Vision Research 126 (2016) 232–241

Contents lists available at ScienceDirect

Vision Research

journal homepage: www.elsevier.com/locate/visres

Discovering hierarchical motion structure

Samuel J. Gershman [a,*], Joshua B. Tenenbaum [a], Frank Jäkel [b]

[a] *Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b] *Institute of Cognitive Science, University of Osnabrück, Germany*

ARTICLE INFO

Article history:
Received 29 November 2014
Received in revised form 2 March 2015
Available online 26 March 2015

Keywords:
Motion perception
Bayesian inference
Structure learning

ABSTRACT

Scenes filled with moving objects are often hierarchically organized: the motion of a migrating goose is nested within the flight pattern of its flock, the motion of a car is nested within the traffic pattern of other cars on the road, the motion of body parts are nested in the motion of the body. Humans perceive hierarchical structure even in stimuli with two or three moving dots. An influential theory of hierarchical motion perception holds that the visual system performs a "vector analysis" of moving objects, decomposing them into common and relative motions. However, this theory does not specify how to resolve ambiguity when a scene admits more than one vector analysis. We describe a Bayesian theory of vector analysis and show that it can account for classic results from dot motion experiments, as well as new experimental data. Our theory takes a step towards understanding how moving scenes are parsed into objects.

1. Introduction

Motion is a powerful cue for understanding the organization of a visual scene. Infants use motion to individuate objects, even when it contradicts property/kind information (Kellman & Spelke, 1983; Xu & Carey, 1996; Xu, Carey, & Welch, 1999). The primacy of motion information is also evident in adult object perception (Burke, 1952; Flombaum & Scholl, 2006; Mitroff & Alvarez, 2007) and non-human primates (Flombaum et al., 2004). For example, in the *tunnel effect* (Burke, 1952; Flombaum & Scholl, 2006; Flombaum et al., 2004), an object passing behind an occluder is perceived as the same object when it reappears despite changes in surface features (e.g., color), as long as it reappears in the time and place stipulated by a spatiotemporally continuous trajectory.

In addition to individuating and tracking objects, motion is used by the visual system to decompose objects into parts. In biological motion, for example, the motion of body parts are nested in the motion of the body. Object motion may be hierarchically organized into multiple layers: an arm's motion may be further decomposed into jointed segments, including the hand, which can itself be decomposed into fingers, and so on (Johansson, 1973).

The hierarchical organization of motion presents a formidable challenge to current models of motion processing. It is widely accepted that the visual system balances motion integration over space and time (necessary for solving the aperture problem) and motion segmentation in order to perceive multiple objects simultaneously (Braddick, 1993). However, it is unclear how simple segmentation mechanisms can be used to build a hierarchically structured representation of a moving scene. Segmentation lacks a notion of *nesting*: when an object moves, its parts should move with it. To understand nesting, it is crucial to represent the underlying dependencies between objects and their parts.

The experimental and theoretical foundations of hierarchical motion perception were laid by the pioneering work of Johansson (1950), who demonstrated that surprisingly complex percepts could arise from simple dot motions. Johansson proposed that the visual system performs a "vector analysis" of moving scenes into common and relative motions between objects (see also, Shum and Wolford (1983)). In the example of biological motion (Johansson, 1973), the global motion of the body is subtracted from the image, revealing the relative motions of body parts; these parts are further decomposed by the same subtraction operation.

While the vector analysis theory provides a compelling explanation of numerous motion phenomena (we describe several below), it is incomplete from a computational point of view, since it relies on the theorist to provide the underlying motion components and their organization; it lacks a mechanism for *discovering* a hierarchical decomposition from sensory data. This is especially important in complex scenes where many different vector analyzes are consistent with the scene. Various principles have been proposed for how the visual system resolves this ambiguity. For example, Restle (1979) proposed a "minimum principle," according to

[*] Corresponding author.
E-mail addresses: sjgershm@mit.edu (S.J. Gershman), jbt@mit.edu (J.B. Tenenbaum), fjaekel@uos.de (F. Jäkel).

http://dx.doi.org/10.1016/j.visres.2015.03.004
0042-6989/© 2015 Elsevier Ltd. All rights reserved.

which simpler motion interpretations (i.e., those with a shorter description length) are preferred over more complex ones (see also, Attneave (1954) and Hochberg and McAlister (1953)). While such description length approaches are formally related to the Bayesian approach described below, Restle only developed his model to explain a small set of parametrized motions under noiseless conditions. Gogel (1974) argued for an "adjacency principle," according to which the motion interpretation is determined by relative motion cues between nearby points. The "belongingness principle" (DiVita & Rock, 1997) holds that relative motion is determined by the perceived coplanarity of objects and their potential reference frames. However, there is still no unified computational theory that can encompass all these ideas.

In this paper, we recast Johansson's vector analysis theory in terms of a Bayesian model of motion perception—*Bayesian vector analysis*. The model discovers the hierarchical structure of a moving scene, resolving the ambiguity of multiple vector analyses using a set of probabilistic constraints. We show that this model can account qualitatively for several classic phenomena in the motion perception literature that are challenging for existing models. We then report a new experiment to demonstrate that the model can also provide a good quantitative fit to human data.

