and aggregate differences between subsequent frames of an action sequence [11, 10]. This results in a motion history image (MHI). Other silhouettes-based works are [223, 9, 213, 182]. Instead of silhouette shape, motion information can be used. Efros et al. [38] calculated optical flow centering around human in very low-resolution videos.

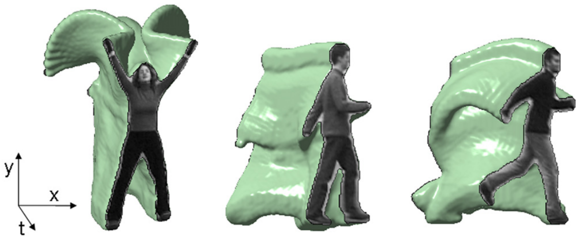


Figure 3-1: Global space-time shapes of ”jumping-jack”, ”walking”, and ”running”. Figure reprinted from [9]

3.2.2 Local representation

Local representations describe the observation as a collection of local descriptors or patches (bag of words) [85, 165, 36, 43, 121, 87, 123, 164, 86, 5], as shown in Figure 3-2. The procedure is as follows: fist spatio-temporal points are sampled or detected at regions of interest, and a descriptor is applied to represent a small patch around these points. Finally, the patch representations are combined into a vector representation for the whole clip. A benchmark paper [210] compares many types of descriptors and evaluates their performance on a set of datasets such as KTH, UCF sports and HollyWood2. Local representations are less sensitive to noise, changes in viewpoint, person appearance, and partial occlusion. It doesn't not strictly require accurate localization, background subtraction or tracking. However, the simplicity of local unordered representation will prevent it from being discriminative when the number of action categories increases.

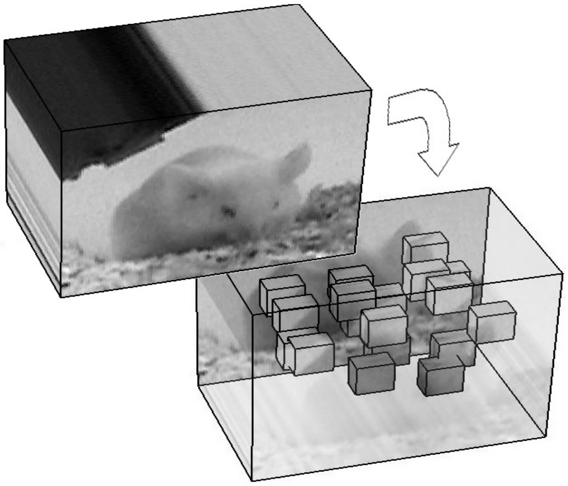


Figure 3-2: Local space-time interest points detected from a space-time shape of a mouse. Figure reprinted from [36]

3.3 System Overview

The system follows a standard procedure for pattern recognition, it firstly converts an input video from gray pixel values into a set of feature representations, then uses the supervised learning technique to train a classifier from a set of feature vectors as well as their labels.

The feature representation is based on the four layer hierarchical model($S_1 \rightarrow C_1 \rightarrow S_2 \rightarrow C_2$) described in Chapter 2. By alternating between a maximum operation in the C stage to gain the invariance, and a template matching operation in the S stage to gain the selectivity and complexity of features, the model gradually builds a representation which is tolerant to 2D changes (*e.g.* the variation of position or appearance of an actor in the visual field) yet specific enough so as to allow fine discrimination between similar actions (*e.g.* jogging and running) . We also experimented with adding two extra layers $S_3 \rightarrow C_3$ on top of the C_2 layer to account for the selectivity and invariance in time. Here We consider two types of features. One is the C_2 output computed for each frame. Note that a C_2 feature

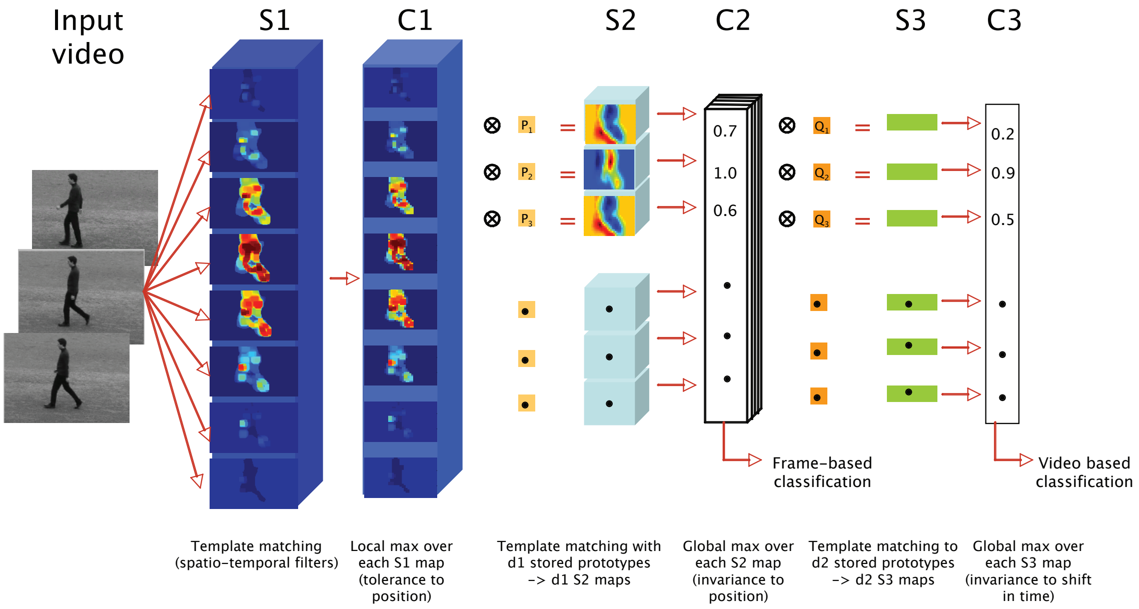

Input video
S1
C1
S2
C2
S3
C3
P1
P2
P3
0.7
1.0
0.6
Q1
Q2
Q3
0.2
0.9
0.5
Frame-based classification
Video based classification
Template matching (spatio-temporal filters)
Local max over each S1 map (tolerance to position)
Template matching with d1 stored prototypes -> d1 S2 maps
Global max over each S2 map (invariance to position)
Template matching to d2 stored prototypes -> d2 S3 maps
Global max over each S3 map (invariance to shift in time)

Figure 3-3: Sketch of the model for action recognition (see text for details).

vector is computed for each frame, but the computation in the S_1 stage already incorporates information in the temporal dimension. Another one is the C_3 output computed for each video. The classification is done with a support vector machine (SVM). The model is illustrated in Figure 3-3

3.4 Representation stage

A general implementation and function of the model is described in Chapter 2, here we experimented with different implementations and parameter settings in the S_1 and S_2 stage, with the goal of building a robust feature representation of actions. On top of the $S_1 \rightarrow C_1 \rightarrow S_2 \rightarrow C_2$ stages, we also experimented with adding one more simple and complex layer $S_3 \rightarrow C_3$ to increase the specificity and invariance in the time domain.

3.4.1 Motion-direction sensitive S_1 units

In order to extend HMAX from representing objects to representing actions, we experimented with 3 types of motion-direction sensitive S_1 units: gradient based units, optical flow based units, and space-time oriented filters [177], which have been shown to be good models for the motion-sensitive simple cells in the primary visual cortex [117].

Space-time gradient-based S_1 units: This type of S_1 units is based on space and time gradients, which were used for instance, in the system by Zelnik-Manor & Irani [223]. The spatial gradients along the x and y axis for each frame are denoted as I_x and I_y, and the temporal gradient of adjacent frames as I_t. Motivated by the optical-flow algorithms that are based on the constant-brightness assumption, we consider two types of S_1 units: $|I_t/(I_x + 1)|$ and $|I_t/(I_y + 1)|$. The absolute value is taken to make features invariant to contrast reversal.

Optical flow based S_1 units: We also experimented with S_1 units that compute responses based on the optical flow of the input image sequence using the Lucas & Kanade's algorithm [95]. We denote θ and ν, the direction and magnitude of the optical flow at each pixel position at the current frame. As in [52], the response of a S_1 unit was obtained by applying the following equation:

$$b(\theta, \theta_p) = \{\frac{1}{2}[1 + cos(\theta - \theta_p)]\}^q \times exp^{(-|\nu - \nu_p|)}, \tag{3.1}$$

where θ_p is the preferred direction of the S_1 unit and ν_p is the preferred speed. We use totally 8 types of units with 4 preferred directions $\theta_p = 0^o, 90^o, 180^o, 270^o$ and two preferred speeds, an intermediate one ($\nu_p = 3$) and a higher one ($\nu_p = 6$ pixels / frame). The constant q, which controls the bandwidth of the direction tuning curve, is set to $q = 2$ as in [52, 24].

Space-time oriented S_1 units: These units constitute the most direct extension to the object recognition systems by Serre *et al.* [171, 119]. In [171, 119], S_1 units are $2D$ Gabor filters at multiple orientations. A natural way to extend these filters to the analysis of motion is to add a third dimension (temporal dimension) to their receptive fields.

Several specific implementations of such motion-direction selective cells have been proposed [52]. Here we used the implementation by Simoncelli & Heeger which uses (3^{rd}) derivatives of Gaussians [177]. These filters have been shown to agree quantitatively with the RF profiles of some direction selective simple cells in the primary visual cortex [117]. As for the optical flow based S_1 units, we used 8 space-time filters tuned to 4 directions

$(0^o, 90^o, 180^o, 270^o)$ and 2 speeds (3 and 6 pixels/ frame). The receptive field size of each S_1 unit is 9 pixels $\times 9$ pixels $\times 9$ frames (a typical video frame rate is 25fps). The filter outputs were half-wave rectified.

3.4.2 C_1 stage

At the C_1 stage, a local max is computed over an 8×8 grid of S_1 units with the same selectivity. C_1 units are computed at every 4 pixels, which is designed to be half the size of the pooling region.

3.4.3 S_2 templates of intermediate complexity

A S_2 template is a collection of responses of spatially neighboring C_1 units that are tuned to all possible selectivity, it could be treated as motion patterns learned from training videos. A template's size is n_c (pixels) $\times$ n_c (pixels)$\times$ n_l (types). n_l is the number of S_1/C_1 selectivity, the value is $2, 8, 8$ for the gradient based, optical flow based, and space-time oriented S_1 units, respectively. n_c is the number of spatially neighboring C_1 units converging into a S_2 unit. To include motion patterns (encoded as C_1 responses) in a range of spatial scales, we use four sizes, $n_c = 4, 8, 12, 16$ (pixels).

To obtain the S_2 templates from all types of actions, we randomly extract 500 patches from C_1 outputs of training videos for each action category and for each template size. This leads to 2,000 stored templates per action category and a total number of templates $d_1 = 10,000 - 18,000$ for a dataset containing 5-9 action categories.

3.4.4 S_2 template matching

Recently, Mutch & Lowe showed that S_2 features can be sparsified leading to a significant gain in performance [119] on standard object recognition datasets (see also [148]). Motivated by this finding, we experiment with the dense as well as sparse S_2 features.

In our experiments (Section 3.6.4), we compare two alternative S_2 template representations: the dense Euclidean distance adapted from [171] and the sparse normalized dot-product suggested by [119]. For the dense template [171], the template matching is

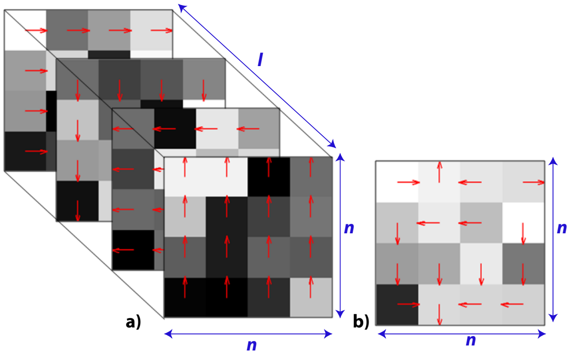


Figure 3-4: An illustration of a dense S_2 template [171] (a) *vs.* a sparse S_2 template [119] (b).

computed over all $n_l \times n_c^2$ coefficients of the template. For the sparse template, only the strongest coefficients among all the S_1/C_1 types are stored for each pixel location of the template. Thus only n_c^2 sparse coefficients are stored for matching. The difference between dense and sparse S_2 features is illustrated in Figure 3-4.

3.4.5 C_2 feature selection with zero-norm SVM

In the same publication of Mutch & Lowe [119], they also showed that applying a simple feature selection technique to the C_2 features can lead to an efficient system which can perform better with less features. Here we experiment with the zero-norm SVM [215] feature selection technique.

The SVM classifier tries to optimize the objective function:

$$||\mathbf{w}||_0 + \mathbf{C}\sum_{i=1}^{N} \zeta_i, \text{ such that } (\mathbf{w}^T\mathbf{x_i} + b) > 1 - \zeta_i \tag{3.2}$$

The zero norm $||\mathbf{w}||_0$ here indicates the count of the features used. In practice this is done in multiple rounds. At each round a SVM classifier is trained on a pool of C_2 vectors randomly selected from the training set, and each dimension of the vectors is then re-weighted using the weights of the hyperplane computed by the SVM. Typically this leads to sparser C_2 vectors at each round. As described in Chapter 2, each value of a C_2 vector corresponds to the best matching with a motion template. At each round, a set of features (motion templates) that receives weights higher than some threshold is selected. In Section 3.6.4, we compare the performance at each round using the selected features.

3.4.6 S_3 and C_3 stage

The perception of actions is selective to temporal order: randomization of the temporal order usually destroys the perception of the movement. While the C_2 vectors achieve a high degree of selectivity to complex motion patterns and spatial invariance, they lack selectivity to temporal-orders and invariance to shifts in time. We experimented with the addition of a S layer (S_3) that adds the temporal order selectivity, and a C layer C_3 that adds the temporal invariance to the features representations.

A S_3 template encodes the temporal order of an action by collecting C_2 vectors computed from 7 consecutive frames, resulting in a size d_1 dimensions × 7 frames (d_1 is the dimension of a C_2 vector as well as the number of S_2 templates). Similarly to the sparse features described in section 3.4.4, here for each C_2 vector, only the top 50% coefficients with largest values are stored for matching. We select $d_2 = 300$ S_3 templates at random frames from random training clips.

For each frame of an input video, we perform a template matching between d_2 S_3 templates and 7 consecutive C_2 vectors centering at current frame, the S_3 response is computed according to Equation 2.2. The C_3 response is then computed as maximum S_3 response over the whole duration of the video for each S_3 template. This results in d_2 C_3 responses, this d_2 dimensional vector is then used to represent the whole clip. The C_3 representation is selective to the temporal order and invariant to the shifts in time, meaning, in order for two video sequences to have similar C_3 vectors, they have to have the same temporal order, but aren't necessary aligned in time.

3.5 Classification stage

The final classification stage is a linear multi-class SVM classifier trained using the all-pairs method. We experimented with classifying outputs of the C_2 stage and the C_3 stage.

When passing C_2 features into the classifier, each training point is a C_2 vector computed for a random frame, the label of the point is the action category frame which the frame is sampled. For a test clip, we thus obtain a classification label for each frame as well. The predicted label for the entire clip was obtained by a majority voting across predictions for all its frames.

When passing C_3 features into the classifier, each training point is a C_3 vector computed for a random clip, the label of the point is the action category of the clip. For a test clip, a single prediction is obtained for the entire clip.

3.6 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). For each dataset, the system's performance is the average of 5 random splits. The KTH Human Set and the Weizmann Human Set were recorded under static background and the actions are whole-body motion. The UCSD Mice Behavior Set is the most challenging one because the actions of the mice are minute (see Figure 3-5 for examples) and because the background of the video is typically noisy (due to the litter in the cage).

3.6.1 Preprocessing

Instead of computing features on a whole frame, we speed-up the experiment by computing features on a bounding box surrounding the moving subject. For the KTH human and UCSD mice datasets we used the openCV GMM background subtraction technique based on [184]. In short, a mixture of Gaussians are modeled at each spatial (pixel) location over the entire clip to identify whether the current pixel belongs to the foreground. For each frame, we compute the center of all the foreground pixels, denoted as $c(x, y)$, and then compute a bounding box (full height and half the width of the frame) centering at $c(x, y)$. For the Weizmann Human dataset, the bounding boxes were extracted directly from the foreground masks provided with the dataset. Figure 3-6 shows snapshots of the actions in the three datasets.

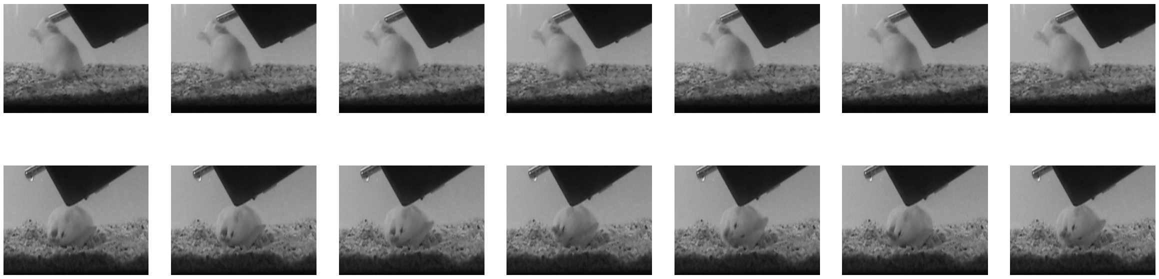


Figure 3-5: Sample videos from the mice dataset (1 out 10 frames displayed with a frame rate of 15 Hz) to illustrate the fact that the mice behavior is minute.

3.6.2 Datasets

KTH human: The KTH Human Set [165] contains 600 clips (6 types of human actions × 25 human subjects × 4 recording conditions). The six types of human actions are walking, jogging, running, boxing, hand waving and hand clapping. These actions are performed several times by twenty-five subjects in four different conditions: outdoors (s_1), outdoors with scale variation (s_2), outdoors with different clothes (s_3) and indoors with lighting variation (s_4). The sequences are about 4 seconds in length and were down-sampled to a spatial resolution of 160×120 pixels. We split the dataset as: actions of 16 randomly drawn subjects for training and that of the remaining 9 subjects for testing.

Weizmann human: The Weizmann Human Set [9] contains 81 clips (9 types of human actions × 9 human subjects) 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 bending. The sequences are about 4 seconds in length and with spatial resolution of 180×144 pixels. We split the dataset as: actions of 6 randomly drawn subjects for training and of the remaining 3 subjects for testing. The size of a subject in this dataset is about half the size of a 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 Set [36] contains 435 clips (5 actions of 7 mice subjects, each being recorded at different points of time in a day such that multiple occurrences of actions within each subset vary substantially). There are five actions in

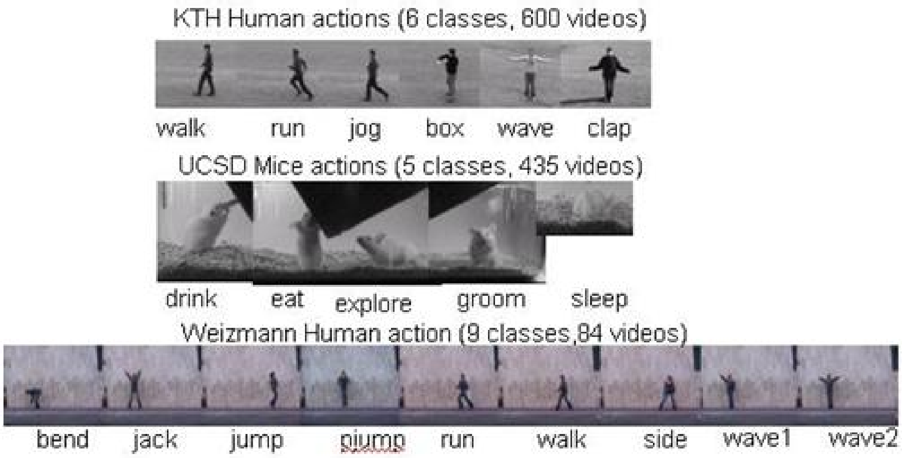

KTH Human actions (6 classes, 600 videos)
walk
run
jog
box
wave
clap
UCSD Mice actions (5 classes, 435 videos)
drink
eat
explore
groom
sleep
Weizmann Human action (9 classes,84 videos)
bend
jack
jump
pjump
run
walk
side
wave1
wave2

Figure 3-6: Illustration of KTH, Weizmann, and UCSD dataset.

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 Figure 3-5 for examples) and second the background of the video is typically noisy (due to the litter in the cage). Each split, we randomly choose 4 subsets for training and the remaining 3 subsets for testing.

3.6.3 Benchmark algorithms

For benchmark we use the algorithm by Dollar *et al.* which has been compared favorably to several other approaches [223, 38, 121] on the KTH human and UCSD mice datasets described earlier. In short, the approach detects interest points in the spatio-temporal domain and extracts *cuboids*, *i.e.* spatio-temporal windows of pixel values, around each point detected. These cuboids are further matched to a dictionary of cuboid-prototypes learned from sequences in the training set. Finally, a vector description is obtained by computing the histogram of cuboid-types of each video, and a SVM classifier is used for classification. The code for was graciously provided by Piotr Dollar.

3.6.4 Results

We have studied several aspects and design alternatives for the system. First we show that the 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 compare 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. For computational reason the performance reported is based on a single split of the KTH dataset. In the first iteration, all $12,000$ motion patterns extracted from the training set were used to compute the C_2 features. In each of the following iteration, only features (motion patterns) with a weight $|w_i| > 10^{-3}$ were selected.

		1	5	10	15	20
$s1$	No. feat.	12000	3188	250	177	158
	accu.	91.7	91.7	89.3	88.9	90.3
S_2	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 3.1: Selecting features: System performance for different number of selected C_2 features at rounds 1, 5, 10, 15 and 20 (see text for details).

Table 3.1 compares the performance of each round. In agreement with previous results on object recognition [119], 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 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 the category of the action. In the following, feature selection was performed on the C_2 features for all the results reported.

Comparing different C_2 feature-types with baseline

Table 3.2 gives a comparison between C_2 features computed from three S_1/C_1 types: 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 [171] *vs.* sparse [119] C_2 features (see Section 3.3 for details). $s_1, \dots s_4$ corresponds to the different conditions of the KTH database (see Section 3.6.2).

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 [119] for object recognition.

	[36]	GrC_2	OfC_2	StC_2
KTH s_1	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 s_2	68.3	86.0 / **86.8**	80.7 / **83.1**	81.3 / **86.1**
s.e.m. S_2	±2.1	±3.9 / ±3.9	±4.0 / ±3.9	±4.2 / ±4.6
KTH s_3	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 s_4	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 3.2: Comparison between three types of C_2 features (gradient based GrC_2, optical flow based OfC_2 and space-time oriented StC_2) and between dense *vs.* sparse C_2 features. In each column, the number on the left *vs.* the right corresponds to the performance of dense [171] *vs.* sparse [119] C_2 features. *Avg* is the mean performance across the 4 conditions $s_1, \dots s_4$. Below the performance on each dataset, we indicate the standard error of the mean (s.e.m.).

Comparing different C_3 feature-types

We have started to experiment with high-level C_3 features. Table 3.3 shows some initial results with C_3 features computed from three S_1/C_1 types: gradient based GrC_3, optical flow based OfC_3 and space-time oriented StC_3 features. In each column, the number to the left *vs.* the right corresponds to the performance of C_3 features computed from dense [171] *vs.* sparse [119] C_2 features. For the KTH dataset, the results are based on the performance on a single split. 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 set (UCSD).

	GrC_3	OfC_3	StC_3
KTH s_1	**92.1** / 91.3	84.8 / **92.3**	89.8 / **96.0**
KTH s_2	81.0 / **87.2**	80.1 / **82.9**	81.0 / **86.1**
KTH s_3	89.8 / **90.3**	84.4 / **91.7**	80.6 / **89.8**
KTH s_4	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 3.3: Comparison between three types of C_3 features (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 [171] *vs.* sparse [119] C_2 features. *Avg* is the mean performance across the 4 conditions $s_1, \ldots s_4$. Below the performance on each dataset, we indicate the standard error of the mean (s.e.m.).

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 [184]. This allows us to discard about 50% of the 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.

3.7 Conclusion

We have applied a biological model of motion processing to the recognition of human and animal actions. 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 also been suggested [52] that another part of the visual system, the ventral stream of the visual cortex, involved with the analysis of shape may also be important for the recognition of motion (consistent with recent work in computer vision [121] which has shown the benefit of using shape features in addition to motion features for the recognition of actions). Future work will extend the present approach to integrate shape and motion information from the two pathways. Another extension is to incorporate top-down effects, known to play an important role for the recognition of motion (*e.g.* [174]), to the present feedforward architecture.

Chapter 4

HMDB: A Large Video Database for Human Motion Recognition

This chapter is currently under submission [82].

Abstract

With nearly one billion online videos viewed everyday, an emerging new frontier in computer vision research is recognition and search in video. While much effort has been devoted to the collection and annotation of large scalable static image datasets containing thousands of image categories, human action datasets lack far behind. Current action recognition databases contain on the order of ten different action categories collected under fairly controlled conditions. State-of-the-art performance on these datasets is now near ceiling and thus there is a need for the design and creation of new benchmarks. Here we collected the largest action video database to-date with 51 action categories and around 7,000 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube. We use this database to evaluate the performance of two representative computer vision systems for action recognition and explore the robustness of these methods under various conditions such as camera motion, viewpoint, video quality and occlusion.

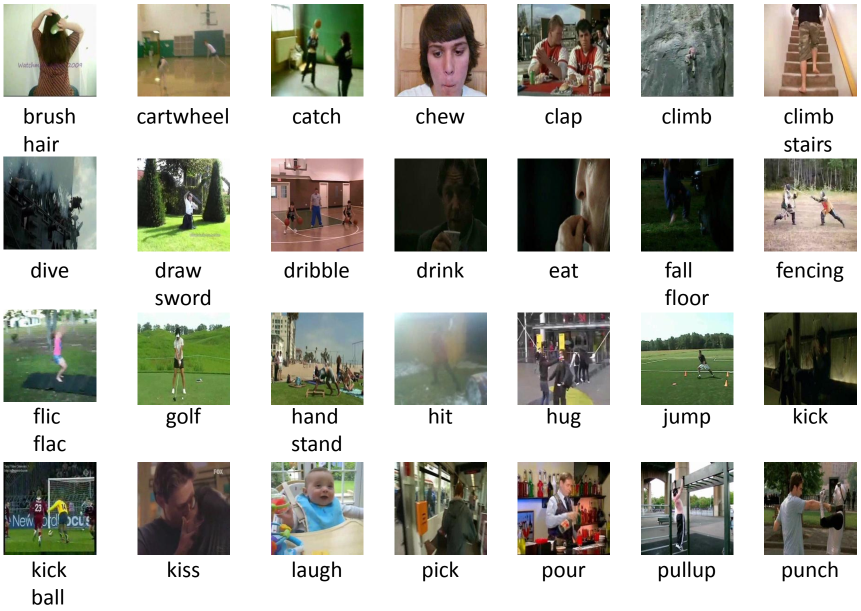


Figure 4-1: Illustrations of the 51 actions in the HMDB51, part I.

4.1 Introduction

With several billions of videos currently available on the internet and approximately 24 hours of additional video uploaded to YouTube every minute, there is an immediate need for robust algorithms that could help organize, summarize and retrieve this massive amount of data. While much effort has been devoted to the collection of realistic internet-scale static image databases [159, 193, 194, 217, 35, 41], current action recognition datasets lag far behind. The three most popular benchmark databases (*i.e.* KTH [165], Weizmann [9] and the IXMAS [213]) datasets contain around 6-11 actions each (see Table 4.1 for a comparison between existing action recognition datasets). These databases are not quite representative of the richness and complexity of real-world action videos as they are fairly well constrained in terms of illumination and camera position. A typical video clip contains a single (staged) actor with no occlusion and very limited clutter.

Recognition rates on these datasets tend to be very high. For instance, a recent survey of action recognition system comparison [214] reported that 12 out of the 21 systems tested perform better than 90% on the KTH dataset. For the Weizmann dataset, 14 out of 16 tested systems perform at 90% or better, 8 out 16 better than 95%, and 3 out of 16 scored a perfect

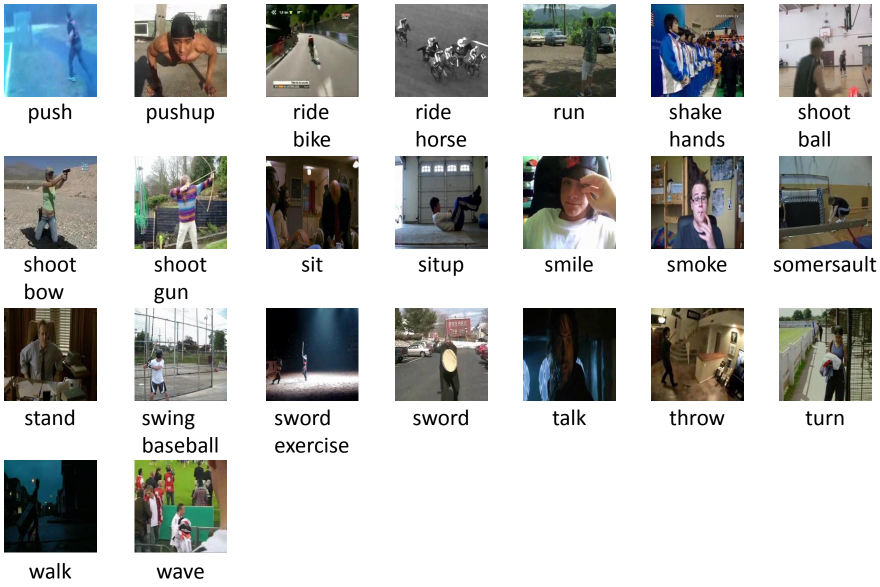


Figure 4-2: Illustrations of the 51 actions in the HMDB51, part II.

100% recognition rate. In this context, we describe an effort to advance the field with the design and collection of a large video database containing 51 distinct action categories, dubbed the Human Motion DataBase (HMDB51), that tries to capture the richness and complexity of human actions (see Figure 4-1, 4-2).

The Hollywood2 and UCF50 datasets are two examples of a recent effort to try to build more realistic action recognition datasets by considering video clips taken from HollyWood movies and YouTube. These datasets are more challenging due to large variations in camera motion, object appearance, changes in the position, scale and viewpoint of the actor(s) as well as cluttered background. The UCF50 and a preliminary version UCF Sports Action Dataset as well as a recently introduced Olympic sports dataset [122] contain mostly sports videos from YouTube. These types of actions are relatively unambiguous (as a result of searching for specific titles on YouTube), and are highly distinguishable from shape cues alone (such as the raw positions of the joints or the silhouette extracted from single frames).

To demonstrate this point, we conducted a simple experiment: Using Amazon mechanical Turk, we manually annotated stick-figures from 5 random clips for each of the 13 action categories on the UCF YouTube Sport Dataset, as illustrated in Figure 4-3. Using

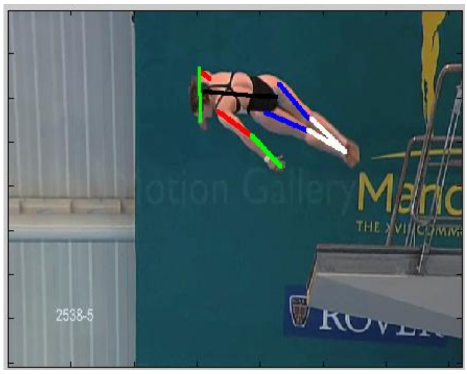


Figure 4-3: An stick-figure annotated on YouTube Action Dataset. The nine line segments correspond to the two upper arms (red), the two lower arms(green), the two upper legs (blue), the two lower legs (white), and the body trunk (black).

a leave-one-clip-out procedure, classifying the raw joint locations from single frames lead to a recognition rate above 98% (chance level 8%). This would suggest that kinematics does not play any role in the recognition of biological motion and does not seem realistic of real-world scenarios. For instance, using a point-light walker stimulus, Johansson famously demonstrated decades ago that joint kinematics play a critical role for the recognition of biological motion by human observers [76].

We conducted a very similar experiment on the proposed HMDB51 database described in this paper where we drew from 10 action categories similar to those used in the UCF (*e.g.* climb, climb-stairs, run, walk, jump, *etc.* .) and manually annotated the joint locations for a set of over 1,100 random clips. The accuracy reached by a classifier using the joint location computed from single frames as inputs reached only 35% this time (chance level 10%) and performed below the level of performance of the same classifier, using instead motion cues (*e.g.* the HOG/HOF features described below performed at 54% on the HMDB and 66% on the UCF50). Such a dataset may thus be a better indicator of the capability of real-world action recognition systems and the relative contributions of motion *vs.* shape cues, which are known to play a critical role in the recognition of actions in biological vision [191].

4.2 Background: existing action datasets

This section and Table 4.1 summarize the existing action datasets. Also see a recent paper for a similar summarization [141].

KTH Action Dataset The KTH Action Dataset [165] contains 600 clips (6 types of human actions × 25 human subjects × 4 recording conditions). The six types of human actions are walking, jogging, running, boxing, hand waving and hand clapping. These actions are performed several times by twenty-five subjects in four different conditions: outdoors, outdoors with zooming, outdoors with different clothes and indoors with lighting variation. The sequences are about 4 seconds in length and were down-sampled to a spatial resolution of 160×120 pixels. The backgrounds are relatively static with only slight camera movement.

Weizmann Action Dataset The Weizmann Action Dataset [9] contains 81 clips (9 types of human actions × 9 human subjects) 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 bending. The sequences are about 4 seconds in length and with spatial resolution of 180×144 pixels. The backgrounds are static and foreground silhouettes are included in the dataset. The viewpoint is also static.

INRIA XMAS multi-view dataset The IXMAS dataset [213] contains 11 actions captured from five viewpoints, each performed 3 times by 10 actors (5 males / 5 females). The 11 actions are check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, and pick up. The backgrounds illumination settings are static. Silhouettes and volumetric voxel representations are part of the dataset.

UCF Sports Action Dataset The UCF Sports Action Dataset [156] contains a set of actions from various sports featured on broadcast television channels such as the BBC and ESPN. The 9 actions in this dataset include diving, golf swinging, kicking, lifting, horse-back riding, running, skating, swinging a baseball bat, and pole vaulting. The dataset contains over 200 video sequences at a resolution of 720×480 pixels. Bounding boxes of the human figure are provided with the dataset. For most action classes, there is considerable variation in action performance, human appearance, camera movement, viewpoint, illumination and background.

UCF YouTube Action Dataset The UCF YouTube Action Dataset [91] contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The clips are collected from YouTube and contain variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. For each action category, there are 25 groups of videos with more than 4 clips per group. The clips in the same group are performed by the same actor and have similar background and viewpoint.

Hollywood human action dataset & Hollywood2 The Hollywood human action dataset [86] contains eight actions (answer phone, get out of car, handshake, hug, kiss, sit down, sit up and stand up), extracted from Hollywood movies. The second version of the dataset [99] includes four additional actions (drive car, eat, fight, run) and an increased number of clips. There is a huge within class variation, and occlusions, camera movements and dynamic backgrounds make this dataset challenging. Most of the samples are at the scale of the upper-body but some show the entire body or a close-up of the face.

Olympic Sports Dataset The Olympic Sports Dataset [122] contains 50 YouTube videos from each of 16 classes: high jump, long jump, triple jump, pole vault, discus throw, hammer throw, javelin throw, shot put, basketball layup, bowling, tennis serve, platform (diving), springboard (diving), snatch (weightlifting), clean and jerk (weightlifting) and vault (gymnastics). The clips contain occlusions and camera movements and the motion is the composition of many short actions. For instance, sequences from the long-jump action class, show an athlete first standing still, in preparation for his/her jump, followed by running, jumping, landing and finally standing up.

Table 4.1: Comparison between existing datasets.

Dataset	Year	Actions	Clips
KTH	2004	6	100
Weizmann	2005	9	9
IXMAS	2006	11	33
Hollywood	2008	8	30-129
UCF Sports	2009	9	14-35
Hollywood2	2009	12	61-278
UCF YouTube	2009	11	100
Olympic	2010	16	50
UCF50	2010	50	min. 100
HMDB51	2011	51	min. 101

UCF50 Dataset The UCF50 Dataset contains 50 action categories collected from YouTube: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot,Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo. For each action category, there are 25 groups of videos with more than 4 clips per group. The clips in the same group are performed by the same actor and have similar background and viewpoint. The clips are collected from YouTube and contain variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc.

4.3 The Human Motion DataBase (HMDB51)

4.3.1 Database collection

It has been estimated that there are over 1,000 human action categories. In order to isolate human actions that are representative of everyday actions, we first asked a group of students to watch videos from various internet sources and Hollywood movies while using a subtitle annotation tool to annotate any segment of these videos that they deemed to represent a single non-ambiguous human action. Students were asked to consider a minimum quality standard (*i.e.* a single action per clip, a minimum of 60 pixels in height for the main actor, minimum contrast level, minimum action length of about 1 second and acceptable compression artifacts). Students considered videos from three sources: digitized movies available on the internet, public databases such as the Prelinger archive, and YouTube and Google videos. A first set of annotations was thus generated in this way with over 60 action categories. To further guarantee that we would be able to populate all action categories with at least 101 different video clips we considered the top 51 action categories and further asked students to specifically look for these types of actions.

