ELSEVIER

Artificial Intelligence 112 (1999) 147–179

Artificial Intelligence

www.elsevier.com/locate/artint

Understanding mechanical motion: From images to behaviors

Tzachi Dar [a,1], Leo Joskowicz [b,*], Ehud Rivlin [c,2]

[a] *Faculty of Electrical Engineering, Technion, Israel Institute of Technology, Haifa 32000, Israel*
[b] *Institute of Computer Science, The Hebrew University, Jerusalem 91904, Israel*
[c] *Computer Science Department, Technion, Israel Institute of Technology, Haifa 32000, Israel*

Received 2 June 1998; received in revised form 5 March 1999

Abstract

We present an algorithm for producing behavior descriptions of planar fixed axes mechanical motions from image sequences using a formal behavior language. The language, which covers the most important class of mechanical motions, symbolically captures the qualitative aspects of objects that translate and rotate along axes that are fixed in space. The algorithm exploits the structure of these motions to robustly recover the objects behaviors. It starts by identifying the independently moving objects, their motion parameters, and their variation with respect to time using normal optical flow analysis, iterative motion segmentation, and motion parameter estimation. It then produces a formal description of their behavior by identifying individual uniform motion events and simultaneous motion changes, and parsing them with a motion grammar. We demonstrate the algorithm on three sets of image sequences: mechanisms, everyday situations, and a robot manipulation scenario.

Keywords: Image sequence analysis; Function-based analysis; Qualitative reasoning; Mechanical motion; Mechanical devices

1. Introduction

Understanding how actions are originated, constrained, and how they determine what will happen in the immediate future is an important aspect of visual classification and a major tasks in artificial intelligence. An essential prerequisite of understanding action is

* Corresponding author. Email: josko@cs.huji.ac.il.
[1] Email: tzachid@tx.technion.ac.il.
[2] Email: ehudr@cs.technion.ac.il.

0004-3702/99/$ – see front matter
PII: S0004-3702(99)00040-5

(a) (b)

Fig. 1. Two images from a video sequence of an athlete working out on an exercise machine in a gym: (a) pulling down, and (b) pushing up.

determining how things move. Given a sequence of images showing moving objects, the goal is to produce a high-level description of their behavior, which is defined by the object motions and their relationships. For example, a description of a video sequence showing an athlete working out on an exercise machine in a gym (Fig. 1) is:

> as the athlete repeatedly lowers and raises his arms, the handle translates down and up, and the weights translate up and down by the same amount.

What makes this description intuitively appealing is that it meaningfully groups together frame sequences according to uniform motion events (motions, their directions, and their duration), that it focuses on change (there is no mention of the frame or other static objects in the room), that it relates the motion of the different objects (up and down, down and up by the same amount), and that it ignores small variations in speed and acceleration. This qualitative description is invariant to different camera poses, lighting conditions, and image capture noise. Producing short, abstract, semantically meaningful qualitative descriptions of image sequences has important applications in robotics, surveillance, manufacturing, and video databases.

Going from image sequences to behavior descriptions is a difficult and computationally expensive task. First, representation issues must be addressed: how are motions and their relations described? What constitutes an adequate image sequence segmentation? Then, the algorithmic issues must be addressed: how is object motion identified? How is motion segmented? What constitutes motion change? Clearly, these issues are in great part domain and application specific. The challenge is to identify an important and useful class of behaviors, and develop robust algorithms for producing such descriptions from image sequences.

Our central premise is that producing behavior descriptions from image sequences is a parsing process whose aim is to recover the internal structure of the object motions and their relations. In structured domains, such as classical ballet, soccer, and mechanical machines, object motions tell a story which is best understood in terms of the domain's motion language. Elementary motions correspond to words, which combine

into grammatical sentences according to predefined motion rules. Sentences combine into paragraphs, describing complex behaviors. Having defined the domain's motion language, the image sequence understanding process consists of identifying specific patterns of elementary motions and their combinations.

This paper presents an algorithm for producing formal behavior descriptions of basic mechanical motions based on a simple and expressive formal language derived from first principles [21,22]. The language symbolically captures the important aspects of changing contact, coordinated, rigid, multi-body kinematics and simple dynamics of *planar fixed-axes mechanical motions*: structured motions where objects translate and rotate along axes that are fixed in space. Fixed-axes mechanical motions constitute the most important class of mechanical motions according to our survey of 2,500 mechanisms in a comprehensive encyclopedia [23]. They are very common in everyday artifacts, such as door handles and staplers, in manufacturing and assembly cells, and in mechanisms, such as locks and car transmissions. Unlike natural motions, such as human body motions, a downhill rock slide, or loosely coupled motions, such as motor vehicle traffic patterns, they are highly structured and thus more amenable to robust automated analysis. The formal behavior descriptions can be used as input to programs that automate other tasks, such explanation generation, and automatic comparison and classification of image sequences.

The algorithm starts by identifying the independently moving objects, their motion parameters, and their variation with respect to time using normal optical flow analysis, iterative motion segmentation, and motion parameter estimation. It isolates individual moving objects by finding rectangular image regions containing their motion envelopes. It then produces a description of their behavior by identifying individual uniform motion events and simultaneous motion changes, and parsing them with a motion grammar which captures the semantics of the domain. The distinguishing characteristics of the algorithm are that it performs all the steps of the image analysis and description generation processes, that it is generative and does not rely on object shapes, that it exploits the constrained structure of coupled fixed-axes mechanical motions, and that its scope is defined by a simple and expressive motion grammar derived from first principles. The algorithm has been implemented and tested on a variety of challenging examples in three categories: mechanisms, everyday situations, and a robotic cell.

The paper is organized as follows: the next section surveys related work in computer vision on extracting semantic descriptions from image sequences. Section 3 motivates planar fixed-axes mechanical motion and reviews previous work on describing mechanical motion. Section 4 presents the language grammar. Section 5 outlines the two steps of the algorithm—motions from image sequences and behaviors from motions and illustrates them with a simple example. Section 6 presents the image and rigid body motion models and describes how motions are extracted from image sequences by normal optical flow computation and iterative motion segmentation and motion parameter estimation. Section 7 describes the derivation of behaviors from motions by uniform motion events and simultaneous motion changes identification followed by parsing. Section 8 presents experimental results of our implementation on three types of examples: mechanisms, everyday situations, and robot manipulation. Section 9 concludes with extensions and future work.

2. Previous work

Extracting high-level behavior information from still images and video sequences requires a series of image processing steps, such as edge detection, optical flow computation, motion segmentation, and model matching. Both the individual steps and the entire analysis process have been the subject of continuous research over the last three decades. In this section, we only review previous work that addresses the entire analysis problem. We will review relevant individual steps in the following sections as appropriate. We first survey work on event-based recognition, which includes motion-based recognition algorithms, and then work on function-based recognition.

Event-based recognition attempts to segment image sequences by identifying events happening in a given context. One approach to simplify the task is to restrict events to a fixed set of predefined events [35,44]. Another is to restrict the shape of the objects and the context in which they appear [6,8]. It is also possible to restrict both [2,30]. For example, Nagel et al. [25,26,30] produce qualitative descriptions of road traffic scenes using both shape constraints and motion information. Their program detects and tracks moving vehicles in road traffic scenes and produces natural language descriptions of trajectory segments. Although similar in spirit to the work presented here, the domain of moving cars is very different from the domain of mechanical object motion. Moving cars exhibit loosely coupled, non-repetitive motions, while object motions in mechanisms are tightly coupled and repetitive. Semantic descriptions of car motions, such as turning left or making a "U" turn are inappropriate for describing mechanical object motions.

Siskind [35] describes ABIGAIL, a system that produces semantic descriptions of events occurring in an animated line drawing movie. ABIGAIL uses notions of support contact and attachment as the basis for the grounding language. Its drawback is that it is limited to a very small set of actions, such as place and pick up, and does not work on real images. Recently, the system was extended to handle real image sequences in a restrictive set-up in which it recognized small set of actions [36]. Yacoob and Davis [44] describe an approach for image motion estimation that uses learned models of temporal-flows to recognize actions, such as various types of walking. Activities are learned from the temporal-flow models and represented as a set of orthogonal temporal-flow bases that are learned using principal component analysis of instantaneous flow measurements. Spatial constraints on the temporal flow were developed for modeling the motion of regions in rigid and constrained motion. The method is based on a learning stage for creating a database of simple activities. Motion-based recognition algorithms identify events by extracting the objects' motion parameters from successive images. They estimate the moving object's trajectory by accumulating the information obtained from the motion analysis of sequences. This approach usually assumes that the number of objects is known, is very sensitive to noise, and does not produce high-level descriptions. For a survey of motion-based recognition algorithms, see [9].