2. Bayesian vector analysis

In this section, we describe our computational model formally.[1] We start by describing a probabilistic generative model of motion—a set of assumptions about the environment that we impute to the observer. The generative model can be thought of as stochastic "recipe" for generating moving images, consisting of two parts: a probability distribution over trees, and a probability distribution over data (image sequences) given a particular tree. We then describe how Bayesian inference can be used to invert this generative model and recover the underlying hierarchical structure from observations of moving images. Specifically, the goal of inference is to find the motion tree with highest posterior probability. According to Bayes' rule, the posterior $P(\text{tree}|\text{data})$ is proportional to the product of the likelihood $P(\text{data}|\text{tree})$ and the prior $P(\text{tree})$. The likelihood encodes the fit between the data and a hypothetical tree, while the prior encodes the "goodness" (in Gestalt terms) of the tree.

2.1. Generative model

The generative model describes the process by which a sequence of two-dimensional visual element positions $\{\mathbf{s}_n(t)\}_{n=1}^N$ is generated, where $\mathbf{s}_n(t) = [s_n^x(t), s_n^y(t)]$ encodes the x and y position of element n at time step t. Most experimental demonstrations of vector analysis have used moving dot displays. A good example are point-light walkers. For these demonstrations each moving dot is naturally represented by its 2-d position on the screen at each time point. This representation, of course, assumes that basic perceptual preprocessing has taken place and the correspondence problem has been solved. Although we will only model moving dot displays in this paper, and hence $\mathbf{s}_n(t)$ is usually the position of the nth dot at time t, $\mathbf{s}_n(t)$ could also be the position of an object, a visual part, or a feature. In the following, we will simply refer to the elements whose movement we want to analyze as either dots or objects.

The object positions are modeled as arising from a tree-structured configuration of motion components; we refer to this representation as the *motion tree*. Each motion component is a transformation that maps the current object position to a new position. An illustration of a motion tree is shown in Fig. 1. Each node in the tree corresponds to a motion component. The motion of the train relative to the background is represented by the top-level node. The motions of Spiderman and Dr. Octopus relative to the train are represented at the second-level nodes. Finally, the motions of each body part relative to the body are represented at the third-level nodes. The observed motion of Spiderman's hand can then be modeled as the superposition of the motions along the path that runs from the top node to the hand-specific node. The aim for our model is to get as inputs the retinal motion of pre-segmented objects—in this example, the motion of hands, feet, torsos, windows, etc.—and output a hierarchical grouping that reflects the composition of the moving scene.

The motion tree can capture the underlying motion structure of many real-world scenes, but inferring which motion tree generated a particular scene is challenging because different trees may be consistent with the same scene. To address this problem, we need to introduce a prior distribution over motion trees that expresses our inductive biases about what kinds of trees are likely to occur in the world. This prior should be flexible enough to accommodate many different structures while also preferring simpler structures (i.e., parsimonious explanations of the sensory data). These desiderata are satisfied by a nonparametric distribution over trees known as the *nested Chinese restaurant process* (nCRP; Blei, Griffiths, & Jordan, 2010). The nCRP is a generalization of the *Chinese restaurant process* (Aldous, 1985; Pitman, 2002), a distribution over partitions of objects. A tree can be understood as a nested partition of objects, where each layer of the tree defines a partition of objects, and thus a distribution over trees can be constructed by recursively sampling from a distribution over partitions. This is the logic underlying the nCRP construction.

The nCRP generates a motion tree by drawing, for each object n, a sequence of motion components, denoted by $\mathbf{c}_n = [c_{n1}, \ldots, c_{nD}]$, where D is the maximal tree depth.[2] The probability of assigning object n to component j at depth d is proportional to the number of previous objects assigned to component j (M_j). This induces a simplicity bias, whereby most objects tend to be assigned to a small number of motion components. With probability proportional to γ, an object can always be assigned to a new (previously unused) motion component. Thus, the model has "infinite capacity" in the sense that it can generate arbitrarily complex motion structures, but will probabilistically favor simpler structures. Mathematically, we can write the component assignment process as:

$$P(c_{nd} = j|\mathbf{c}_{1:n-1}) = \begin{cases} \frac{M_j}{n-1+\gamma} & \text{if } j \leqslant J \\ \frac{\gamma}{n-1+\gamma} & \text{if } j = J+1 \end{cases} \tag{1}$$

where J is the number of components currently in use (i.e., those for which $M_j > 0$). Importantly, the assignment at depth d is restricted to a unique set of components specific to the component assigned at depth $d-1$. In this way, the components form a tree structure, and $\mathbf{c}_n$ is a path through the tree. The parameter $\gamma \geqslant 0$ controls the branching factor of the motion tree. As γ decreases, different objects will tend to share the same motion components. Thus, the nCRP exhibits a preference for trees that use a small number of motion components.

Fig. 2 (top panel) shows how a tree is generated by successively adding objects. Starting from the left, a single object follows a path (indicated by orange shading) through 3 layers of the motion tree. Note that the initial object always follows a chain since no other branches have yet been created. The second object creates a new

[1] Matlab code implementing the model is available at https://github.com/sjgershm/hierarchical_motion.

[2] As described in Blei et al. (2010), trees drawn from the nCRP can be infinitely deep, but we impose a maximal depth for simplicity.