The actions categories can be grouped in five types: a) General facial actions: *smile, laugh, chew, talk*; b) Facial actions with object manipulation: *smoke, eat, drink*; c) General body movements: *cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave*; Body movements with object interaction: *brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw*; Body movements for human interaction: *fencing, hug, kick someone, kiss, punch, shake hands, sword fight*.

4.3.2 Annotations

In addition to action category labels, each clip was annotated with meta-data to allow for a more precise evaluation of the limitation of current computer vision systems. This meta-data contains six additional fields for the following properties:

- visible body parts / occlusions: head, upper body, lower body, full body.
- camera motion : moving or static
- camera view point relative to the actor : front, back, left, or right
- the number of people involved in performing the action: single, two, or multiple

- video quality ranging : high (*i.e.* detailed visual elements such as the fingers and eyes of the main actor identifiable through most of the clip, limited motion blur and compression artifacts), medium (*i.e.* larger body parts like the upper and lower arms and legs identifiable through most of the clip), or low (*i.e.* even larger body parts not identifiable due in part to the presence of motion blur and compression artifacts).

4.3.3 Training and test set generation

For evaluation purposes, three distinct training and test splits were generated from the database. The sets were built to ensure that the same video source could not be used for both training and testing and that the relative proportion of possible conditions such as camera position, video quality, motion, *etc.* (see above) would be balanced across the training and test sets. For example, each action category in our dataset contains 100 clips and instead of randomly drawing 70/30 clips for training/testing, we selected a particular set of 70/30 clips such that they make up 70%/30% of the high quality clips, 70%/30% of the bad quality clips, 70%/30% of the front-view camera, 70%/30% of the side-view camera, and so on with the added constraint that clips in the training and test set could not come from the same source.

To do so, we implemented a very simple constraint satisfaction algorithm to select the subsets of clips that best satisfy these criteria from a very large number of randomly generated splits. To ensure that the various splits were not too similar, we implemented a greedy approach where we first picked the split with the best parameter distribution and subsequently chose the second and third best splits that would be least correlated as measured by a normalized Hamming distance. Note that because different numbers of clips under various conditions might have been selected from the different sources it is not always possible to find an exact solution but we found that in practice the simple approach described above provided reasonable splits.

4.3.4 Videos normalization

The original video sources used to extract the action video clips varied in size and frame rates. To ensure consistency across the database, we resized all clips to a height of 240 pixels. The width of the clips was scaled accordingly so as to maintain the original aspect ratio. We further normalized all video frame rates to ensure a frame rate of 30 fps for all clips. All clips were compressed using the *DviX 5.0* codec with the *ffmpeg* video library.

4.3.5 Videos stabilization

One significant challenge associated with the use of video clips extracted from real-world videos is the potential presence of significant camera/background motion (about 2/3 of the clips in our database). Such camera motion is assumed to interfere with the local motion computation and should potentially be corrected. Video stabilization is thus a key pre-processing step. We used a simple algorithm for camera motion based on standard image stitching techniques to align successive frames according to the camera motion. In this approach, a background plane is estimated by first detecting and then matching salient features between adjacent frames. Correspondences are then computed using a distance measure that includes both the absolute pixel differences and the Euler distance of the corner points. Points with a minimum distance are then matched and the RANSAC algorithm is used to estimate the geometric transformation between all neighboring frames (independently for every pair of frames). Using this estimate, the single frames are warped and combined to achieve a stabilized video. We visually inspected a large number of the resulting stabilized clips and found that the corresponding approach works surprisingly well. For the evaluation of the action recognition algorithms, all tests were conducted with both the original clips as well as with the stabilized videos. An example of a stabilized video is shown in Figure 4-4

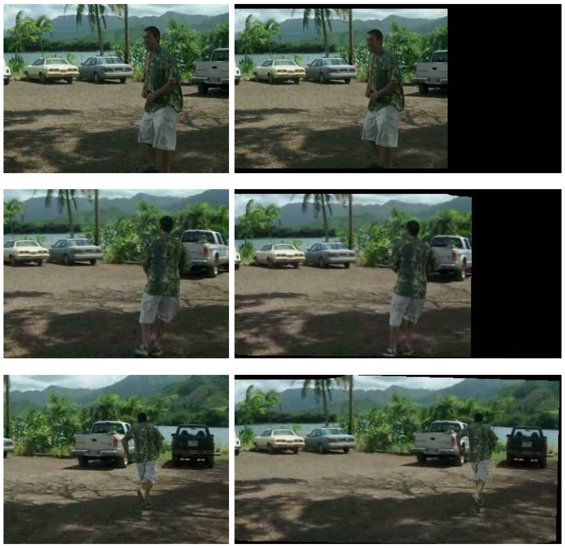


Figure 4-4: Examples of videos stabilized over 50 frames.

4.4 Comparison with other action datasets

Here we compare the HMDB51 database to other similar databases (Hollywood, UCF sports, Hollywood2 and UCF50). To assess the discriminative power of various low-level cues on these datasets, we extracted a number of very simple visual features, which, in principle should not be predictive of a high-level action category. This included a measure of color based on the mean color in HSV space computed for each frame over a 12×16 grid arrangement. We also tried to use a combination of color and gray value information as well as PCA to further reduce the feature dimension. Last we computed a gist vector for every frame (*i.e.* a coarse orientation-based representation of an image that has been shown to capture well the contextual information about objects in natural scenes and shown to perform quite well on a variety of recognition tasks, see [127]). We also benchmark the performance using one of the state-of-the-art action recognition system (HOG/HOF) [86] that uses motion features extracted from local spatio-temporal volumes to do classification. Table 4.4 shows the results.

Results obtained by classifying these very simple features suggest that the UCF Sports dataset is closer to a scene recognition dataset than an action recognition dataset as both color and low-level global scene information is more predictive than mid-level spatio-temporal features. While we were expecting color cues to be predictive of the action category on a sport dataset (*e.g.* green grass for ball games, blue water for swimming sports, white snow for skiing, *etc.*), it seems that the problem remains in the UCF50 where the gist descriptors achieve almost as well as the HOG/HOF. This might be due to low-level biases (*e.g.* preferred vantage points and camera positions for amateur directors) for videos on YouTube. In comparison the performance of these low-level cues is much reduced in comparison to the mid-level motion descriptors and certainly reveal that databases generated from YouTube do not capture the large array of appearances of action videos present in Hollywood movies and HMDB51.

Table 4.2: Mean recognition performance of low-level shape/color cues for different action databases.

	Color	Color+ PCA	Color+ Gray	Color+ Gray+ PCA	Gist	HOG/ HOF
Hollywood	21.4%	21.4%	19.9%	26.9%	25.9%	30.3%
UCF Sports	82.7%	84.3%	89.6%	89.6%	90.0%	78.1%
Hollywood2	13.9%	15.7%	18.9%	16.1%	23.8%	45.2%
UCF50	34.6%	39.2%	44.2%	41.9%	55.5%	56.4%
HMDB51	6.1%	5.9%	8.7%	8.8%	14.3%	20.4%

4.5 Benchmark systems

To evaluate the discriminability of our 51 action categories we focus on the class of algorithms for action recognition based on the extraction of local space-time information from videos, which have become the dominant trend in the past five years [210]. Local space-time based approaches mainly differ in the implementation of the spatio-temporal filters used and in the number of spatio-temporal points sampled (dense *vs.* sparse). Wang and colleagues have grouped these descriptors into six types and evaluated their performance on the KTH, UCF sports and Hollywood2 datasets [210] and shown that Laptev's HOG/HOF descriptors performed best for the Hollywood2 and UCF sports. Because these datasets are the most similar to the proposed HMDB51 (the HMDB51 contains both Hollywood movies like the Hollywood2 dataset and YouTube videos much like the UCF sports database), we selected the algorithm by Latptev and colleagues [86] as one of our benchmarks. To expand over [210], we chose for our second benchmark the bio-inspired approach by Jhuang, Serre and colleagues [75] because it was not included in the original study by Wang *et al.* The approach uses a hierarchical architecture that was modeled after the dorsal stream of the visual cortex of the primate cortex.

Here we provide a detailed comparison between these algorithms, looking in particular at the robustness of the two approaches with respect to various nuisance factors including the quality of the video, the presence of occluders and camera motion, as well as changes in the position, scale and viewpoint of the main actor(s). In addition the range of actions included in the HMDB51 allows comparison of the two approaches over the types of actions considered (*e.g.* face vs. body motion and whether or not the action involves any interaction with an object).

4.5.1 HOG/HOF features

Local space-time features have recently become a popular video representation for action recognition. Much like their static local spatial features counterpart for the recognition of objects and scenes, they have been shown to achieve state-of-the-art performance on several standard action recognition databases [86, 210]. An extensive comparison between existing methods (feature detectors and local descriptors) for the computation of space-time features in a common experimental setup was described in [210]. We implemented a system based on one of the most commonly used system configurations using a combination of the *Harris3D* detector and the HOG/HOF descriptors. For every clip we detected 3D Harris corners and computed combinations of histograms of oriented gradients (HOG) and oriented flows (HOF) as local descriptors.

To evaluate the best code book size, we sampled 100,000 space-time interest-point descriptors from a training dataset and the k-means clustering with $k = 2,000-10,000$ was applied on the sample set. For every clip in the training set, the space-time interest-point descriptors were matched to the nearest prototype as returned by k-means clustering and a histogram was built over the index of the codebook entries. This lead to a k-dimensional feature vector where k is the number of clusters learned from k-means. This feature vector was then used as input to an SVM classifier for final classification.

As described in [86], we used a support vector machine with an RBF kernel $K(u, v) = \exp(-\gamma * |u - v|^2))$. The parameters of the RBF kernel (cost term and γ) were optimized using a greedy search with a 5-fold cross-validation on the training set. The best results for the original clips was reached for $k = 8,000$ whereas the best results for stabilized clips was at $k = 2000$ (see Section 4.6.1). To validate our re-implementation of Laptev's system, we evaluated the performance of the system on the KTH dataset and were able to reproduce the 92.1% reported in [210].

4.5.2 C2 features

Two versions of the C2 features have been described in the literature. The former corresponds to a model of the ventral stream of the visual cortex described in [171, 173] (assumed to be critically involved in the processing of shape information and invariant object recognition). The model starts with a pyramid of Gabor filters (S1 units at different orientations and scales (see [171, 173] for details). These mimic processing by the so-called simple cells in the primary visual cortex. The next stage correspond to complex cells, which pool together the activity of S1 units via a local neighborhood in both space and spatial frequency to build some tolerance to 2D transformations (translation and scale).

Next S2 maps are computed by convolution with a dictionary of features/prototypes learned by random sampling from a training set of images. Unlike the bag-of-words approach described above that uses vector quantization, the final C2 vector is obtained by computing the best match between an input image and each feature of the dictionary across all positions and scales. This final stage has been shown to account well for the properties of cells in the inferotemporal cortex, which is the highest purely visual area of the primate brain.

Building on the work described above, Jhuang *et al.* described a model of the dorsal stream of the visual cortex (thought to be critically involved in the processing of motion information in the primate brain). The model starts with spatio-temporal filters modeled after motion-sensitive cells in the primary visual cortex [177]. Just like the V1-like simple units in the model of the ventral stream described above, these units are tuned to specific

Table 4.3: Performance of the benchmark systems on the HMDB51.

System	Original clips	Stabilized clips
HOG/HOF	20.44%	21.96%
C2	**22.83%**	**23.18%**

orientations. As opposed to those in the model of the ventral stream, which respond best to static stimuli, simple units in the model of the dorsal stream have non-separable spatio-temporal receptive fields and respond best to a bar moving in a direction orthogonal to their preferred orientation.

Consistent with these models of the ventral and dorsal streams, it has been suggested that motion-direction sensitive cells and shape-tuned cells constitute two ”channels” of processing, the former projecting to the dorsal stream and the latter to the ventral stream (see [75]). In intermediate stages of the model of the dorsal stream, S2 units are now tuned to optic-flow patterns corresponding to combinations of several complex cell receptive fields (tuned to different directions of motion instead of spatial orientations in the model of the ventral stream and learned via sampling of a training set) and the final C2 vector is obtained by computing the best match between an input frame (or series of frames) and each feature in the dictionary.

4.6 Evaluation

4.6.1 Overall recognition performance

We first evaluated the overall performance of both systems on the HMDB51 (averaged over the three random splits described in Section 4.3.3). Both systems exhibited very comparable levels of performance slightly over 20% (chance level 2%). The confusion matrix for both systems is shown in Figure 4-5 and Figure 4-6. Errors seem to be randomly distributed across category labels with no apparent trends. The most surprising result is that the performance of the two systems improved only marginally after stabilization for camera motion despite the apparent effectiveness of the algorithm (as revealed by visual inspection of the stabilized videos).

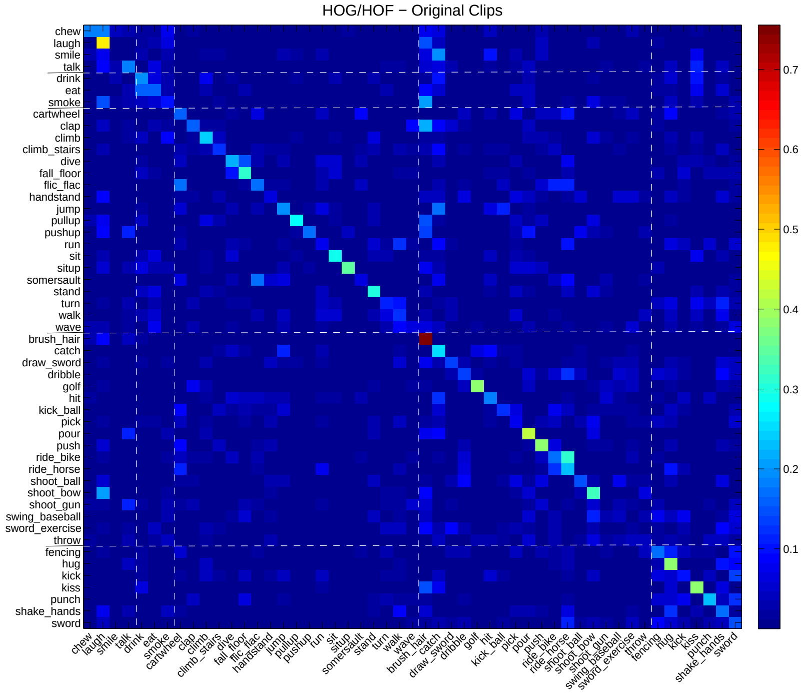


Figure 4-5: Confusion Matrix for the HOG/HOF features

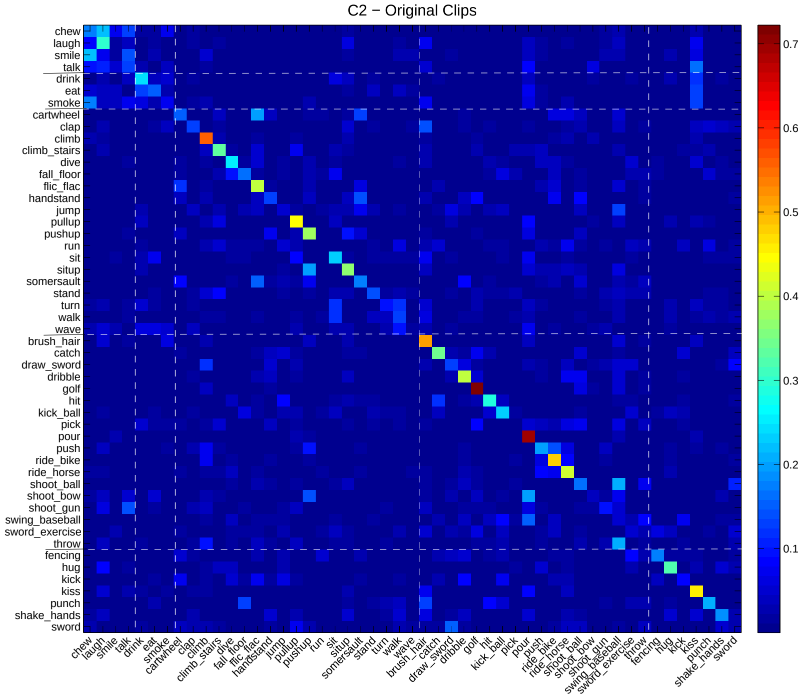


Figure 4-6: Confusion Matrix for the C2 features

Table 4.4: Mean recognition performance as a function of camera motion and clip quality.

	Camera motion		Quality		
	yes	no	low	med	high
HOG/HOF	19.84%	19.99%	17.18%	18.68%	27.90%
C2	25.20%	19.13%	17.54%	23.10%	28.62%

4.6.2 Robustness of the benchmarks

In order to asses the relative strengths and weaknesses of the two benchmark systems on the HMDB51 in the context of various nuisance factors, we broke down their performance in terms of the presence/absence of 1) occlusions and 2) camera motion, 3) viewpoint/ camera position and 4) the quality of the video clips. Surprisingly we found that, the presence/absence of occlusions and the camera position did not seem to influence performance. A major factor for the performance of the two systems, however, was the clip quality. As shown on Table 4.4, from high to low quality videos, the two systems registered a drop in performance of about 10% (from 27.90%/28.62% for the HOG/HOF vs. the C2 features down to 17.18%/17.54% respectively for the low quality clips). Camera motion was one of the factors that differentially affected the two systems: Whereas the HOG/HOF performance was stable with the presence/absence of camera motion, surprisingly, the performance of the C2 features actually improved with the presence of camera motion. We suspect that camera motion might actually improve the response of the low-level S1 motion detectors. An alternative explanation is that the camera motion by itself (*e.g.* its direction) might be correlated with the action category. To evaluate whether camera motion alone can be predictive of the performed action, we tried to classify the mean parameters returned by the video stabilization algorithm (estimated frame-by-frame motion). The result of 5.29% recognition shows that at least camera motion alone does not provide significant information.

To further investigate how various nuisance factors may affect the recognition performance of the two systems, we conducted a logistic regression analysis to try to predict whether each of the two systems will be correct *vs.* incorrect for specific conditions. The logistic regression model was built as follow: The correctness of the predicted label was used as binary dependent variable, the camera viewpoints were split into one group for front and back views (because of similar appearances; front, back =0) and one group for

Table 4.5: Results of the logistic regression analysis on the key factors influencing the performance of the two systems.

HOG/HOF			
Coefficient	Coef. est. b	p	odds ratio
Intercept	-1.60	0.000	0.20
Occluders	0.07	0.427	1.06
Camera motion	-0.12	0.132	0.88
View point	0.09	0.267	1.09
Med. quality	0.11	0.254	1.12
High quality	0.65	0.000	1.91

C2			
Coefficient	Coef. est. b	p	odds ratio
Intercept	-1.52	0.000	0.22
Occluders	-0.22	0.007	0.81
Camera motion	-0.43	0.000	0.65
View point	0.19	0.009	1.21
Med. quality	0.47	0.000	1.60
High quality	0.97	0.000	2.65

side views (left, right = 1). The occluded condition was split into full body view (=0) and occluded views (head, upper or lower body only =1). The video quality label was converted into binary variables for medium and high quality. Labels 10, 01 and 00 thus corresponded to a high, medium and low quality video respectively.

The estimated β coefficients for the two systems are shown in Table 4.5. The largest factor influencing performance for both systems remained the quality of the video clips. On average the systems were predicted to be nearly twice as likely to be correct on high vs. medium quality videos. This is the strongest influence factor by far. However this regression analysis also confirmed the counterintuitive effect of camera motion reported earlier whereby camera motion either lead to stable or improved performance. Consistent with the previous analysis based on error rates, this trend is only significant for the C2 features. The additional factors, occlusion as well as camera view point, did not have a significant influence on the results of the HOG/HOF approach.

4.6.3 Shape vs. motion information

The role of shape *vs.* motion cues for the recognition of biological motion has been the subject of an intense debate. Computer vision could provide critical insight to this question

Table 4.6: Average performance for shape vs. motion cues.

HOG/HOF	HOGHOF	HOG	HOF
Original	**20.44**%	15.01%	17.95%
Stabilized	21.96%	15.47%	**22.48**%
C2	Motion+Shape	Shape	Motion
Original	**22.83**%	13.40%	21.96%
Stabilized	**23.18**%	13.44%	22.73%

as various approaches have been proposed that rely not just on motion cues like the two systems we have tested thus far but also on single-frame shape-based cues, such as posture [221] and shape ([165, 121]), as well as contextual information [99].

We here study the relative contributions of shape *vs.* motion cues for the recognition of actions on the HMDB51. We compared the HOG/HOF descriptor with the recognition of a shape-only HOG descriptor and a motion-only HOF descriptor. We also contrasted the performance of the previously mentioned motion-based C2 feature to those of a shape-based C2 descriptor. Table 4.6 shows a comparison between the performance of the various descriptors.

In general we find that shape cues alone perform much worse than motion cues alone and that their combination tend to improve recognition performance very moderately (the effect seems to be stronger for the original *vs.* stabilized clips). An earlier study [164] suggested that ”shape is enough to recognize actions”. The results described above suggest that this might be true for simple actions as is the case for the KTH database but motion cues do seem to be more powerful than shape cues for the recognition of complex actions like the ones on the HMDB51.

4.7 Conclusion

We described an effort to advance the field of action recognition with the design of what is, to our knowledge, currently the largest action recognition database. With currently 51 action categories and a little under 7,000 video clips, the proposed database is still far from capturing the richness and the full complexity of video clips commonly found on the internet. However given the level of performance of representative state-of-the-art computer vision algorithms (*i.e.* about 25% correct classification with chance level at 2%), this initial database is arguably a good place to start (performance on the CalTech-101 database for object recognition started around 16% [44]). Furthermore our exhaustive evaluation of two state-of-the-art systems suggest that performance is not significantly affected over a range of factors such as camera position and motion as well as occlusions. This suggests that current methods are fairly robust with respect to these low-level video degradations but remain limited in their representative power in order to capture the complexity of human actions.

Chapter 5

A Vision-based Computer System for Automated Home-Cage Behavioral Phenotyping of Mice

A preliminary version of this chapter has been firstly published as an abstract at Neuroscience [74] and a technical report at MIT [170] in 2009. A complete version is published as a journal paper in Nature communications [73] and a short version is published as a conference paper in Measuring Behavior [72] in 2010. This work has also been presented at the workshop of ”Visual Observation and Analysis of Animal and Insect Behavior” in 2010 and CSHL (Cold Spring Harbor Laboratory) conference on ”Automated Imaging & High-Throughput Phenotyping” in 2010.

Abstract

Neurobehavioural analysis of mouse phenotypes requires the monitoring of mouse behavior over long periods of time. In this study, we describe a trainable computer vision system enabling the automated analysis of complex mouse behaviors. We provide software and an extensive manually annotated video database used for training and testing the system. Our system performs on par with human scoring, as measured from ground-truth manual annotations of thousands of clips of freely behaving mice. As a validation of the system, we characterized the home-cage behaviors of two standard inbred and two non-standard mouse strains. From these data, we were able to predict in a blind test the strain identity of individual animals with high accuracy. Our video-based software will complement ex-

isting sensor-based automated approaches and enable an adaptable, comprehensive, high-throughput, fine-grained, automated analysis of mouse behavior.

5.1 Introduction

Automated quantitative analysis of mouse behavior will play a significant role in comprehensive phenotypic analysis - both on the small scale of detailed characterization of individual gene mutants and on the large scale of assigning gene functions across the entire mouse genome [7]. One key benefit of automating behavioral analysis arises from inherent limitations of human assessment: namely cost, time, and reproducibility. Although automation in and of itself is not a panacea for neurobehavioral experiments [28], it allows for addressing an entirely new set of questions about mouse behavior such as conducting experiments on time scales that are orders of magnitude larger than traditionally assayed. For example, reported tests of grooming behavior span time scales of minutes [57, 102] whereas an automated analysis will allow for analysis of this behavior over hours or even days.

Indeed, the significance of alterations in home cage behavior has recently gained attention as an effective means to detect perturbations in neural circuit function - both in the context of disease detection and more generally to measure food consumption and activity parameters [158, 26, 185, 55, 34]. Most previous automated systems [55, 34, 71, 124] rely mostly on the use of sensors to monitor behavior. The physical measurements obtained from these sensor-based approaches limit the complexity of the behavior that can be measured. This problem remains even for expensive commercial systems using transponder technologies such as the IntelliCage system by NewBehavior Inc. While such systems can be effectively used to monitor the locomotion activity of an animal and even perform operant conditioning, they cannot be directly used to study natural behaviors such as grooming, hanging, sniffing or rearing.

Recent advances in computer vision and machine learning yielded robust computer vision systems for the recognition of objects [29, 209] and human actions [109]. The use of vision-based approaches is already bearing fruit for the automated tracking [208, 48, 77] and recognition of behaviors in insects [16, 30]. Several computer-vision systems for the tracking of animals have been developed [124, 15, 183]. Such systems are particularly suitable for studies involving spatial measurements such as the distance covered by an animal or its speed. These tracking techniques have the same limitations as the sensor-based approaches and are not suitable for the analysis of fine animal behaviors such as micro-movements of the head, grooming or rearing.

A few computer-vision systems for the recognition of mice behaviors have been recently described, including a commercial system (CleverSys, Inc) and two prototypes from academic groups [36, 219]. They have not been tested yet in a real-world lab setting using long uninterrupted video sequences and containing potentially ambiguous behaviors or at least comprehensively evaluated against human manual annotations on large databases of video sequences using different animals and different recording sessions.

In this chapter, we describe a trainable, general-purpose, automated and potentially high-throughput system for the behavioral analysis of mice in their home-cage. We characterize its performance against human labeling and other systems. In an effort to motivate further work and set benchmarks for evaluating progress in the field, we also provide a very large database of manually annotated video sequences of mouse behaviors. Developed from a computational model of motion processing in the primate visual cortex [52, 75], our system consists of several steps: first a video input sequence is converted into a representation suitable for the accurate recognition of the underlying behavior based on the detection of space-time motion templates. After this feature computation step a statistical classifier is trained from labeled examples with manually annotated behaviors of interest and used to analyze automatically new recordings containing hours of freely behaving animals. The full system provides an output label (behavior of interest) for every frame of a video-sequence. The resulting time sequence of labels can be further used to construct ethograms of the animal behavior and fit statistical models to characterize behavior. As a proof of concept, we trained the system on eight behaviors of interest (eat, drink, groom, hang, micro-move, rear, rest and walk, see Figure 5-1 for an illustration) and demonstrate that the resulting system performs on par with humans for the scoring of these behaviors. Using the resulting system, we analyze the home-cage behavior of several mouse strains, including the commonly used strains C57BL/6J, DBA/2J, the BTBR strain that displays autistic-like behaviors, and a wild-derived strain CAST/EiJ. We characterize differences in the behaviors of these strains and use these profiles to predict the strain type of an animal.

5.2 Background: automated Systems for Mice Behavior Analysis

Sensor-based approaches

Previous automated systems [55, 34, 71, 124, 224] have relied on the use of sensors to monitor behavior by deriving patterns from trajectories of an animal. Popular sensor-based approaches include the use of PVDF sensors [105], infrared sensors [34, 23, 188, 189], RFID transponders [90] as well as photobeams [55]. Such approaches have been successfully applied to the analysis of coarse locomotion activity as a proxy to measure global behavioral states such as active vs. resting. Other studies have successfully used sensors for the study of food and water intake [224, 50]. However the physical measurements obtained from these sensor-based approaches limit the complexity of the behavior that can be measured. This problem remains even for commercial systems using transponder technologies such as the IntelliCage system (NewBehavior Inc). While such systems can be effectively used to monitor the locomotion activity of an animal as well as other pre-programmed activities via operant conditioning units located in the corners of the cage, such systems alone cannot be used to study natural behaviors such as grooming, sniffing, rearing or hanging, etc.

Video-based approaches

One of the possible solutions to address the problems described above is to rely on vision-based techniques. In fact such approaches are already bearing fruit for the automated tracking [208, 48, 77] and recognition of behaviors in insects [16, 30]. Several computer-vision systems for the tracking of mice have been developed [124, 15, 183, 206, 200, 89, 225]. As for sensor-based approaches, such systems are particularly suitable for studies involving coarse locomotion activity based on spatial measurements such as the distance covered by an animal or its speed [108, 31, 12, 37]. Video-tracking based approaches tend to be more flexible and much more cost efficient. However, as in the case of sensor-based approaches, these systems alone are not suitable for the analysis of fine animal activities such as grooming, sniffing, rearing or hanging. Most of the existing computer vision systems for human action recognition, however, cannot be applied to mice actions because they rely on the articulation of body structures (Ramanan & Forsyth, 2003), whereas mice lack clearly visible limbs or joints, therefore these approaches can not be directly apply to mice actions.

The first effort to build an automated computer vision system for the recognition of mouse behaviors was initiated at USC. As part of this SmartVivarium project, an initial computer-vision system was developed for both the tracking [15] of the animal as well as the recognition of five behaviors (eating, drinking, grooming, exploring and resting) [36]. Xue & Henderson recently described an approach [218, 219] for the analysis of rodent behaviors; however, the system was only tested on synthetic data [66] and a very limited number of behaviors. Overall, none of the existing systems [36, 218, 219] have been tested in a real-world lab setting using long uninterrupted video sequences containing potentially ambiguous behaviors or at least evaluated against human manual annotations on large databases of video sequences using different animals and different recording sessions.

Recently a commercial system (HomeCageScan by CleverSys, Inc) was also introduced and the system was successfully used in several behavioral studies [55, 158, 26, 185]. This commercial product relies on the contour shape of an animal and simple heuristics such as the position of the animal in the cage to infer behavior. It thus remains limited in its scope (tracking of simple behaviors) and error-prone (See [185] and Table 5.6 for a comparison against our manual annotations). In addition, the software packages are proprietary: there is no simple way for the end user to improve its performance or to customize it to specific needs.

5.3 Dataset collection and its challenges

We video recorded singly housed mice from an angle perpendicular to the side of the cage (see Figure 5-1 for examples of video frames). In order to create a robust detection system we varied the camera angles as well as the lighting conditions by placing the cage in different positions with respect to the overhead lighting. In addition, we used many mice of different size, gender, and coat color. Several investigators were trained to score the mouse behavior using two different scoring techniques.

5.3.1 Behavior of interest and definition

We annotate 8 types of common behaviors of inbred mice: drinking (defined by an animal attaching its mouth on the tip of the drinking tube), eating (defined by an animal reaching and acquiring food from the foodhopper), grooming (defined by a fore- or hind-limbs sweeping across the face or torso, typically the animal is reared up), hanging (defined by a grasping of the wire bars with the fore-limbs and/or hind-limbs with at least two limbs off the ground), rearing (defined by an upright posture and forelimbs off the ground, and standing against a wall cage), resting (defined by inactivity or nearly complete stillness), walking (defined by ambulation) and micro-movements (defined by small movements of the animal's head or limbs).

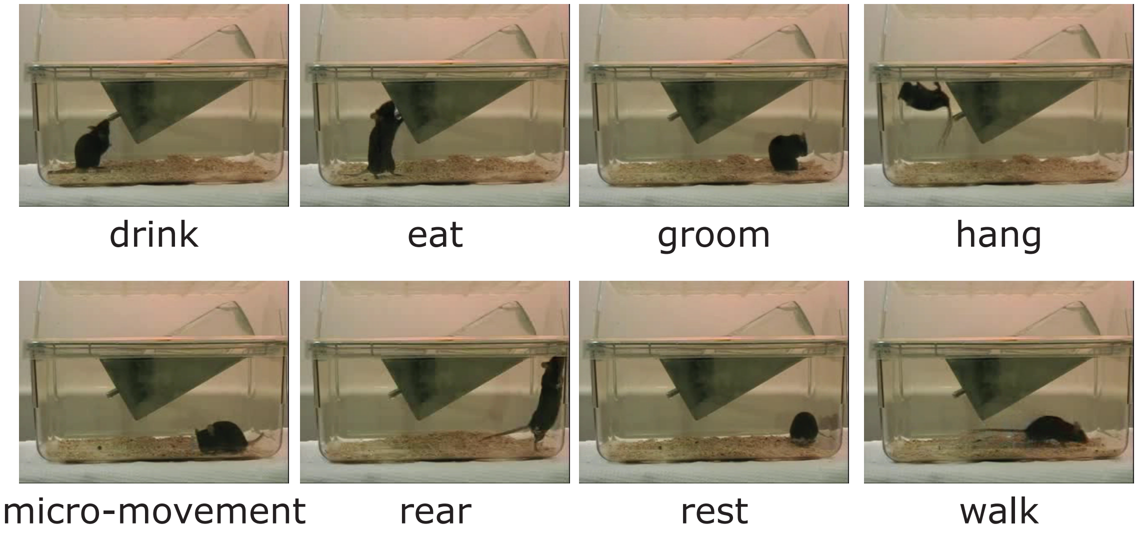


Figure 5-1: Snapshots taken from representative videos for the eight home-cage behaviors of interest.

5.3.2 Datasets

Currently, the only public dataset for mice behaviors is limited in the scope: it contains 435 clips and 6 types of actions [36]. In order to train and test our system on a real-world lab setting where mice behaviors are continuously observed and scored over hours or even days, Two types of datasets are collected. The *clipped database* contains clips with the most exemplary instances of each behavior and is used to train and tune the feature computation module of our system as described in Section 5.5.2 The *full database* was used to train and test the classification module of our system as described in Section 5.5.3. To compare the performance of the system against human performance, we compiled *set B*, a subset of the *full database*, where each frame is assigned a second annotation.

clipped database The first type of dataset denoted as the *clipped database* includs only clips scored with very high stringency, best and most exemplary instances of each behavior from 12 videos. These videos contain different mice (differ in coat color, size, gender, etc) recorded at different times during day and night during 12 separate sessions. Each clip contains one single behavior. Through this style of annotation we created more than $9,000$ short clips, each containing a unique annotation. To avoid errors, this database was then curated by one human annotator who watched all $9,000$ clips again, retaining only the most accurate and unambiguous assessments, leaving $4,200$ clips ($262,360$ frames corresponding to about 2.5 hours) from 12 distinct videos. This database is significantly larger than the currently publicly available, clip-based dataset [36], which contains only 5 behaviors (eating, drinking, grooming, exploring and resting) for a total of 435 clips. Figure 5-2 shows the distribution of labels for the *clipped database*.

full database The second dataset, called the *full database* involved labeling every frame for 12 distinct videos (different from the 12 videos used in the *clipped database*). Each video is $30-60$ min in length, resulting in a total of over 10 hours of continuously annotated videos. As in the *clipped database*, these videos are chosen from different mice at different times to maximize generalization of the dataset. These labels are less stringency than in the *clipped database*. Currently there is no other publicly available dataset with continuously labels like the *full database*. By making such a database available and comparing the performance against human labeling and other vision-based systems, we hope to further motivate the development of such computer vision systems for behavioral phenotyping applications. Figure 5-3(A) shows the distribution of labels for the *full database*.

set B We considered a small subset of the *full database* corresponding to many short video segments which are randomly selected from the *full database*. Each segment is $5-10$ min long and makes of a total of about 1.6 hours of dataset. Each frame of the *set B* is assigned a second human annotation. We estimate an average human labeler's performance by computing the average agreement between the second set of human annotations with the first set of human annotations (ground truth). Ground truth (first human annotation) of the *full database* is not 100% accurate mostly due to frames containing ambiguous actions arising during the transition of two actions (as described in detail below), therefore we use the human labeler's performance as a close-to-upper bound of performance since the system relies on these human annotations to learn to recognize behaviors. Figure 5-3(B) shows the distribution of labels for the *set B*.

5.3.3 Annotation

All the 24 videos (12 in the *clipped database* + 12 in the *fill database*) were annotated using a freeware subtitle editing tool, Subtitle Workshop freeware subtitle editing tool from UroWorks available at `http://www.urusoft.net/products.php?cat=sw&lang=1`. A team of 8 investigators: 'Annotators group 1' was trained to annotate mouse home cage behaviors. *Set B* was annotated by 4 human annotators randomly selected from 'Annotators group 1', denoted as 'Annotators group 2'. Some segments of *set B* have the first and second set of annotations made by the same annotator. For the *full database* to be annotated, every hour of videos took about 22 hours of manual labor for a total of 230 hours of work. For the *clipped database* it took approximately 50 hours to manually score $9,600$ clips. The second screening used to remove ambiguous clips took around 40 hours.

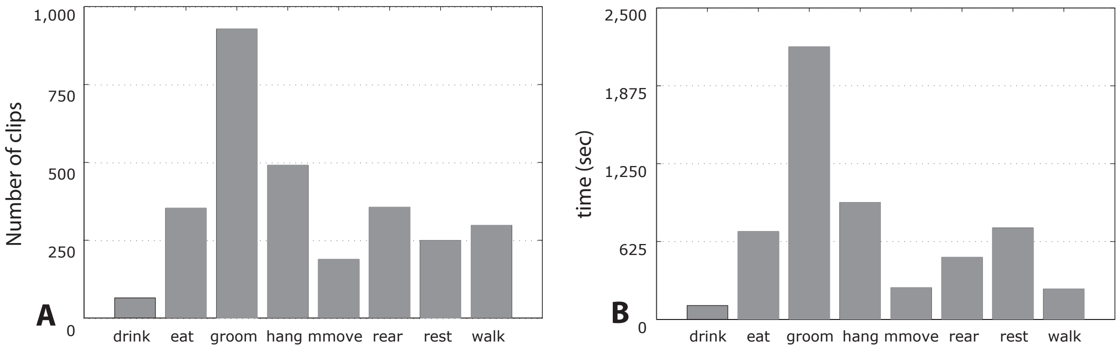


Figure 5-2: Distributions of behavior labels for the *clipped database* over (A) the number of clips and (B) the total time.