Bruckstein et al. [6,7] use known object models to recover the object's trajectory and orientation. They show that five images are enough to recover the motion of a rigid rod or a disk in accordance with physical laws. The existence of solutions to the resulting polynomial motion equations is determined with techniques from algebraic geometry. Engel and Rubin [11], and similarly Gould and Shah [14] use motion characteristics

obtained by tracking representative points on an object to identify important events corresponding to changes in direction, speed and acceleration in the object's motion. Work has also been done on higher-level descriptions of object trajectories in terms of concepts such as stopping/starting, object interactions, and motion verbs [8,20,26]. Bobick et al. [2, 18] propose a method that restricts the context of the scenes to be analyzed: moving objects are modeled weakly and are tracked under the closed world assumption. It presupposes a space-time region of an image sequence in which the complete taxonomy of objects is known and in which all pixels can be explained as belonging to one of those objects. This approach was successfully tested in the football domain for annotating video sequences by tracking football players. although there is no attempt to segment images or produce higher-level behavior descriptions.

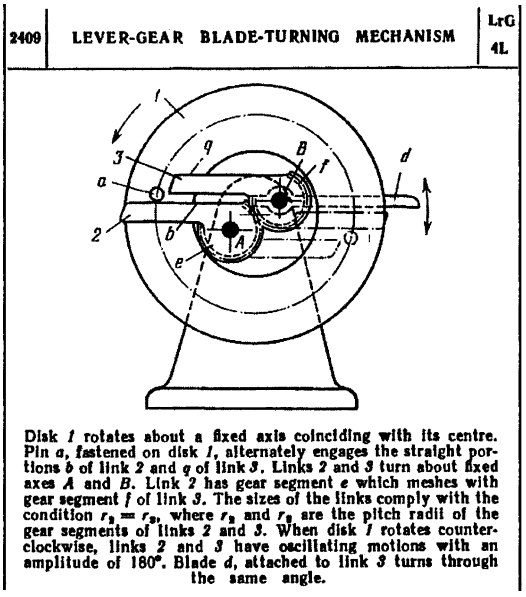
In contrast to the event-based approach, the function-based approach attempts to recognize objects and their motions based on their function. An object category is defined in terms of properties that an object must have in order to function as an instance of that category [40]. Recognizing an object functionally provides a potential behavior. Attempts to recognize objects in a single image following this approach are described in [3,4] and in [31,38,39]. Brand et al. [3,4] describe a system, called SPROCKET, that recovers the causal structure of simple machines in a single image. It incrementally builds a scene model through interleaved sensing and analysis using precompiled qualitative knowledge about rigid body interactions. It integrates diverse visual cues into an explanation of a machine's design and function. Because SPROCKET works on a single image, it cannot take advantage of motion information and must rely on object shape. Duric et al. [10] present a method to determine the function of a known tool in an image sequence from its motion while performing a task. They show that the motion of a tool, when combined with information its uses provides strong constraints on the possible function being performed. However, their flow-based analysis treated relatively short sequences. The understanding of the motion analysis phase required a database of various typical tool behaviors.

None of this work adequately addresses the problem of producing high-level behavior descriptions of mechanical motions from image sequences. In the next section, we justify our focus on fixed-axes mechanical motion, survey existing mechanical motion description languages, and present the language we propose.

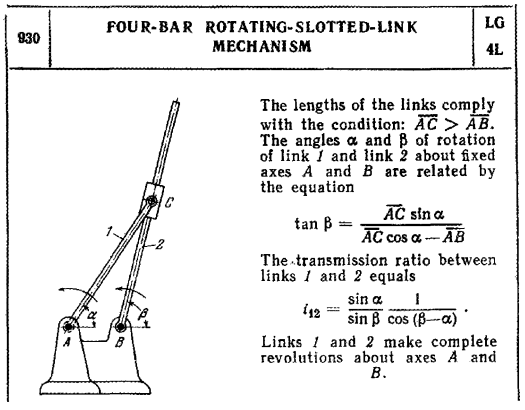
3. Classification and description of mechanical motion

We begin by addressing two key issues in producing descriptions of mechanical motions from image sequences: (1) identifying the most common types of mechanical motions and (2) determining the appropriate language in which to describe them.

Based on the mechanical engineering literature and on our own studies, we identify six classes of mechanical motions, each corresponding to a class of mechanisms: fixed axes, linkages, and general, each of which can be planar or spatial. Fixed-axes motions are rotations and/or translations along axes that are fixed in space, possibly with contact changes between objects (Fig. 2(a)). Linkage motions are coupled motions along curves produced resulting from parts permanently connected by standard joints (Fig. 2(b)).



(a)



(b)

Fig. 2. Examples of planar fixed axes and linkage mechanisms and their descriptions, as they appear in Artobolevsky's encyclopedia. (a) Fixed axes. (b) Linkage.

General motions are all motions which are neither fixed-axes nor linkage motions. Planar motions are those that can be embedded on a single plane.

Fixed-axes motions is the largest, most important, and most common class. Our survey of 2,500 mechanisms from Artobolevsky's encyclopedia [23] shows that 85% of object motions are planar, that 32% are fixed axes, that 27% are linkages, and that 34% are general motions, of which 20% are a combination of fixed axes and linkages motions. Of all pairwise motions, 89% are fixed axes motions, and 66% are planar fixed axes. Most of

the input/output behavior of general mechanisms can be described with fixed-axes motions. We also observe that fixed-axes motions are very common in operating daily artifacts, such as turning door handles, and opening lids. The practical importance of fixed-axes motion class justifies focusing on it first.

The next issue is how to describe object motions and their relationships. The motion of an assembly of rigid objects in space can be described by six configuration parameters (or their associated transformation matrix) and their first and second derivatives with respect to time for each moving object. Relations between object motions can be described implicitly, through their dependence on time, or directly, by equations relating the motion parameters. While fully general, this quantitative representation is often too complex and detailed, and fails to reveal the structure of simple motions, the relations between several moving objects, and repetitive, tightly interrelated patterns of behavior. These descriptions are also highly sensitive to noise and do not explicitly identify nearly simultaneous object motion changes. Unlike natural and unstructured motions, it is both possible and desirable to describe them at a symbolic, qualitative level of abstraction, as in the descriptions of Fig. 2.

Previous work describes languages for qualitative mechanical motion. Kota and Chiou [27] describe fixed-axes motions with qualitative motion constraint matrices and define a symbolic matrix algebra to compose them. Kannapan and Marshek [24] propose a hybrid algebraic and predicate logic language. Both languages provide a single level of abstraction and are restricted to one degree of freedom, permanent contact mechanisms, which we consider too restrictive since our survey indicates that over 20% of mechanical motions have multiple behavior modes resulting from contact changes. Linkage motions are difficult to describe qualitatively because objects move along complex paths (see description of Fig. 2(b)). Freudenstein and Maki [13] classify linkage behavior from their kinematic structure according to their degrees of freedom only, but this is too coarse for our purposes. Shrobe [34] presents a language for describing simple, one-degree of freedom planar linkage behavior based on qualitative features of the curve shapes. No symbolic language has been proposed for general motions, perhaps because those are most difficult to describe verbally.

Configuration space [28] provides a first-principles framework for developing mechanical motion description languages. All collision-free rigid object motions in an assembly can be described as paths in free pairwise configuration space regions. Regions correspond to pairwise behavioral modes, and region adjacencies correspond to transitions between modes. Joskowicz' region diagrams [21] and Faltings' place vocabularies [12] embody this idea. Configuration space based representations have been used for fixed axes qualitative simulation explanation [33,37], for mechanism concept retrieval [29], and for kinematic motion synthesis [41]. The main drawback of representations derived from configuration spaces is that they describe every contact change, which produces overly detailed descriptions in most cases.

4. A language for fixed-axes mechanical motion

We base our planar fixed-axis mechanical motion language on the fixed-axes mechanisms language described in Joskowicz and Neville [22] and on Joskowicz simplification

and abstraction operators [21]. The language is based on the configuration space representation and describes the behavior of fixed axes mechanisms by means of predicates and algebraic relations that define the configurations and motions of objects. The language is simple and comprehensive, and captures the important aspects of the kinematics and simple dynamics of mechanisms. It distinguishes between structural and behavior information, and allows both quantitative, accurate and complete descriptions, and partial, qualitative, descriptions. Because it is defined as a BNF grammar, it provides a precise, formal characterization of its scope and is amenable to computation. We chose to produce formal behavioral descriptions and not natural language-like descriptions because formal descriptions can be used to automate other tasks, such explanation generation, and comparison and classification of image sequences [21].

Fixed-axes mechanical motions are described as sequences of rotations and translations of objects. Transitions from one motion to another in the sequence reflect contact changes between objects (e.g., gears that stop meshing) or actuation changes (e.g., the driving motor reversing its rotation direction). The language, which was originally developed to specify design requirements, describe known mechanisms, and catalog them according to their behavior, was adapted for the task of mechanical motion understanding from image sequences. First, causal relations, specifying which objects drive other objects must be discarded because they cannot in general be inferred from the image sequences alone. For example, from an image sequence of two turning meshed gears, it is impossible to deduce which is the driver and which is the driven gear. Similarly, it is not possible to distinguish between objects that hold other objects or are permanently stationary. To keep spatial and temporal coherence, and track objects that have stop and go motion, we introduce the **no-motion** descriptor: it indicates that a certain object is temporarily stationary. Relative object velocity, and not absolute object configuration, better reflects object behavior and is directly obtainable for the image sequences. Finally, names of objects, axes, and motion parameters can only be selected syntactically, without reflecting the object function, which is not known a priori. We use meaningful names in our descriptions to facilitate comprehension.