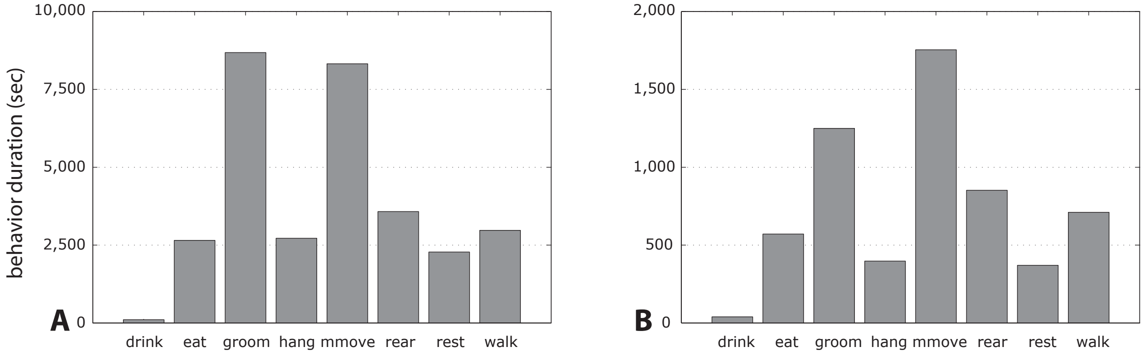


Figure 5-3: Distribution of behavior labels on the (A) *full database* annotated by 'Annotator group 1' and the (B) *set B* (a subset of the *full database*), which was annotated by one more annotator from 'Annotator group 2' (a subset of 'Annotator group 1') to evaluate the agreement between two independent annotators.

5.3.4 Challenge

Context dependency

Labeling of actions can not be made on a frame-by-frame basis. Contextual information from nearby frames are required for both robust behavior annotation and recognition. An example is illustrated in Figure 5-4.

Ambiguous actions

The accuracy of the system depends mostly on the quality of the ground truth human annotations. Given the definition in Section 5.3.1, we observe from the labeled examples that the main confusion for a human annotator is between:

- (1) eat *vs.* rear: at the instance when a mouse stands against the back side of a cage (rearing), it looks like reaching the foodhopper (eating) because in both cases, the head of the mouse seems to touch the foodhopper when seeing from the front side of the cage where the camera is placed.
- (2) micro-movement *vs.* walk: small movements of a mouse's limbs (micro-movement) sometimes result in slow and continuous changes of positions, and therefore being annotated as ”walking”.
- (3) micro-movement *vs.* grooming: when sitting back to the camera during grooming, the mouse seems to only move its head slowly and therefore annotated as ”micro-movement”.
- (4) grooming *vs.* eating: chewing (eating) is usually followed by acquiring food from the foodhopper (eating). If the temporal association is neglected, the appearance of chewing (rearing up with fore-limb sweeping across the face) indeed looks similar to grooming. Apparently, some annotators assign the most suitable category for each frame independently without considering the temporal association.

These confusions arise from the limited resolution, the limited viewpoint((1),(3)), and ambiguity of actions per se ((2),(4)).

Transition of actions

The annotators have to assign a label for each frame even when the underlying action is ambiguous. The main disagreement between human annotators is the misaligned boundary between two actions. For example, a mouse typically takes around 10 milliseconds to transit from a well-defined walking to a well-defined eating, these transition frames can fall into both categories and therefore are usually disagreed between two human labelers.

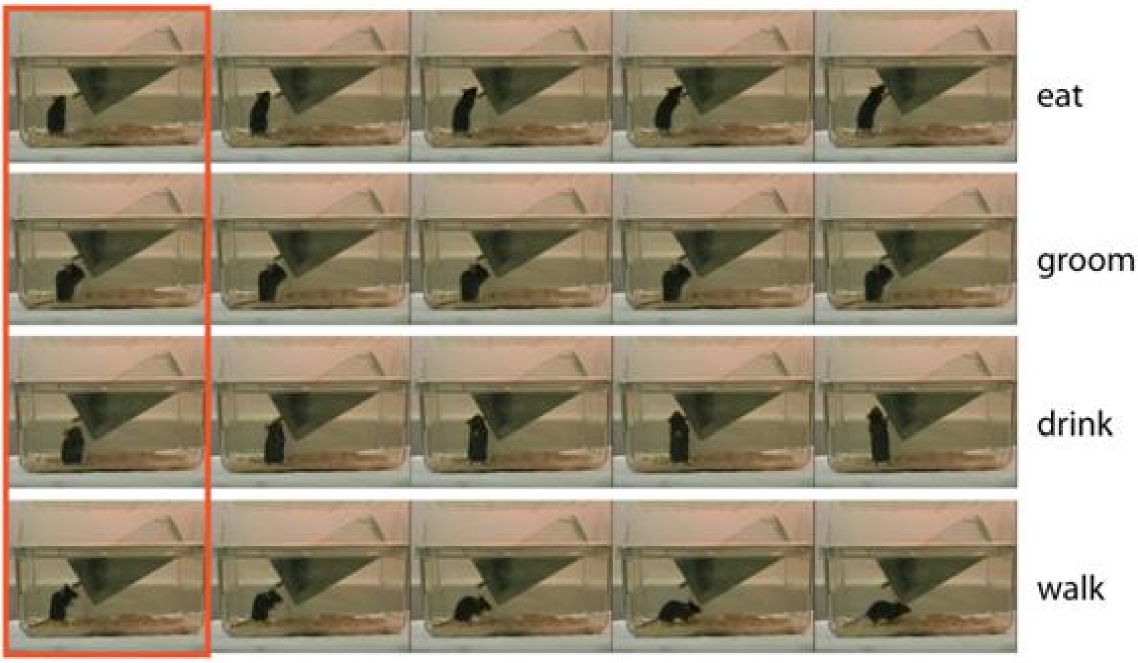


Figure 5-4: Single frames are ambiguous. Each row corresponds to a short video clip. While the leftmost frames (red bounding box) all look quite similar, they each correspond to a different behavior (text on the right side). Because of this ambiguity, frame-based behavior recognition is unreliable and temporal models that integrate the local temporal context over adjacent frames are needed for robust behavior recognition.

5.4 System Overview

Our system is a trainable, general-purpose, automated, quantitative and potentially high-throughput system for the behavioral analysis of mice in their home-cage. Our system, developed from a computational model of motion processing in the primate visual cortex [75] consists of two modules: (1) a feature computation module, and (2) a classification module. In the feature computation module, a set of 304 space-time motion templates that are learned from most exemplary clips in the *clipped database* are used to convert an input video sequence into a representation, which is then passed to a classifier to reliably classify every frame into a behavior of interest. In the classification module, the classifier is trained from continuously labeled temporal sequences in the *full database* and outputs a label (as one of the 8 types of behaviors) for every frame of a input video-sequence. The resulting time sequence of labels can be further used to construct ethograms of mouse behavior to characterize mouse strain. The system modules are illustrated in Figure 5-15.

5.4.1 Feature Computation

The feature computation module takes as input a video sequence and outputs for each frame a feature vectors of 316 dimensions (a concatenation of 304 motion features + 12 position- and velocity-based features). A background subtraction procedure is first applied to an input video to compute a segmentation mask for pixels belonging to the animal vs. the cage based on the instantaneous location of the animal in the cage (Figure 5-15(A)). This is adapted from our previous work for the recognition of human actions [75]. A bounding box centering on the animal is derived from the segmentation mask (Figure 5-15(B)). Two types of features are then computed: position- and velocity-based features as well as motion features. Position- and velocity-based features are computed directly from the segmentation mask (Figure 5-15(C)). In order to speed-up the computation, motion-features are performed on the bounding-box within a hierarchical architecture (Figure 5-15(D-F)).

Motion features The system models the organization of the dorsal stream in the visual cortex, which has been critically linked to the processing of motion information [13]. The model computes features for the space-time volume centering at every frame of an input video sequence via a hierarchy of processing stages, whereby features become increasingly complex and invariant with respect to $2D$ transformations along the hierarchy. The model starts with the S_1/C_1 stage consisting of an array of spatio-temporal filters tuned to 4 directions of motion and modeling after motion-direction-sensitive cells in the primary visual cortex (V1) [177]. By convolving the input sequence with these filters, we obtain the outputs of the S_1/C_1 stage as a sequence of C_1 maps, each corresponding to motion present at a frame along the 4 directions (Figure 5-15(E)). In the S_2/C_2 stage, we computed for every C_1 map, a vector of matching scores that measure the similarity between the motion present in the current map and each of the 304 motion templates (Figure 5-15(F)). More specifically, at every spatial position of a C_1 map, we perform a template matching between a motion template and a patch of the map centering at the current position with the same size of the template and then we obtain a matching score. The C_2 output is the global maximum pooled over the matching scores computed at all the spatial locations of one frame.

We repeat this procedure for all the motion templates, and obtain a C_2 feature vector, for each frame, of 304 dimensions.

Learning the dictionary of motion templates The motion templates used in the S_2/C_2 stage are extracted from the *clipped database* because this set contains the most exemplary instances of each behavior. We draw $12,000$ motion templates, each as a local patch of a C_1 map randomly selected from 3 videos in the *clipped database*. In order to select templates that are useful for discriminating between actions and speed up the experiment, we perform feature selection on a set of $4,000$ C_2 feature vectors computed from frames which are randomly selected from the 3 videos. As in [75], the zero-norm SVM [215] of Weston *et al.* is used for feature selection. The algorithm is described as below.

A SVM classifier is trained on the pool of C_2 vectors and returns a hyperplane that maximizes the margin between pairs of behavior categories (8 in this case). The hyperplane is a vector of $12,000$ dimensions, each corresponding to the significance (how well it discriminates between categories) of one motion template. Each dimension of the C_2 vectors is then reweighed using the coefficient of the hyperplane in the same dimension. The reweighed data is then used for training another SVM. By repeating this procedure, the weights of the hyperplane corresponding to motion templates that highly discriminate between behavior categories increase, whereas the weight corresponding to other templates gradually decrease to zero. Finally, we select 304 highly-weighted templates that lead to a good performance without taking too much time to compute. Detailed results are described in Section 5.5.2.

Evaluation of the motion features on the *clipped database* In order to evaluate the quality of our motion features (C_2 feature vectors) for the recognition of high-quality unambiguous behaviors, we trained and tested a multi-class Support Vector Machine (SVM) on motion features of single frames from the *clipped database* using the all-pair multi-class classification strategy. This approach does not rely on the temporal context of behaviors beyond the computation of low-level motion signals in the S_1/C_1 stage and classifies each frame independently. We also rely on the performance on the *clipped database* to optimize some parameters of the model. The parameters include preferred directions of the filters, the nature of the non-linear transfer function, and the video resolution. The results are described in Section 5.5.2. The optimized motion features led to 93% accuracy (chance level 12.5% for 8-class classification). This suggests that the representation provided by the dictionary of 304 motion templates is suitable for the recognition of the behaviors of interest, even under conditions when the global temporal structure, temporal structure beyond the computation of low-level motion signals, of a temporal sequence is completely discarded.

Position- and velocity-based feature computation In addition to the motion features described above, we computed an additional set of features derived from the instantaneous location of the animal in the cage (Figure 5-15(C)). To derive these features, we first computed a bounding box for each frame centering at the animal by subtracting off the video background. For a static camera as used here, the video background can be well approximated by a median frame in which each pixel value is the median value across all the frames at the same pixel location (day and night frames under red lights were processed in separate videos). Position- and velocity-based measurements were estimated for each frame based on the $2D$ coordinates (x, y) of the bounding box. These include the position and the aspect ratio of the bounding box (indicating whether the animal is in a horizontal or vertical posture), the distance of the animal from the feeder as well as the instantaneous velocity and acceleration. Figure 5-15(C) illustrates 6 types of features. A complete description of the 12 types of features is listed in Table 5.1.

The position and size of the cage vary between videos due to the variations in the camera angle and the distance between the camera and the cage. To make position- and velocity-based features comparable between videos, we calibrate these features with respect to the x and y coordinates of the top, bottom, left and right sides of the cage.

5.4.2 Classification

Performing a reliable phenotyping of an animal requires more than the mere detection of stereotypical non-ambiguous behaviors. In particular, the present system aims at classifying every frame of a video sequence even for those frames whose underlying actions are

C_x	x coordinate of the center of the mouse
C_y	y coordinate of the center of the mouse
w	width of the mouse
h	height of the mouse
h/w	aspect ratio of the mouse
f_d	nearest distance from the mouse to the feeder
t_d1	nearest distance from the mouth of the mouse to the far tip of the drinking spout
t_d2	nearest distance from the mouth of the mouse to the near tip of the drinking spout
V_x	speed of the mouse in the x direction
V_y	speed of the mouse in the y direction
sV_x	smoothed speed of the mouse in the x direction
sf_d	smoothed nearest distance from the mouse to the feeder

Table 5.1: A list of 12 position- and velocity-based features, where $V_x(t) = |C_x(t) - C_x(t-1)|$,$V_y(t) = |C_y(t) - C_y(t-1)|$, $sV_x(t) = |C_x(t) - C_x(t-2)|$, and $sf_d(t) = \frac{(f_d(t-2)+f_d(t-1)+f_d(t))}{3}$

difficult to categorize, as described in Section 5.3.4. For this challenging task, the temporal context of a specific behavior becomes an essential source of information; thus, learning an accurate temporal model for the recognition of actions becomes critical. Here we used a Hidden Markov Support Vector Machine(SVMHMM) [195, 196], which is an extension of the Support Vector Machine classifier for sequence tagging. This temporal model was trained on the 12 continuously labeled videos of the *full database*. SVMHMM takes input as a sequence of C_2 features vectors of an input video and their annotations, and outputs a predicted label (behavior of interest) for each frame (Figure 5-15(G)).

Hidden Markov Support Vector Machine(SVMHMM) SVMHMM combines the advantage of SVM and HMM by discriminatively training models that are similar to a hidden Markov model. The general setting of SVMHMM allows for learning a kth-order hidden Markov model. Here we use the first-order transition model. Given an input sequence $\mathbf{X} = (\mathbf{x}_1 \ldots \mathbf{x}_T)$ of feature vectors, the model predicts a sequence of labels $\mathbf{y} = (y_1 \ldots y_T)$ according to the following linear discriminant function:

$$\mathbf{y} = \text{argmax}_{\mathbf{y}} \sum_{t=1}^{T} [\mathbf{x}_t \cdot \mathbf{w}_{y_t} + I_{trans}(y_{t-1}, y_t) \cdot \mathbf{w}_{trans}] \tag{5.1}$$

$\mathbf{w}_{y_t}$ is an emission weight vector for the label y_t and $\mathbf{w}_{trans}$ is a transition weight vector for the transition between the label y_{t-1} and y_t. $I_{trans}(y_{t-1}, y_t)$ is an indicator vector that has exactly one entry set for the sequence (y_{t-1}, y_t).

During training, SVMHMM was given a set of training examples of sequences of feature vectors, $\mathbf{X}^1 \dots \mathbf{X}^N$ with their label sequences $\mathbf{y}^1 \dots \mathbf{y}^N$ and tries to minimize a loss function $\Delta(\mathbf{y}^i, \mathbf{y})$ which is defined as the number of misclassified labels used in a sequence.

$$\min \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{n}\sum_{i=1}^{N}\xi_i \tag{5.2}$$

s.t. for all $\mathbf{y}$ and $i = 1 \dots N$:

$$\textstyle\sum_t(\mathbf{x}_t^i \cdot \mathbf{w}_{y_t^i} + I_{trans}(y_{t-1}^i, y_t^i) \cdot \mathbf{w}_{trans}) \geq \tag{5.3}$$

$$\textstyle\sum_t(\mathbf{x}_t^i \cdot \mathbf{w}_{y_t} + I_{trans}(y_{t-1}, y_t) \cdot \mathbf{w}_{trans}) + \Delta(\mathbf{y}^i, \mathbf{y}) - \xi_i \tag{5.4}$$

$\mathbf{w}$ is the concatenation of the emission and transition weight vector. C is a parameter that trades off margin size and training error.

5.5 Experiments and the results

5.5.1 Training and Testing the system

The evaluation on the *full database* and the *set B* was obtained using a leave-one-video-out cross-validation procedure. This consists in using all except one videos to train the system and the left out video to evaluate the system and repeating this procedure $n = 12$ times for all the videos. The system's predictions as well as ground truth annotations for all the videos are then concatenated to compute the overall accuracy as

$$\text{accuracy} = \frac{\text{\# total frames correctly predicted by the system}}{\text{\# total frames}} \tag{5.5}$$

Human labelers' performance is computed similarly as

$$\text{accuracy} = \frac{\text{\# total frames correctly labeled by the system}}{\text{\# total frames}} \tag{5.6}$$

Here a prediction or a label is ’correct’ if it matches ground truth made by ’Annotators group 1’. Such a procedure has been shown to provide the best estimate of the performance of a classifier and is standard in computer vision. This guarantees that the system is not just recognizing memorized examples but generalizing to previously unseen examples.

For the *clipped database*, the leave-one-video-out procedure is performed on 9 videos that were not used to extract motion templates. The clips from all except one video are used to train the system while testing is performed on the clips of the remaining video. This procedure is repeated $n = 9$ times. A single prediction was obtained for each clip via voting across frames as in [75], and predictions for all the clips are then concatenated to compute the overall accuracy as

$$\text{accuracy} = \frac{\text{\# total clips correctly predicted by the system}}{\text{\# total clips}}. \tag{5.7}$$

In addition to measure the accuracy of the system as above, we also use a confusion matrix to visualize the system’s performance for each individual behavioral category. A confusion matrix is a common visualization tool used in muti-class classification problem. Each row of the matrix represents a true class, and each column represents a predicted class. Each entry $C(x, y)$ in a confusion matrix is the probability with which a behavior of type y (along rows) is classified as type x (along column), as computed by

$$C(x, y) = \frac{\text{\# total frames annotated as type y and predicted as type x}}{\text{\# total frames annotated as type y}} \tag{5.8}$$

The higher probabilities along the diagonal and the lower off-diagonal values indicate successful classification for all behavioral types.

5.5.2 Results for the feature computation module

The *clipped database* contains the best exemplary instances of each behavior, therefore it is suitable for optimizing parameters of the feature computation module. Particularly, in the S_1/C_1 stage, the parameters are the preferred directions of the spatio-temporal filters, the nature of the non-linear transfer function used, and the video resolution. We also optimize the S_2/C_2 stage by selecting a set of motion templates that are useful for discriminating different action categories. Finally, we compare our optimized system with a computer vision system developed by Dollar *et al.* [36].

Optimization of the S_1/C_1 stage

When optimizing the S_1/C_1 stage, we evaluate the system's performance by training and testing a multi-class Support Vector Machine (SVM) classifier on C_1 feature vectors computed from single frames of the 9 videos that were not used to extract motion templates. The training/testing is done on a leave-one-video-out procedure, and the accuracy is computed as in Equation 5.7. To speed up the evaluation, we experiment with a subset of $36,000$ C_1 feature vectors computed from random frames of the 9 videos.

Comparison of 7 types of S_1 units The animal as well as the background color vary between videos and the lighting condition changes with time; it is white in the day and red in the night. In order to find the best S_1 units that are invariant to these contrast changes, we experimented with 7 types of S_1 units as shown in Table 5.2. Table 5.2 shows very close recognition rates, 76% 77% for all types. We choose the first type, *i.e.* , after convolving an input sequence with the 4 direction-selective filters, we normalize these S_1 responses along each direction with respect to the summation of responses across all the directions.

Comparison of video resolutions We also experimented with video resolutions in order to find one in which motion can be best captured by the fixed-sized spatio-temporal filters. We start from the original video resolution 480 pixels $\times$ 720 pixels then down to 0.75, 0.5, 0.375, 0.25, 0.187 times of the original resolution. The results are shown in Table 5.3. We found that the medium resolution 180×270 leads to the best performance. However, position features might not be accurate computed at a low video resolution, we therefore choose a slightly higher resolution: 240×360.

Comparison of the number of preferred directions of the S_1 units With the use of S_1 units tuned to more directions, motion can be computed more accurately, however, the computation increases correspondingly. Here we tried to determine the number of directions

	normalization formula	accuracy
1	$S_1(d)' = F \cdot X$ $S_1(d) = \frac{S_1'(d)^2}{\sum_d S_1'(d)^2}$	77.1%
2	$S_1 = F \cdot X$	76.8%
3	$S_1 = \frac{F \cdot X}{\|\|X\|\|}$	76.8%
4	$S_1 = \frac{F \cdot X}{\|\|X\|\|_2}$	76.8%
5	apply a z transform on every 3×3(pixels) patch of the frame $S_1 = F \cdot X$	76.8%
6	apply a z transform on every 9×9(pixels) patch of the frame $S_1 = F \cdot X$	76.1%
7	do a histogram equalization within the bounding box S1 $= F \cdot X$	75.9%

Table 5.2: 7 types of S_1 types we experimented with and the accuracy evaluated on the $C1$ features computed from the 7 types. F is a spatio-temporal filter with size 9 pixels × 9 pixels × 9 frames. X is a space-time patch of the same size. $S_1(d)$ is the convolution between the patch X and the F that is tuned to the $d - th$ direction.

resolution	480×720	360×540	240×360	180×270	120×180	90×135
accuracy	74%	76.5%	76.8%	79.5%	78%	77%

Table 5.3: All the video resolutions we experimented with (unit: pixel) and the accuracy evaluated on the $C1$ features.

that best balance the tradeoff between accuracy and the computation. We experimented with n tuned directions that are equally spaced between 0^o and 360^o, $n = 1, 2, 4, 8$. As shown in Table 5.4, the accuracy grows with n, the number of directions, as expected, but the growth rate decreases after $n = 4$. That is, accuracy increases by 3% when n is doubled from 1 to 2 and from 2 to 4, but only increases by 1.5% from $n = 4$ to $n = 8$. We choose $n = 4$ to compromise between computation and performance.

n	1	2	4	8
accuracy	69.5%	72.5%	76.0%	77.5%

Table 5.4: n, the number of tuned directions of the filters and the accuracy evaluated on the $C1$ features.

Selection of motion templates

The goal of the feature selection stage is to choose from an initial set of motion templates a subset of that is useful for discriminating between behavioral categories. The initial set of $12,000$ patches were randomly drawn from the C_1 maps of 3 videos in the *clipped database* and used to compute C_2 vectors of $4,000$ frames, which are randomly drawn from the same 3 videos.

We applied the feature selection technique, zero-norm SVM, on these C_2 vectors. Figure 5-5 shows the number of motion templates that receive weights higher than some threshold for the first 50 rounds of the zero-norm SVM. The number drops quickly, from $12,000$ down to $2,000$ in the first 6 rounds and remains steadily around 300 after the $15th$ round.

We next evaluate the system's accuracy as a function of the number of selected templates. We select 6 rounds and report the system's accuracy in Table 5.5. The accuracy remains 93% for 954 down to 304 templates. We therefore conclude that the 304 motion templates are very significant in discriminating actions, and they will be used to compute motion features for the *full database*.

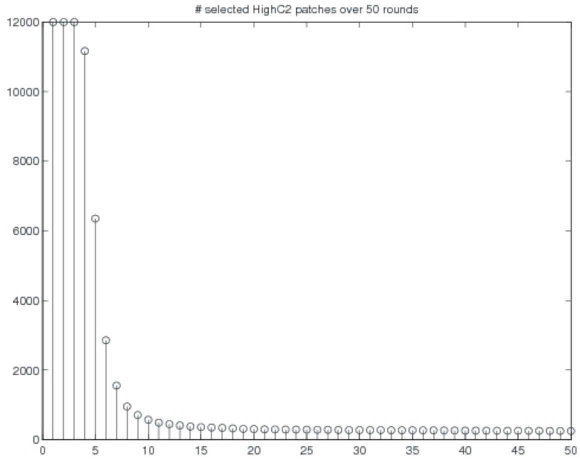


Figure 5-5: The number of selected motion templates in each round of the zero-norm SVM.

round	8	9	10	11	12	20
# templates	954	709	573	490	407	304
Accuracy	93.4%	93.3%	92.9%	92.8%	92.9%	92.9%

Table 5.5: The number of selected motion templates in each round of the zero-norm SVM and the accuracy evaluated on the $C2$ features.

Comparison with a computer-vision system on the *clipped database*

The computer vision system used here for benchmark is the system developed by [36] at the University of California (San Diego) as part of the SmartVivarium project [8]. The system has been shown to outperform several other computer vision systems on several standard computer vision databases and was tested for both the recognition of human and rodent behaviors [36]. The authors graciously provided the source code for their system. Training and testing of their system was done in the same way as for our system using a leave-one-video-out procedure on the *clipped database*. Here we attempted to maximize the performance of the baseline system [36] by tuning some of the key parameters such as the number of features and the resolution of the videos used. Nevertheless we found that the default parameters (50 features, a 320×240 video resolution as used for our system) led to the best performance (81% for their system vs. 93% for our system). It is possible however that further refinement of the corresponding algorithm could nevertheless improve its performance.

5.5.3 Results for the classification module

The *full database* contains a set of continuously labeled videos and therefore is suitable for learning the temporal transition between frames and for optimizing the sequential tagging algorithm, SVMHMM. We will compare our optimized system with the human annotators ('Annotator group 2') and with a commercial software (HomeCageScan). In this section, we will also evaluate some aspects of the system, such as the contribution of the position features to the system's performance and the number of annotated examples required to train the system.

Optimization of the classifier

Here we evaluate the system's performance by training and testing a SVMHMM classifier on C_2 feature vectors computed from the 12 videos of the *full database*. The training/testing is done on a leave-one-video-out procedure. For each leave-one-out trial, we draw from each training video 5 video segments, each being 1-min long, for training the SVMHMM, and the testing is still done for the whole length of the left video. The accuracy is computed as in Equation 5.9.

Optimizing C In the SVMHMM setting, the parameter C trades off margin size and training error. We expect a large C increases the penalty for misclassified labels and could therefore lead to a better performance. We tried a range of C values, from 1 to 10, the system's accuracy as well as required computation time are shown in Figure 5-6. The accuracy remains quite consistent for all the values we tried, but the computation time increases almost linearly with C, we therefore use $C = 1$ for the rest of the experiments.

Optimizing the length of training sequences SVMHMM takes as input a set of sequences. In applications such as speech tagging, an input sequence is a sentence, whereas in our videos, there is no analogous concept or structure to that of a sentence. Our solution is to divide a training video into many video segments of equal length, each as a training example. During testing, the whole testing video is treated as a single example. Here we experimented with the length of the video segments.

We firstly train a SVMHMM using the 1-min long video segments described above; each training example is a 1-min segment ($1,800$ frames). We then repeatedly divide these segments into shorter segments down to segment length of 1 frame (3 ms) and train a SVMHMM for each segment length. Figure 5-7 shows the accuracy increases with segment length and remains stable (72%) for length longer than 100 frames. We conclude that for the recognition of mouse behavior, a sequence of at least 3 seconds (100 frames) is required in order for a SVMHMM to learn a good model of the temporal transition. For the future experiments, we use 1-min as the length for each training example.

In addition, we trained a SVM classifier on single frames of the set of video segments and compared against the SVMHMM. The accuracy of the SVM (red cross in Figure 5-7) is 62%, about 10% lower than the 72% achieved by SVMHMM. This suggests that learning of temporal transition is significant to the recognition of mouse behavior in videos.

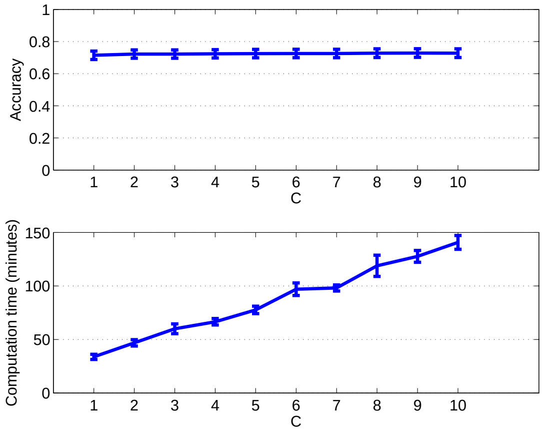


Figure 5-6: (Top) The accuracy of the system evaluated on the *full database* and (Bottom) the required computation time as a function of the C parameter.

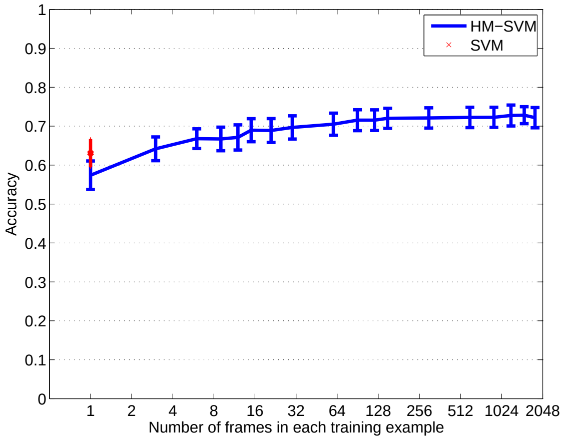


Figure 5-7: The accuracy of the system evaluated on the *full database* as a function of length of training video segment. The red cross indicates the performance of the system when a SVM, instead of SVMHMM, classifier is used.

Comparison with a commercial software *vs.* human performance on the *full database*

Here we evaluate the system's performance on the doubly annotated *setB*. The system is compared against a commercial software (HomeCageScan 2.0, CleverSys, Inc) for mouse home cage behavior classification and against human manual scoring. In order to compare our system with the commercial software HomeCageScan 2.0 (CleverSys Inc), we manually matched the 38 output labels from the HomeCageScan to the 8 behaviors used in our system. For instance, actions such as 'slow walking', 'walking left' and 'walking right' were all re-assigned to the 'walking' label to match against our annotations. With the exception of behaviors such as 'jump', 'urinate', 'unknown behavior' which don't occur in the two datasets we collected, we matched all other HomeCageScan output behaviors to one of the 8 behaviors of interest (see Table 5.7 for a list of matches between the labels of the HomeCageScan and our system). It is possible that further fine-tuning of HomeCageScan parameters could have improved upon the accuracy of the scoring.

Note that the annotations made by initial 8 annotators ('Annotators 1') are used as ground truth to train and test the system, and the second set of annotations made by 'Annotators group 2' on *set B* is used only for computing the performance of human manual scoring.

Table 5.6 shows the comparison. Overall we found that our system achieves 76.6% agreement with human labelers('Annotator group 1') on the *set B*, a result significantly higher than the HomeCageScan 2.0 system (60.9%) and on par with humans ('Annotator group 2') (71.8%). Figure 5-8 shows the confusion matrices for the system, humans ('Annotators group 2'), and HomeCageScan. Two online videos demonstrating the automatic scoring of the system are at `http://techtv.mit.edu/videos/5561` and `http://techtv.mit.edu/videos/5562`. Two online videos demonstrating the annotations of 'Annotators group 1' *vs.* 'Annotators group 2' are at `http://techtv.mit.edu/videos/5562` and `http://techtv.mit.edu/videos/5563`.

Generalization with few training examples

When the system is used under a novel setting, such as behaviors other than the existing 8 types, videos taken from a top-view camera, environment other than home-cage (for example, fear-conditioning box), or to detect behaviors in rats, it is critical to know how many annotated examples are required by the system to reach reasonable performance. We investigate this issue by varying the number of training examples for evaluation of the *full*

	Our system	CleverSys commercial system	Human ('Annotator group 2')
set B (1.6 hours of video)	77.3%/76.4%	60.90%/64.0%	71.6%/75.7%
full databse (over 10 hours of video)	78.3%/77.1%	61.0%/65.8%	

Table 5.6: Accuracy of our system, human annotators and HomeCageScan 2.0 CleverSys system evaluated on the *set B* and the *full database* for the recognition of 8 behaviors. Using 'Annotator group 1' as ground truth, accuracy is computed as percentage of frames correctly classified by a system (chance level is 12.5% for the 8-class classification problem). For the *set B*, we also report the average of diagonal terms of confusion matrices shown in Figure 5-8, see underlined numbers.

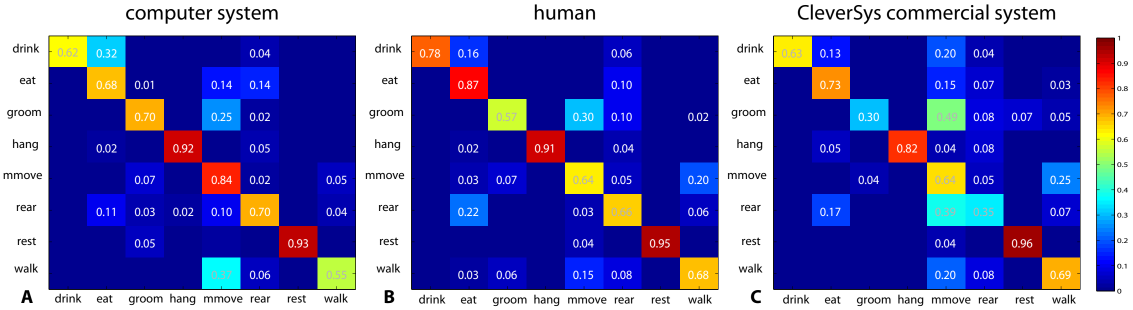


Figure 5-8: Confusion matrices evaluated on the doubly annotated *set B* for (A) system to human scoring, (B) human to human scoring, and (C) CleverSys system to human scoring. Using 'Annotator group 1' as ground truth, the confusion matrices ware obtained for measuring the agreement between the ground truth (row) with the system (computer system), with 'Annotator group 2'(human) and with the baseline software (CleverSys commercial system). For a less cluttered visualization, entry with value less than 0.01 is not shown. The color bar indicates the percent agreement, with more intense shades of red indicating agreements close to 100% and lighter shades of blue indicating small fractions of agreement.

database. When performing the leave-one-video-out procedure on the *full database*, only a representative set of x minutes (x 1-minute video segments) from each training video is used for training, and testing is done on the whole length of the left-out video. A representative set is selected such that all types of actions are included. Average accuracy of the 12 videos as a function of x is shown in Figure 5-9(A). (Note in Table 5.6, overall accuracy is computed by concatenating predictions across videos, here we compute accuracy for each video to obtain the standard deviation). It shows with annotating only 2 mins for each training video, the system is already able to achieve 90% of the performance obtained using 30 minutes. With 10-15 minutes of annotation for each training video, the system's average accuracy reaches the optimal level that is obtained by using all the minutes for training. Therefore, we expect the system can generalize well to previously unseen videos with around 22 minutes of training examples (2 mins × 11 training video). We therefore expect the system is able to scale up to many types of behaviors with hours of annotated examples.

Although the goal of the present study was to create a behavior detection tool that would generalize well in many other laboratories, this is not always necessary. In such cases where generalization is not required, the most efficient approach is to train the system on the first few minutes of the same video and then let the system complete the rest of that video. In Figure 5-10(A), we show that by training on a representative set of 3 minutes of a video, the system is able to achieve performance with 90% level of performance obtained using a representative set of 30 minutes.

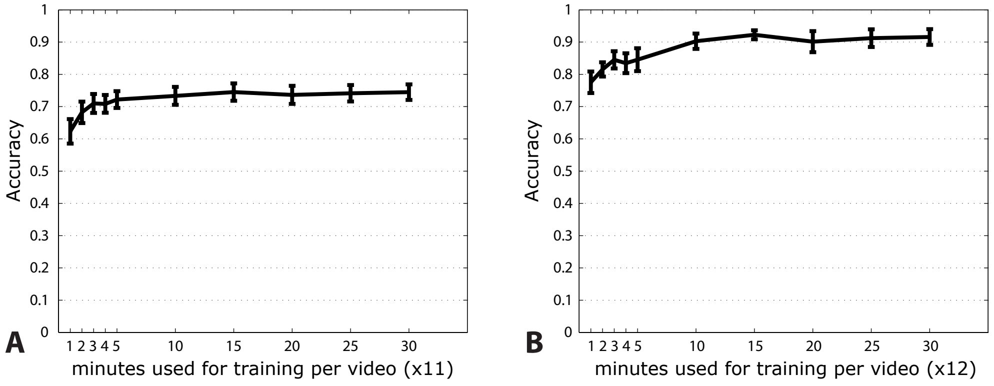


Figure 5-9: Average accuracy of the system for the (A) *full database* and the (B) *wheel-interaction set* as a function of minutes of videos used for training. For each leave-one-out trial, the system is trained on a representative set of x (x axis of the figure) minutes selected from each training video and tested on the whole length of the left-out video. A representative set consisting of x 1-minute segment is selected to maximize types of actions present in the set. (A) Average accuracy and standard error across the 12 trials, each for one video in *full database*. (B) Average accuracy and standard error across the 13 trials, each for one video in the *wheel-interaction set*.

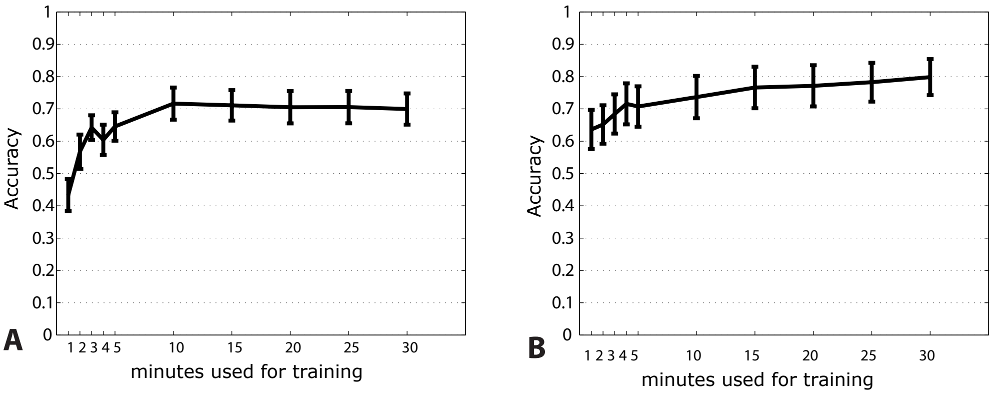


Figure 5-10: Average accuracy of the system for the (A) *full database* and the (B) *wheel-interaction set* as a function of minutes of videos used for training. A representative set of video segments is selected from $0 - 30$th min of each video for training and testing is done on 30th min- end of the same video. A representative set consisting of x (x axis of the figure) 1-minute segment is selected to maximize types of actions present in the set. (A) Average accuracy and standard error across the 12 trials, each for one video in the *full database*. (B) Average accuracy and standard error across the 13 trials, each for one video in the *wheel-interaction set*.

The effects of adding position features

Some behaviors exhibit similar motion and without the locations of occurrence, these behaviors are hard to distinguish for a human labeler. For example, 'drinking', 'eating', and 'rearing' all have upward motion, but usually occur at different locations. 'Drinking' occurs near the water bottle spout when an animal attaches its mouth to the tip of a drinking tube; 'eating' occurs when an animal reaches the foodhopper; and 'rearing' occurs when an animal reaches against the wall. Our solution for removing these ambiguities is to compute a set of 12 position- and velocity-based features such as the distance from a mouse to a drinking tube or foodhooper. Table 5.1 lists the 12 types of features. To quantify the effects of the 12 features, we remove them from the feature computation module and train and test the system on the *set B* using motion-only features. Figure 5-11 shows the confusion matrix evaluated for the system evaluated on the *set B*. Comparing Figure 5-8 (motion + pos) with Figure 5-11 (motion), we found the system's performance for the most static actions benefits most from the addition of the position features. The improvement of accuracy is 62% for 'drinking' and 28% for 'resting'. Accuracy for 'eating' also increases by 8%.

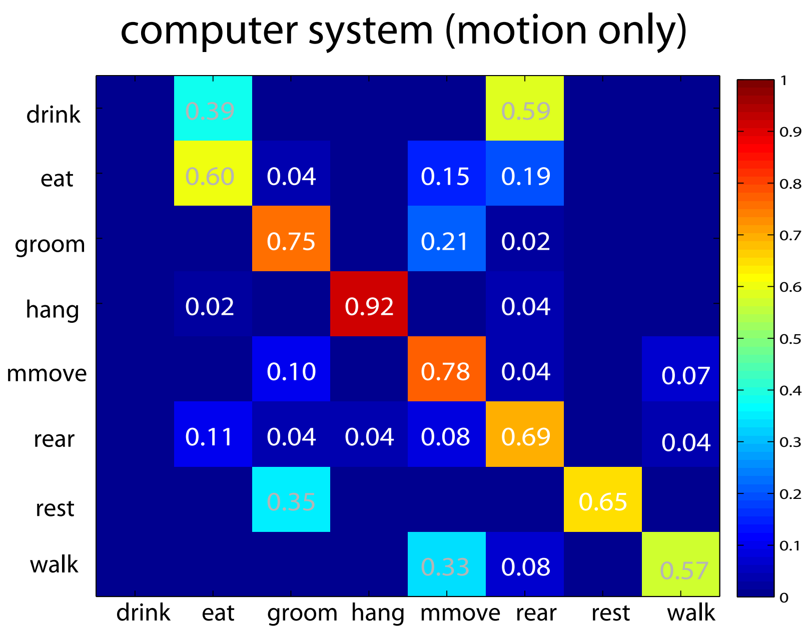


Figure 5-11: Confusion matrices evaluated on the doubly annotated set *set B* for system to human scoring. Here only motion features are used in the feature computation module. For a less cluttered visualization, entry with value less than 0.01 is not shown. The color bar indicates the percent agreement, with more intense shades of red indicating agreements close to 100% and lighter shades of blue indicating small fractions of agreement.

5.6 Application

5.6.1 Characterizing the home-cage behavior of diverse inbred mouse strains

To demonstrate the applicability of this vision-based approach to large-scale phenotypic analysis, we characterized the home-cage behavior of four strains of mice, including the wild-derived strain CAST/EiJ, the BTBR strain, which is a potential model of autism [102] as well as two of the most popular inbred mouse strains C57BL/6J and DBA/2J. We video recorded $n = 7$ mice of each strain during one 24-hour session, encompassing a complete light-dark cycle. An example of an ethogram obtained over a 24-hour continuous recording period for one of the CAST/EiJ (wild-derived) strain is shown in Figure 5-15(H). One obvious feature was that the level of activity of the animal decreased significantly during the day ($12-24$ hr) as compared to night time ($0-12$hr). The mean activity peak of the CAST/EiJ mice shows a much higher night activity peak in terms of walking and hanging than any of the other strains tested (Figure 5-12). As compared to the CAST/EiJ mice, DBA/2J strain showed an equally high level of hanging at the beginning of the night time but this activity quickly dampened to that of the other strains C57BL/6J and BTBR.

We also found that the resting behavior of this CAST/EiJ strain differed significantly from the others: while all four strains tended to rest for the same total amount of time (except BTBR which rested significantly more than C57BL/6J), we found that the CAST/EiJ tended to rest for fewer but longer stretches. Their resting bouts(a continuous duration with one single predicted label) lasted almost three times longer than those of any other strain. (Figure 5-16(A-B)).

As BTBR mice have been reported to hyper-groom [102], we next examined the grooming behavior of BTBR mice. In the study of McFarlane *et al.* [102], grooming was who manually scored during the 10th-20th minute after placing mouse into a novel cage. Under the same condition, our system detected that the BTBR strain spent approximately 150 seconds grooming compared to the C57BL/6J mice which spend a little more than 85 seconds grooming. For a sanity check, two annotators, ('H','A'), are randomly selected from 'Annotators 1' (see Section 5.3.3) to annotate the same videos independently. The behavior difference detected by the system were able to be reproduced by both annotators (Figure 5-16(C)). Moreover, using annotator 'H' as ground truth, frame-wised accuracy of the system is 89% and frame-wised accuracy of annotator 'A' is 91%. This shows the system can detect grooming behavior with nearly human performance. Here we show that using our system we were able to reproduce the key results that the BTBR strain grooms more than the C57BL/6J strain when placed in a novel cage environment. Note that in the present study the C57BL/6J mice were approximately 90 days old(+/- 7 days) while the BTBR mice were approximately 80 days old (+/-7 days). In the McFarlane study younger mice were used (and repeated testing was performed), but our results essentially validate their findings.

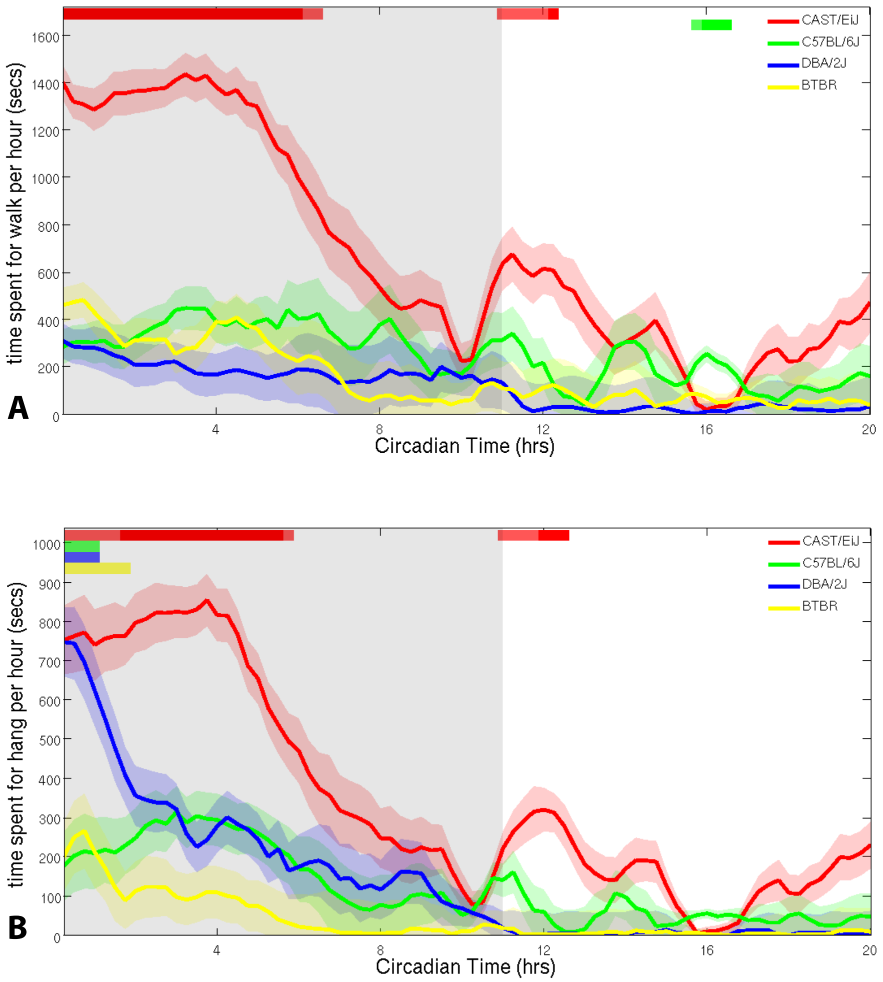


Figure 5-12: Average time spent for (A) 'hanging' and (B) 'walking' behaviors for each of the four strains of mice over 20 hours. The plots begin at the onset of the dark cycle, which persists for 11 hours (indicated by the gray region), followed by 9 hours of the light cycle. For (A), at every 15 minute of the 20-hour length, we compute the total time one mouse spent for 'hanging' within a one-hour temporal window centering at current time. For (B), the same procedure as in (A) is done for 'walking' behavior. The CAST/EiJ (wild-derived) strain is much more active than the three other strains as measured by their walking and hanging behaviors. Shaded areas correspond to 95% confidence intervals and the darker line corresponds to the mean. The intensity of the colored bars on the top corresponds to the number of strains that exhibit a statistically significant difference (*P¡0.01 by ANOVA with Tukey's post test) with the corresponding strain (indicated by the color of the bar). The intensity of one color is proportional to $(N-1)$, where N is the number of groups whose mean is significantly different from the corresponding strain of the color. For example, CAST/EiJ at hour $0-7$ for walking is significantly higher than the three other strains so N is 3 and the red is the highest intensity.

5.6.2 Patterns of behaviors of multiple strains

To demonstrate the patterns of behaviors can be used to characterize mice strains. We experimented with system predictions for the 7 24-hour video for the 4 mice strains as described in Section 5.6.1. Patterns of behaviors were computed from the system output by segmenting the predictions for each 24-hour video into 4 non-overlapping 6-hour long segments (corresponding to the first and second halves of the night, first and second halves of the day, respectively) and calculating the histogram of 8 types of behaviors for each segment. The resulting 8-dimensional (one for each behavior) vectors of the 4 segments were then concatenated to obtain one single 32-dimensional vector (8 dimensions 4 vectors) as pattern of behavior for each animal. The pattern of behavior corresponds to the relative frequency of each of the 8 behaviors of interest, as predicted by the system, over a 24-hour period.

To visualize the data, we computed dissimilarity of behavioral pattern between all pairs of animals by calculating the Euclidean distance between all pairs of 32-dimensional vectors. The Euclidean distance is then scaled by non-metric Multidimensional Scaling (MDS) analysis. MDS is a common statistical technique for visualizing dissimilarity of data. It takes as input point-point similarities, and assigns each point a new location in a N-dimensional space such that the relative point-point distance is maintained. Here we choose $N = 3$. This analysis was done using the matlab command 'mdscale' with the Kruskal's normalized stress1 normalization criterion. Although in this relatively low dimensional space, individual animals tend to cluster by strains suggesting that different strains exhibit unique patterns of behaviors that are characteristic of their strain-type (Figure 5-16(D)). The exception is 2 BTBR mice that tended to behave more like DBA/2J.

To quantify this statement, we conducted a pattern classification analysis on the patterns of behaviors by training and testing a linear SVM classifier directly on these patterns of behaviors and their labels (as one of the 4 strains). This supervised learning procedure was conducted using a leave-one-animal out approach, whereby 27 animals were used to train a classifier to predict the strain of the remaining animal. The procedure was repeated $n = 28$ times, one for each animal. Accuracy for the 28 strain predictions is computed as:

$$\text{accuracy} = \frac{\text{\# total animals whose strain is predicted correctly}}{\text{\# total animals = 28}} \tag{5.9}$$

The SVM classifier is able to predict the genotype of individual animals with accuracy of 90% (chance level is 25% for this 4-class classification problem). Figure 5-16(E) shows a confusion matrix for the resulting classifier that indicates the probability with which an input strain (along the rows) was classified as each of the 4 strains (along the columns). For example, the value of 1 for C57BL/6J means that this strain was perfectly classified. The higher probabilities along the diagonal and the lower off-diagonal values indicate successful classification for all strains. Using a leave-one-animal-out procedure, we found that the resulting classifier was able to predict the strain of all animals with an accuracy of 90%.

5.7 Extension of the system to more complex behaviors and environments

5.7.1 Extension to complex behaviors

To train and evaluate the performance of the system we chose the eight behaviors described above to capture essentially all home-cage behaviors. We next asked if the system can be extended to other more complex behaviors based on motion features. No additional features have to be designed for the system to adapt to new actions and the system can automatically learn from examples of new behaviors. We demonstrate this by training and testing the system on a set of videos of mice interacting with ”low profile” running wheels (Figure 5-13(A)). The *wheel-interaction set* contains 13 fully annotated one-hour videos taken from six C57BL/6J mice. The four actions of interest are: ”running on the wheel” (defined as all 4 paws on the wheel and the wheel to be rotating), ”interacting with the wheel but not running” (any other behavior on the wheel), ”awake but not interacting with wheel”, and ”rest outside the wheel”. Snapshots of these actions are shown in Figure 5-13(A) and in the video `http://techtv.mit.edu/videos/5567`. Using the leave-one-video-out procedure and accuracy formulation as for the *full database*, the system achieves 92.8%

of accuracy. The confusion matrix is shown in Figure 5-13(B) and indicates that the system can discriminate between interacting with the wheel from running on the wheel. See also an online video `http://techtv.mit.edu/videos/5567` for a demonstration of the system scoring the wheel-interaction behaviors.

In order to understand how many annotated examples is required to reach this performance,we repeat the same experiment of varying the number of training examples as described in Section 5.5.3. For the leave-one-video-out experiment, using 2 minutes of annotation for each training video, the system achieves 90% of performance obtained using 30 minutes, as shown in Figure 5-9(B). Interestingly, although with different types of actions and different number of videos, the result for the *wheel-interaction set* matches that for the *full database*(Figure 5-9(A)). When training and testing on the same video of the *wheel-interaction set*, the system's accuracy keeps increasing and doesn't reach optimal even when all first 30 minutes are used for training, as shown in Figure 5-10(B). This may be due to the large within-class variation of the action ”awake but not interacting with wheel”: all the actions that are performed outside the wheel such as walking, grooming, eating, rearing all fall into this category. A mouse may perform ”awake but not interacting with wheel” in first 30 minutes in a way different from the way in the rest of the video.

5.7.2 Extension to more complex environments

For home-cage behavior detection, the two video databases used to train the system contain very little nesting materials, as shown in Figure 5-1. We next asked how the system would perform detecting behavior under more natural nesting conditions, including more bedding. Since this system relies on motion (as opposed to shape), which is mostly visible under partial occlusion, we expected that it could still perform well. For example, when grooming, a mouse sweeps its fore- or hind-limbs across the face or torso, which can still be recognized by the system as long as the limbs and face of the mouse is visible. To validate this point, we apply the system to a one-hour video taken from a cage (with more bedding than the videos in the *full database*) to demonstrate the recognition of actions remains robust. The two videos and predictions of the system are available online at `http://techtv.mit.`

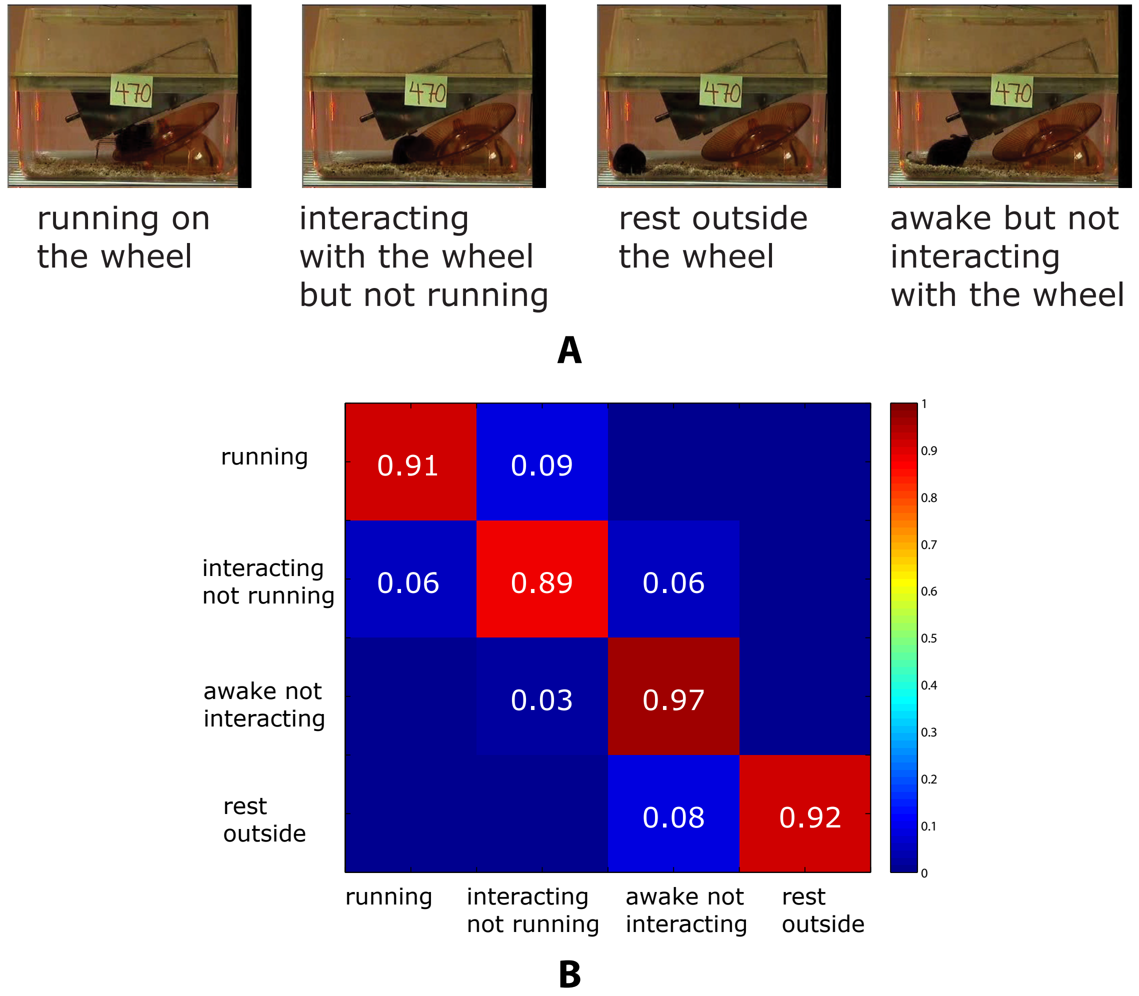


Figure 5-13: (A) Snapshots taken from the *wheel-interaction set* for the four types of interaction behaviors of interest: resting outside of the wheel, awake but not interacting with the wheel, running on the wheel, and interacting with (but not running on) the wheel. (B) Confusion matrices for system to human scoring.

`edu/videos/5566` and `http://techtv.mit.edu/videos/5565`.

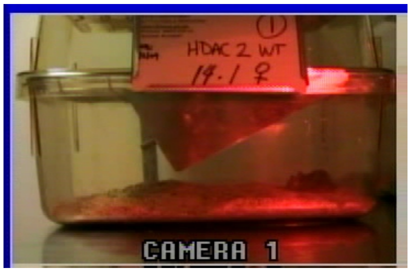


Figure 5-14: Snapshot taken from a one-hour video within natural home-cage environment.

5.8 Conclusion

In this chapter we describe the development and implementation of a trainable computer vision system capable of capturing the behavior of a single mouse in the home-cage environment. Importantly, as opposed to several proof-of-concept computer vision studies [36, 218], our system has been demonstrated with a ”real-world” application, characterizing the behavior of several mouse strains and discovering strain-specific features. We provide software as well as the large database that we have collected and annotated in hope that it may further encourage the development of similar vision-based systems. The search for ”behavioral genes” requires cost effective and high-throughput methodologies to find aberrations in normal behaviors [190]. From the manual scoring of mouse videos described in Section 5.3.3, we have estimated that it requires about 22 person hours of work to manually score every frame of a one-hour video. Thus, we estimate that the 24-hour behavioral analysis conducted above with our system for the 28 animals studied would have required almost $15,000$ person hours (i.e., almost 8 years of work for one person working full-time) of manual scoring. An automated computer-vision system permits behavioral analysis that would simply be impossible using manual scoring by a human experimenter. The system is implemented using GPU (graphical processing unit) based on a framework of (Mutch & Poggio, in prep) and performs in real time for the computation of motion and position- and velocity-based features (it takes about 1 second to process 30 frames).

In principle, our approach can be extended to other behaviors such as dyskinetic behaviors in the study of Parkinson’s disease models, seizures for the study of epilepsy, mice with bipolar disorder. Future developments of our learning and vision approach could deal with the quantitative characterization of social behavior involving two or more freely behaving animals. This will require a tracking module for identifying location and identity of each mouse prior to recognition of its behavior. In conclusion, our study shows the promise of learning-based and vision-based-techniques in complementing existing approaches towards a complete quantitative phenotyping of complex behavior.

5.9 Future work

Our system can be extended to recognize behaviors of multiple animals by adding a tracking module for identifying the location of each individual animal prior to the recognition of behaviors. Depending on whether a group of mice are interacting (from their relative positions and shapes), the system either recognizes actions of individual animal, or social actions of the group. Under the conditions of multiple mice in a cage, it has been shown tracking identity of three mice can be achieved by simple heuristic rules [15]. In the more complicated case when multiple animals exist and stack on or spin with each other, the main difficulty will be occlusions aroused from nesting materials and other animals. To deal with occlusions, we will use at least two cameras: one from top view for better tracking of animal positions and one from side view for seeing contour (limbs and body) and motion of mouse. Mouse identity can be inferred by combining positions computed from these two sources of images [218]. Under the complicated experimental setting with multiple animal and multiple cameras, an abundant amount of accurate annotations will be a key for the success of the system. The scoring process under multiple camera/mice will become more time-consuming, therefore it is critical to develop an annotation tool that makes the best use of our current system, meaning the trained system can predict all the labels in advance so annotators only have to correct wrong predictions. The system can also be extended to do incremental learning [25]: during the process of annotation, the system simultaneously learns from examples that were corrected by annotators to make more accurate predictions for the subsequent frames.

System label	HCS label
drink	drink
eat	eat chew
groom	groom
hang	hang cuddle hang vertically hang vertically from hang cuddled hang vertically from rear up remain hang cuddled remain hang vertically
micro-movement	awaken pause remain low sniff twitch
rear	come down come down from partially reared come down to partially reared stretch body, land vertically rear up rear up from partially reared rear up to partially reared remain partially reared remain rear up
rest	sleep stationary
walk	circle turn walk left walk right walk slowly
not processed	dig forage jump repetitive jumping unknown behavior urinate

Table 5.7: Matching between 8 types of labels in our system and labels in the HomeCageScan.

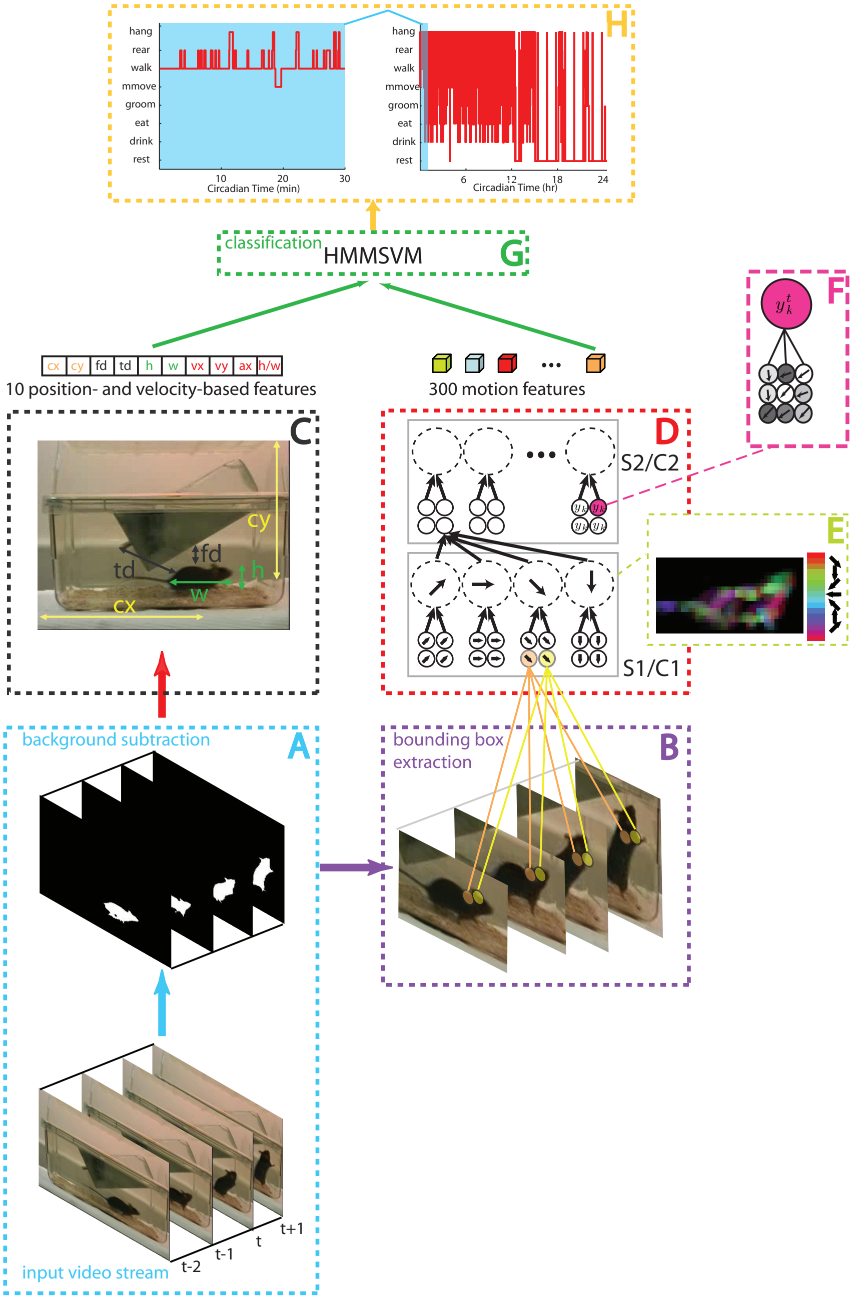


Figure 5-15: Overview of the proposed system for recognizing the home-cage behavior of mice. The system consists of a feature computation module(A-F) and a classification module(G). (A) The background subtraction technique is performed on each frame to obtain a foreground mask. (B) A bounding box centering at the animal is computed from the foreground mask. (C) Position- and velocity-based features are computed from the foreground mask. (D) Motion-features are computed from the bounding-box within a hierarchical architecture (D-F).(G) HMMSVM. (H) An ethogram of time sequence of labels predicted by the system from a 24-hr continuous recording session for one of the CAST/EiJ mice. The right panel shows the ethogram for 24 hours, and the left panel provides a zoom-in version corresponding to the first 30 minutes of recording. The animal is highly active as a human experimenter just placed the mouse in a new cage prior to starting the video recording. The animal's behavior alternates between 'walking', 'rearing' and 'hanging' as it explores its new cage.

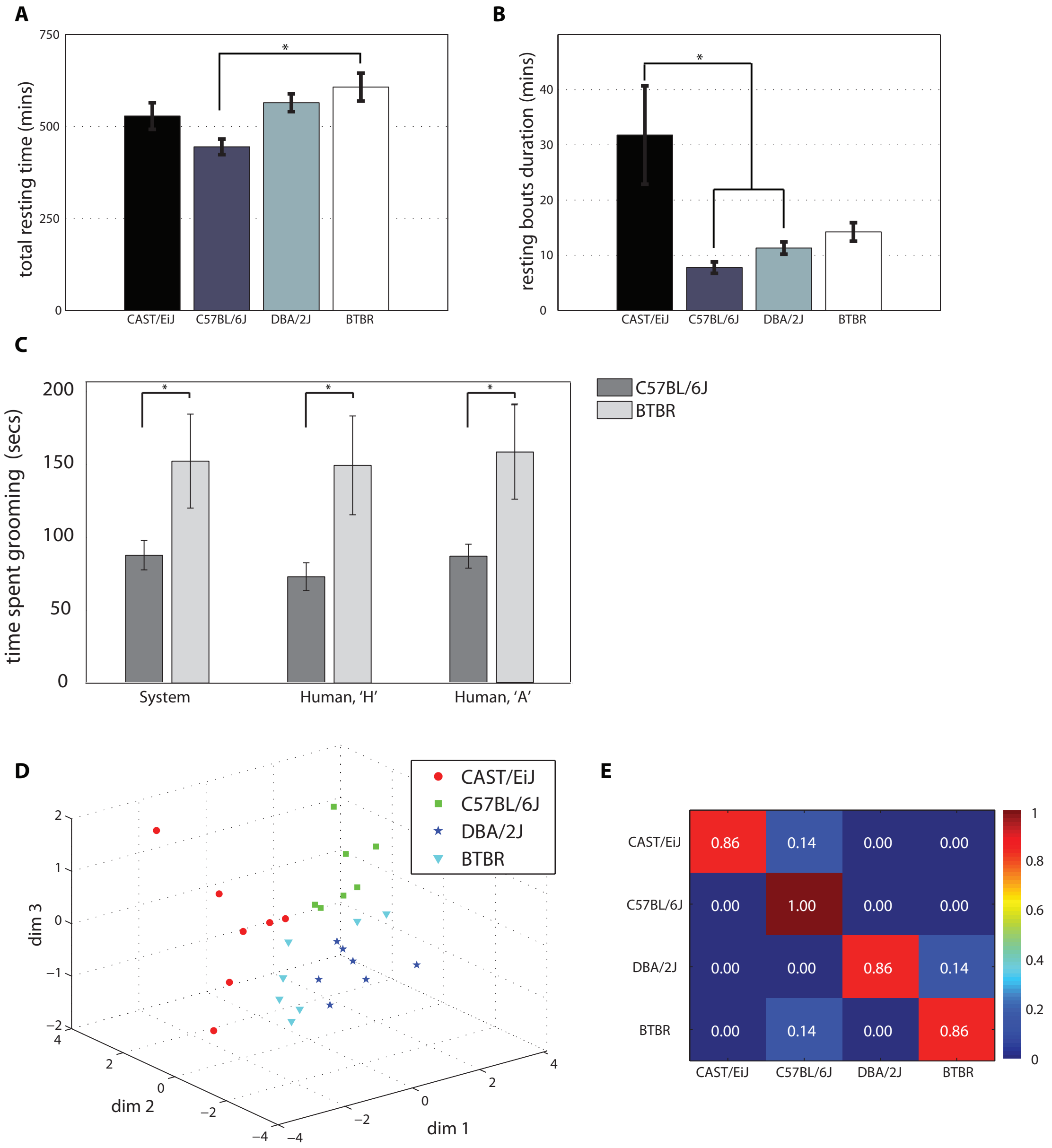


Figure 5-16: (A) Average total resting time for each of the four strains of mice over 24 hours. (B) Average duration of resting bouts (defined as a continuous duration with one single label). Mean +/- SEM are shown, $*P < 0.01$ by ANOVA with Tukey's post test. (C) Total time spent for grooming exhibited by the BTBR strain as compared to the C57BL/6J strain within 10th-20th minute after placing the animals in a novel cage. Mean +/- SEM are shown, $*P < 0.05$ by Student's T test, one-tailed. (P = 0.04 for System and P =0.0254 for human 'H', P = 0.0273 for human 'A'). (D-E) Characterizing the genotype of individual animals based on the patterns of behavior measured by the computer system. (D) Multi-Dimensional Scaling (MDS) analysis performed on the patterns of behaviors computed from the system output over a 24-hour session for the 4 strains. (E) The confusion matrix for the SVM classifier trained on the patterns of behavior using a leave-one-animal out procedure.

Chapter 6

Towards a Biologically Plausible Dorsal Stream Model

This chapter is now under preparation for a journal submission.

Abstract

A substantial amount of data about the neural substrates of action recognition and motion perception is accumulating in neurophysiology, psychophysics and functional imaging, but the underlying computational mechanisms remain largely unknown, and it also remains unclear how different experimental evidence is related. A computational model constrained by experimental results will help organize the known physiological facts as well as suggest novel experiments and predict the neuronal responses, which, if verified, could be used to further refine or constrain the model.

In this work we present a hierarchical model for the motion processing in the dorsal stream and action selectivity in the area STP. This model has been shown to perform on par or outperforms computer vision algorithms for the recognition of human actions [75] as well as mice behaviors in videos [73]. By comparing the model outputs with the neuronal responses, we show that the model can explain motion processing in the area V1 and area MT as well as action selectivity in the area STP. Specifically, the first two layers of the model match the spatial and temporal frequency tuning of V1 cells. The latter two layers match the distribution of pattern and component sensitivity [115], local motion integration [97], and speed-tuning [144] of MT cells. The model, when combining with the ventral stream model [173], could also explain the action and actor selectivity in the STP area, a high level cortical area receiving inputs from both the ventral and the dorsal stream.

6.1 Introduction

6.1.1 Organization of the dorsal stream

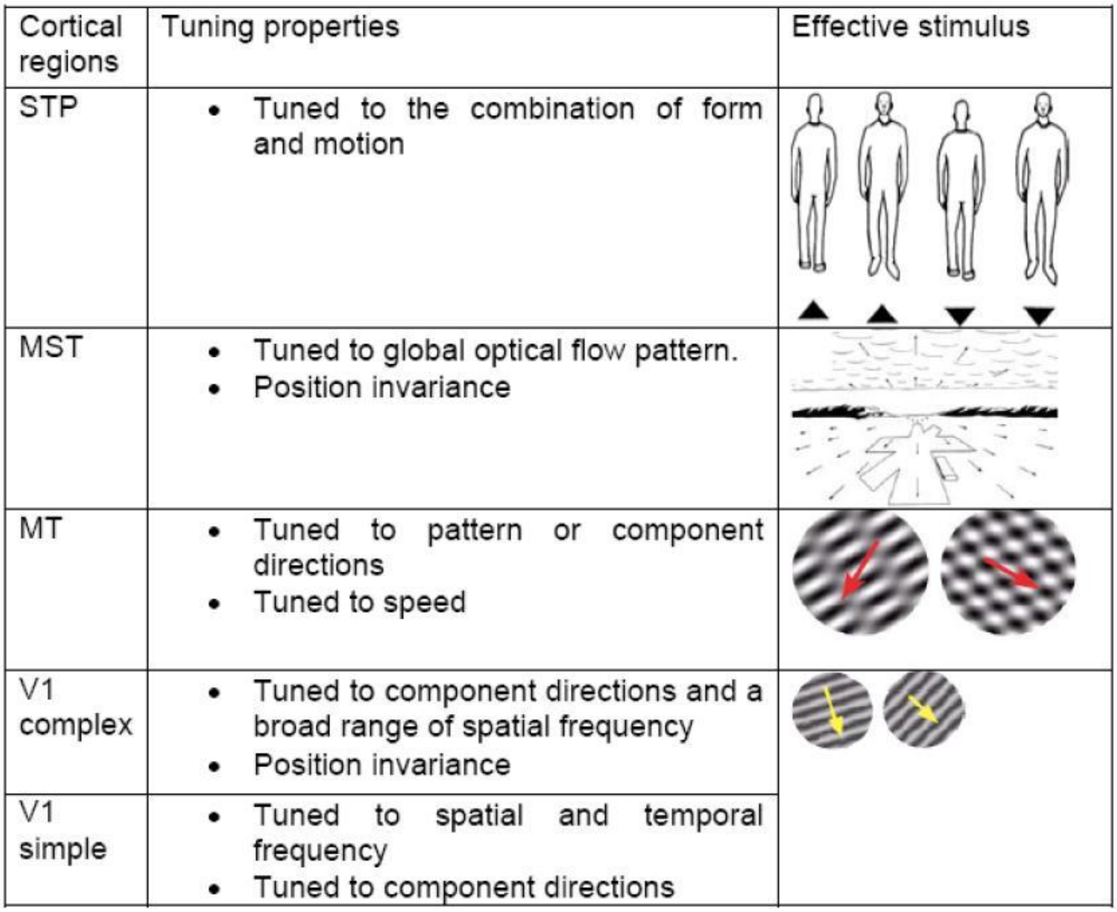
The dorsal stream is a functionally specialized pathway for processing visual signals received from retina. It is called ”where pathway” because it is involved in space perception, such as measuring the distance to an object or the depth of a scene. It is also called ”motion pathway” because it is involved in the analysis of motion signals [202, 54], such as perception of motion and recognition of actions. In this chapter we will focus on the motion aspect.