Table 1 shows the new BNF grammar for describing fixed-axes mechanical behaviors. We explain it next, starting from the derivation at the top. A behavior description is a sequence of one or more motion sequences. A motion sequence is composed of sequential and simultaneous motions of single objects. Sequential motions occur one after the other in the order indicated by the sequence. Simultaneous motions occur in parallel. The single motion clause contains the motion information associated with an individual object. It consists of a unique object name, motion type, axis, motion parameter, and the interval of the motion. The motion parameter is a velocity parameter. The motion describes a continuous motion along the axis: **translation**, **rotation**, or **no-motion** for temporarily stationary objects. Repetitive motion patterns are expressed with a motion modifier. The most common are alternation and dwell: **alternate** indicates a constant change in the direction of motion, such as the motion of windshield wipers; **with-dwell** indicates a rest period in a motion with constant direction, such as stop-and-go motions; **alternate-with-dwell** indicates an alternating motion with a dwell period in between. The motion relation can be a constant (constant velocity), a qualitative value (positive or negative), or

Table 1
BNF description of the planar fixed-axis mechanical behavior language. Symbols enclosed by brackets, e.g., <MOTION> are non-terminals. Bold symbols, e.g., **translation** are terminals. Other symbols, i.e., SYMBOLIC-EXPRESSION stand for classes of terminals defined in a separate dictionary. $^+$ is an abbreviation for one or more symbols

<BEHAVIOR-DESCRIPTION>	::=	<MOTION-SEQUENCE>$^+$
<MOTION-SEQUENCE>	::=	<SINGLE-MOTION> | <SEQUENTIAL-MOTIONS> | <SIMULTANEOUS-MOTIONS>
<SEQUENTIAL-MOTIONS>	::=	<SINGLE-MOTION> **then** <MOTION-SEQUENCE>
<SIMULTANEOUS-MOTIONS>	::=	<SINGLE-MOTION> **and** <MOTION-SEQUENCE>
<SINGLE-MOTION>	::=	<OBJECT>: <MOTION-TYPE>,<AXIS>, <MOTION-PARAMETER>= <MOTION-RELATION>,<TIME-INTERVAL>
<MOTION-TYPE>	::=	<MOTION> | <MOTION> <MOTION-MODIFIER>
<MOTION>	::=	**translation** | **rotation** | **no-motion**
<MOTION-MODIFIER>	::=	**alternate** | **with-dwell** | **alternate-with-dwell**
<MOTION-RELATION>	::=	<AMOUNT> | + | − | SYMBOLIC-EXPRESSION
<TIME-INTERVAL>	::=	$t \in$ [<AMOUNT>,<AMOUNT>]
<AMOUNT>	::=	INTEGER | REAL-VALUE | CONSTANT | VARIABLE
<OBJECT>	::=	OBJECT-NAME
<AXIS>	::=	AXIS-NAME
<MOTION-PARAMETER>	::=	MOTION-PARAMETER-NAME

a symbolic expression describing the parameter variation as a function of time or relative to another parameter.

5. Algorithm

We present now an overview of our algorithm and highlight its distinguishing characteristics. The algorithm proceeds in two steps (Table 2): it first extracts object motions from the image sequence, and then constructs behavior descriptions from the resulting object motions. Object motion extraction identifies the independently moving objects, their motion parameters, and their variation with respect to time, which it describes with motion graphs. Behavior descriptions are constructed from the motion graphs by first partitioning them into sequences of individual uniform motion events, identifying simultaneous object motion changes, and then parsing the motion event sequences with the motion grammar.

In the first step, the algorithm starts by computing the normal optical flow image sequence from pairs of consecutive images. Next, it iteratively performs motion parameter estimation and motion segmentation on each image. Initially, the algorithm assumes that there is a single moving object and computes its motion parameters. The hypothesized

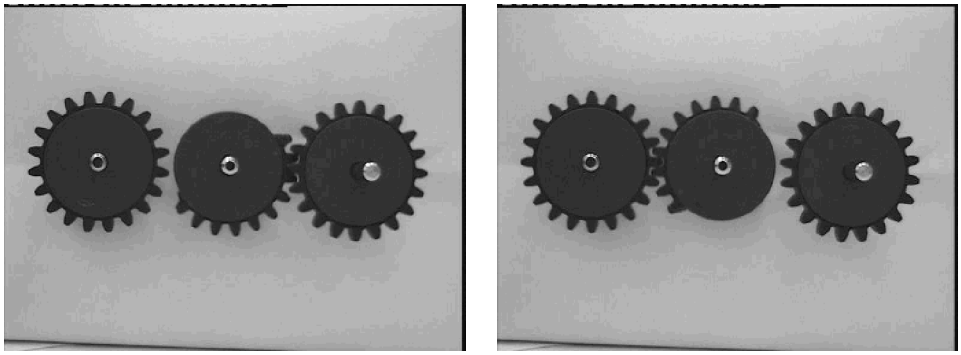
Table 2
Outline of the algorithm

I. Motions from image sequences
(i) Compute normal optical flow image sequence.
(ii) For each image region do (initially the entire image is a single region):
  (a) estimate the single motion parameters;
  (b) compute the hypothesized normal optical flow;
  (c) motion segmentation:
    – compare the actual and hypothesized normal optical flow images;
    – if they are not similar, divide the region and repeat step (ii);
  (d) construct motion graphs for each region.

II. Behaviors from motions
(i) Identify individual uniform motion events:
partition each motion graph along the parameter axis and extract the start and end time, average motion parameter values.
(ii) Identify simultaneous object motion changes: concurrently adjust motion events intervals by looking for nearby motion changes along the time axis.
(iii) Parse the resulting motion event sequences using the motion grammar.

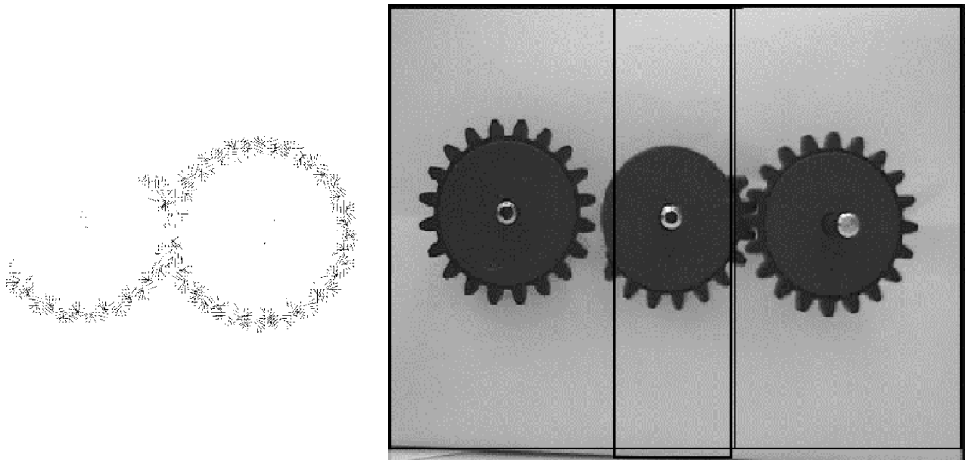
motion induces a normal optical flow on the image points, which is computed and compared with the original one. If the original normal optical flow image indeed contains a single motion, the two flows will be very similar, and the segmentation is completed. Otherwise, the algorithm divides the image into axis-aligned rectangular regions, which it hypothesizes to be single motion regions, and recursively repeats the above computation until each region is left with a single motion. The result of this computation is the identification of the independently moving objects (one per region) and its motion parameters for each normal optical flow image. The final regions contain the motion envelope of each moving object. Repeating this process over all the normal optical flow images yields the motion description of the independently moving objects, which is represented by a motion graph showing the value of each motion parameter at each frame. To speed up the computation, the algorithm uses the motion description results (regions and motion parameters) of one normal optical flow image as the initial guess for the motion segmentation on the next image.

In the second step, the algorithm partitions the motion graphs into individual uniform motion events, identifies simultaneous motion changes, and parses the resulting motion events sequence with the motion grammar to obtain the behavior description. The partitioning is performed by individually thresholding the motion graph of each object along the parameter axis to determine where uniform motion events begin and end and to obtain the average motion parameter values in that interval. Different thresholds are used for different motion parameters and different objects, e.g., some objects might be stationary while others might be rotating with no translation. Next, the algorithm identifies simultaneous events—object motion changes that occur almost at the same time—by correlating all the motion graphs and locating nearby parameter value changes on the time axis. The algorithm uses the start and end times of the motion events and a predefined time window to locate and determine simultaneous events. The result is a sequence of motion events for each moving object, which is viewed as words in a sentence of the motion



(a) (b)

Fig. 3. Two images from the video sequence showing the driver half-gear meshed with (a) the driven right gear and (b) the driven left gear.