The dorsal stream starts at direction selective cells in the primary visual cortex (V1) [62]. These cells then project to middle temporal area (MT/V5) [203], where most of the neurons are direction and speed sensitive and the receptive fields are $2-3$ times larger than the V1 afferents [107]. MT neurons then project to the medial superior temporal area (MST), where neurons are tuned to complex optical-flow patterns over a large portion of the visual field, and are invariant to the position of the moving stimulus [56]. The dorsal stream is thought to include area V1, MT and MST. Dorsal stream signals are then integrated with ventral stream signals, specifically from inferior temporal cortex (IT), at the superior temporal poly-sensory area (STP). Figure 1-1 indicates the locations of these areas in the dorsal stream. Table 6-1 lists the neuronal tuning properties and illustrates effective stimuli in these areas.

6.1.2 Motion processing in the dorsal stream

The primary visual cortex (V1) is the first area of the dorsal stream, V1 neurons are more studied than higher-level neurons because of their relative simpler RF structures and functions.

V1 simple cells The striate neurons are diverse in terms of receptive field sizes, structures and functions. Simple cell receptive field contains oriented excitatory regions in which presenting an edge stimulus excites the cell and inhibitory regions in which stimulus presentation suppresses responses. The cells respond to oriented stimuli (gratings, bars) whose



Cortical regions	Tuning properties	Effective stimulus
STP	- Tuned to the combination of form and motion	▲ ▲ ▼ ▼
MST	- Tuned to global optical flow pattern. - Position invariance	
MT	- Tuned to pattern or component directions - Tuned to speed	
V1 complex	- Tuned to component directions and a broad range of spatial frequency - Position invariance	
V1 simple	- Tuned to spatial and temporal frequency - Tuned to component directions	

Figure 6-1: The neuronal tuning properties in the dorsal stream and the effective stimuli.

orientation matches that of subregions [68, 70, 163]. There is a long tradition in which simple receptive fields are modeled as linear functions; meaning the response of the cells are a weighted sum of the light intensity distribution of the stimuli.

A subset of V1 simple cells are direction selective (DS). This subset is thought to constitute the first layer of motion processing in the visual system. In these cells, the spatial receptive field changes over time in a way that the subregions are oriented when plotted in the space-time domain (Figure 6-2). Translating stimulus can also be pictured as occupying a space-time orientation [1, 64]; the orientation uniquely determines the speed and direction of the stimulus. Therefore a space-time-oriented simple RF allows the cell responding to motion characterized by the same space-time-orientation [33] and having velocity preference. Moreover, DS simple cells are tuned to the spatial and temporal frequency of the stimulus [47, 96, 114]. The elongated receptive field structure [33] and the tuning to spatio-temporal frequency can be well approximated by localized spatio-temporal filters [78, 222, 64, 177], or learned from natural images [149, 128].

While many aspects of simple cells' responses are consistent with the linear model, there are also violations of the linearity. For example, the responses of a cell scale linearly with the contrast of the stimulus, but saturate at high contrast [2, 3]. Moreover, the responses to an optimally-oriented stimulus can be diminished by superimposing an orthogonal stimulus that is ineffective in driving the cell when presented alone. This phenomenon is called ”cross-orientation-suppression” [32, 143]. Linearity alone also fails to account for direction selectivity of simple cells. In order to account for these nonlinearities, the linear model was extended with rectification and normalization operations [113, 65, 22, 21, 139]. For example, Heeger used divisive normalization in which the response of a cell is normalized by the summed responses of a pool of cells [65, 22, 21]. The pool includes cells tuned to a range of spatial frequencies and directions in order to account for the directional and spatial frequency tuned suppression signals [32].

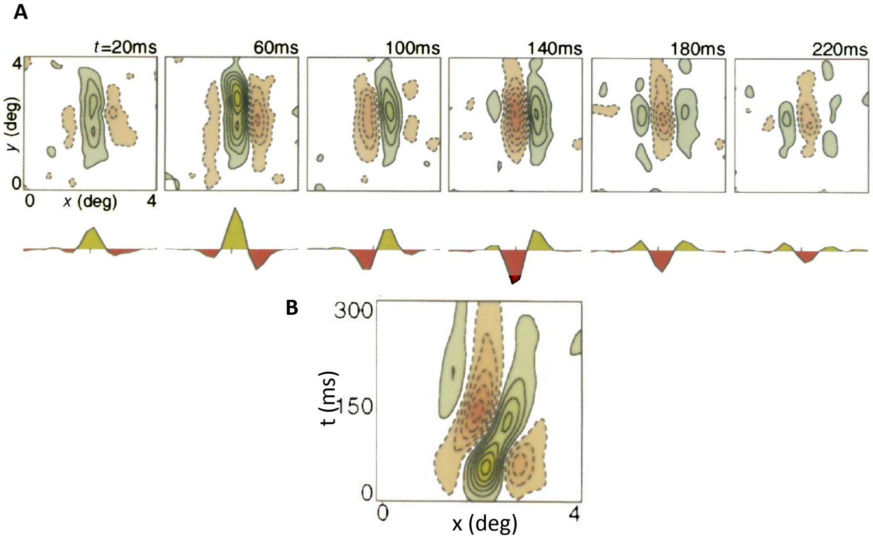


Figure 6-2: A. Dynamics of receptive field of directional selective striate simple cells. Below each contour plot is a 1D RF that is obtained by integrating the 2D RF along the yaxis, which is parallel to the cell's preferred orientation. B. Spatiotemporal receptive field for the same cell. The figure is modified from [33].

V1 complex cells The receptive field of DS complex cells can not be mapped out by the responses to a single stimulus because the cells are insensitive to the polarity and spatial position of the stimulus [68, 70, 163]. The receptive field structure is investigated using two moving bars and the reversed correlation technique, which together revealed that the underlying subunits are elongated and oriented [116], suggesting simple cells as inputs to complex cells. Combined with the fact that complex cells have broader tuning for spatial frequency and larger receptive field than simple cells (around $2 - 3$ times), it is generally accepted that complex cells combine multiple simple cells that are tuned to the same direction and a range of spatial frequencies over a localized spatial region[144]. Such a combination was modeled as a max-pooling [52, 84, 46, 79], a linear weighting [177], or it could be learned from natural image sequences [100, 101]. Adelson & Bergen's energy model and the extended Reichardt detectors were often used to model DS complex cells [151, 1, 207, 40]. The max-pooling was supported by physiological experiments [84, 46], and it can also be approximated with the energy model under some conditions [46].

MT and MST cells Beyond the primary visual cortex, the processing of motion becomes complex within a large receptive fields (MT RF size is least 10^o). In MT area, receptive field structures are rarely studied with a few exceptions [45, 93]. Most of the MT neurons are tuned to direction and speed of motion [4, 83], but these two tuning are not independent properties [83, 155], and each of them also depends on other factors. For example, direction tuning changes with speed [126, 83], spatial configuration [97], spatial frequency [107], and complexity of the stimulus [115]. Speed tuning also changes with the spatial frequency of the stimulus [134, 142, 144]. In Section 6.3.1, we will describe the direction and speed tuning of MT cells. The MST is sometimes divided into MSTI where cells are also tuned to directions like MT afferents, and MSTd where cells are tuned to large optical flow patterns such as spiral motion [56].

6.1.3 The problem

A substantial amount of data about the neural substrates of action recognition and motion perception is accumulating in neurophysiology, psychophysics and functional imaging, but the underlying computational mechanisms remain largely unknown, and it also remains unclear how different experimental evidence is related. A computational model constrained by experimental results will help organize the known physiological facts as well as suggest novel experiments and predict the neuronal responses, which, if verified, could be used to further refine or constrain the model. There are indeed many computational models for motion processing, motion perception, or visual attention in the dorsal stream [177, 197, 157, 58, 18, 20]. On the other hand, computer vision systems have been developed to mimic the functions of the human visual system, such as recognition of faces, objects and actions. However, modeling visual processing in brains and computer vision systems for recognition have been mostly developed independently. We believe an ideal computational model for the visual system should also be applicable to computer vision tasks, and vice versa.

HMMAX is such a model that could explain neurophysiology, human psychophysics as well as recognition of objects. HMAX was built partially based on neuronal recording from V1 and partially on predictions that specificity and invariance are gradually built up along the ventral stream hierarchy with repetitive simple and complex operations. In Chapter 2, we described the extension of HMAX along the time domain to represent actions in videos, and showed that the outputs of the model could be used for recognition of human actions (Chapter 3, Chapter 4) as well as mice behaviors (Chapter 5).

In this chapter we will answer if our proposed model (Chapter 2) for the recognition of actions could also explain physiology in the dorsal stream. In particular, the first two layers (S_1/ C_1) of the model were designed to closely follow the known receptive field profiles of DS V1 cells, and our main goal is to test our prediction that the next two layers (S_2/ C_2) could model the downstream MT cells. We will also go beyond the dorsal stream and try to model the responses of STP neurons. Some STP neurons are shown to be selective to actors, actions, or their combinations, and therefore closely related to the recognition of actions [178]. These types of selectivity were believed to be the result of receiving shape features from ventral afferents and motion features from dorsal afferents. We will propose a way these two types of features are integrated and compare the results with the tuning of STP neurons.

6.1.4 The interpretation of velocity in the frequency domain

Perception of the velocity of moving objects is essential for extracting information about the surrounding environment. For example, animals need to estimate the speed and direction of other species in order to capture prey or avoid being captured. The definition of speed is the total distance traveled per unit of time. This computation is involved in tracking a particular point over time (solving a correspondence problem), then compute the delay and distance. This seems to be implausible within neurons' local spatio-temporal receptive field. Fahle and Poggio [42] and Adelson and Bergen [1] have pointed out that in order to understand neuronal processing in the visual system, motion is better characterized as orientation in space-time, where orientation is a function of direction and speed.

Motion processing in primates starts in the striate cortex, where a group of neurons are tuned to orientations and directions perpendicular to the orientations, and are thought to be pre-processors for extracting motion. The striate neurons process input signals within a spatially localized region and a restricted window of time, and the spatial structure of receptive fields changes as a function of time. For direction-selective cells, the spatial receptive field changes in a way that the ON/OFF subregions are oriented in space-time (Figure 6-2). This allows the cells responding to motion characterized by the same space-time-orientation [33]. Indeed, a simple cell's preferred direction and speed of motion can be predicted reliably from the structure and the slope of the oriented subregions in the space-time domain [32, 103, 104]. The orientation in the space-time domain could be translated as orientation in the Fourier domain.

Here we review the analysis by Watson and Ahumada in [212] (The Fourier analysis of motion has also been discussed in [42, 64]). Consider a two dimensional pattern translating at a constant velocity in a two-dimensional space (x-y). The trajectory of the image can be written as

$$c(x, y, t) = c(x, y)\delta(x - v_x t)\delta(y - v_y t) \tag{6.1}$$

where x and y are vertical and horizontal image coordinates and t is time. The image intensity distribution at time 0 is $c(x, y)$, and v_x and v_y are the velocity in the x and y dimension.

After applying the Fourier transform,

$$\begin{aligned} C(w_x, w_y, w_t) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(x, y, t) exp^{(-i2\pi(xw_x+yw_y+tw_t))} dxdydt \quad (6.2) \\ &= C(w_x, w_y)\delta(w_t + w_x v_x + w_y v_y) \quad (6.3) \end{aligned}$$

the spatial frequency along the x and y axis (w_x and w_y), and temporal frequency (w_t) lie on a common plane ($w_t + w_x v_x + w_y v_y = 0$) in the frequency domain ($w_x - w_y - w_t$), as shown in Figure6-3. Here w_x and w_y are defined in cycles per degree, and w_t is defined in cycles per second.

Consider a one-dimensional signal (such as edges, bars, gratings) moving along the x axis ($v_y = 0$), the frequency spectrum is a line $w_t + w_x v_x = 0$ with a slope $\frac{-1}{v_x}$ in the $w_t - w_x$ space, as shown in Figure 6-8D. In other words, in the case of one-dimensional motion, the speed of the image can be interpreted as the ratio of the spatial to the temporal frequency of the image. A typical example of 1D motion is the motion of sine-wave grating, whose spatial frequency is the inverse of the width of a single sinusoidal cycle and temporal frequency is the inverse of the time required for a single pixel to go through a single sinusoidal cycle.

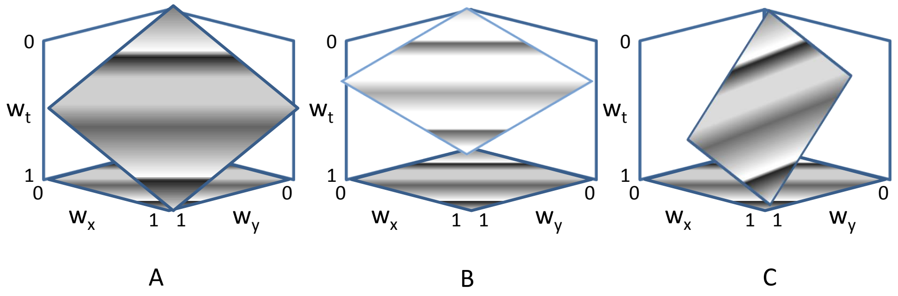


Figure 6-3: The surface indicates the Fourier spectrum of objects translated at a particular velocity. The slant of the plane from the floor is proportional to the speed of motion. The tile of the plane relative to the spatial frequency axis is equal to the direction of motion. The greater slant of A as compared to B indicate a faster speed in A. The motion in A and C have identical speeds but different directions. Figure modified from [59].

6.2 The Model

The general structure of the proposed hierarchical model has been described in Chapter 2. Here we describe the detailed implementation of each stage and their biological correlates.

S_1 **units** S_1 units are designed after the direction selective simple V1 neurons. Let I denote the light intensity distribution of a stimulus, f_i denote the receptive field profile of the $i-th$ S_1 unit. The response S_{1i} is the linear convolution of the stimulus with f_i followed by Heeger's normalization model [65, 22, 21]. Prior to the convolution, the stimulus was normalized to have unit average intensity [177].

$$\begin{aligned} L_i &= f_i \times I & (6.4) \\ S_{1i} &= \frac{L_i^2}{\sum_i L_i^2 + \beta} & (6.5) \end{aligned}$$

In Heeger's normalization model, the linear response (L_i) is squared and then divided by the pooled responses of a large number of cells. The squaring operation was shown to approximate the transformation from the membrane potential to the spike rate [2, 166, 21]. The divisive normalization could account for the nonlinearity and dynamics of simple cell responses [65]. Here the pool contains cells of the same receptive field tuned to 16 different directions equally spaced in the angular space (between 0 and 2π). β is the saturation constant. The normalization model belongs to a big class of canonical-models which could be implemented with neuronal circuits (sigmoid-like model in [80]).

In some works, the transformation from the membrane potential to the spike rate is modeled as rectification. The rectification operation and the squaring operation are similar under some conditions [65, 22, 21].

Here each S_1 unit's receptive field is modeled as a three-dimensional Gabor filter tuned to a particular speed (v) and direction θ.

$$
\begin{aligned}
f(x,y,t) &= exp(-\frac{(x'+vt)^2+\gamma^2 y'^2}{2\sigma^2})\cdot cos(\frac{2\pi}{\lambda}(x'+vt)+\xi)\cdot exp(-\frac{t^2}{2\tau^2}) && (6.6)\\
x' &= xcos\theta + ysin\theta && (6.7)\\
y' &= xsin\theta + ycos\theta && (6.8)
\end{aligned}
$$

In this equation, γ represents the spatial aspect ratio, λ and τ control the spatiotemporal period (inversely proportional to the preferred spatiotemporal frequency) of the filter, λ is also a function of the preferred speed of the filter. Each filter is normalized to be zero mean and unit L_2 norm.

The spatial size of V1 receptive fields has been approximated as a linear function of the eccentricity [107]. Here we model cells of RF sizes $0.6^o - 3.4^o$ at eccentricity $2^o - 15^o$. Let a typical video resolution 240×360 pixels correspond to 45 degrees of a visual field, the RF sizes will correspond to filter sizes $5 - 27$ pixels. λ is set to be proportional to the filter size, therefore a range of filter sizes that were designed to capture motion occurred at different scales (sizes of moving objects) will respond as well to a range of spatial frequencies.

The temporal resolution of a typical simple cell is $300(ms)$ [33], corresponding to 9 frames for a typical video frame rate $29 - 35$ fps. The model contains 12 sizes of filters, see Table 6.1 for a list of parameters.

C_1 **units** C_1 units mimic the tolerance of V1 complex cells to the shift of the stimulus' position and size by computing a maximum response over S_1 units of adjacent two scales in a local spatial region. The spatial pooling size is designed to be at least half of the S_1 filter size, and the pooling step at least half of the pooling size. The parameters of C_1 units are listed in Table 6.1.

Table 6.2 compares a set of basic tuning properties of our S_1 and C_1 population to DS V1 cells. The S_1 and C_1 units match quite well with V1 cells in terms of the tuning to spatial frequency, temporal frequency, and direction.

Table 6.1: Parameters of S1/C1 units. Here λ is set for a preferred speed of 1 pxs/frame.

scale	S1 size	λ	C1 pooling size	c1 pooling step size
1	5×5	3.5	4	3
2	7×7	5.3		
3	9×9	7.1	6	3
4	11×11	8.8		
5	13×13	10.6	8	5
6	15×15	12.3		
7	17×17	14.1	10	5
8	19×19	15.9		
9	21×21	17.7	12	7
10	23×23	19.4		
11	25×25	21.2	14	7
12	27×27	23.0		

Table 6.2: Tuning properties of S1/C1 units and V1 cells

Tuning Property		S1/C1 units	V1 cells	eccentricity	Reference
RF size	range	0.6-3.4 ^{o}	0.6-4^o	2-15^o	[107]
	mean	2^o	2.2^o		[107]
peak spatial frequency	range	0.4-4.4 $c/^o$	<0.75-6 $c/^o$	1.5-3.5^o	[62]
	range	0.4-4.4 $c/^o$	0.5-8 $c/^o$	2-5^o	[47]
	mean	1.4 $c/^o$	2.2 $c/^o$		[47]
peak temporal frequency	range	1.0-15 c/s	0.5-12 c/s	2-5^o	[47]
	mean	4.6 c/s	3.7 c/s		[47]
spatial frequency bandwidth	range	0.5-4.1 octave		2-5^o	[47]
	mean	1.35 octave	1.8 octave		[47]
temporal frequency bandwidth	range	0.5-5.8 octave			
	mean	3.14 octave	2.9 octave	2-5^o	[47]
direction bandwidth	range	45-92^o			
	mean	52^o	68^o		[4]

S_2 **units** We conjecture S_2 and C_2 units could model MT cells because they both integrate directions and spatiotemporal frequencies extracted from a previous stage (Details in Section 6.3.6). Each S_2 unit computes a template matching between inputs and a stored template using a normalized dot product operation (linear kernel).

Each template is normalized to be zero mean in order to account for the suppressive signals within MT receptive field. The majority of MT neurons respond vigorously to stimuli moving in the preferred direction and the responses are suppressed below the spontaneously firing rate when stimuli move in the opposite direction [106]. The responses of a MT cell to random dots moving in its preferred direction can also be suppressed by superimposing random dots moving in the opposite direction [180, 145]. This phenomenon is so called ”motion opponency”. These suppressive signals are present in a local space (they don’t suppress preferred motion far apart in space) and tuned to direction [180] (See [160] for a discussion about the possible origin of suppressive signals). In previous MT models, these suppressive signals were accounted by negative synaptic weights or divisive normalization[177, 94, 52, 161]. In our proposed model, C_1 responses are by default positive, due to the squaring operation in the S_1 stage, we therefore ”generate” negative weights by subtracting the mean of each template. The zero-mean normalization of templates is indeed necessary for modeling direction tuning of MT neurons, as demonstrated in Section 6.3.6.

Here the templates were randomly sampled from C_1 outputs of training sequences. It could also be obtained by learning rules such as Hebbian rule [168] or spike timing dependent plasticity (STDP) [181, 101]. In Lecun’s convolutional networks, weights along the hierarchy are learned through back propagation [88].

For the experiments in this chapter, we sampled S_2 templates from random scales.

C_2 units C_2 units perform a max-pooling operation over S_2 units within its receptive field and over all the S_2 scales. Each C_2 unit’s receptive field is designed to be 189 pixels, corresponding to a 20^o of visual field.

6.3 Comparing C2 units with MT cells

6.3.1 Introduction: directional tuning

Here we reviewed a nice paper for the introduction of the directional tuning in the visual system by Movshon, Adelson, Gizzi and Newsome [115].

Aperture problem and the theoretical solutions Each cell has a specific receptive field, which defines the region of retina over which one can influence the firing of that cell. This receptive field can be treated as a small aperture through which the cell looks at the world. Within this aperture, the motion of a single extended contour (edge, line, linear border, bar) doesn't allow one to determine the motion of the object that contains that contour. For example, Figure 6-4A shows three objects containing an oblique grating moving behind a circular aperture. In all cases, the appearance of the grating, as seen through the aperture, is identical: the gratings appear to move up and to the left, normal to their orientation. This is because, as illustrated in Figure 6-4B, object velocity can be decomposed into two orthogonal vectors, one perpendicular to the orientation of the contour and one parallel to the contour. The parallel vector is invisible because one can not detect the motion of an edge along its own length, therefore we can only perceive the perpendicular vector. The computational problem of estimating the global motion direction of an object from the different local motions apparent through two or more apertures is called the aperture problem.

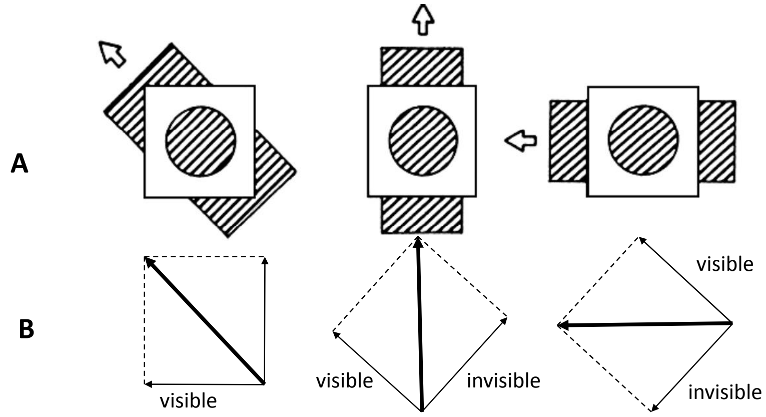


Figure 6-4: A. Three patterns moving in different direction produce the same physical stimulus, as seen through the aperture (Adapted from [115]). B. The object velocity can be decomposed into two orthogonal vectors, one perpendicular to the orientation of the edge and one parallel to the edge. C.

Assuming an object is translating in the image plane (linear motion), the motion is ambiguous when only one edge is visible, but two edges of the object with different orientations should be sufficient to determine its velocity. Consider the object shown in Figure 6-5A moving to the right. The top right edge of the object, appears to move up and to the right, as seen behind the aperture. The percept of the edge motion could be generated by any of the object motions shown by the arrows in Figure 6-5B. This is because motion parallel to the edge is not visible, so all motions that have the same component of motion perpendicular to the edge are possible candidates for the true motion of the object that contains the edge. The set of possible solutions for the true motion lies along a line in the velocity space, as shown in Figure 6-5B. In this velocity space, motion of two edges with different orientations correspond to two non-parallel lines, whose intersection satisfies both constraints and corresponds to the true motion of the object that contains both edges. This is so called intersection of constraints (IOC).

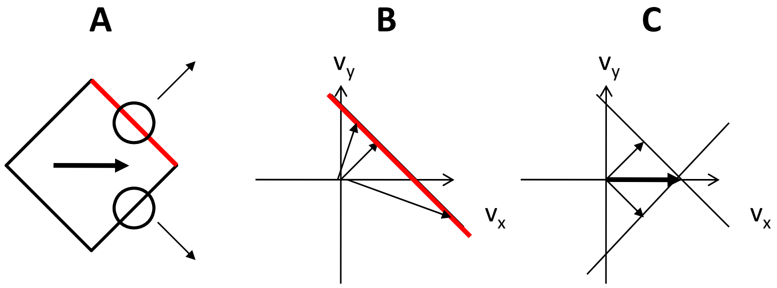


Figure 6-5: A. An object moves to the right, one if its border (colored in red) appears to moves up and to the right behind the aperture. B. Each arrow is a velocity vector that generates the percept of the red border in A, and the set of possible velocity vectors lies along a line (colored in red) in the velocity space (Adapted from [115]). C. Each border of the object provides a constraint- a line in velocity space, and the intersection of the two lines represent the unique true motion of the object that contains both edges.

Solving the aperture problem in the visual system Since most neurons in the primary visual cortex have relatively small receptive fields, they confront the aperture problem when an object larger than their receptive field moves across the visual field. How does the visual system solve the aperture problems and perceive true motion of the objects in this world? Tony Movshon, and his colleagues proposed a two stage model. The initial stage performed orientational filtering. In this stage orientation tuned cortical neurons respond to components of motion perpendicular to their preferred orientation. In the second stage, high-order neurons integrate the component motion analyzed by the first stage and infer the true motion. The hypothesis that motion information in the visual system is processed in two stages was tested by physiological experiments described as below.

In the experiments of Movshon et al [115], two kinds of stimuli were used to investigate how visual system analyze motion: sine wave grating and sine wave plaids. The grating moves in the direction perpendicular to its orientation, and therefore exhibits the ambiguous motion as shown in 6-5B. A sine wave plaid is composed of two overlapping gratings at different orientations, each moving in a direction perpendicular to its orientation. The two gratings have the same spatial frequencies and move at the same speed. The motion of the plaid could be uniquely determined by IOC solutions. Because the two component gratings have identical spatiotemporal properties, the IOC solution is simply the summation of the two velocity vectors.

Two types of directional selectivity have been defined in visual electrophysiology: *component directional selectivity (CDS)* and *pattern directional selectivity (PDS)*. CDS cells respond to the motion of single oriented contours, whether they are presented in isolation or embedded in a more complex 2-dimensional pattern, such as plaids. PDS cells respond to the direction of motion of the overall pattern; therefore, theoretically they have responses that are identical for a grating and a plaid moving in the same direction, even though the underlying gratings have different motion.

Figure 6-6 illustrates the responses of a hypothetical directional-selective cell to a grating and a plaid. The direction tuning curve is shown as a polar plot, in which the moving direction of the stimulus is given by the angle, and the cell's response is given by the radial distance to the origin. When presenting a grating, the direction tuning curve peaks when the grating moves in the optimal direction of the cell, as shown in Figure 6-6C. A cell can be classified as PDS or CDS based on its responses to the plaid.

If the cell is CDS (tuned to the individual motion of gratings), the response to the plaid could be predicted by combining the responses to two gratings that are presented separately. Let α be the plaid angle, defined as the angular difference of the two gratings' directions. In other words, the component gratings move in $\alpha/2$ degrees to either side of the plaid direction. The prediction for the direction tuning curve to a plaid could therefore be obtained by summing the direction tuning curve to the two gratings, each shifted to either side by $\alpha/2$ degrees, resulting in a bi-lobed direction tuning curve. Let $y_c(\theta)$ be the response to a grating moving in direction θ, the response of a CDS cell to the plaid could be predicted as:

$$\hat{y_c}(\theta) = y_c(\theta - \frac{\alpha}{2}) + y_c(\theta + \frac{\alpha}{2}) \tag{6.9}$$

If the cell is PDS (tuned to the overall motion of the stimulus), its responses to the plaid could be predicted to be identical to the responses to the grating:

$$\hat{y_p}(\theta) = y_c(\theta) \tag{6.10}$$

Figure 6-6 D illustrates the CDS prediction (bi-lobed solid line) and PDS prediction (single-peaked dashed line).

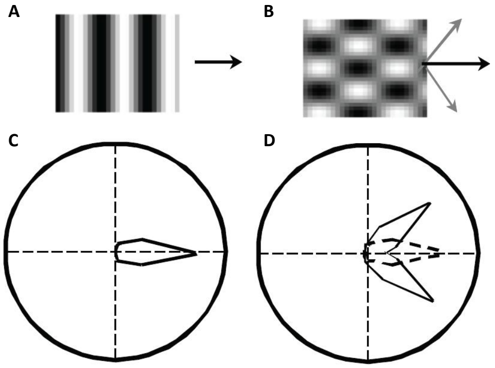


Figure 6-6: Stimulus and direction tuning curve of a hypothetical directional-selective cell. A. sine wave grating. B. sine wave plaid. C. direction tuning curve to a grating D.ideally predicted direction tuning curve to a plaid, a cell is classified as pattern directional selective if the tuning curve is the same as to the grating (shown in dashed line). A cell is classified as pattern directional selective if the tuning curve is as in bi-lobed solid line. Figure modified from [115]

To measure the type of directional selectivity of a neuron (referred alternatively as *pattern direction sensitivity*), the actual neuronal responses to the plaid are correlated with the predicted responses. The partial correlation for the pattern prediction (R_p) and component prediction (R_c) is defined respectively as:

$$R_p = \frac{r_p - r_c r_{pc}}{\sqrt{(1 - r_c^2)(1 - r_{pc}^2)}} \tag{6.11}$$

$$R_c = \frac{r_c - r_p r_{pc}}{\sqrt{(1 - r_p^2)(1 - r_{pc}^2)}} \tag{6.12}$$

where r_c is the correlation between the actual responses to the plaid and the component prediction. r_p is the correlation between the actual responses and the pattern prediction. r_{pc} is the correlation between the component prediction and the pattern prediction.

A cell is classified as a ”pattern cell” if the partial pattern correlation is significantly larger than the partial component correlation ($R_p > R_c$). It is classified as a ”component cell” if the partial component correlation is significantly larger than the partial pattern correlation ($R_c > R_p$). Cells that are intermediate between the two extremes, *i.e.* the two partial correlations don’t significantly differ from each other, are classified as an intermediate type- ”unclassified” (R_p R_c). Figure 6-24A illustrates the boundary of the three cell types.

Most of the directional selective V1 cells are tuned to the motion perpendicular to the optimal orientation. The results in [115] showed a majority of DS V1 neurons are classified as component cells. In MT area, cells’ pattern direction sensitivity (R_p and R_c) forms a broad distribution; $\sim$ 25% of MT cells are classified as ”pattern cells”, whereas $\sim$ 40% of MT cells are classified as ”component cells”, and the rest 35% of cells fall into the ”unclassified” category [115].

This study revealed the analogy of the proposed two-stage model in the visual system. In the initial stage of the model, component motion that is perpendicular to the orientation of edges is extracted, and in the second stage, the motion to more complex pattern is computed by integrating the component motions from the first stage. In the visual system analogy, component directional selective V1 neurons correspond to the first stage, and pattern directional selective MT cells correspond to the second stage.

6.3.2 Introduction: speed tuning

Speed tuning in the visual system IOC could be used to disambiguate motion seen through apertures. For example, consider a zebra moving to the left as in Figure 6-7, two motion vectors computed within two apertures (green and red 1 circle in Figure 6-7) should uniquely determine the velocity of the zebra as long as they signal non-parallel component directions. If we replace the aperture (from red 1 to red 2), where the spatial frequency of the zebra changes, the solution should remain the same. For a visual system to implement IOC, neurons therefore have to ”speed-tuned”, meaning they should respond to a particular speed of motion independent of the spatial composition of the stimulus (Figure 6-8 C), which can be measured as spatial frequency.

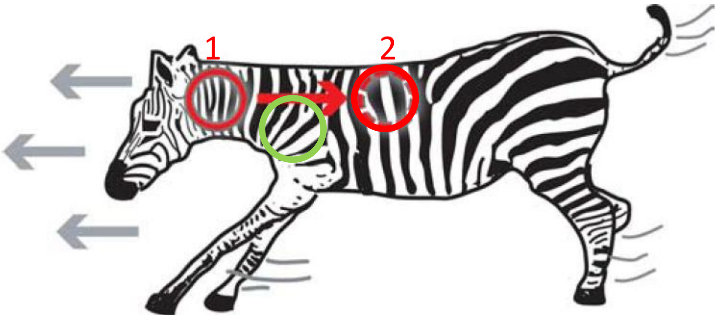


Figure 6-7: A zebra in motion. Modified from [110].

The speed of a one-dimensional motion is given by its temporal frequency divided by its spatial frequency (Section 6.1.4). A speed-tuned neuron with a peak speed v will respond to the spatial frequency w_s maximally when the stimulus moves in the temporal frequency $w_t = v \times w_s$. In other words, the preferred temporal frequency of the neuron changes with the stimulus’ spatial frequency. When plotting the responses as a function of the spatial and temporal frequency in a 2D plot ($w_t - w_s$), the preferred spatio-temporal frequency will lie on a line with slope v, equivalent to the neuron's preferred speed (Figure 6-8D). Alternatively, a neuron could have independent (or separable) tuning for spatial and temporal frequency (Figure 6-8B), meaning the preferred temporal frequency is independent of the stimulus' spatial frequency, and as a result, the preferred speed changes with the stimulus' spatial frequency (Figure 6-8A).

Most of the DS V1 cells are sensitive to the speed [131] of the stimulus. These neurons are however tuned independently to the spatial and temporal frequencies (Figure 6-8B), meaning they are not speed-tuned [192, 47, 96, 114].

Perrone and Thiele firstly showed that some MT cells are speed tuned [134] (Brief accounts of such experiment have been reported [120, 112]). They measured the spatiotemporal frequency responses of MT cells using sinusoidal gratings with thirty different spatiotemporal frequency combinations moving in the preferred direction of a cell (temporal frequencies, 1, 2, 4, 8 or 16 Hz; spatial frequencies, 0.2, 0.4, 0.7, 1.4, 2.8 or 5.6 cycles/degree). The results showed that some MT cells have inseparable spatio-temporal frequency tuning oriented in the Fourier space (Figure 6-8D, termed as ”spectral receptive field” in [134]) that enables them to respond selectively to particular spatiotemporal frequency combinations, that is, to a certain speed of motion. Priebe et al [142] later confirmed the existence of speed-tuned MT cells and estimated they make up 25% of the MT population.

In a subsequent study [144], Priebe et al measured the spatio-temporal frequency of DS V1 simple and complex cells as well as MT cells. It was known that V1 simple → V1 complex → MT cells constitute a direct pathway for the processing of motion [111, 160, 107] in the visual cortex of primates. A two-dimensional Gaussian is then used to fit the spectral receptive field of each cell:

$$R(sf,tf) = Aexp(\frac{-(log_2 sf - log_2 sf_0)^2}{2\sigma_{sf}^2}) \times \tag{6.13}$$

$$[exp(\frac{-(log_2 tf - log_2 tf_p(sf))^2}{2(\sigma_{tf} + \zeta(log_2 tf - log_2 tf_p(sf)))^2}) - exp(\frac{-1}{\zeta^2})] \tag{6.14}$$

where

$$log_2 tf_p(sf) = \xi(log_2 sf - log_2 sf_0) + log_2 tf_0 \tag{6.15}$$

A is the peak response of the neuron, sf_0 is the preferred spatial frequency averaged across temporal frequencies, $tf_p(sf)$ is the preferred temporal frequency of the neuron for a particular spatial frequency of the stimulus, and ζ is the skew of the temporal frequency tuning curve. The parameter ξ, called *speed-tuning index*, captures the dependence of the preferred temporal frequency (and therefore preferred speed) on the stimulus' spatial frequency. A neuron whose peak temporal frequency is independent of the stimulus' spatial frequency (ideal separable responses) has speed tuning index 0. A neuron whose peak temporal frequency increases or decreases with the spatial frequency has positive or negative speed tuning index, respectively. An ideal speed-tuned neuron has speed tuning index 1, and an ideal neuron with separable spatiotemporal tuning has speed tuning index 0.

Priebe *et al.* [144] showed that most of the V1 simple cells have separable tuning (mean speed-tuning index 0.08). V1 complex and MT cells have more diverse tuning property, ranging from 0 (separable tuning) to 1 (speed tuned). Surprisingly, the distribution and mean value of speed-tuning indices of V1 complex and MT cells are similar (0.44 and 0.48 for V1 complex and MT, respectively), as shown in Figure 6-22.

6.3.3 Summary

We described the transformation of motion processing from DS V1 neurons to MT neurons: V1 simple neurons respond to component directions and have separable spatiotemporal tuning. V1 complex neurons also respond to component directions but start becoming speed-tuned. MT neurons are of diverse tuning properties. In terms of pattern direction selectivity, MT cells range from CDS, unclassified, to PDS. In terms of speed tuning, MT cells range from separable tuning to speed tuning. It is tempting to think that the V1 simple neurons' function is to decompose motion into channels of directions, spatial frequencies, and temporal frequencies, which are then integrated by the PDS and speed-tuned MT cells to determine the velocity of the stimulus.

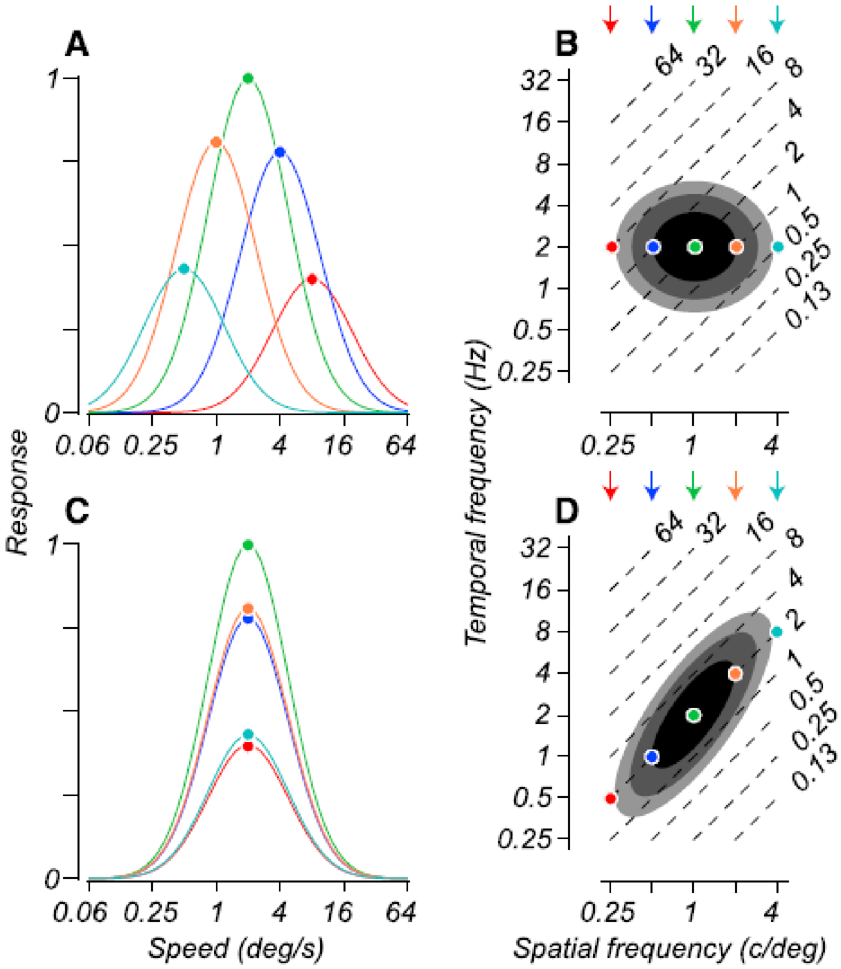


Figure 6-8: Responses and spectral receptive field of hypothetical cells. (A, B) cells of separable spatial and temporal frequency tuning. (C, D) cells that are tuned to speed. Figure reprinted from [144]

6.3.4 Previous computational models

Indeed, MT cells have been modeled as velocity tuned units that integrate multiple elementary features (directions, spatio-temporal frequencies). In the Fourier space, the spectrum of an object in translation lies on a plan, whose slant and tilt uniquely determine the object's velocity (Section 6.1.4 and Figure 6-3). Simoncelli and Heeger's PDS MT model (in some works referred as ”SH model”) [177] sums the responses of V1 cells whose preferred spatio-temporal frequencies lie on such a plane, and therefore achieves velocity tuning (Figure 6-9). Their V1 cells are implemented as 3^{rd} order Gaussian derivative spatio-temporal filters. The MT responses are then squared and normalized with respect to a set of MT population.

$$Q_j = \sum_i w_{ij} C_i \tag{6.16}$$

$$Q_j \leftarrow \frac{K Q_j^2}{\sum_i Q_i^2 + \sigma_2^2} \tag{6.17}$$

In the latest version of the SH model [161], the divisive normalization includes a self-normalization term to account for the suppression outside the receptive field [6]:

$$Q_j \leftarrow \frac{K Q_j^2}{\sum_i Q_i^2 + \sigma_1 Q_j + \sigma_2^2} \tag{6.18}$$

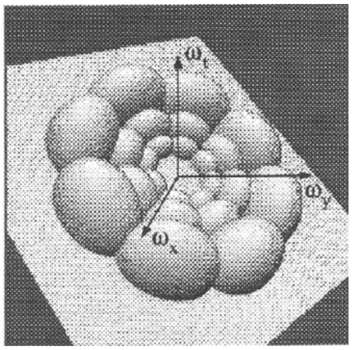


Figure 6-9: Construction of MT pattern cell from combination of V1 complex cell afferents, shown in the Fourier domain. Figure reprinted from [177]

Perrone's MT model [138, 135, 136, 137] combines two V1 neurons; one with a low pass temporal frequency tuning (sustained, S) and another with a band-pass temporal frequency tuning (transient, T). In primates, the S type has a unimodal temporal response profile that extends for the duration of the stimulus and the T type has a biphasic profile with the response primarily at stimulus onset and offset [47, 63]. The S type and T type have separable spatio-temporal tuning, and they are combined in a way such that the resulting MT spectral receptive field is tilted (inseparable). The response of their MT model to a spatio-temporal frequency is computed as

$$WIM(sf,tf) = \frac{log(\phi T(sf,tf) + S(sf,tf) + \alpha)}{|log\phi T(sf,tf) - logS(sf,tf)| + \delta} \tag{6.19}$$

There are more algorithms that combine outputs of a set of densely sampled spatio-temporal filters to approximate the image velocity [64, 60]. PDS responses have also been modeled as a combination of directional signals [167, 125, 211].

6.3.5 Two constraints for modeling MT PDS cells

MT pattern cells might be strongly related to the human's ability to estimate velocity of the motion of the surrounding environment, which could further help understand higher-order decision making or cognitive functions. Therefore the modeling of MT PDS cells has been a topic of great interest. To the best of our knowledge, there are at least 10 models [64, 59, 167, 125, 211, 177, 138, 135, 161, 198] simulating the PDS cells.

Most of the existing computational models for PDS cells could be summarized as a general class of a linear-nonlinear (LN) model, in which a linear combination of V1 CDS cells followed by a non-linear normalization could explain both PDS and CDS cells. These linear coefficients are derived based on the IOC in the Simoncelli and Heeger's model [177], learned from moving gratings and plaids in the Nowlan and Snowjeski's work [125], and chosen from a parameter space to fit neuronal responses in the work by Rust *et al.* [161].

Here we propose two constraints for the modeling of MT PDS cells based on physiological results.

Constraint 1: diverse directional selectivity of MT cells In MT area, pattern direction sensitivity (R_p and R_c in Equation 6.12) forms a continuous distribution, as shown in Figure 6-24. A large number of cells couldn't be classified into either PDS or CDS, but instead into ”unclassified” type. Moreover, PDS and CDS cells also range from strong PDS/CDS to the ”unclassified” type. We propose that, if direction selectivity of all the MT cells is generated through the same mechanism, a PDS/CDS model should also explain the unclassified type as well as the continuous distribution of R_p and R_c. Table 6.4 summarizes from various experiments the percentage of MT direction selective cells that are classified into CDS, unclassified, or PDS type. Although the percentage numbers vary across experiments, they show consistently that a majority of cells are CDS, and $20\% - 30\%$ of cells are PDS.

Constraint 2: local directional integration of MT cells Previous experiments for motion integration in MT have used stimuli that fills the receptive field, and thus do not test whether directional features are really integrated across the whole area. Most of the existing MT models have also assumed the V1 afferent weights were homogeneous across the MT receptive field [64, 59, 167, 125, 211, 177, 138, 135, 161, 198]. The work of Majaj, Carandidi and Movshon provides a spatial constraint for the direction integration within the receptive filed of MT cells [97].

In their study, the direction selectivity of each cell is tested using three stimuli: gratings, plaids, and pseudoplaids, referring to plaids whose grating components delivered separately in space . For each MT neuron, they identified two regions (”patches”) within the receptive field that were approximately equally effective in driving responses to gratings. Pattern selectivity was measured for each region separately by presenting gratings (Figure 6-10 (a,d)) and plaids (Figure 6-10 (b,e)) confined in the region. They then measured responses to pseudo-plaids (Figure 6-10 (c,f)), which have the same component gratings as plaids except the grating were separated in the two patches. If MT cells simply pooled all the inputs across the receptive field, the spatial separation of motion signals would not affect the responses, and the pattern direction sensitivity measured using pseudo-plaids should be identical to plaids.

The results show that PDS cells, which respond to the plaid direction, respond instead to the individual grating direction (CDS) when the gratings separated in two patches. These indicate that the computation of plaid direction, or integration of component directions in MT, is processed within a scale that is smaller than the whole receptive field.

6.3.6 Why our model could explain MT cells

The first two stages of our model, S1 and C1, extract component motion at particular spatial and temporal frequency scales. S_2 units detect motion features with intermediate complexity by performing a template matching (normalized dot product) between inputs encoded in the previous C_1 layer and a set of templates (prototypes) extracted also from the C_1 layer during a training phase. In the perspective of computational modeling, the $S_1 \rightarrow C_1 \rightarrow S_2$

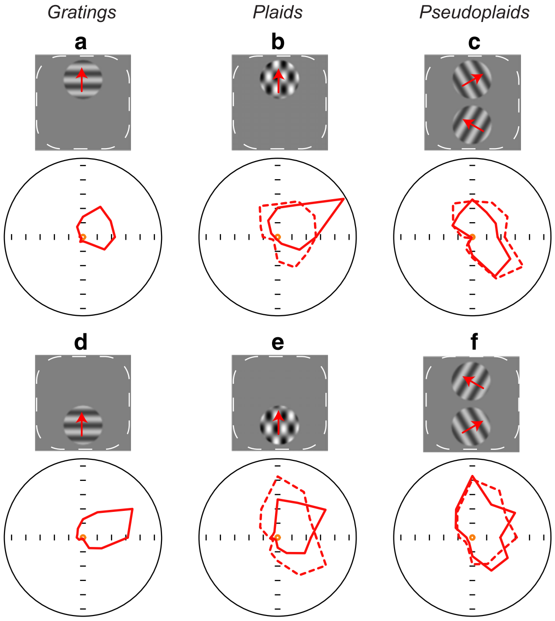


Figure 6-10: Responses of MT cells to (a,d) gratings, (b,e) plaids, and (c,f) pseudo-plaids. In (a,d), the grating covers one of the two patches within the receptive field, as indicated by the stimulus icons. In (b, e), solid curves indicate responses to small plaids with plaid angle 120^o. Dashed curves indicate the CDS prediction to the small plaids. The prediction in (b), (e) is obtained from (a), (c), respectively using equation 6.10. In (c, f), solid curves show responses to pseudo-plaids; dashed curves show the CDS prediction based on the two grating tuning curves in (a,d). Reprinted from [97].

connection falls into the class of LN models, where a MT cell is modeled as a linear combination of V1 complex cells followed by a nonlinear operation. We claim the S_2 units can model MT cells as well. Moreover, it has mechanisms that render some properties that were not accounted in previous MT models, as described below.

S_2/C_2 units can explain the continuous pattern direction sensitivity of MT cells The operation of template matching in the S_2 stage is useful for the recognition task, in which inputs are classified based on the similarity to training data in a feature space (in this case C_1 space). In previous works, PDS MT neurons have been modeled as functional units that solve IOC to compute the true velocity of inputs [64, 59, 177]. Although the perspective of template matching is very different from the perspective of computing image velocity, we claim S_2 units can explain PDS cells as well. Moreover, S_2 units could explain CDS and unclassified types using the same template matching mechanism.

Figure 6-11 shows the responses of 16 C_1 units at the same location tuned to 16 directions equally spaced in the angular space between 0 and 2π. A grating in translation activates one single C_1 unit (Figure 6-11 1), while a plaid with two component gratings activates two (Figure 6-11 2).

To obtain the direction tuning curve of a template, we move the stimulus in all 16 directions and compute a normalized-dot-product between the C_1 responses (Figure 6-11B) with the template (Figure 6-12B). If the template peaks in one single direction, the matching will result in a single-peaked direction tuning for gratings and double-peaked for plaids (Figure 6-12 C-D). This is identical to a MT CDS cell (Figure 6-12 row 1), and we predict that this type of templates could be learned from the motion of oriented stimuli, such as edges, bars, or gratings.

If the template peaks in a broad range of continuous directions, the matching will result in a single-peaked broad tuning curve for both gratings and plaids. This is identical to a MT PDS cell (Figure 6-12 row 2-3). This template could be learned from the coherent motion of textured patterns, which contain many orientations and therefore activate many C_1 cells. Random dots or multiple gratings are a kind of textured patterns. Note that if the activated C_1 cells' preferred directions are not continuous (*i.e.* the template has multiple peaks), the resulting tuning curve to gratings will consist of multiple peaks; the quantitative measurement (using Equation 6.12) shows this will fall into the CDS or unclassified type. It is NOT a PDS cell(Figure 6-12 row 4).

It was found when presenting stimuli at a particular high or low speed, that some MT cells exhibit bi-modal direction tuning [126]. We predict in our model that these cells could be learned from bi-modal directional motion, such as movements of a plaid (Figure 6-12 row 4).

C_2 units can account for the local directional integration of MT cells The computation of plaid direction, or integration of component directions in MT, is processed within a scale that is smaller than the whole MT receptive field [97]. An integration of directions within a spatially localized region followed by a global pooling (for example, summation, average or max) of responses over space could explain this phenomenon. Indeed, the global-pooling

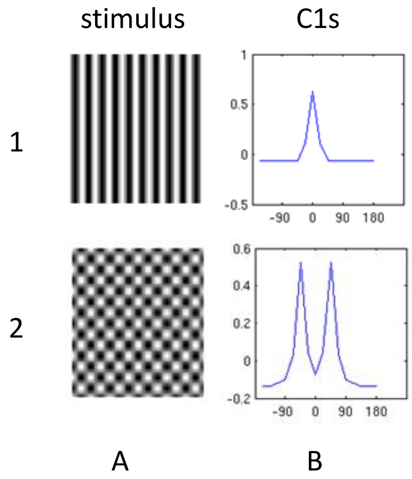


Figure 6-11: (A) Stimulus (B) Responses of C_1 units to stimuli in (A). Here we concatenate responses of C_1 units tuned to 16 directions in one plot. The x axis specifies the preferred direction of each C_1 unit.

has been used in previous works [137, 198]. For example, In the MT model proposed by Perrone & Krauzlis, nine pattern motion detectors are mapped out over the receptive field of the model neuron. The outputs from all of the 9 detectors are summed to give an overall response for the model neuron (Figure 6-13).

In our model, a S_2 unit performs directional integration with a template matching operation, a C_2 unit then pools a maximum response of S_2 units of the same weights (*i.e.* the same template) in all the spatial positions with the C_2 RF. Figure 6-14 shows that the max-pooling operation in C_2 units allows them to simulate the experimental results of Majaj *et al.* [97].

The global pooling operation within MT receptive field is indeed supported by physiological evidence. Britten et al [19] measured the response of pairs of gratings moving in two local regions of MT receptive fields, denoted as R here. They then compared it with the responses obtained by presenting the two gratings alone, denoted as r_1 and r_2 here. They concluded a power-law summation model with divisive normalization could explain the spatial summation of responses (also see [150]).

$$R = a(r_1^n + r_2^n)^{(1/n)} + b \tag{6.20}$$

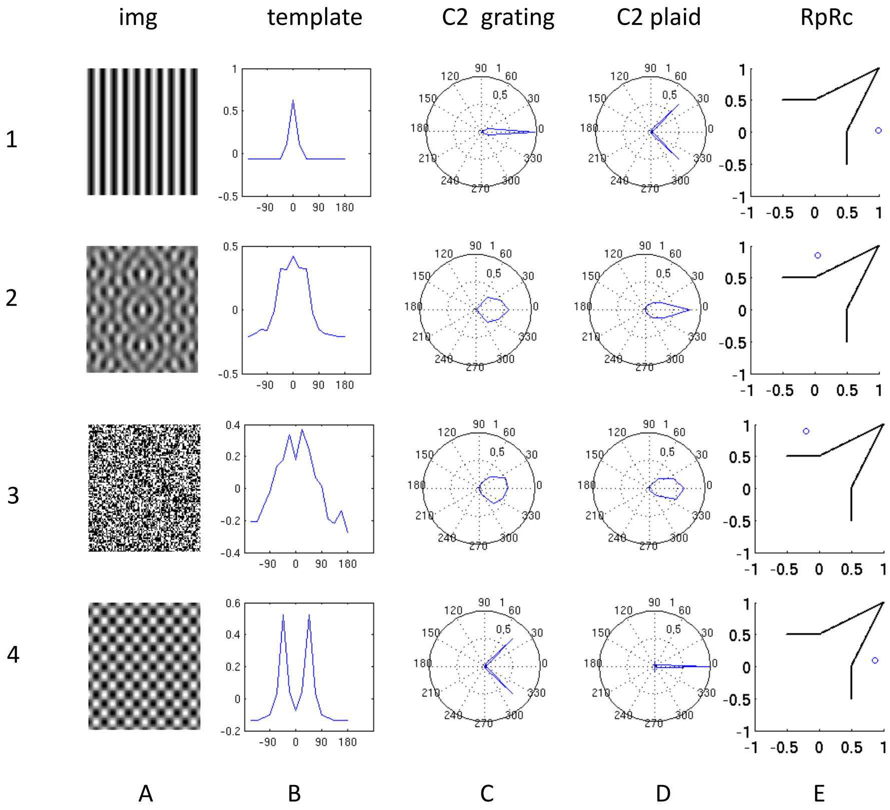


Figure 6-12: Responses of ideal S2 units that model MT component and pattern cells. (A)Templates in image space. (B) templates. (C) Directional tuning of templates to gratings (D) Directional tuning of templates to plaids. (E) The pattern sensitivity of the templates. From top to bottom, the templates were ”learned” from single grating moving in 0^o, five superimposed gratings moving in $45^o, 22.5^o, 0^o, 22.5^o, 45^o$, random dots moving in 0^o, plaid with component gratings moving in 45^o, -45^o.

The mean value of the exponent n was reported as 2.72. On the other hand, Kouh and Poggio have pointed out that such a power-law summation model with exponent $n = 3$ (called ”canonical circuit” in [80]) is an approximation of the max-operation [80], which is used in our C_2 layer.

Zero-mean normalization of S_2 templates could account for directional suppressive signals Suppressive signals within MT receptive field have been shown to contribute to a range of neuronal properties.

Mikami *et al.* found the dominant mechanism of direction selectivity in MT was a pronounced suppression of response for motion in the null direction (the opposite of the

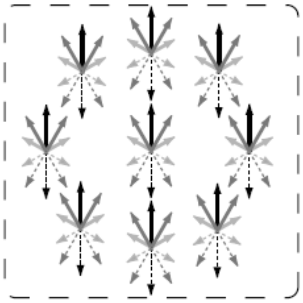


Figure 6-13: Each arrow cluster is a MT subunit where the pattern direction is computed using a WIM model[135]. A MT cell is modeled by the set of nine clusters equally distribute within its receptive filed, and the response is obtained by summing the responses of the 9 clusters. Reprinted from [137].

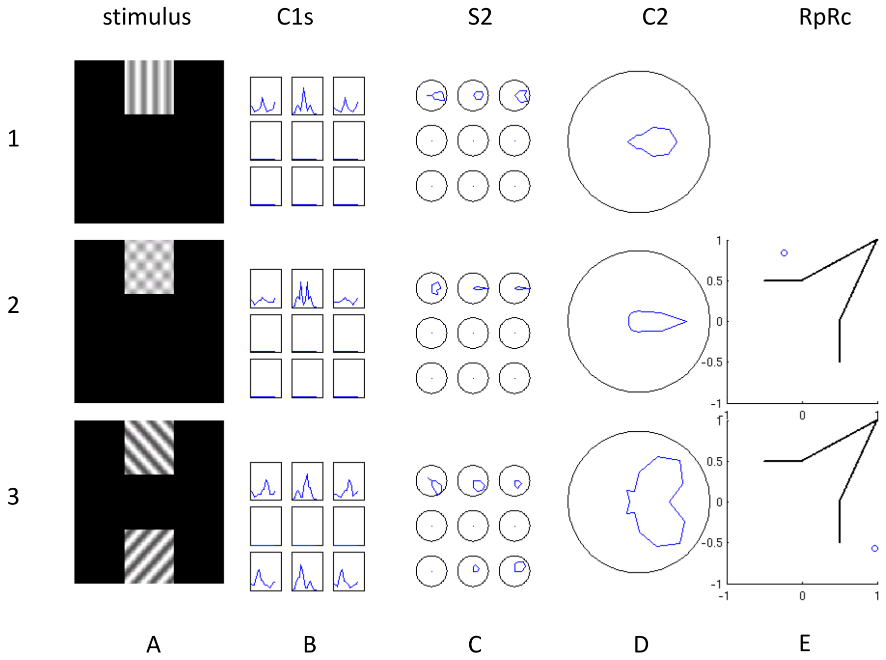


Figure 6-14: Responses of an ideal C_2 unit that models MT pattern cells. (A) Stimuli moving in the direction 0^o. (B) Responses of C_1 units to stimuli in (A). Here we show responses of C_1 units tuned to 16 directions and sampled at 9 locations. (C) Responses of S_2 units sampled at 9 locations, all the 9 S_2 units store the same template learned from the coherent motion of the random dots, as shown in 6-12, row 3 , column A. (D) Direction tuning of the C_2 unit to the stimuli moving in 16 directions from 0 to 2π. The C_2 unit computes the maximum responses of the 9 S_2 units in (C). (E) R_p and R_c for the C_2 unit.

preferred direction). The direction selectivity is measured as *direction index (DI)*

$$DI = 1 - \frac{\text{response in the null direction}}{\text{response in the preferred direction}} \tag{6.21}$$

A facilitation in the preferred direction or suppression in the null direction will both result in a large direction selectivity [106]. Snowden *et al.* found that the responses of a MT cell to the preferred direction can also be suppressed by superimposing motion in the null direction [180]. Qian *et al.* 's psychophysics results suggested that the motion signals of different directions, of the same disparity and spatial frequency contents, locally inhibit each other. They therefore suggest the suppression signals could reduce noise and segment motion in different depth and scales, which is a key to solve motion transparency problem [145].

Here we demonstrate S_2 units can explain the direction tuning of MT cells, and are also important for shaping the pattern direction selectivity of MT pattern cells. Figure 6-15 2B shows a "raw" template sampled from C_1 responses of moving dots(Figure 6-15 2A). The direction tuning of the C_2 unit to a grating is a single-peaked curve tuned to a broad range of directions, the tuning to a plaid is similar (Figure 6-15 2C-D). Although both tuning curves are single-peaked, a quantitative measurement showed that the response to the plaid is higher correlated with the component prediction (Equation 6.9) than with the pattern prediction (Equation 6.10). The cell is therefore a component cell or an unclassified cell (Figure 6-15 2E).

With the zero-mean normalization applied to the template (Figure 6-15 2A), the motion that is "far from the preferred direction" in the angular space contributes negatively to the normalized dot product. This sharpens the direction tuning to gratings and plaids (Figure 6-15 2C-D), which in turn, decreases the correlation between the component prediction and the plaid response. This results in a pattern cell (Figure 6-15 2E).

C_1/S_2/C_2 units can explain speed-tuned V1 complex and MT cells The emergence of velocity selectivity in MT neurons through a suitable combination of V1 afferents has been described in [64, 60, 177]. A simplified version of this construction is shown in Figure 6-16, in which a hypothetical MT neuron sums the responses of of three hypothetical V1 neurons with separable spatiotemporal tuning and the same ratio for the peak temporal

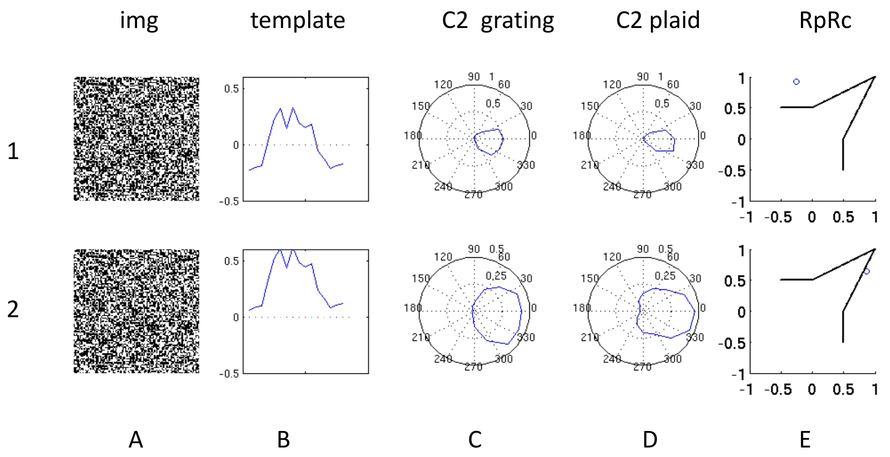


Figure 6-15: Why normalizing each S_2 template's mean

to spatial frequency. The resulting MT neuron has a tilted spectral receptive field and is therefore speed tuned.

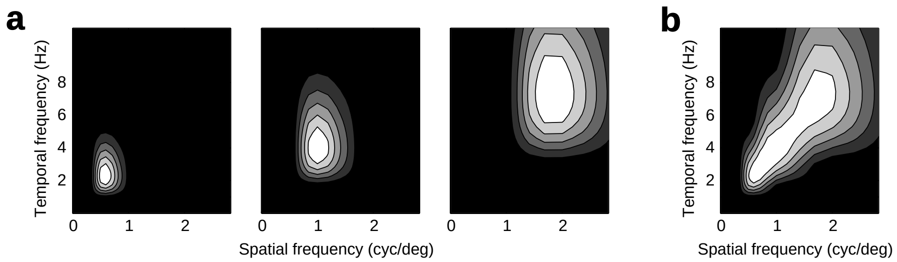


Figure 6-16: Construction of a speed-tuned MT neuron from V1 neurons.(A) Spectral receptive field for three hypothetical V1 neurons (B) Spectral receptive field for a hypothetical MT neuron. Reprinted from [176].

6.3.7 Results

In Section 6.3.6, we showed that with S_2 templates imprinted from synthetic motion, C_2 units can explain a range of MT neuronal properties such as pattern direction sensitivity, local direction integration, and speed-tuning. Here we sampled a large number of S_2 templates from natural image sequences to show that the responses of the S_2/C_2 population can model the MT population activity as well.

A random sampling of S_2 templates results in many patches that don't contain motion. It was shown that direction selectivity is a prominent property of MT cells. We therefore apply a filtering approach to remove noisy S_2 templates based on the direction selectivity of MT neurons reported in [4]. It was reported in [4] that the distribution of direction index (DI, Equation 6.21) clusters around 1, and the directional tuning bandwidth is between 32^o - 186^o with a mean value 95^o. Here we filter out templates with DI smaller than 0.9 or bandwidth lower than 57^o. This filtering step will be implicitly accounted if we replace the random sampling with leaning rules like Hebbian learning.

Figure 6-17 shows three types of video sources from which S_2 templates were sampled. Figure 6-17A illustrates videos obtained by a camera mounted to a cat's head, so they should approximate the natural input to the cat's visual system. The videos' sampling rate is 25 frames per second and the video resolution is 240 × 320 pixels [39]. Figure 6-17B illustrates broadcasts recorded from Dutch, British and German television. The videos' sampling rate is 25 frames per second and the video resolution is 128 × 128 pixels. The videos were originally collected to analyze the statistics of nature scenes [205]. Figure 6-17C illustrates a large human motion database (HMDB) clipped from HollyWood movies. These videos were collected to test action recognition systems [82].

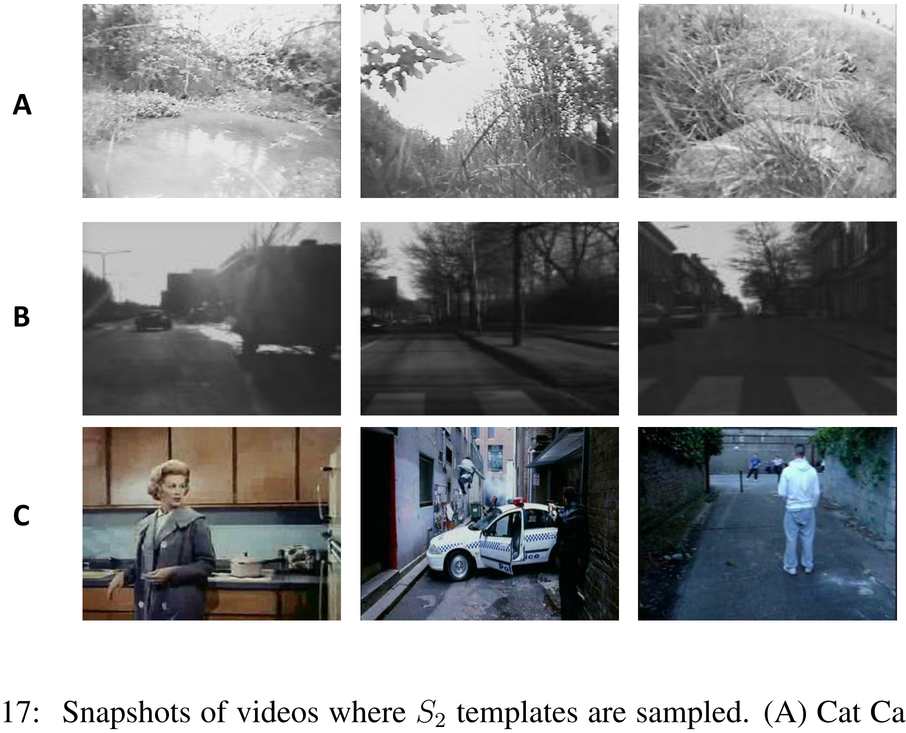


Figure 6-17: Snapshots of videos where S_2 templates are sampled. (A) Cat Camera. (B) Street Scenes. (C) HMDB.

Table 6.3: Sampling of S_2 templates

Experiment	S_1 scale	S_1 preferred speed	S_2 template size	video source
Local direction integration	5-6	1 pxs/frame	1, 3, 5, 7, 9, 11	A
Motion opponency	1-2	1 pxs/frame	1 ,3, 5, 7, 9, 11	A
Speed tuning	1-12	0.5 pxs/frame	1,3,5	C
Pattern direction sensitivity	1-12	1 pxs/frame	1, 3, 5, 7, 9, 11	A,B,C

Local direction integration

Majaj *et al.* used pseudo-plaids (plaids with component gratings separated in space) to measure the pattern and component direction sensitivity of MT cells. [97], and they found that MT cells that were tuned to the pattern direction tune instead to the component direction when tested with pseudo-plaids(Figure 6-19). Figure 6-18 shows that C_2 units can explain their results.

In our experiments, the S_2 template size is a key factor for this effect. Pattern and component direction selectivity are sometimes combined to give one single measurement called *pattern index (PI)* , which is defined as

$$PI = Zp - Zc \tag{6.22}$$

$$Z_p = 0.5\sqrt{n-3}(ln\frac{1+R_p}{1-R_p}) \tag{6.23}$$

$$Z_c = 0.5\sqrt{n-3}(ln\frac{1+R_c}{1-R_c}) \tag{6.24}$$

where Z_p and Z_c are R-to-Z transformed R_p and R_c. PI larger than 1.25 indicates pattern cells; PI smaller than -1.25 indicates component cells; PI between 1.25 and -1.25 is the unclassified type. As in [97], the pattern-to-component effect is characterized as the change of PI, (ΔPI = PI measured with pseudo-plaids − PI measured with small plaids). We observe that ΔPI is a function of the S_2 template size (See Table 6.3 for a list of sizes). For a very small template size (<< size of a local patch in Figure 6-14A), all the pattern and unclassified cells become component cells ($\Delta PI < 0$ as shown in Figure 6-20B). For an intermediate size similar to the patch size, a majority of pattern cells become either unclassified or component cells ($\Delta PI < 0$) with a few some exceptions with higher pattern index for a pseudo-plaid ($\Delta PI > 0$). A few component cells become unclassified cells and a few unclassified cells become pattern cells.

This validates the finding of Majaj *et al.* . In their experiment, a majority of pattern and component cells become component cells; one component cell becomes unclassified cell and one unclassified cell becomes pattern cell, as shown in Figure 6-20A. We therefore suggest that the size of the local region where directions are integrated forms a continuous distribution in MT area. A quantitative match between the ΔPI of MT cells and various S_2 sizes will help predict the size of the local region within MT receptive field.

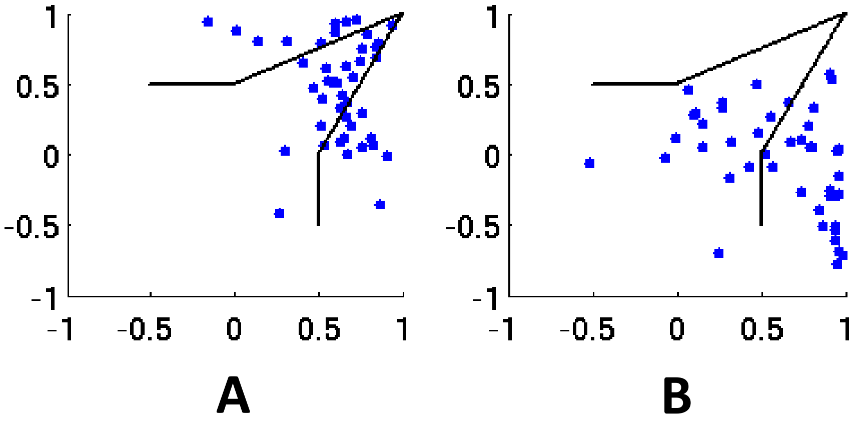


Figure 6-18: Scatter plot of R_p and R_c of C2 units. (A) Measured using small plaids. (B) Measured using pseudo-plaids.

Motion opponency

Snowden *et al.* [180] investigated the suppression in the area MT by recording MT neuronal responses to superimposed pairs of random dot fields filling in the MT receptive field. One field moves in the preferred direction and the other field moves with the same speed in the null direction. In order to measure the magnitude of suppression, they defined a term *suppression index* as

$$I_s = 1 - \frac{\alpha}{P} \tag{6.25}$$

where P is the response of the field moving in the preferred direction, α is the response

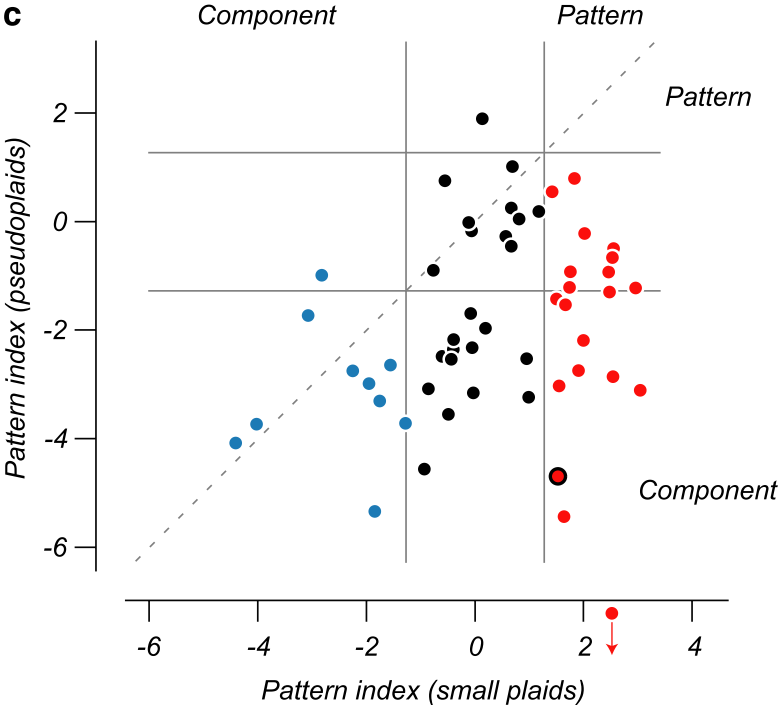


Figure 6-19: The pattern indices of MT cells measured using pseudo-plaids and small plaids. Reprinted from [97].

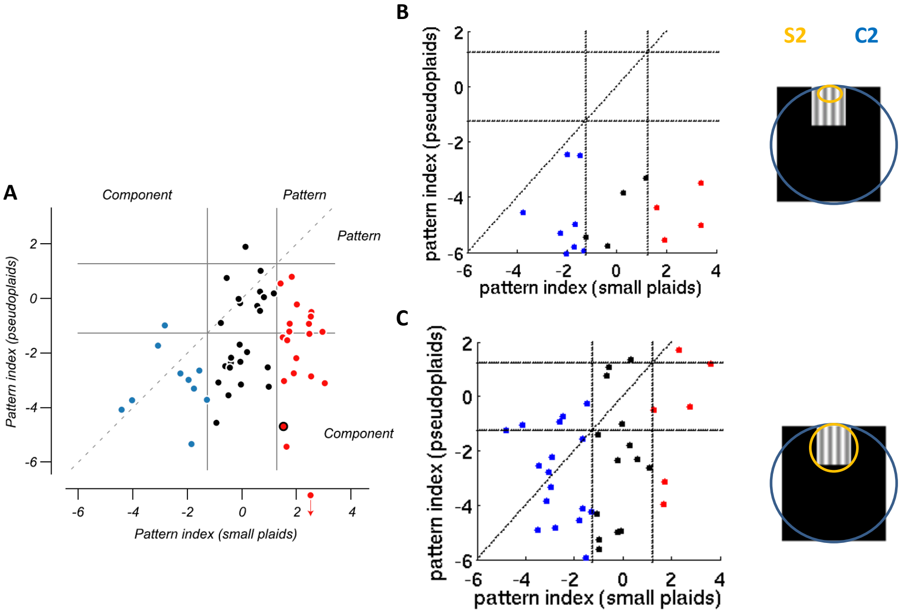


Figure 6-20: The pattern indices of C_2 units measured using pseudo-plaids and small plaids. (A) Reprinted from [97]. (B) C_2 units with small template sizes. (C) C_2 units with intermediate template sizes.

after superimposing the null field. I_s 0 indicates the null motion has no effects on the preferred motion. I_s 1 indicates the null motion completely silenced the preferred motion.