(a) (b)

Fig. 4. Results of the iterative motion segmentation. (a) Normal flow for meshed right gear. (b) Single motion regions.

language. The behavior description, which is the structure of the sentence, is obtained by parsing the motion event sequences according to the motion language grammar.

We illustrate the algorithm on a simple example. Fig. 3 shows a mechanism consisting of a driver half-gear (center) and two driven full gears (left and right). The driver half-gear continuously rotates clockwise at constant speed, alternately meshing with the left and right gears and turning them counterclockwise proportionally for half a turn. Fig. 4(a) shows the normal optical flow image of the half-gear meshed with the right gear (there is no flow for the left gear, which is stationary). Fig. 4(b) shows the three single motion regions, which closely match the gear diameters. Fig. 5 shows a detail of the right gear

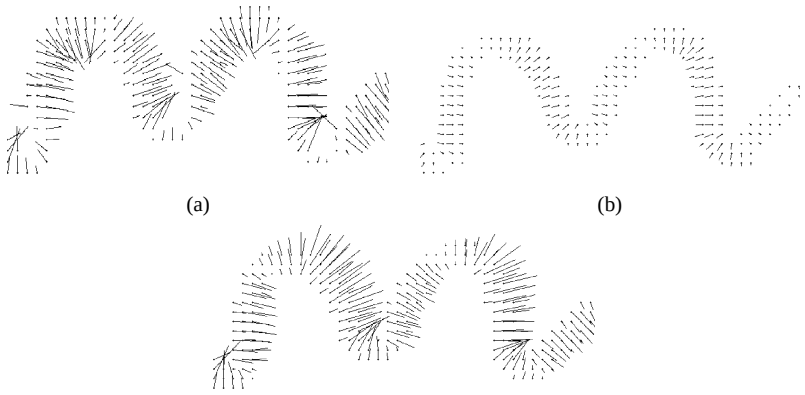


Fig. 5. Detail of the right gear flow: (a) actual flow; (b) hypothesized flow assuming a single moving object in the entire image (wrong) and (c) hypothesized flow after region division (correct). Note that flows (a) and (c) are much more similar than (a) and (b).

actual and hypothesized normal optical flows. Fig. 6 shows the motion graphs of the three gears.

In the second step, the individual motion event identification determines that the translational velocities of all three gears are negligible when compared to their angular velocities, and are thus taken to be zero throughout the sequence. The angular velocity graphs are partitioned into three segments:

(1) a no motion event followed by a positive angular velocity event for the left gear, repeated three times;
(2) a single constant negative angular velocity event for the center gear, and;
(3) a positive angular velocity event followed by a no motion event for the right gear.

Next, the algorithm determines that the beginning of the left gear rotation events coincide with the end of the right gear rotation events, and that the beginning of the right gear rotation events coincides with the end of the left gear rotation event. The vertical lines in Fig. 6 show the resulting event partition. The final parsing step produces the behavior description in Table 3.

Distinguishing characteristics

Our method has several distinguishing characteristics that contribute to research in understanding motion from image sequences. First, the algorithm performs *all* the analysis process, from low-level image processing to high-level behavior description, on a commonly occurring and formally defined class of motions. It uses a small number of predefined parameter thresholds for comparing optical flows and determining when a motion is present based on velocity ratios. The algorithm is designed to exploit the constrained structure of fixed-axes mechanical motions, defined by a simple and expressive motion grammar. Because it works directly with object motions and their changes, it does not rely on object shapes and thus does not require shape modeling, segmentation, or

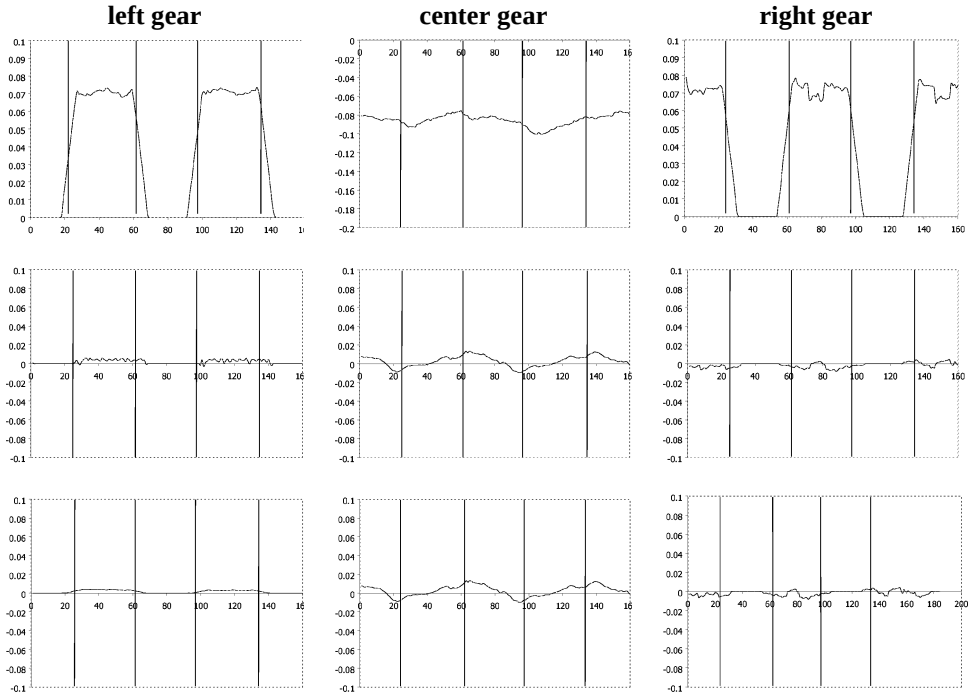


Fig. 6. Motion graphs of the left, center, and right gears for the angular velocities c (top), and horizontal translational velocity u (middle), and vertical translational velocities v. The horizontal axis indicates the frame number (time), and the vertical axis the parameter magnitude. The vertical lines indicate simultaneous events partition.

Table 3
Behavior description of the gear mechanism. O_c, O_r, O_l and cc, cr, cl are the axes names and angular velocity parameters of the center, right, and left gears, $\alpha = 0.075$ rad/frame on average, and T_i are frame intervals from the sequence $0, 24, 61, 97, 134, 166$

S_{2i}	$t \in T_{2i}$			
	left-gear:	**no-motion**		
	center-gear:	**rotation**,	**axis** = O_c,	$cc = -\alpha$
	right-gear:	**rotation**,	**axis** = O_r,	$cr = +\alpha$
then				
S_{2i+1}	$t \in T_{2i+1}$			
	left-gear:	**rotation**,	**axis** = O_l,	$cl = +\alpha$
	center-gear:	**rotation**,	**axis** = O_c,	$cc = -\alpha$
	right-gear:	**no-motion**		

recognition. It uses an iterative scheme for motion segmentation and motion parameter estimation to identify and classify individual object motions based on velocity profiles. This scheme makes no a-priori assumption on the number of moving objects, their size, or their motion sequences. It is capable of identifying and keeping track of objects with stop and go motions, correctly identifying them as the same object. It segments the image sequence by adaptively identifying individual motion events and simultaneous motion changes, which occur frequently and convey meaningful behavioral information. Because it uses a generative approach to motion event identification and behavior description, it does not require a predefined library of motions and behaviors.

6. From images to motions

The first step of the algorithm identifies the moving objects and their motion characteristics by iterative motion parameter estimation and motion segmentation. The input is the unedited sequence of images. The output is a list of moving objects, their axis of motion, and three velocity graphs for each part describing the planar angular, horizontal and vertical velocities as a function of time.

There are two approaches for obtaining motion information from image sequences: discrete feature-based methods and differential optical flow methods. Feature-based methods [5,19,43] first find correspondences between moving points in subsequent images, then estimate the motion parameters and the scene structure from these correspondences. The methods require identifying distinguishing points for each moving object, such as points on their boundary obtained by edge detection. They also require a large number of point feature correspondences and stable tracking over the entire sequence to achieve robustness. Because the number of moving objects and their shapes is unknown, and the object can be stationary temporarily, feature-based methods are unsuitable for our problem.

Optical flow methods [10,17] recover the motion information by computing and interpreting *optical flow fields*. When an observed object moves, it induces a velocity for each projected point in the image. The set of all image point velocities creates a motion field. The motion field cannot be computed directly; it is estimated with the optical flow field, which is obtained by taking the pixel difference between pairs of successive images. In fact, as [16,42] show, only the *normal optical flow*, indicating the flow in the direction of the grey-level gradient, can be computed for each image. A dense normal optical flow field is obtained by analyzing a neighborhood of each image point. The field can be computed by filtering [15] or by global minimization of a predetermined function [16]. We choose to work directly with the normal optical flow without trying to recover first the optical flow. The advantages of flow-based methods are that they do not require feature detection, feature tracking, or maintaining feature correspondences. They are especially suited for sequences of images with small time intervals between them, for which establishing stable feature correspondences is difficult.