Figure 6-21A shows the suppression index of a population of V1 and MT neurons. Figure 6-21B shows the suppression index of C_1 and C_2 units. The C_2 units have higher I_s than C_1 units, just like MT cells have higher I_s than V1 cells, and the two sets of distributions roughly cluster at the same value. It is 0.1 for V1 and C_1 units, and 0.5 for MT and C_2 units.

In our model, the directional suppression signal within MT receptive field is modeled using the zero-mean normalization of S_2 templates, we suggest a more refined model for the suppression signal will be necessary to get a closer match of the I_s distribution between the C_1/C_2 population and the V1/MT population.

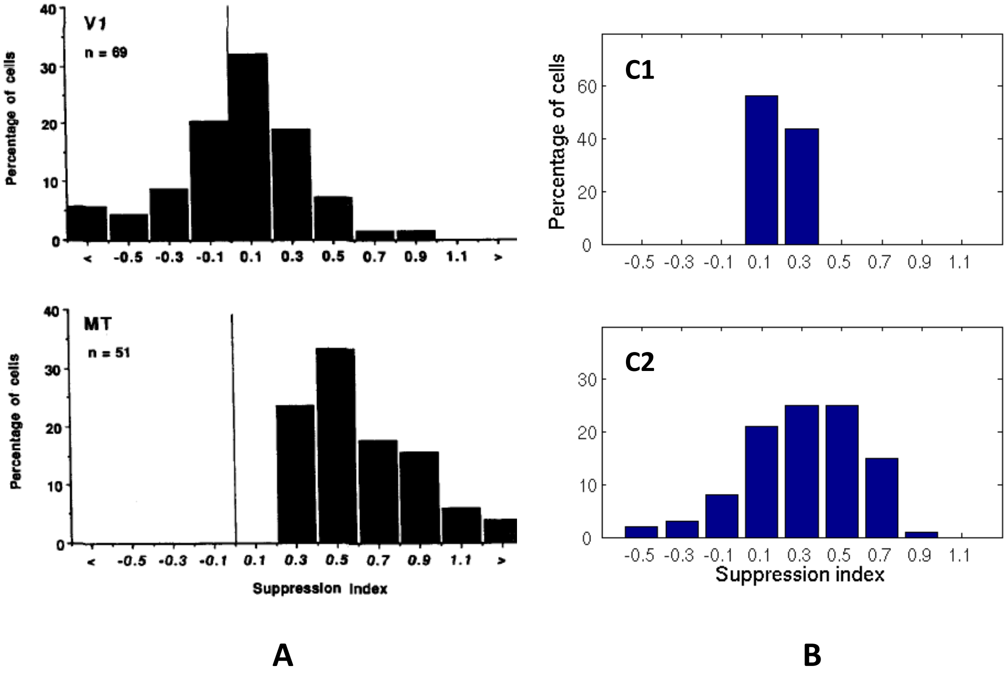


Figure 6-21: Suppression index of MT cells and C2 units. (A) Histogram of suppression indices of V1 and MT cells. (B) Histogram of suppression indices of C_1 and C_2 cells.

From separable spatiotemporal tuning to speed tuning

Priebe *et al.* [144] mapped the spectral receptive fields of DS V1 simple, V1 complex, and MT cells, and defined the "speed-tuning index" ξ to measure the the tilt of the fields (Equation 6.14). ξ 1 indicates ideal speed-tung, and ξ 0 indicates ideal separable spatiotemporal tuning. The distribution of ξ of these cells is shown in Figure 6-22A.

In our simulation, we mapped the spectral receptive field using 180 possible combinations of 15 spatial frequencies and 12 temporal frequencies. The spatial frequencies range from 0.05 (cdeg) to 6.4 (cdeg) and the temporal frequency ranges from 0.05 (Hz) to 2.3 (Hz), both equally spaced in the logarithm space. We also exclude C_2 units that have ξ outside the range of $(-1, 1)$, which are mostly because the spectral receptive field could not be fitted using a two-dimensional gaussian function. The resulting distribution is shown in Figure 6-22B.

The mean value for the two sets of distribution is very close, it is 0.08, 0.44, 0.48 for V1 simple, V1 complex, and MT cells, and -0.03, 0.42, 0.42 for S_1, C_1, and C_2 units. However, the distributions shows that the model units are able to simulate the continuous distribution of MT cells, but not of the V1 complex cells.

In our experiment, the speed tuning index is a function of the number of scales pooled in a complex layer. Complex units (C_1 or C_2) become more speed tuned (larger ξ) as they pool across more scales. Here each C_1 unit pools over two scales of S_1 units, and C_2 unit pools over all the scales of S_2 units. We suggest the speed tuning indices of more V1 complex cells (there are only 33 cells in [144]) will help refine more accurately the ”right” number of scales pooled in the model's complex layer.

Figure 6-23 illustrates the spectral receptive fields of representative S_1, C_1, and C_2 units.

Continuous pattern/component direction selectivity

Many experiments already suggested in MT area, the selectivity to pattern direction sensitivity, measured as R_p and R_c (Equation 6.12), forms a continuum from pattern cells, unclassified cells to component cells. Figure 6-24A-B illustrate this continuum. Here we randomly sampled a set of S_2 templates at all 12 possible scales from three types of video sources at 6 template sizes. (See Table 6.3 for a list of parameters). After removing the templates without motion ($DI < 0.9$ or directional tuning bandwidth $< 57^o$), we obtain 9100 C_2 units, whose pattern direction selectivity is shown in Figure 6-24C.

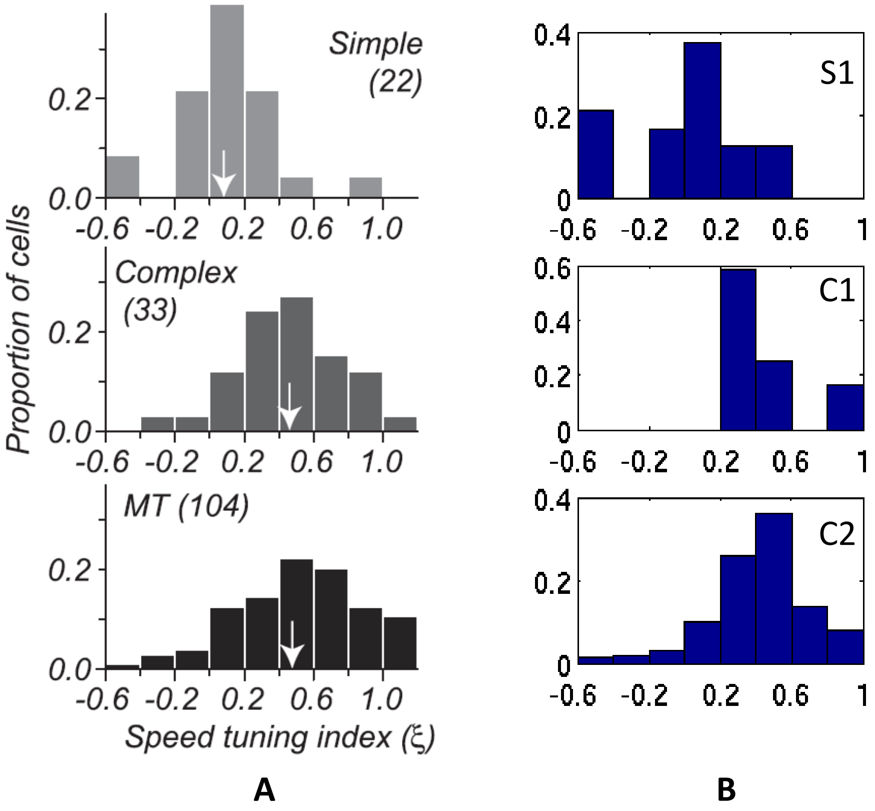


Figure 6-22: Histograms of speed tuning indices for DS neurons and model units. (A) speed tuning indices of directional selective neurons V1 simple, V1 complex, and MT neurons, reprinted from [144] (B) speed tuning indices of S1, C1, and C2 units.

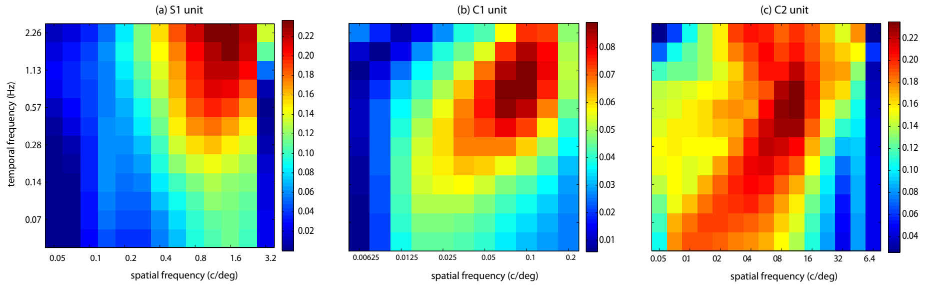


Figure 6-23: Spectral receptive fields for three representative model units.(A) S_1 unit (B) C_1 unit (C) C_2 unit.

Figure 6-25 shows the image sequences where a typical pattern, component, and unclassified C_2 unit is imprinted. Here we confirm the prediction in Section 6.3.6; a pattern cell is learned from the motion of a textured pattern, a component cell is from the motion of edges/bars, the rest of motion will result in unclassified cells.

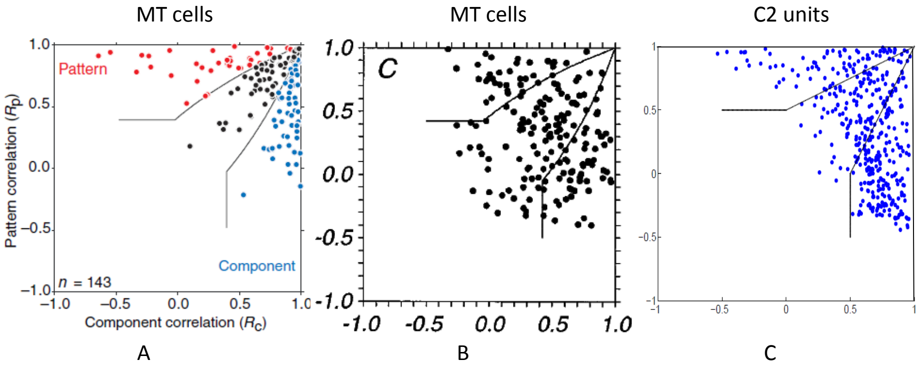


Figure 6-24: Scatter plot of R_p and R_c of MT neurons and C_2 units. (A) [179] (B) [111] (C) 2034 C_2 units

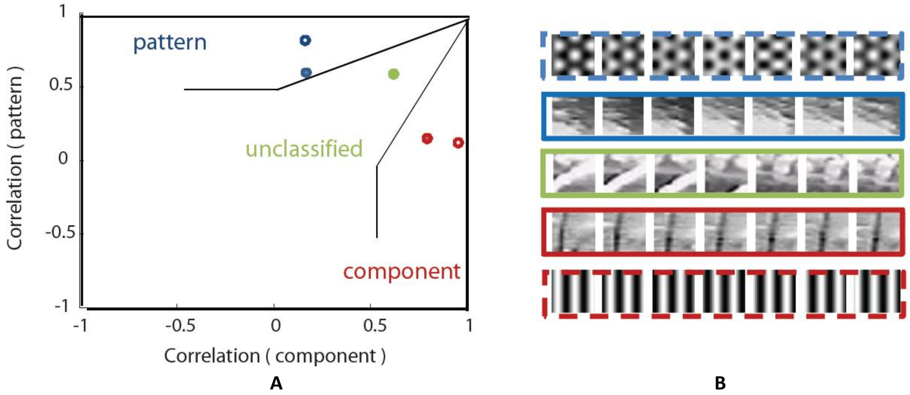


Figure 6-25: What makes a component/pattern cell. (A) R_p and R_c plot. (B) Local sptio-temporal image patches where pattern and component cells are learned. Dotted boxes indicate the synthetic image sequences, and closed boxes indicate images from the Cat video as shown in Figure 6-17A.

In order to quantify the fitness of the distribution of C_2 units to previous experimental results of MT neurons, we group the C_2 units into a three-bin normalized histogram, the three bins are the proportion of CDS cells, the proportion of unclassified cells, and the proportion of PDS cells. We compute the same histograms for MT neurons recorded in four previous experiments [115], [154], [179], and [161] (See Table 6.4 for the histograms).

We then use the chi-square distance to compute the distance between two histograms h and k:

$$1 - \sum_{i=1}^{3} 2\frac{(h_i - k_i)^2}{h_i + k_i} \tag{6.26}$$

A distance 1 indicates the two histograms are identical, a low value indicates two histograms are highly dissimilar. For each previous experiment as well as for our model, a final score is computed as the average chi-square distance between the histogram and the four previous experiments [115, 154, 179, 161]. The last column of Table 6.4 shows such scores. We conclude that the distribution of C_2 units are quite consistent with that of MT neurons.

Table 6.4: Distribution of R_p and R_c of MT cells and C2 units

Reference	# of cells	plaid angle	CDS (%)	unclassified (%)	PDS (%)	score
[115]	108	135^o	40%	35%	25%	0.97
[154]	33	135^o	33%	36%	30%	0.94
[161]	50	120^o	56%	25%	19%	0.90
[179]	143	135^o	41%	34%	25%	0.97
C_2 using RpRc	9100	135^o	47%	35%	19%	0.95

We next ask what determines the ratio of the CDS *vs.* unclassified *vs.* PDS cells. For the 9100 sampled S_2 templates, we group them according to the video data source (Figure 6-26A-B), the size/scale of the S_1 filters (Figure 6-26C-D), and the S_2 template size (Figure 6-26E-F). In Figure 6-26A,C,E, we plot the matching score of the histogram for each group, as determined using Equation 6.26, and we also plot the proportion of CDS *vs.* unclassified *vs.* PDS cells for each group in Figure 6-26B,D,F. Overall we found the S_2 template size is the dominant factor. For a small template size (such as 1×1), almost all the C_2 units are classified as component cells, when the template size increases, the proportion of PDS cells increases accordingly.

As explained in Section 6.3.6, a pattern cell is tuned to a broad range of continuous directions. Here we suggest a larger spatial region is more likely to contain a broad range of motion directions than a small region, that is why we get more pattern C_2 units for a large S_2 template size. Previous computational models for the MT area focused on the integration of signals in the directional space, but not over the spatial (x-y) domain. Here we suggest the integration of directional signals over the x-y space is important for the development of MT pattern cells.

We already claimed that the tuning to a broad range of directions is what makes a pattern cell, we next quantify this statement by measuring the directional tuning bandwidth of C_2

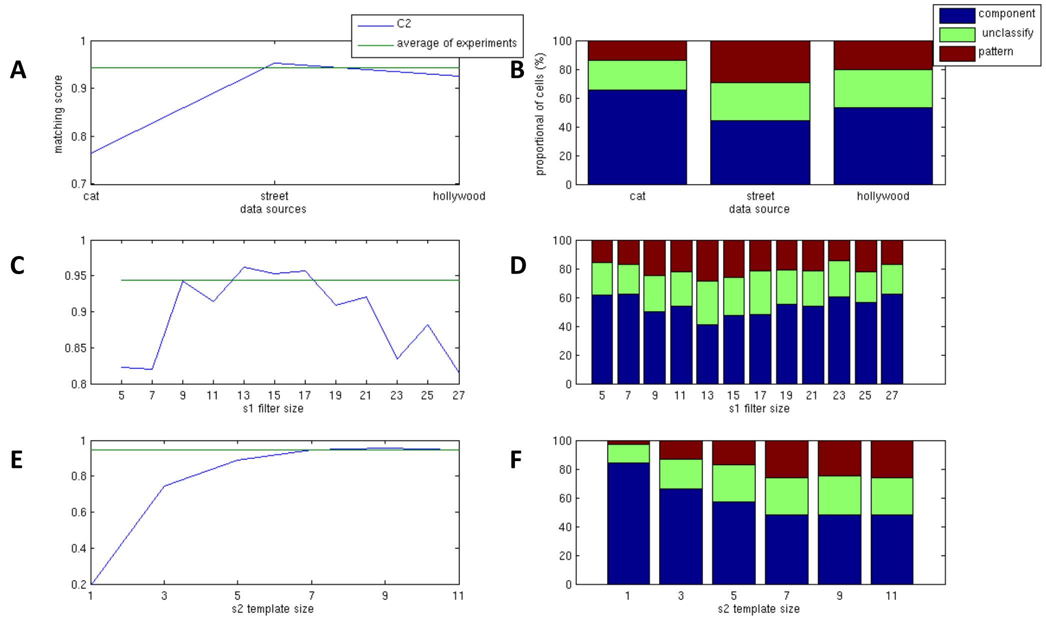


Figure 6-26: Some factors that determine the proportion of CDS, PDS, and unclassified C_2 units. (A,C,E) The matching score as a function of video type (A), S_1 filter size (C), andS_2 template size (E). (B,D,F) The proportion of component (blue), unclassified (green), and pattern (red) cells as a function of the video type(B), the S_1 filter size (D), and the S_2 template size (F).

units and plot the pattern index (PI, defined in Equation 6.3.7) as a function of bandwidth in Figure 6-27A. We observed that there is a positive correlation between the bandwidth and the PI when the bandwidth is smaller than 2.2 octave.

We also consider the ”composition” of S_2 templates. Similarly to the analysis by Rust *et al.* [161], here we count the proportion of reliable positive weights in each template that exceeds 20% of the maximum positive weight, and the proportion of reliable negative weights that are smaller than 20% of the most negative weight. Figure 6-27B-C show the pattern index as a function of the two measurements. We observed that there is a positive correlation between the proportion of positive weights and the pattern index. Except for a few outliers, most of the pattern cells have at least 20% of the positive weights. On the other hand, the proportion of negative weights has almost no effects on the pattern index, confirming the results in [161],

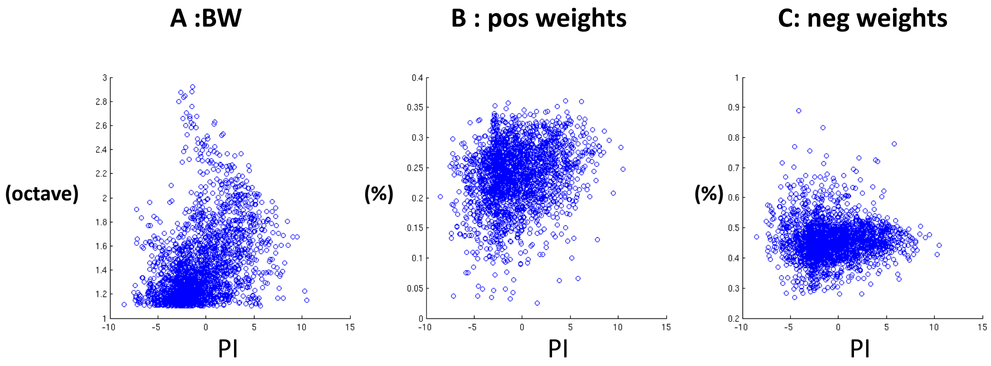


Figure 6-27: Some factors that determine the pattern index of C_2 units. (A) directional tuning bandwidth vs PI. (B) The proportion of positive coefficients in each S_2 templates vs PI. (C) The proportion of negative coefficients in each S_2 templates vs PI.

6.3.8 Comparison with other MT models

To the best of our knowledge, there are at least 10 models [64, 59, 167, 125, 211, 177, 138, 135, 161, 198] simulating the velocity tuning of MT neurons and especially MT pattern cells.

Some of these models make an explicit assumption that one of the functions of MT neurons is to compute the velocity of the moving stimulus [64, 59, 177, 138, 135]. They therefore propose to integrate directional signals from V1 neurons whose frequency spectrum in the Fourier space lie on a common plane (See Section 6.1.4 for details). The similarity between our S_2/C_2 units and previous MT models is that they both compute a linear weighting of the input directional signals followed by a nonlinear normalization.

However, S_2/C_2 units are designed to perform visual recognition task, and they are "learned" from the motion of the natural image sequences. With this learning procedure, the model is able to generate a large number of S_2/C_2 units to model the population activity in the MT area. The model also presents a possibility that the neural circuitry in MT makes no explicit attempt to compute the true velocity of moving stimuli. Instead, MT neurons derive pattern direction selectivity from learning a " motion pattern" of a broad range of directions that comprise most of the moving objects in natural scenes.

Another main difference between our model and previous MT models is that our model considers the integration not only over directions but also over space. Indeed our model predicts that the MT cells don't have uniform RF profiles like assumed in other models and that the preferred direction of a MT cell might change as a function of the stimulus position within RF when testing with a stimulus in a scale less than a S_2 template size. This prediction has been confirmed by Thomas Albright (personal communication).

6.3.9 Discussion

Coherency *vs.* transparency of a plaid An important use of motion information is to segment a complex visual scene into surfaces and objects. Transparent motion, referring to the motion field where more than one velocity vector occurs at each local region in the image, is usually used in psychophysics [146] or physiology [180, 145] experiments to test the human's ability to perform visual segmentation.

The plaid made of two superimposed gratings is one example of the transparent motion. Movshon *et al.* [115] already showed that in MT area, pattern cells are able to signal the motion of a perceptually coherent plaid pattern, while component cells signal the motion of individual gratings (transparency).

Qian, Andersen & Adelson [146] tested human's perception for transparency *vs.* coherency using two sets of patterns moving in different directions. These patterns include lines, random dots, or square waves. They found that humans perceive coherent motion when the two sets of patterns have similar spatial frequencies and the same disparity, and are displayed in a locally balanced manner. Snowden *et al.* [180] also found that the spatial separation of two sets of velocity vectors within RF determines whether velocity vectors inhibit each other. Stoner, Albright and Ramachandran [186] found in their psychophysics experiment that the human's tendency to see pattern motion depends on the luminance of the intersections relative to that of the gratings. They also found that responses of MT neurons can be modified by the same factor known to influence the perceptual decision [187].

In the first layer of our model, motion is extracted into a set of directional and spatial frequency channels at many spatial locations. The integration of directions in the S_2 layer is done for each individual spatial frequency channel at local spatial regions. Therefore our model will simulate the fact that human perceive coherent motion when the objects in motion have similar spatial frequencies and are presented with spatial overlapping. However, our current model cannot explain the dependence of coherent plaid motion on the luminance of the intersection of the gratings. In the current setting, the lighting intensity of the stimulus is normalized to be unit average.

Effects of terminators The directional tuning in the V1 and MT areas has been mostly studied using one-dimensional stimuli or stimuli that are homogenous in space, such as random dots, extended gratings, or plaids. These stimuli present ambiguous motion (aperture problem), while two dimensional stimuli containing terminators (end points, corners, or intersections) allow accurate velocity measurements. A series of experimenters performed by Pack and his colleagues have shown that the direction tuning of V1 and MT neurons are modulated by the geometry of the two dimensional stimuli (such as ”barber pole”) as well as the geometry of the terminators [132, 133]. These V1 neurons are the so called ”end-stopped cells”. They proposed a MT model that combines end-stopped V1 cells to explain the direction tuning of MT neurons to the two-dimensional stimuli [198]. In their model, the end-stopped cells are modeled with a divisive normalization from surrounding signals *i.e.* , a V1 cell’s response is suppressed when the stimulus’s length exceeds the preferred length.

Our current model accounts for the feedforward pathway of the motion signals (within the first 200 ms after the onset of the stimulus), while the neurophysiological experiments by Pack *et al.* account for the tuning of the MT neurons from the first 200 ms to 800ms after the onset of the stimulus. The end-stopped V1 cells have also been considered as the product of feedback signals from the high-level areas [149]. A Bayesian model accounting for the feedback pathway of HMAX has been proposed to explain visual attention [27]; whether this model could be extended to explain the direction tuning of MT neurons after the first 200ms of the stimulus onset will require further investigation.

Contrast Some tuning properties of V1 cells change as a function of the stimulus contrast. For example, the receptive field organization of V1 simple as well as complex cells change in a way that the optimal speed increases for the high-contrast stimuli [92]. The spectral receptive field (tuning to the spatial and temporal frequency) of V1 complex cells show a shift from separable tuning to speed tuning when the contrast of the stimulus increases [144]. It was also shown that the surround suppression of V1 cells is highly dependent on stimulus contrast, such that suppression is reduced or eliminated as contrast decreases [162]. In our model, the outputs of S_1 units are normalized with respect to the summed responses of a pool of S_1 units with the same receptive field but tuned to different directions. It is possible to incorporate into this pool the S_1 units from the surround in order to account for the suppression as well as for the dependence of the suppression on stimulus contrast. Indeed, a divisive normalization has been used to model the dependence of the suppression on stimulus contrast [198]. However, it requires further experiments to answer whether the divisive normalization can account for the dependence of the speed-tuning and optimal speed on stimuls contrast.

6.4 Comparing ”ventral + dorsal C_2 units” with STP cells

6.4.1 Introduction

The superior temporal polysensory area (STP) receives input from the inferior temporal cortex (IT), as well as more posterior dorsal-stream sources, such as area MST. STP neurons are selective to complex action sequences that encode both form and motion, such as specific body view and direction combinations [130, 129], but the roles of the ventral and dorsal pathways in action selective neurons remain unclear.

The role of ventral stream in shaping action selectivity is supported by the fact that many actions are easily recognizable when performed under a strobe light or even single images. A human fMRI study found a great increase of signals in MT/MST when a subject looked at static images of moving people, animals, or natural scenes [81]. It might be that some neurons are selective for actions because they are selective for particular poses that arise when those actions are performed (these neurons are called ”snapshot neurons” in [52]). Action selectivity might also be built from motion selectivity in the dorsal stream.

Integration of signals from neurons that are selective to relatively complex local motion patterns could give rise to the selectivity for motion vector fields that matched those generated by particular sorts of actions [67, 24, 75]. A compromising point of view is that STP neurons obtain the action selectivity by combining the form selectivity from the ventral and the motion selectivity from the dorsal stream [52]. Stimuli that contain only motion or form have been used to test such a hypothesis. The ”point-light walker” that applies lights to several points on otherwise invisible actors allows actions to be presented with greatly reduced form information [76]. The ”formless dot field” that is made of thousands of dots contains motion but no static form information [178]. (Also see more discussions about these stimuli in [178].)

To study the integration of shape and motion carried by the ventral and dorsal streams in STP neurons, we compare the responses of ”ventral + dorsal C_2 units” with that of STP neurons to complex action sequences that encode both form and motion. The 64 action sequences we used contain 8 types of actors performing 8 types of complex actions, generated by [178] and illustrated in Figure 6-28. In [178], monkeys were trained to recognize actions in these sequences, and a population of neurons was recorded from STP. To quantify the information conveyed by the population of cells, each neuron is treated as a rate-varying Poisson process with a mean firing rate λ, *i.e.* the number of spikes per unit of time. λ is estimated for each neuron from the responses to each stimulus i during each time bin of 40 ms, t, and therefore it is a function of time and stimulus. Such a model is then used to predict the probability of observing n spikes of the same neuron at the same time bin in response to each of the 64 stimuli j, as computed as follows:

$$P_(i, j, t) = \frac{e^{-\lambda(t,i)} \times \lambda(t,i)^{n(t,j)}}{n(t,j)} \; i = 1 \cdots 64, j = 1 \cdots 64 \qquad (6.27)$$

The probabilities are then combined over all the time bins and of all the neurons to obtain a 64×64 matrix, as shown in Figure 6-29. Each (x, y) entry in the matrix is the probability of obtaining response for stimulus y based on the responses for stimulus x. Along each axis of the confusion matrix, the 64 stimuli is arranged such that the first 8 stimuli are 8 actions performed by the same actor, the next 8 stimuli are 8 actions performed by another actor, and so on. These 8 types of actions are sorted in the same order in each block of the 8 actors. If a population of neurons is selective to actors regardless of the actions performed, the neuronal responses to one action can predict perfectly the responses to another action as long as they are performed by the same actor. Therefore, this population will result in a matrix with high values along each 8×8 block along the main diagonal (Figure 6-29: ”ideal data- actor selective”). Whereas for a population of neurons that is selective to actions regardless of the actor performing, the neuronal responses to one actor can predict perfectly the responses to another actor as long as they perform the same action. Therefore, this population will result in a matrix with high values at every 8 pixels along each column (or each row) (Figure 6-29: ”ideal data- action selective”). A population that contains both types of neurons will result in a matrix as a combination of the two ideal types of matrices. In [178], a population of cells was recorded from area STP of two monkeys. For one monkey, 31, 6 and 18 cells are selective to actors, actions, and pairs, respectively; therefore the matrix has stronger magnitude in 8×8 main-diagonal blocks, as shown in Figure 6-29: ”Monkey data - monkey Simon”. For another monkey, the numbers of cells of three types are 14, 3, 33, resulting in a matrix with stronger off-diagonal values, as shown in ”Monkey data - monkey Gal”.

6.4.2 Results

We computed the confusion matrix in a similar way with the firing rate replaced by the C_2 unit response and a time bin replaced by one frame (see Table 6.5 for a comparison). We choose randomly a population of 100-800 C_2 units from our model and the precedent ventral stream model [173]. We vary the number of C_2 units from each model and the resulting matrices lie approximately in a continuum, from high actor selectivity to high action selectivity (Figure 6-29, bottom row). Using ventral-only C_2 units results in a strong ”actor selective” matrix and the matrix gradually shifts to be more ”action selective” as more dorsal-C_2 units are added. Interestingly, the matrices obtained from STP neurons of the two monkeys seem to lie in the same continuum: the matrix for ”monkey Simon” is between that of (800, 0) C_2 units and that of (100, 0) C_2 units, and the matrix for ”monkey

Gal" between that of (100, 0) C_2 units and that of (100, 50) C_2 units. The similarity of the matrices derived from the monkeys and from our model suggests three things. First, the combination of C_2 stages of our dorsal stream model and the ventral stream model [173] might be a good model for some STP neurons. Second, the ventral and dorsal stream might both contribute to the selectivity of STP neurons to these actions sequences. Lastly, the selectivity to actions obtained for dorsal-only C_2 units is too high to be compatible with the physiology data, therefore the ventral stream may play a more significant role in forming the selectivity of STP neurons to these sequences of actions.

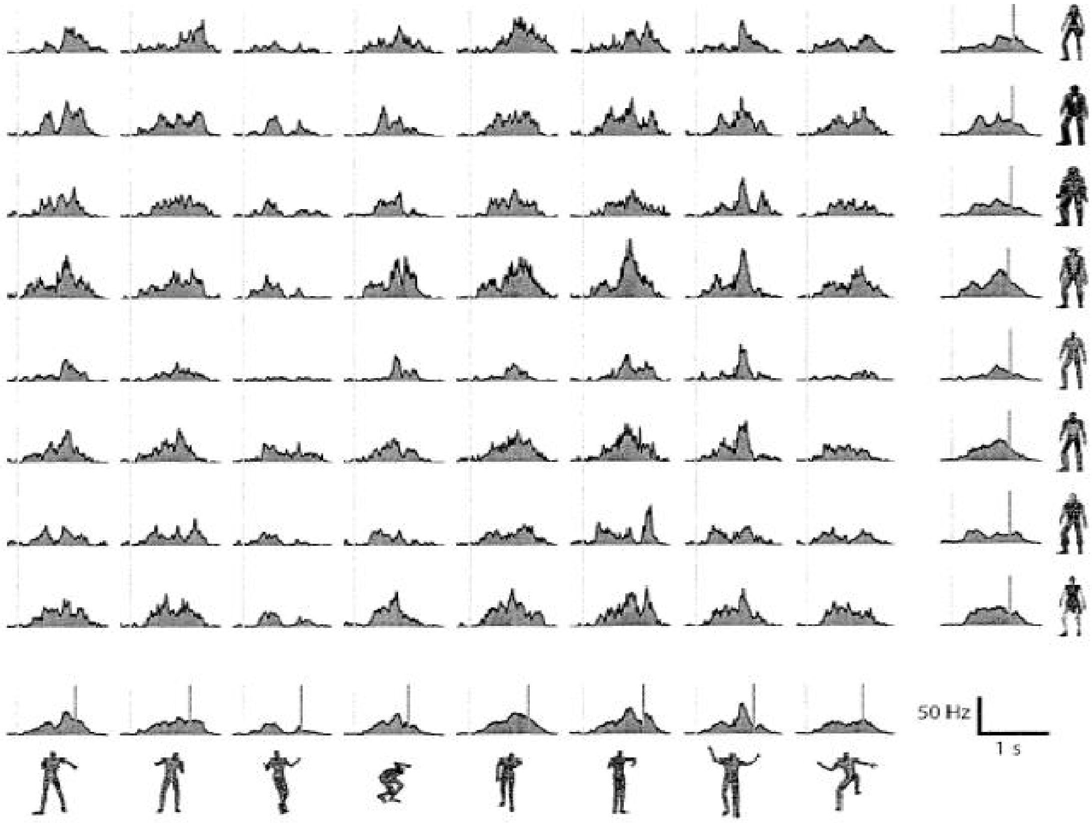


Figure 6-28: Response of a single cell from the lower bank of monkey STP. The main 8×8 grid of traces shows the mean firing rate in response to each specific combination of character and action. Characters are in rows, actions in columns. The actors appeared in a neutral pose at 0ms, and began to move at 300ms. Reprinted from [178].

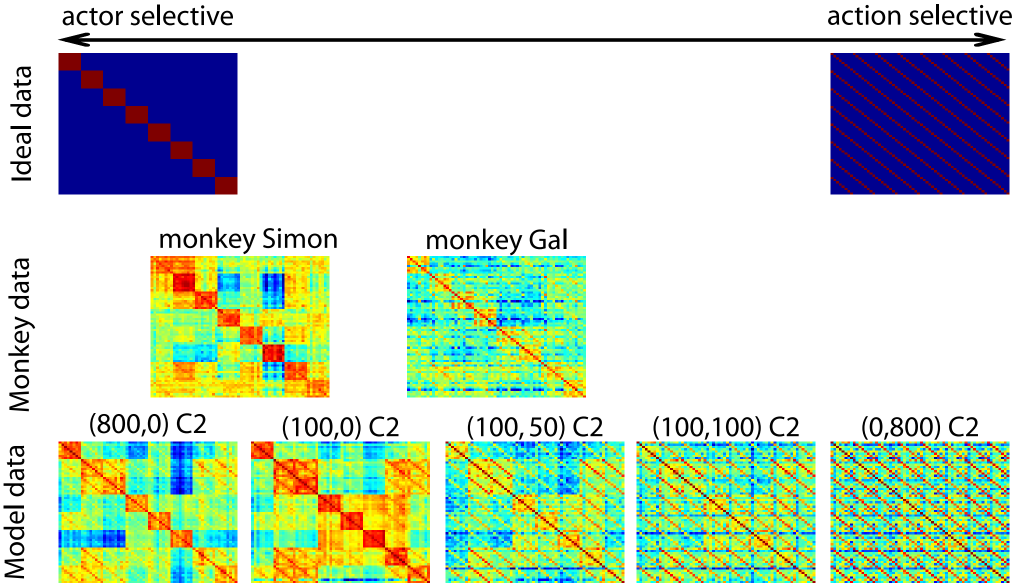


Figure 6-29: A comparison between the electrophysiology and model data. Each subplot corresponds to a confusion matrix obtained from ideal data (top row), monkey data (middle row) and the computational models (bottom row). High probability is indicated by deep red. Stimuli are sorted first by character and then by action. Ideal data (top row) describes the ideal case where the population of cells is tuned to characters (as indicated by 8×8 blocks of high probability on the main, left panel); single-pixel diagonal lines of high probability indicate correct classification of actions (right panel). High probability on the main diagonal indicates good performance at pair of character and action. The monkey data is shown in the middle row. Model data (bottom row), (n_1, n_2) C_2 corresponds to a combination of n_1 C_2 units of the ventral and n_2 C_2 units of the dorsal stream model.

	$\lambda(t, i)$	$n(t, j)$
Singer, 2009 [178]	Mean spiking rates estimated at time bin t in response to stimuli i	Average spike counts at time bin t in response to stimuli j
Our model	The response of a C_2 unit to the frame t of stimuli i	The response of a C_2 unit to the frame t of stimuli j

Table 6.5: The parameters of the Poisson model used in [178] and in our experiment.