This section presents the details of the new iterative motion parameter estimation and motion segmentation based on normal optical flow computation and comparison. We begin by describing the imaging model and developing the equations of the motion field and the

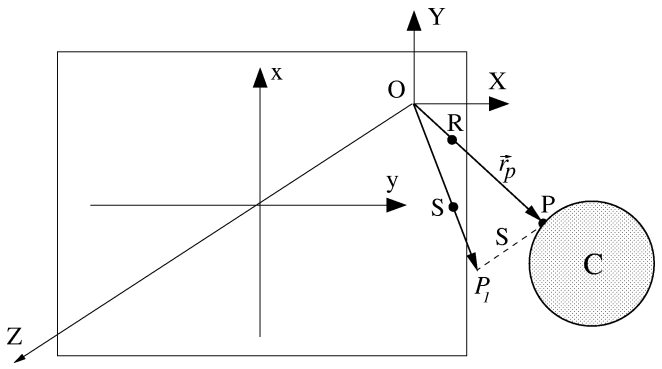


Fig. 7. The plane perspective projection image of P is $R = f(X/Z, Y/Z, 1)$; the weak perspective projection image of P is obtained through the plane perspective projection of the intermediate point $P_1 = (X, Y, Z_c)$ and is given by $S = f(X/Z_c, Y/Z_c, 1)$.

normal optical flow. We then describe planar motion parameter estimation and motion segmentation.

6.1. Imaging model

We model the relations between the moving object and the imaging system with respect to two orthonormal coordinate frames: the fixed camera (observer) coordinate system, $Oxyz$, and the moving coordinate frame, $Cx_1y_1z_1$, which is fixed to the moving object's origin.[3] The unit vectors $\vec{i}_1$, $\vec{j}_1$, and $\vec{k}_1$ are in the directions of the Cx_1, Cy_1, and Cz_1 axes. The position of the moving frame's origin C, with respect to the fixed coordinate system at any instant is given by the position $\vec{d}_c = (X_c, Y_c, Z_c)^{\mathrm{T}}$. Its orientation is defined by the nine direction cosines of the axes of the moving frame with respect to the fixed frame. For a given position $\vec{p}$ of point P in $Cx_1y_1z_1$, the position $\vec{r}_p$ of P in $Oxyz$ is determined by

$$\vec{r}_p = R\vec{p} + \vec{d}_c, \tag{1}$$

where R is the matrix of the direction cosines.

Let (X, Y, Z) denote the Cartesian coordinates of a scene point with respect to the fixed camera frame (Fig. 7), and let (x, y) denote the corresponding coordinates in the image plane. The equation of the image plane is $Z = f$, where f is the focal length of the camera. The perspective projection is given by $x = fX/Z$ and $y = fY/Z$. For weak perspective projection, an additional reference point (X_c, Y_c, Z_c) is needed. To obtain the image coordinates of a scene point (X, Y, Z), we first project it onto the point (X, Y, Z_c), and then project it onto the image point (x, y) through plane perspective projection. Usually,

[3] We use capital letters for the world coordinates, and small letters for coordinates in the image. Quantities that are in the moving (object) frame, and thus not in the fixed camera (observer) system are identified by the subscript 1.

Z_c is taken to be the average of the Z values of the object points. The projection equations are then given by

$$x = \frac{X}{Z_c} f, \qquad y = \frac{Y}{Z_c} f. \tag{2}$$

6.2. Motion field and normal optical flow

We now turn to object motion. The velocity of a point $\vec{r}_p$ on a rigid object is obtained by differentiating (1):

$$\dot{\vec{r}}_p = \vec{\omega} \times \left(\vec{r}_p - \vec{d}_c\right) + \vec{T},$$

where $\vec{\omega} = (A, B, C)^{\mathrm{T}}$ is the angular velocity of the moving frame and

$$\dot{\vec{d}}_c = (\dot{X}_c, \dot{Y}_c, \dot{Z}_c)^{\mathrm{T}} \equiv (U, V, W)^{\mathrm{T}} \equiv \vec{T}$$

is the translational velocity of the point C, which can be re-written as

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} 0 & -C & B \\ C & 0 & -A \\ -B & A & 0 \end{pmatrix} \begin{pmatrix} X - X_c \\ Y - Y_c \\ Z - Z_c \end{pmatrix} + \begin{pmatrix} U \\ V \\ W \end{pmatrix}. \tag{3}$$

To compute the instantaneous velocity of the projection of an object point into the image point (x, y) under weak perspective projection, we take derivatives of (1) with respect to time and use (3):

$$\begin{aligned} \dot{x} &= f\frac{\dot{X}Z_c - X\dot{Z}_c}{Z_c^2} = f\frac{[-C(Y - Y_c) + B(Z - Z_c) + U]Z_c - XW}{Z_c^2} \\ &= \frac{Uf - xW}{Z_c} - C(y - y_c) + f\,B\left(\frac{Z}{Z_c} - 1\right), \end{aligned} \tag{4}$$

$$\begin{aligned} \dot{y} &= f\frac{\dot{Y}Z_c - Y\dot{Z}_c}{Z_c^2} = f\frac{[C(X - X_c) - A(Z - Z_c) + V]Z_c - YW}{Z_c^2} \\ &= \frac{Vf - yW}{Z_c} + C(x - x_c) - f\,A\left(\frac{Z}{Z_c} - 1\right), \end{aligned} \tag{5}$$

where $(x_c, y_c) = (fX_c/Z_c, fY_c/Z_c)$ is the image of the point C. Let $\vec{\imath}$ and $\vec{\jmath}$ be the unit vectors in the x and y directions, respectively. The projected motion field at the point $\vec{r} = x\vec{\imath} + y\vec{\jmath}$ is $\dot{\vec{r}} = \dot{x}\vec{\imath} + \dot{y}\vec{\jmath}$. We choose a unit direction vector $\vec{n}_r$ at the image point $\vec{r}$ and call it the normal direction. We define the *normal motion field* at image point $\vec{r}$ as $\dot{\vec{r}}_n = (\dot{\vec{r}} \cdot \vec{n}_r)\vec{n}_r$, where $\vec{n}_r$ is a unit normal direction vector at $\vec{r}$. The vector $\vec{n}_r$ can be chosen in various ways. The usual choice, which we adopt, is the direction of the image intensity gradient.

Let $I(x, y, t)$ be the image intensity function. The time derivative of I can be written as

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial I}{\partial t} = \left(I_x\vec{\imath} + I_y\vec{\jmath}\right) \cdot \left(\dot{x}\vec{\imath} + \dot{y}\vec{\jmath}\right) + I_t = \nabla I \cdot \dot{\vec{r}} + I_t,$$

where ∇I is the image gradient and the subscripts denote partial derivatives. Assuming that the image intensity does not vary with time, i.e., $\mathrm{d}I/\mathrm{d}t = 0$ [16], we get $\nabla I \cdot \vec{u} + I_t = 0$.

The vector field $\vec{u}$ in this expression is called the *optical flow*. If we choose the normal direction $\vec{n}_r$ to be the image gradient direction, i.e., $\vec{n}_r \equiv \nabla I / \|\nabla I\|$, we obtain

$$\vec{u}_n = (\vec{u} \cdot \vec{n}_r)\vec{n}_r = \frac{-I_t \nabla I}{\|\nabla I\|^2}, \tag{6}$$

where $\vec{u}_n$ is the *normal optical flow* at a point.

It was shown in [42] that the magnitude of the difference between $\vec{u}_n$ and the normal motion field $\vec{r}_n$ is inversely proportional to the magnitude of the image gradient. Hence $\dot{\vec{r}}_n \approx \vec{u}_n$ when $\|\nabla I\|$ is large. Eq. (6) thus provides an approximate relationship between the three-dimensional motion and the image derivatives.

6.3. Estimating planar motion parameters from normal optical flow

For an object moving in the plane, the previous equation simplifies as follows [10]. Let $\vec{T} = (U, V, W)$ be the translation of a point in space, $\vec{T}_1 = (U_1, V_1, 0)^{\mathrm{T}}$ be its translation in the moving frame, and its rotation in the fixed frame is given by

$$\vec{\omega} = (A, B, C)^{\mathrm{T}} = C_1 R\vec{k}_1 = C_1\vec{N}, \tag{7}$$

where $\vec{N}$ is the normal to the object motion plane. Note that $\vec{\omega}_1 = C_1\vec{k}_1$, and that $\vec{T}$ and $\vec{T}_1$ are related by a rotation matrix.

We now consider the term $(Z - Z_c)/Z_c$ for the points on the moving object. The equation of the plane orthogonal to $\vec{N} = R\vec{k}_1$ which contains the point (X_c, Y_c, Z_c) in the $Oxyz$ frame coordinates is:

$$(X - X_c)N_x + (Y - Y_c)N_y + (Z - Z_c)N_z = 0.$$

Multiplying by $f(Z_c N_z)^{-1}$ we obtain

$$f\frac{Z - Z_c}{Z_c} = -(x - x_c)N_x/N_z - (y - y_c)N_y/N_z. \tag{8}$$

From (5), and (8) we obtain the equations of projected motion for points on the moving object under weak perspective:

$$\dot{x} = \frac{Uf - xW}{Z_c} - C_1(y - y_c)N_z - C_1\big[(x - x_c)N_x N_y/N_z + (y - y_c)N_y^2/N_z\big], \tag{9}$$

$$\dot{y} = \frac{Vf - yW}{Z_c} + C_1(x - x_c)N_z + C_1\big[(x - x_c)N_x^2/N_z + (y - y_c)N_x N_y/N_z\big]. \tag{10}$$

These two equations relate the projected image motion field and (x_c, y_c) to the scaled translational velocity $Z_c^{-1}\vec{T} = Z_c^{-1}(U\ V\ W)^{\mathrm{T}}$, the rotational parameter C_1, and the normal to the object motion plane $\vec{N} = (N_x, N_y, N_z)^{\mathrm{T}}$.

The normal motion field for projected points that are on the moving object is

$$\begin{aligned}\vec{r} \cdot \vec{n} &= \vec{n}_x\dot{x} + \vec{n}_y\dot{y} \\ &= n_x f\big[U/Z_c + (x_c/f)\,C_1 N_x N_y/N_z\big] - n_x x\big(W/Z_c + C_1 N_x N_y/N_z\big) \\ &\quad - n_x(y - y_c)C_1\big(N_z + N_y^2/N_z\big) + n_y f\big[V/Z_c - (y_c/f)C_1 N_x N_y/N_z\big] \\ &\quad - n_y y\big(W/Z_c - C_1 N_x N_y/N_z\big) + n_y(x - x_c)C_1\big(N_z + N_x^2/N_z\big),\end{aligned} \tag{11}$$

where and $n_x\vec{i} + n_y\vec{j}$ is the normal direction and (x_c, y_c) is the position of the object reference point C, which is chosen to be the object center of mass computed by averaging the coordinates of all edge points for which the normal flow was computed. We can re-write this relation as

$$\vec{r} \cdot \vec{n} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{c}, \tag{12}$$

where

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} \equiv \begin{pmatrix} n_x f \\ -n_x x \\ -n_x(y - y_c) \\ n_y f \\ -n_y y \\ n_y(x - x_c) \end{pmatrix},$$

$$\boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix} \equiv \begin{pmatrix} U/Z_c + (x_c/f)\, C_1 N_x N_y/N_z \\ W/Z_c + C_1 N_x N_y/N_z \\ C_1(N_z + N_y^2/N_z) \\ V/Z_c - (y_c/f) C_1 N_x N_y/N_z \\ W/Z_c - C_1 N_x N_y/N_z \\ C_1(N_z + N_x^2/N_z) \end{pmatrix}. \tag{13}$$

To separate between directly observable quantities (vector $\boldsymbol{a}$) and quantities that need to be estimated (vector $\boldsymbol{c}$).

We compute an estimate for $\boldsymbol{c}$ as follows. A good approximation for the normal motion field is the normal flow of the image points (x_i, y_i), $i = 1, \ldots, m$, at which the magnitude of the image gradient $\|\nabla I(x_i, y_i, t)\|$ is large. We replace the left hand side of (12) by the normal flow $-I_t/\|\nabla I\|$. As a result, we obtain the system of equations $A\boldsymbol{c} \approx \boldsymbol{b}$, where $\boldsymbol{c}$ is the vector with six unknowns, A is a $m \times 6$ matrix whose rows are the vectors $\boldsymbol{a}_i$, and $\boldsymbol{b}$ is an m-vector whose elements are

$$-\big(\partial I(x_i, y_i, t)/\partial t\big)/\big\|\nabla I(x_i, y_i, t)\big\|.$$

The solution to this linear system of equations formulated over six image points yields the desired result.

When more than six points are used ($m > 6$) the system becomes over-constrained, so we look for a solution that minimizes the norm of the difference $\|\boldsymbol{b} - A\boldsymbol{c}\|$. The solution to this system is identical to the solution of the system

$$A^{\mathrm{T}}A\boldsymbol{c} = A^{\mathrm{T}}\boldsymbol{b} \equiv \boldsymbol{d}.$$

We solve the system

$$A^{\mathrm{T}}A\boldsymbol{c} = \boldsymbol{d}$$

using the Cholesky decomposition. Because $A^{\mathrm{T}}A$ is a positive definite 6×6 matrix, we compute the lower triangular matrix L such that $L\,L^{\mathrm{T}} = A^{\mathrm{T}}A$. Substituting, we get $L\,L^{\mathrm{T}}\boldsymbol{c} = \boldsymbol{d}$, for which we solve two triangular systems $L\boldsymbol{e} = \boldsymbol{d}$ and $L^{\mathrm{T}}\boldsymbol{c} = \boldsymbol{e}$ to obtain the parameter vector $\boldsymbol{c}$.

Having estimated $\boldsymbol{c}$, we obtain an estimate for the planar motion parameters $\vec{T}/Z_c$ and C_1 by applying (13) to obtain:

$$\begin{aligned} \frac{U}{Z_c} &= c_1 - \frac{x_c c_7}{f}, \qquad & \frac{V}{Z_c} &= c_4 + \frac{x_c c_7}{f}, \\ \frac{W}{Z_c} &= \frac{c_2 + c_5}{2}, \qquad & C_1 &= \operatorname{sign}(c_6)\sqrt{c_3 c_6 - c_7^2}, \end{aligned} \tag{14}$$

where $c_7 = (c_2 - c_5)/2$.

Under the assumption of planar motion viewed at a fronto-parallel plane slanted by at most 30°, the translation parameters in the moving object's coordinate system U_1/Z_c and V_1/Z_c can be estimated from the translation in the camera's (observer) coordinate system $(U/Z_c, V/Z_c, W/Z_c)$ [10]. This assumption is justified because a nearly frontal viewpoint is the best way to show a planar mechanism. We apply this procedure here to complete the recovery of the three motion parameters of the moving object.

6.4. Motion segmentation

The motion parameter estimation procedure that we just described assumes a single planar motion over the full image frame and over the entire image sequence. Clearly, this is not a valid assumption for the situations we want to handle. It is, however, an effective procedure for the image regions and image sub-sequences within which there is indeed a single object moving. If we can automatically identify single motion regions in the image sequence, we can directly determine the number of moving objects (one per region), their locations and motion axes, and their planar translational and angular velocities at each frame. The single motion regions is an estimate of the motion envelope of the object: the set of points occupied by the object as it moves. Three motion graphs per object are obtained by plotting the value of each velocity parameter as a function of time, measured at frame intervals.

To automatically identify single motion regions without any a priori knowledge of the number of moving objects and their shapes, we propose the following top-down iterative method (unlike [1,32], which use a bottom-up approach). Initially, the algorithm hypothesizes that there is a single region covering the entire image with a single moving object. It computes its motion parameters under this assumption as described in Section 6.3. The hypothesized motion induces a normal optical flow on the image points, which is computed using the equations in Section 6.2. Each point in the image region has now associated with it two normal optical flows: the actual one, which was computed from the original sequence of images, and the hypothesized one. If the image sequence indeed contains a single rigid motion in the region, most of the normal optical flow values at the image points should be very similar (note that they will not be all exactly identical because the computation model is approximate, e.g., $\dot{\vec{r}}_n \approx \vec{u}_n$ when $\|\nabla I\|$ is large). When the two optical flows are very similar, the hypothesis is valid and the segmentation is completed. Otherwise, the algorithm divides the image into axis-aligned rectangular regions, which it hypothesizes to be single motion regions, and recursively repeats the above computation until each region is left with a single motion.

Computing the hypothesized normal optical flow

We derive the hypothesized normal optical flow at each point from Eqs. (4) and (5). Since we assume planar motion, the first two components of the angular velocity vector $\vec{\omega} = (A, B, C)^{\mathrm{T}}$, A and B are zero. Substituting in these equations and rearranging, we obtain the hypothesized translational velocities U'_c and V'_c of an image point (x, y):

$$\begin{aligned} U'_c &= fU/Z_c - xW/Z_c - C_1(y - y_c), \\ V'_c &= fV/Z_c - yW/Z_c + C_1(x - x_c). \end{aligned} \tag{15}$$

Since we can calculate U/Z_c and V/Z_c and C_1 from Eq. (14), and x_c and y_c are known, we can compute U'_c and V'_c. To obtain the hypothesized normal flow, we project this flow on the normal direction.

Comparing normal optical flows

To determine if the actual and the hypothesized normal optical flows are close, we establish a similarity measure. The measure is the ratio between the image points for which the relative difference between the actual and hypothesized normal optical flows is smaller than a predetermined threshold and the total number of points

$$\mathit{similarity} = \frac{1}{n}\left(\sum_{i=1}^{n}\left(\left|\frac{(\mathit{actual}(\vec{u}_n, i) - \mathit{hypothesized}(\vec{u}_n, i))}{\max_i(\mathit{actual}(\vec{u}_n, i))}\right| < \mathit{threshold}\right)\right), \tag{16}$$

where $\mathit{actual}(\vec{u}_n, i)$ and $\mathit{hypothesized}(\vec{u}_n, i)$ are the actual and hypothesized normal flow vector $\vec{u}_n$ magnitudes at pixel i. The similarity measure tends to zero when the flows are very different (many moving objects, for which the flow appears as non-rigid motion of a single object) and to one when they are very similar (a single moving object). For example, for the flows in Fig. 5, the similarity measure between (a) and (b) is 0.32, and between (a) and (c) is 0.70 for a threshold of 5.1. Alternative robust statistical estimate methods based on histogramming can also be used.