6.5 Conclusion

HMAX [153, 173], a feedforward hierarchical architecture, was firstly designed for the recognition of objects and later on shown to explain the neuronal responses in several cor-

Area	Experiment	physiology data	Section	Figure
STP	action and actor selectivity	[178]	Sec. 6.4	Fig. 6-29
MT	direction tuning	[4]	Sec. 6.3.6	Fig. 6-15
MT	pattern direction sensitivity	[115]	Sec. 6.3.7	Fig. 6-24
MT	motion opponency	[180]	Sec. 6.3.7	Fig. 6-21
MT	local directional integration	[97]	Sec. 6.3.7	Fig. 6-24
MT	speed tuned	[144]	Sec. 6.3.7	Fig. 6-22
V1	speed tuned	[144]	Sec. 6.3.7	Fig. 6-22
V1	spatial and temporal frequency tuning	[47]	Sec. 6.2	Tab. 6.2
V1	direction tuning	[4]	Sec. 6.2	Tab. 6.2

Table 6.6: Physiological data the model can explain

tical areas of the ventral stream. In previous chapters, we have extended HMAX in the temporal dimension for the recognition of actions in videos. The model has been shown to perform on par or outperforms computer vision algorithms for the recognition of human actions [75, 82] as well as mice behaviors in videos [73]. In this chapter, we prove that the model can explain the neuronal responses in V1 and MT. When combining the C_2 outputs of the model with the C_2 outputs of HMAX, we can also explain the neuronal responses in STP. Specifically, the first two layers of the model mimic the V1 simple and V1 complex cells. We designed a population of spatio-temporal filters whose spatial and temporal frequency tuning closely match that of V1 cells. The latter two layers mimic the motion-sensitive MT cells. We showed that with a template matching operation in the S_2 and a max operation in the C_2 stage, the model can simulate the continuous distribution of pattern sensitivity [115], integration of directional signals in a local scale [97], and the tuning to the speed of the stimulus [144].

Table 6.6 shows a list of physiological experiments the proposed model can explain.

There are very few models that could explain neurophysiology as well as be applied to the real-world computer vision tasks. Our model is one that agrees with (or processes) data at different levels: from computer vision algorithm, practical software, to neuroscience.

Bibliography

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

[2] D G Albrecht and D B Hamilton. Striate cortex of monkey and cat: Contrast response function. *Journal of Neurophysiology*, 48(1):48217–231, 1982.

[3] Duane G Albrecht and Wilson S Geisler. Motion selectivity and the contrast-response function of simple cells in the visual cortex. *Visual Neuroscience*, 7:531–546, 1991.

[4] T D Albright. Direction and orientation selectivity of neurons in visual area MT the macaque. *Jounal of Neurophysiology*, 52(6), 1984.

[5] Kl Alexander, Marcin Marszalek, and Schmid Cordelia. A Spatio-Temporal Descriptor Based on 3D-Gradients. In *BMVC*, 2008.

[6] John Allman, Francis Miezin, and Evelynn Mcguinness. Neurophysiological Mechanisms for Local-Global Comparisons in Visual Neurons. *Annual review of neuroscience*, 8:407–430, 1985.

[7] Johan Auwerx and At El. The European dimension for the mouse genome mutagenesis program. *Nature Genetics*, 36(9):925–927, 2004.

[8] Serge Belongie, Kristin Branson, Piotr Dollár, and Vincent Rabaud. Monitoring Animal Behavior in the Smart Vivarium. *Measuring Behavior*, 2005.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

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

[11] A.F. Bobick. Movement , activity and action : the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London B*, 1997.

[12] Stephen J Bonasera, A Katrin Schenk, Evan J. Luxenberg, and Laurence H Tecott. A Novel Method for Automatic Quantification of Psychostimulant-evoked Route-tracing Stereotypy: A pplications to Mus Musculus. *J. Psychopharmacology*, 196(4):591–602, 2008.

[13] Richard T Born and David C Bradley. Structure and Function of Visual Area MT. *Annual review of neuroscience*, 28:157–189, 2005.

[14] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden Markov models for Complex Action Recognition. In *CVPR*, 1997.

[15] K Branson and S Belongie. Tracking multiple mouse contours (without too many samples). In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[16] Kristin Branson, Alice A Robie, John Bender, Pietro Perona, and Michael H Dickinson. High-throughput ethomics in large groups of Drosophila. *Nature Methods*, 6(6), 2009.

[17] Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist Based Acquisition and Tracking of Animal and Human Kinematics. *IJCV*, 56(3):179–194, February 2004.

[18] K H Britten. *The middle temporal area: motion processing and the link to perception*, pages 1203–1216. MIT Press, Cambridge, 2003.

[19] K H Britten and H W Heuer. Spatial summation in the receptive fields of MT neurons. *Journal of Neuroscience*, 19(12):5074–5084, 1999.

[20] K H Britten, W T Newsome, MN Shadlen, S Celebrini, and JA Movshon. A relationship between behavioral choice and the visual responses of neurons in macaque. *Vision Research*, 13(1):87–100, 1996.

[21] M Carandini, D J Heeger, and J A Movshon. Linearity and normalization in simple cells of the macaque primary visual cortex. *Journal of Neuroscience*, 17(21):8621–8644, 1997.

[22] Matteo Carandini and David J Heeger. Summation and Division by Neurons in Primate Visual Cortex. *Science*, 264(5163):1333–1336, 1994.

[23] Gemma Casadesus, Barbara Shukitt-hale, and James A Joseph. Automated measurement of age-related changes in the locomotor response to environmental novelty and home-cage activity. *Mechanisms of Ageing and Development*, 122:1887–1897, 2001.

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

[25] Gert Cauwenberghs and Tomaso Poggio. Incremental and Decremental Support Vector Machine Learning. *Neural Information Processing Systems*, 2001.

[26] Danica Chen, Andrew D Steele, Susan Lindquist, and Leonard Guarente. Increase in Activity During Calorie Restriction Requires Sirt1. *Science*, 310, 2005.

[27] S. Chikkerur, T. Serre, C. Tan, and T. Poggio. What and Where: A Bayesian inference theory of visual attention. *Vision Research*, 50(22), 2010.

[28] John C Crabbe, Douglas Wahlsten, and Bruce C Dudek. Genetics of mouse behavior: interactions with laboratory environment. *Science*, 284, 1999.

[29] N Dalal and B Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition.*, 2001.

[30] Heiko Dankert, Liming Wang, Eric D Hoopfer, David J Anderson, and Pietro Perona. Automated monitoring and analysis of social behavior in Drosophila. *Nature Methods*, 6(4), 2009.

[31] L De Visser, R Van Den Bos, W W Kuurman, M J H Kas, and B M Spruijt. Novel approach to the behavioural characterization of inbred mice: automated home cage observations. *Genes, brain, and behavior*, 5(6):458–66, 2006.

[32] G C Deangelis, J G Robson, I Ohzawa, and R D Freeman. Organization of Suppression in Receptive Fields of Neurons in Cat Visual Cortex. *Journal of Neurophysiology*, 68(1), 1992.

[33] Gregory C Deangelis, Izumi Ohzawa, and Ralph D Freeman. Receptive-field dynamics in the central visual pathways. *Trends in Neurosciences*, 18(10):451–458, 1995.

[34] Giacomo Dell'Omo, Elisabetta Vannoni, Alexei L Vyssotski, Michele Angelo Di Bari, Romolo Nonno, Umberto Agrimi, and Hans-peter Lipp. Early behavioural changes in mice infected with BSE and scrapie: automated home cage monitoring reveals prion strain differences. *European Journal of Neuroscience*, 16:735–742, 2002.

[35] J Deng, W Dong, R Socher, LJ Li, K Li, and L Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009.

[36] P Dollar, V Rabaud, G Cottrell, and S Belongie. Behavior recognition via sparse spatio-temporal features. In *visual surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[37] Kevin D Donohue, Dharshan C Medonza, Eli R Crane, and Bruce F O'Hara. Assessment of a non-invasive high-throughput classifier for behaviours associated with sleep and wake in mice. *Biomedical engineering online*, 7:14, 2008.

[38] A A Efros, A C Berg, G Mori, and J Malik. Recognizing action at a distance. In *ICCV*, volume 12345, 2003.

[39] Wolfgang Einhauser, Christoph Kayser, Peter Konig, and Konard Kordig. Learning the invariance properties of complex cells from their responses to natural stimuli. *European Journal of Neuroscience*, 15:457–486, 2002.

[40] Rober C. Emerson, James R Bergen, and Edward H Adelson. Directionally selective complex cells and the computation of motion energy in cat visual cortex. *Vision Research*, 32(2):203–218, 1992.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[42] M Fahle and T Poggio. Visual Hyperacuity : Spatiotemporal Interpolation in Human Vision. *Proc. R. Soc. Land*, 213(1193):451–477, 1981.

[43] C Fanti, L Zelnik-manor, and P Perona. Hybrid models for human motion recognition. In *CVPR*, volume 2, 2005.

[44] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *IEEE CVPR Workshop on Generative-Model Based Vision*, 2004.

[45] Daniel J. Felleman and Jon H. Kaas. Receptive-Field Properties of Neurons in Middle Temporal Visual Area (MT) of Owl Monkeys. *Journal of Neurophysiology*, 52(3), 1984.

[46] Ian M Finn and David Ferster. Computational Diversity in Complex Cells of Cat Primary Visual Cortex. *Journal of Neuroscience*, 27(36):9638–9647, 2007.

[47] K H Foster, J P Gaska, M Nagler, and D A Pollenn. Spatial and temporal frequency selectivity of neurones in visual cortical areas V1 and V2 of the macaque monkey. *Jounal of Physiology*, 365:331–363, 1985.

[48] Steven N Fry, Nicola Rohrseitz, Andrew D Straw, and Michael H Dickinson. TrackFly: virtual reality for a behavioral system analysis in free-flying fruit flies. *Journal of Neuroscience Methods*, 171:110–117, 2008.

[49] Kunihiko 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.

[50] K S Gannon, J C Smith, R Henderson, and P Hendrick. A system for studying the microstructure of ingestive behavior in mice. *Physiology & Behavior*, 51(3):515–21, March 1992.

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

[52] M A Giese and T Poggio. Neural Mechanisms for the Recognition of Biological Movements. *Nature Review Neuroscience*, 4:179–192, 2003.

[53] M A Goodale. Action Insight: The Role of the Dorsal Stream in the Perception of Grasping. *Neuron*, 47(3):328–329, 2005.

[54] M A Goodale and A D Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[55] Evan H Goulding, A Katrin Schenk, Punita Juneja, Adrienne W Mackay, Jennifer M Wade, and Laurence H Tecott. A robust automated system elucidates mouse home cage behavioral structure. *Proceedings of the National Academy of Sciences*, 105(52), 2008.

[56] S A Graziano, A Andersen, and Robert J Snowden. Tuning of MST Neurons to Spiral Motions. *Journal of Neuroscience*, 14(1):54–67, 1994.

[57] Joy M Greer and Mario R Capecchi. Hoxb8 Is Required for Normal Grooming Behavior in Mice. *Neuron*, 33:23–34, 2002.

[58] S Grossberg, E Mingolla, and L Viswanathan. Neural dynamics of motion integration and segmentation within and across apertures. *Vision Research*, 41:2521–2553, 2001.

[59] N.M. Grzywacz, J.M. Harris, and F.R. Amthor. *Computational and Neural Constraints for the Measurement of Local Visual Motion*, chapter 2, pages 19–50. Academic Press, London, 1994.

[60] N.M. Grzywacz and A.L. Yuille. A Model for the Estimate of Local Velocity. *Lecture Notes in Computer Science*, 427:331–335, 1990.

[61] Y. Guo, G Xu, and S. Tsuji. Tracking Human Body Motion Based on a Stick Figure Model. *JVCIR*, 5:1–9, 1994.

[62] Michael J Hawken, A.J. Parker, and Jennifer S Lund. Laminar organization and contrast sensitivity of direction-selective cells in the striate cortex of the old world monkey. *October*, 8(10):3541–3548, 1988.

[63] MJ Hawken, RM Shapley, and DH. Grosof. Temporal-frequency selectivity in monkey visual cortex. *Visual Neuroscience*, 13(3):477–492, 1996.

[64] David J Heeger. Model for the extraction of image flow. *Journal of Optical Society of America A*, 4(8):1455–1474, 1987.

[65] David J Heeger. Modeling Simple-Cell Direction Selectivity With Normalized, Half-Squared, Linear Operators. *Jounal of Neurophysiology*, 70(5), 1993.

[66] Thomas C Henderson and Xinwei Xue. Constructing Comprehensive Behaviors : A Simulation Study. In *International Conference on Computer Applications in Industry and Engineering*, 2005.

[67] D D Hoffman and B E Flinchbaugh. The Interpretation of Biological Motion. *Biological Cybernetics*, 42:195–204, 1982.

[68] D H Hubel and T N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *Jounal of Physiology*, 148:574–591, 1959.

[69] D H Hubel and T N Wiesel. Receptive fields, binocular interaction and functional architecture in the cats visual cortex. *Jounal of Physiology*, 160:106–54, 1962.

[70] D H Hubel and T N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of physiology*, 195:215–243, 1968.

[71] Walker S Jackson, Sara J Tallaksen-greene, Roger L Albin, and Peter J Detloff. Nucleocytoplasmic transport signals affect the age at onset of abnormalities in knock-in mice expressing polyglutamine within an ectopic protein context. *Human Molecular Genetics*, 12(13):1621–1629, 2003.

[72] H. Jhuang, E. Garrote, N. Edelman, T. Poggio, T. Serre, and A. Steele. Trainable, Vision-Based Automated Home Cage Behavioral Phenotyping. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, 2010.

[73] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A Steele, and T Serre. Automated Home-Cage Behavioral Phenotyping of Mice. *Nature communications*, 2010.

[74] H. Jhuang, T. Serre, and T. Poggio. Computational Mechanisms for the Motion Processing in Visual Area MT. In *Program No 731.11, 2010 Neuroscience Meeting Planner*, San Diego, CA: Society for Neuroscience, 2010.

[75] H Jhuang, T Serre, L Wolf, and T Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[76] Gunnra Johansson. Visual Perception of Biological Motion and a Model for Its Analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[77] Zia Khan, Tucker Balch, and Frank Dellaert. MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. *Pattern Analysis and Machine Intelligence*, 27:1805–1819, 2005.

[78] J J Koenderink and Doorn A J Van. Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55:367–375, 1987.

[79] Minjoon Kouh. *Toward a More Biologically Plausible Model of Object Recognition*. PhD thesis, MIT, 2007.

[80] Minjoon Kouh and Tomaso Poggio. A Canonical Neural Circuit for Cortical Nonlinear Operations. *Neural Computation*, 20:1427–1451, 2008.

[81] Zoe Kourtzi and Nancy Kanwisher. Activation in human MT/MST by static images with implied motion. *Journal of Cognitive Neuroscience*, 12(1):48–55, 2000.

[82] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB : A Large Video Database for Human Motion Recognition. In *In submission*, 2011.

[83] 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.

[84] Ilan Lampl, David Ferster, Tomaso Poggio, and Maximilian Riesenhuber. Intracellular Measurements of Spatial Integration and the MAX Operation in Complex Cells of the Cat Primary Visual Cortex. *Journal of Neurophysiology*, 92:2704 –2713, 2004.

[85] I Laptev and T Lindeberg. Space-Time Interest Points. In *ICCV*, 2003.

[86] I. Laptev, M. Marszaek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[87] I Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007.

[88] Yann Lecun and Yoshua Bengio. *Convolutional Networks for Images , Speech , and Time-Series*. MIT Press, 1995.

[89] Toon Leroy, Stijn Stroobants, Jean-Marie Aerts, Rudi D'Hooge, and Daniel Berckmans. Automatic analysis of altered gait in arylsulphatase A-deficient mice in the open field. *Behavior Research Methods*, 41:787–794, 2009.

[90] Lars Lewejohann, Anne Marie Hoppmann, Philipp Kegel, Mareike Kritzler, Antonio Krüger, and Norbert Sachser. Behavioral phenotyping of a murine model of Alzheimer's disease in a seminaturalistic environment using RFID tracking. *Behavior Research Methods*, 41:850–856, 2009.

[91] J. Liu, J. Luo, and M. Shah. Recognizing Realistic Actions from Videos ”in the Wild”. In *ICCV*, 2009.

[92] Margaret S Livingstone and Bevil R Conway. Contrast Affects Speed Tuning , Space-Time Slant , and Receptive-Field Organization of Simple Cells in Macaque V1. *Journal of Neurophysiology*, 97:849–857, 2007.

[93] Margaret S Livingstone, Christopher C Pack, and Richard T Born. Two-Dimensional Substructure of MT Receptive Fields. *Neuron*, 30:781–793, 2001.

[94] Gunter Loffler and Harry S Orbach. Modeling the integration of motion signals across space. *Journal of Optical Society of America A*, 20(8):1472–1489, 2003.

[95] B D Lucas and T Kanade. An iterative image registration technique with an application to stereo vision. In *Joinnt Conf. on Art. Intell*, pages 121–130, 1981.

[96] Lamberto Maffei and Adriana Fiorentini. The visual cortex as a spatial analyser frequency. *Vision Research*, 13:1255–1267, 1973.

[97] Najib J Majaj, Matteo Carandini, and J Anthony Movshon. Motion Integration by Neurons in Macaque MT Is Local, Not Global. *Journal of Neuroscience*, 27(2):366 –370, 2007.

[98] D. Marr and Lucia Vaina. Representation and Recognition of the Movements of Shapes. *Biological Sciences*, 214(1197):501–524, 1982.

[99] M Marszaek, I Laptev, and C Schmid. Actions in context. In *CVPR*, 2009.

[100] Timothee Masquelier, Thomas Serre, Simon Thorpe, and Tomaso Poggio. Learning complex cell invariance from natural videos: A plausibility proof, 2007.

[101] Timothee Masquelier and Simon J Thorpe. Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. *PLoS computational biology*, 3(2), 2007.

[102] H G Mcfarlane, G K Kusek, M Yang, J L Phoenix, V J Bolivar, and J N Crawley. Autism-like behavioral phenotypes in BTBR T1tf/J mice. *Genes, Brain and Behavior*, 7:152–163, 2008.

[103] J. McLean and L. A. Palmera. Contribution of linear spatiotemporal receptive field structure to velocity selectivity of simple cells in area 17 of the cat. *Vision Research*, 29:657–679, 1989.

[104] J. McLean, S. Raaba, and L. A. Palmera. Contribution of linear mechanisms to the specification of local motion by simple cells in areas 17 and 18 of the cat. *Visual Neuroscience*, 11:295–306, 1994.

[105] A.A.H.P. Megens, J. Voeten, J. Rombouts, T.F. Meert, and C.J.E. Niemegeers. Behavioural activity of rats measured by a new method based on the piezo-electric principle. *Psychopharmacology*, 93(3):382–388, November 1987.

[106] A Mikami, W T Newsome, and R H Wurtz. Motion selectivity in macaque visual cortex. I. Mechanisms of direction and speed selectivity in extrastriate area MT. *Journal of Neurophysiology*, 55(6):1308–1327, 1986.

[107] Akichika Mikami, William T Newsome, and Robert H Wurtz. Motion Selectivity in Macaque Visual Cortex. II. Spatiotemporal Range of Directional Interactions in MT and Vl. *Journal of Neurophysiology*, 55(6), 1986.

[108] Magali Millecamps, Didier Jourdan, Sabine Leger, Monique Etienne, Alain Eschalier, and Denis Ardid. Circadian pattern of spontaneous behavior in monarthritic rats: a novel global approach to evaluation of chronic pain and treatment effectiveness. *Arthritis and rheumatism*, 52(11):3470–8, 2005.

[109] Thomas B Moeslund, Adrian Hilton, and Volker Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[110] Bartlett D Moore Iv. Speed Selectivity in V1 : A Complex Affair. *Journal of Neuroscience*, 26(29):7543–7544, 2006.

[111] J A Movshon and W T Newsome. Visual response properties of striate cortical neurons projecting to area MT in macaque monkeys. *Journal of Neuroscience*, 16(23):7733–7741, 1996.

[112] J A Movshon, W T Newsome, M S Gizzi, and J B Levitt. spatio-temporal tuning and speed sensitivity in macaque cortical neurons. *Invest. Ophthalmol. Vis. Sci. Suppl.*, 29:327, 1988.

[113] J A Movshon, I D Thompson, and D J Tolhurst. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *Jounal of Physiology*, 283:79–99, 1978.

[114] J Anthony Movshon. The Velocity Tuning of Single Uints in Cat Striate Cortex. *Journal of physiology*, 249:445–468, 1975.

[115] J Anthony Movshon, Edward H Adelson, Martin S Gizzi, and William T Newsome. *Tha analysis of moving visual patterns*, pages 116–151. Rome: Vatican Press., 1983.

[116] J Anthony Movshon, I D Thompson, and D J Tolhurst. Receptive field organization of complex cells in the cat's striate cortex. *Jounal of Physiology*, 283:79–99, 1978.

[117] J Anthony Movshon, I D Thompson, and D J Tolhurst. Spatial And Temporal Contrast Sensitivity Of Neurones In Areas 17 And 18 Of The Cat's Visual Cortex. *Jounal of Physiology*, 283:101–120, 1978.

[118] J. Mutch, U. Knoblich, and T. Poggio. CNS: a GPU-based framework for simulating cortically-organized networks., 2010.

[119] Jim Mutch and David Lowe. Multiclass Object Recognition Using Sparse, Localized Features. In *CVPR*, 2006.

[120] W T Newsome, M S Gizzi, and J A Movshon. Spatial and temporal properties of neurons in macaque MT. *Invest. Ophthalmol. Vis. Sci. Suppl.*, 24:106, 1983.

[121] J. Niebles, H. Wang, and L. Fei Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[122] Juan Carlos Niebles, Chih-wei Chen, and Li Fei-fei. Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In *ECCV*, pages 1–14, 2010.

[123] Juan Carlos Niebles and Li Fei-Fei. A Hierarchical Model of Shape and Appearance for Human Action Classification. *CVPR*, June 2007.

[124] L P Noldus, A J Spink, and R A Tegelenbosch. EthoVision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods, Instruments, Computers*, 33(3):398–414, August 2001.

[125] Steven Nowlan and Terrence J Sejnowski. A Selection Model for Motion Processing in Area MT of Primates. *Journal of Neuroscience*, 15(2):1195–1214, 1995.

[126] Hiroaki Okamoto, Susumu Kawakami, Hide-aki Saito, Eiki Hida, Keiichi Odajima, Daichi Tamanoi, and Hiroshi Ohno. MT neurons in the macaque exhibited two types of bimodal direction tuning as predicted by a model for visual motion detection. *Vision Research*, 39(20):2331–55, 1999.

[127] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[128] Bruno A Olshausen. Learning Sparse, Overcomplete Representations of Time-Varying Natural Images. In *ICIP*, number 2, 2003.

[129] M W Oram and D I Perrett. Integration of form and motion in the anterior superior temporal polysensory area (STPa) of the macaque monkey. *Jounal of Neurophysiology*, 76(1):109–129, 1996.

[130] W Oram and D I Perrett. Responses of Anterior Superior Temporal Polysensory (STPa) Neurons to ”Biological Motion” stimuli. *Journal of Cognitive Neuroscience*, 6(2):99–116, 1994.

[131] G A Orban, H Kennedy, and J Bullier. Velocity sensitivity and direction selectivity of neurons in areas VI and V2 of the monkey: influence of eccentricity. *Journal of Neurophysiology*, 56(2):462–480, 1986.

[132] C C Pack, M S Livingstone, K R Duffy, and R T Born. End-stopping and the aperture problem: Two-dimensional motion signals in macaque V1. *Neuron*, 39:671–680, 2003.

[133] Christopher C Pack, Andrew J Gartland, and Richard T Born. Integration of Contour and Terminator Signals in Visual Area MT of Alert Macaque. *Journal of Neuroscience*, 24(13):3268–80, 2004.

[134] J A Perrone and A Thiele. Speed skills: measuring the visual speed analyzing properties of primate MT neurons. *Nature Neuroscience*, 4(5):526–532, 2001.

[135] John A Perrone. A visual motion sensor based on the properties of V1 and MT neurons. *Vision Research*, 44:1733–1755, 2004.

[136] John A Perrone. A Single Mechanism Can Explain the Speed Tuning Properties of MT and V1 Complex Neurons. *Journal of Neuroscience*, 26(46):11987–11991, 2006.

[137] John A Perrone and Richard J Krauzlis. Spatial integration by MT pattern neurons : A closer look at pattern-to-component effects and the role of speed tuning. *Journal of Vision*, 8(9):1–14, 2008.

[138] John A Perrone and Alexander Thiele. A model of speed tuning in MT neurons. *Vision Research*, 42:1035–1051, 2002.

[139] T. Poggio and W. Reichardt. Visual Fixation and Tracking by Flies: Mathematical Properties of Simple Control Systems. *Biological Cybernetics*, 40:101–112, 1981.

[140] R. Polana and R. Nelson. Detecting activities. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–7. IEEE Comput. Soc. Press, 1993.

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

[142] Nicholas J Priebe, Carlos R Cassanello, and Stephen G Lisberger. The Neural Representation of Speed in Macaque Area MT/V5. *Journal of Neuroscience*, 23(13):5650 –5661, 2003.

[143] Nicholas J Priebe and David Ferster. Mechanisms underlying cross-orientation suppression in cat visual cortex. *Nature Neuroscience*, 9(4):552–561, 2006.

[144] Nicholas J Priebe, Stephen G Lisberger, and J Anthony Movshon. Tuning for Spatiotemporal Frequency and Speed in Directionally Selective Neurons of Macaque Striate Cortex. *Journal of Neuroscience*, 26(11):2941–2950, 2006.

[145] Ning Qian and A Andersen. Transparent Motion Perception as Detection of Unbalanced Motion Signals. II. Physiology. *Journal of Neuroscience*, 14(12):7357–7380, 1994.

[146] Ning Qian, Richard A Andersen, and Edward H Adelson. Transparent Motion Perception as Detection of Unbalanced Motion Signals. I. Psychophysics. *Journal of Neuroscience*, 14(12):7357–7366, 1994.

[147] Deva Ramanan and D A Forsyth. Automatic Annotation of Everyday Movements. In *NIPS*, number July, 2003.

[148] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In *CVPR*, June 2007.

[149] Rajesh P N Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999.

[150] G H Recanzone, R H Wurtz, and U Schwarz. Responses of MT and MST Neurons to One and Two Moving Objects in the Receptive Field. *Journal of Neurophysiology*, 78:2904–2915, 1997.

[151] W. Reichardt. *Autocorrelation, a principle for the evaluation of sensory information by the central nervous system*, pages 303–317. MIT Press, New York, 1961.

[152] M Riesenhuber and T Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(24):1019–1025, 1999.

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

[154] H R Rodman and T D Albright. Single-unit analysis of pattern-motion selective properties in the middle temporal visual area (MT). *Experimental Brain Research*, 75:53–64, 1989.

[155] Hillary E Rodman and Thomas D Albright. Coding of visual stimulus velocity area MT of the macaque. *Vision Research*, 27(12):2035–2048, 1987.

[156] M. Rodriguez, J. Ahmed, and M. Shah. Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition. In *CVPR*, 2008.

[157] E T Rolls and S M Stringer. Invariant global motion recognition in the dorsal visual system: A unifying theory. *Neural Computation*, 19(1):139–169, 2007.

[158] J V Roughan, S L Wright-Williams, and P A Flecknell. Automated analysis of postoperative behaviour: assessment of HomeCageScan as a novel method to rapidly identify pain and analgesic effects in mice. *Lab Animal*, 2008.

[159] BC Russell, A Torralba, KP Murphy, and WT Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1):157–173, 2008.

[160] NiCole Rust. *Signal transmission, feature representation and computation in areas V1 and MT of the macaque monkey*. Phd, NYU, 2004.

[161] Nicole C Rust, Valerio Mante, Eero P Simoncelli, and J Anthony Movshon. How MT cells analyze the motion of visual patterns. *Nature Neuroscience*, 9(11):1421–1431, 2006.

[162] Michael P Sceniak, Dario L Ringach, Michael J Hawken, and Robert Shapley. Contrast s effect on spatial summation by macaque V1 neurons. *Nature Neuroscience*, 2(8):733–739, 1999.

[163] Peter H Schiller, Barbara L Finlay, and Susan F Volman. Quantitative studies of single-cell properties in monkey striate cortex. I. Spatiotemporal organization of receptive fields. *Jounal of Neurophysiology*, 39(6):1288–1319, 1976.

[164] Konrad Schindler and Luc Van Gool. Action Snippets: How many frames does human action recognition require? In *CVPR*, 2008.

[165] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, pages 32–36, 2004.

[166] G Sclar, J H R Maunsell, and P Lennie. Coding of image contrast in central visual pathways of the macaque monkey. *Vision Research*, 30(1):1–10, 1990.

[167] Margaret E Sereno. *Neural Computation of Pattern Motion: Modeling stages of motion analysis in the primate visual cortex.* MIT Press/Bradford Books., Cambridge, 1993.

[168] Martin I Sereno. Learning The Solution To The Aperture Problem For Pattern Motion With A Hebb Rule. In *Neural Information Processing Systems*, 2001.

[169] Martin I Sereno and Margaret E Sereno. Learning to See Rotation and Dilation with a Hebb Rule. In *Neural Information Processing Systems*, pages 320–326, 1991.

[170] T.* Serre, H.* Jhuang, E. Garrote, T. Poggio, and Steele A. Automatic recognition of rodent behavior: A tool for systematic phenotypic analysis, 2009.

[171] T Serre, L Wolf, and T Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, volume 127, 2005.

[172] Thomas Serre. *Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines*. PhD thesis, MIT, 2006.

[173] Thomas Serre, Lior Wolf, Stanley Bileschi, and Maximilian Riesenhuber. Robust Object Recognition with Cortex-Like Mechanisms. *Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[174] Maggie Shiffrar and Jennifer J Freyd. Shiffirar_PsychologyScience_90.pdf. *Psychological Science*, 1(4):257–264, 1990.

[175] Rodrigo Sigala, Thomas Serre, Tomaso Poggio, and Martin Giese. Learning Features of Intermediate Complexity for the Recognition of Biological Motion. In *ICANN*, 2005.

[176] E Simoncelli and D J Heeger. Representing retinal image speed in visual cortex. *Nature Neuroscience*, 4(5):461–462, 2001.

[177] E P Simoncelli and D J Heeger. A Model of Neuronal Responses in Visual Area MT. *Vision Research*, 38(5):743–761, 1997.

[178] Jedediah Miller Singer. *Vision Over Time: Temporal Integration in the Temporal Lobe*. phd, Brown Univeristy, 2009.

[179] Matthew A Smith, Najib J Majaj, and J Anthony Movshon. Dynamics of motion signaling by neurons in macaque area MT. *Nature Neuroscience*, 8(2):220–228, 2005.

[180] Robert J Snowden, Stefan Treue, Roger G Erickson, and A Andersen. The Response of Area MT and V1 Neurons to Transparent Motion. *Journal of Neuroscience*, 11(9):2768–2785, 1991.

[181] Sen Song, Kenneth D Miller, and L F Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, 2000.

[182] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *CVPR*, 2008.

[183] A J Spink, R A J Tegelenbosch, M O S Buma, and L P J J Noldus. The EthoVision video tracking system-A tool for behavioral phenotyping of transgenic mice. *Physiology & Behavior*, 73:719–730, 2001.

[184] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*. I, 1998.

[185] Andrew D Steele, Walker S Jackson, Oliver D King, and Susan Lindquist. The power of automated high-resolution behavior analysis revealed by its application to mouse models of Huntingtons and prion diseases. *Proceedings of the National Academy of Sciences*, 104(6):1983–1988, 2007.

[186] G.R. Stoner, T.D. Albright, and V.S. Ramachandran. Transparency and coherence in human motion perception. *Nature*, 344(6262):153–155, 1990.

[187] G.R. Stoner and Thomas D Albright. Neural correlates of perceptual motion coherence. *Nature*, 358(30):412–414, 1992.

[188] P. Tamborini, H. Sigg, and Zbinden G. Quantitative analysis of rat activity in the home cage by infrared monitoring. Application to the acute toxicity testing of acetanilide and phenylmercuric acetate. *Archives of Toxicology*, 63:85–96, 1989.

[189] Xiangdong Tang and Larry D Sanford. Home cage activity and activity-based measures of anxiety in 129P3/J, 129X1/SvJ and C57BL/6J mice. *Physiology & behavior*, 84(1):105–15, 2005.

[190] Laurence H Tecott and Eric J Nestler. Neurobehavioral assessment in the information age. *Nature Neuroscience*, 7(5):462–466, 2004.

[191] M. Thirkettle, C.P. Benton, and N.E. Scott-Samuel. Contributions of form, motion and task to biological motion perception. *Journal of Vision*, 9(3):1–11, 2009.

[192] D J Tolhurst and J Anthony Movshon. Spatial and temporal contrast sensitivity of striate cortical neurones. *Nature*, 257:674–675, 1975.

[193] A Torralba, R Fergus, and WT Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Ieee T Pattern Anal*, pages 1958–1970, 2008.

[194] A Torralba, BC Russell, and J Yuen. Labelme: online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

[195] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support Vector Machine Learning for Interdependent and Structured Output Spaces. *International Conference on Machine Learning*, 2004.

[196] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[197] John K Tsotsos, Yueju Liu, Julio C Martinez-trujillo, Marc Pomplun, Evgueni Simine, and Kunhao Zhou. Attending to visual motion. *Computer Vision and Image Understanding*, 100:3–40, 2005.

[198] James M G Tsui, J Nicholas Hunter, Richard T Born, and Christopher C Pack. The Role of V1 Surround Suppression in MT Motion Integration. *Journal of Neurophysiology*, 103:3123–3138, 2010.

[199] Pavan Turaga, Student Member, Rama Chellappa, and V S Subrahmanian. Machine Recognition of Human Activities : A survey. *Circuits and Systems for Video Technology*, 18(11):1473 – 1488, 2008.

[200] C. J. Twining, C. J. Taylor, and P. Courtney. Robust tracking and posture description for laboratory rodents using active shape models. *Behavior Research Methods, Instruments, & Computers*, 33(3):381–391, 2001.

[201] Leslie G Ungerleider, Susan M Courtney, and James V Haxby. A neural system for human visual working memory. *Proceedings of the National Academy of Sciences*, 95:883–890, 1998.

[202] Leslie G Ungerleider and M Mishkin. *Two cortical visual systems*, pages 549–586. MIT Press, Cambridge, 1982.

[203] LG Ungerleider and Robert Desimone. Cortical connections of visual area MT in the macaque. *Journal of Comparative Neurology*, 248(2):190–222, 1986.

[204] David C Van Essen and Jack Gallant. Neural Mechanisms of Form and Motion Processing in the Primate Visual System. *Neuron*, 13:1–10, 1994.

[205] J H Van Hateren and D L Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proc. Royal Society ser. B*, 265:2315–2320, 1998.

[206] P.B.A Van Lochem, M.O.S Buma, J.B.I Rousseau, and L.P.J.J Noldus. Automatic recognition of behavioral patterns of rats using video imaging and statistical classification. In *Measuring Behavior*, 1998.

[207] Jan P H Van Santen and George Sperling. Elaborated Reichardt detectors. *Journal of Optical Society of America A*, 2(2):300–321, 1985.

[208] Ashok Veeraraghavan, Rama Chellappa, and Mandyam Srinivasan. Shape-and-behavior encoded tracking of bee dances. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):463–76, March 2008.

[209] Paul Viola and Michael Jones. Robust Real-time Object Detection. In *ICCV*, 2001.

[210] Heng Wang, Muhammad Muneed Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[211] Ruye Wang. A Network Model of Motion Processing in Area MT of Primates. *Journal of Computational Neuroscience*, 4:287–308, 1997.

[212] Watson and Ahumada. Model of human visual-motion sensing. *Journal of Optical Society of America A*, 2(2):322, 1985.

[213] D. Weinland, R. Ronfard, and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. *Computer Vision and Image Understanding*, 104:249–257, 2006.

[214] D. Weinland, R. Ronfard, and E. Boyer. A Survey of Vision-Based Methods for Action Representation, Segmentation and Recognition. *Computer Vision and Image Understanding*, 2010.

[215] Jason Weston, Andre Elisseeff, Bernhard Scholkopf, and Mike Tipping. Use of the Zero-Norm with Linear Models and Kernel Methods. *Journal of Machine Learning Research*, 3(7-8):1439–1461, October 2003.

[216] Kylie J Wheaton, Andrew Pipingas, Richard B Silberstein, and Aina Puce. Human neural responses elicited to observing the actions of others. *Visual Neuroscience*, 18:401–406, 2001.

[217] J Xiao, J Hays, KA Ehinger, A Oliva, and A Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3485–3492, 2010.

[218] Xinwei Xue and Thomas C Henderson. Video Based Animal Behavior Analysis From Multiple Cameras. *International Conference On Multisensor Fusion and Integration for Intelligent Systems*, pages 335–340, 2006.

[219] Xinwei Xue and Thomas C. Henderson. Feature fusion for basic behavior unit segmentation from video sequences. *Robotics and Autonomous Systems*, 57(3):239–248, March 2009.

[220] Yaser Yacoob and Michael J Black. Parameterized Modeling and Recognition of Activities. In *CVIU*, 1999.

[221] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[222] Richard A Young and Ronald M Lesperance. The Gaussian Derivative model for spatial-temporal vision: II. Cortical data. *Spatial Vision*, 14(3):321– 389, 2001.

[223] L Zelnik-manor and M Irani. Event-based analysis of video. In *CVPR*, 2001.

[224] Eric P Zorrilla, Koki Inoue, Eva M Fekete, Antoine Tabarin, Glenn R Valdez, and George F Koob. Measuring meals: structure of prandial food and water intake of rats. *American journal of physiology. Regulatory, integrative and comparative physiology*, 288(6):1450–67, 2005.

[225] Jane Brooks Zurn, Xianhua Jiang, and Yuichi Motai. Video-Based Tracking and Incremental Learning Applied to Rodent Behavioral Activity Under Near-Infrared Illumination. *IEEE Transactions on Instrumentation and Measurement*, 56(6):2804–2813, December 2007.