Region division

When the actual and hypothesized flow differ, the region contains more than one moving object. It must be divided into two or more regions, each containing a single moving object. In general, motion regions can have various shapes, corresponding to the motion envelopes of the moving objects. We approximate the motion envelopes with nearly non-overlapping axis-aligned rectangles containing them. Initially, the entire image rectangle is a single region. The region is subdivided by expanding small rectangles from each one of its four corners. The rectangles are expanded until the similarity measure decreases significantly. When it does, the region is marked as a possible solution, and the algorithm proceeds to search for more single motion regions from the four neighboring sides or the rectangle. The rectangle with the best similarity measure from the four candidates is selected. The algorithm continues the subdivision process recursively until each region is a single object motion region.

This region division strategy assumes that there is little overlap between the object motion envelopes, which is the case in most situations (it is true in all the examples of this paper). The strategy can be modified to handle overlapping cases at the expense of

more computation by, for example, growing regions that are only a few pixels in size, and splitting and merging them as appropriate.

Image sequence motion segmentation

The motion segmentation starts by performing motion segmentation on the first pair of images in the sequence as described above. The resulting single motion regions are then used as initial guesses for the successive pairs of images. In each region, motion parameters are estimated, and the hypothesized normal optical flow is computed. The regions are then updated as follows. If the actual and hypothesized flows are similar, the region remains as is. If the hypothesized flow is very weak (values are below a given threshold), the object in the region is not moving, and the region remains as is. If the actual and monitored flows are dissimilar, a new object has started moving or has entered the region. The region is then split and adjusted as described above. This procedure guarantees the spatial and temporal coherence: objects that have stopped moving will be recognized as the same object when they start moving again.

7. From motions to behaviors

To produce the formal sentences that describe the behaviors of the mechanism, we developed a three-step algorithm. First, the algorithm partitions the motion graphs into individual uniform motion events. Next, it identifies simultaneous motion changes, and then parses the resulting motion events sequence with a motion grammar to obtain the behavior description. For each behavior of each part of the mechanism, it produces a sentence that describes it. When one object changes its motion, a new sentence is written to reflect the new behavior. This process terminates when there is a sentence for each moving object behavior.

7.1. Individual motion events identification

Uniform motion events for each moving object are identified by individually partitioning each motion graph. The partitioning is performed by individually thresholding the motion graph of each object along the parameter axis to determine where uniform motion events begin and end, and to obtain the average motion parameter values in that interval. Different thresholds are used for different motion parameters and different objects: objects have different types of motion and are moving with different very different velocities. The threshold is set to be sensitive to the processed graph, e.g., by some relative portion of the average parameter values. Note that the precise locations on the time axis where the partition takes place are dependent on a predefined threshold.

7.2. Simultaneous motion changes identification

The next step is to identify the relationship between events, relating them on the time axis. To identify simultaneous events, object motion changes that occur almost at the same time, the algorithm correlates all the motion graphs and locates nearby parameter value

changes on the time axis. The algorithm uses the start and end times of the motion events and a predefined time window to locate and determine simultaneous events. Events that fall within the same time window are considered to happen together, and get the average value of the time window as a time stamp. Note that the size of the time window is scenario dependent. For example, when analyzing a series of fast events, we would expect the time window to contain fewer frames. The result is a sequence of motion events for each moving part. These are the building blocks for the next step, which is to take these events/words and combine them into a sentence of a motion language.

7.3. Behavior parsing

The final step is to group the motion events into higher level structures, as specified by the motion grammar. The symbolic form of the individual events of each moving object are obtained directly from the motion graph partition, which determines the sequence of simultaneous events. This straightforward translation yields terminals describing the behavior of each part (rotation, translation no-motion, etc.), their axes, and motion relations. The result is a string of primitive events, which constitute the words of the behavior sentences. The structure of the sentences is recovered by parsing the sentences with a standard bottom up technique. Note that further processing can turn this formal output into natural language sentences (we have not investigated this further).

8. Experimental results

We have implemented the algorithm and tested it on image sequences of examples and scenarios of varying complexity from three categories: mechanisms, everyday life situations, and a robotic cell. We briefly describe the experimental setup and results next.

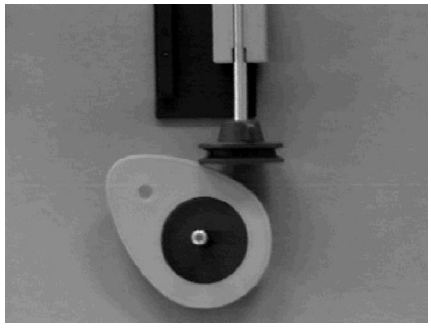
The video sequences consist of 768×576 8-bit gray-level pixel images (440 kB) shot at a rate of 25 images per second. Sequences last from 5 to 60 seconds. The camera image plane is at an angle of at most 30° from the motion plane in all sequences. The camera position and lighting are constant throughout the sequence. Full-resolution images are used in the computation, with no temporal or spatial sub-sampling. Most of the computation time is spent on normal optical flow computation and image file handling. Reading and writing the normal optical flow images to disk takes up to 70% of the computation time. For a 200 image sequence, this process takes about one hour on a SGI IP25 workstation. The motion segmentation and motion parameter estimation take about a minute for the first pair of images. The program uses the regions as estimates for the subsequent images, which is a correct guess for most of the images, as discussed in Section 6.4. Processing the motion graph takes less than 30 seconds in all cases.

The results depend on the values of various thresholds and parameter values. For the first step, the hypothesized and actual normal optical flows are considered similar in a region if the similarity measure is higher than 0.5. Usually, the ratio is 0.7 when there is a single object, and 0.3 or less when there are two or more (Section 6.4). The motion segmentation uses nearly non-overlapping axis-aligned rectangles. The rectangle sizes are

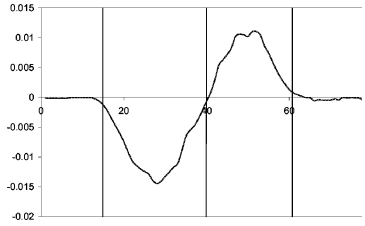
grown in multiples of 20 pixels, e.g., 20 × 60, 40 × 80 and so on. For the second step, the velocity thresholds were chosen individually for each part and for each type of motion. The velocity ratios between different objects were 5:1 or less, which allowed the program to successfully filter out from the velocity graphs the noise and slight velocity variations due to inertia, impacts, and motor fluctuations. We chose to distinguish between three velocity values: positive, negative, and zero. Zero was taken to be values below 10% of the maximum value. Events were considered to be simultaneous if they occur within a half second (12 frame) interval. For the behavior description, we chose qualitative descriptions indicating the velocity sign, not its actual value, and ignored acceleration.

8.1. Mechanisms

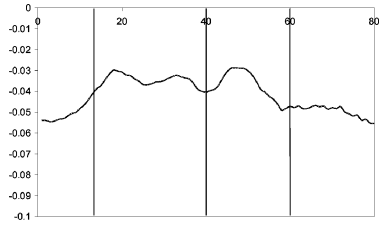
We built, videotaped, and successfully analyzed a dozen fixed-axes mechanisms representing the most common mechanical elements: gear, cam, and belt drives. They were all powered by a small, inexpensive electric motor. The gear drive example in Fig. 3 was discussed in Section 5. Figs. 8–10 show three additional examples: a reciprocating cam piston, a crank slider, and a belt drive from two different views. In the cam piston example, the cam continuously rotates clockwise, driving the piston up and down. In the crank slider mechanism, the slider translates horizontally back and forth as the wheel rotates clockwise



(a)



(b)



(c)

Fig. 8. Cam piston mechanism and its segmented motion graphs. (a) Cam piston. (b) Cam velocity. (c) Piston velocity.

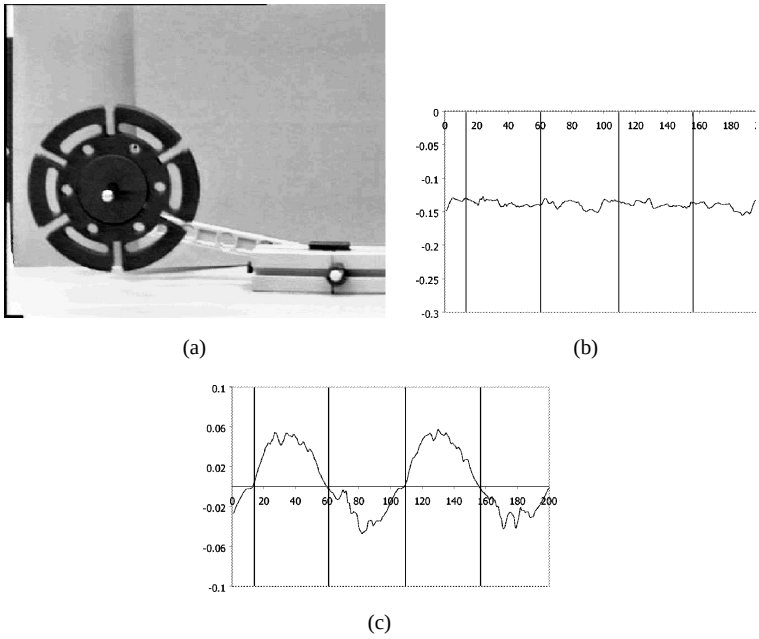


Fig. 9. Crank slider mechanism and its segmented motion graphs. (a) Crank slider. (b) Wheel velocity. (c) Slider velocity.

according to a sine relationship. In both cases, the program identifies the periodicity and includes the modifier *alternate* in the description:

$t \in [i, i + 95]$ for $i = 0, 1, 2$
crank: rotation, c-axis, $c = +$
slider: translation alternate, u-axis, $u = +$

The image sequence includes three cycles that repeat every 95 frames. Each cycle consists of two slider translation segments of equal length, starting with translation in the positive direction. The crank rotates continuously in the positive direction. Note that the intermediate link motion, which is not fixed axes, is ignored. This shows that the program is capable of correctly producing an input/output behavior when not all part motions are fixed axes.

In the belt drive mechanism, the right wheel uniformly rotates at a constant speed clockwise, causing the left wheel to rotate more slowly. The motion of the chain, which is not a rigid body object is correctly ignored. Note that, after thresholding, the belt drive graphs show that the velocities are constants and that the velocity ratios are very similar.

8.2. Everyday situations

Another interesting category are human actuated mechanisms from everyday life situations, such as the gym example in the introduction Fig. 1 and the door handle in

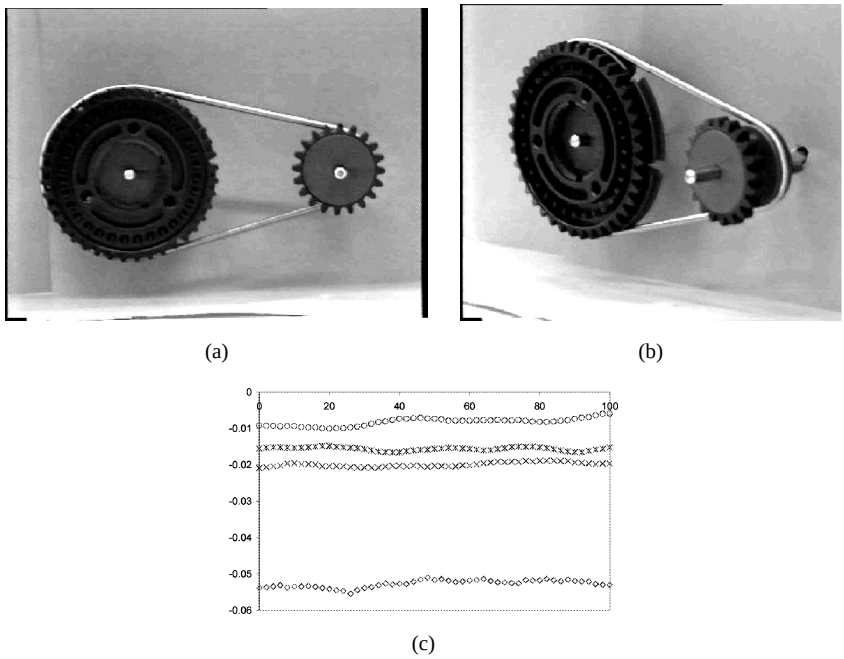


Fig. 10. Belt drive mechanism from two viewpoints and their motion graphs. (a) 0° view. (b) 30° view. (c) Motion graphs. In (c) the top two plots are the velocities of the large and small wheels at 30°, and the bottom two plots are the velocities of the large and small wheels at 0°.

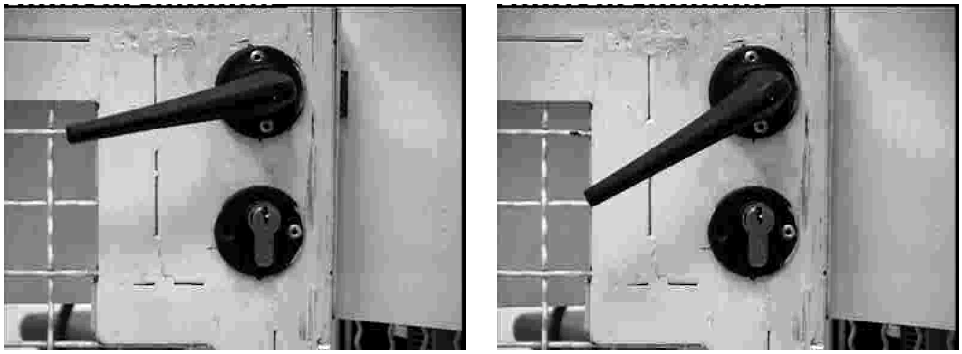
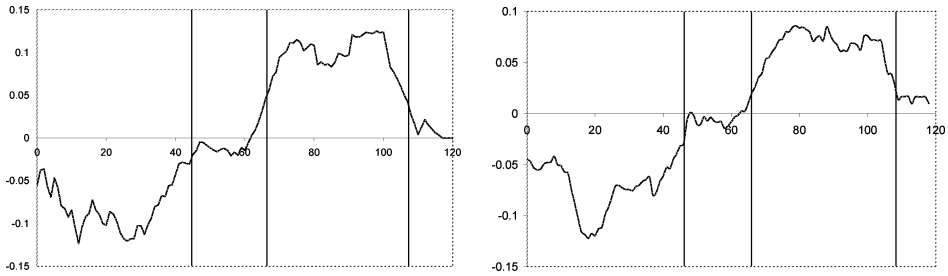


Fig. 11. A door handle sequence. The handle was turned from behind the door. (a) Closed door. (b) Open door.

Fig. 11. Fig. 12 shows the corresponding motion graphs. In these cases, there is more variation in the motion, and thus more noise. Nevertheless, the program correctly identified individual continuous motion events.



(a) (b)

Fig. 12. Door latch graphs. (a) Latch horizontal translation. (b) Handle angular velocity.

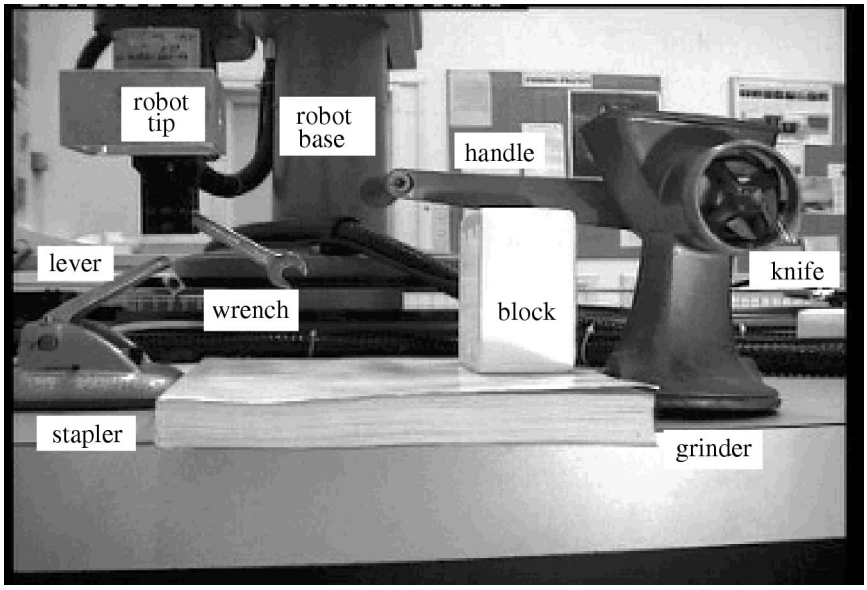


Fig. 13. A robot cell scene consisting of a handle-activated meat grinder (left), a lever-activated stapler, and a moving robot tip rigidly holding a wrench (center). The grinder handle is initially supported by a white rectangular block. The grinder handle and the stapler lever can rotate with respect to their bases. The grinder knife rotates inside the grinder housing. All the robot motions are translational.

8.3. Robot cell

The most complex scenario we analyzed is the robot cell in Fig. 13. There are four moving objects: the robot tip, the stapler lever, the grinder handle, and the grinder knife. Fig. 14 shows six frames from the image sequence. In the initial configuration (a), the wrench side is in contact with the meat grinder handle. The robot performs the following sequence of motions: first, it moves the tip up and down, raising and lowering the grinder handle and causing the grinder knife to rotate clockwise and counterclockwise (b). Then,

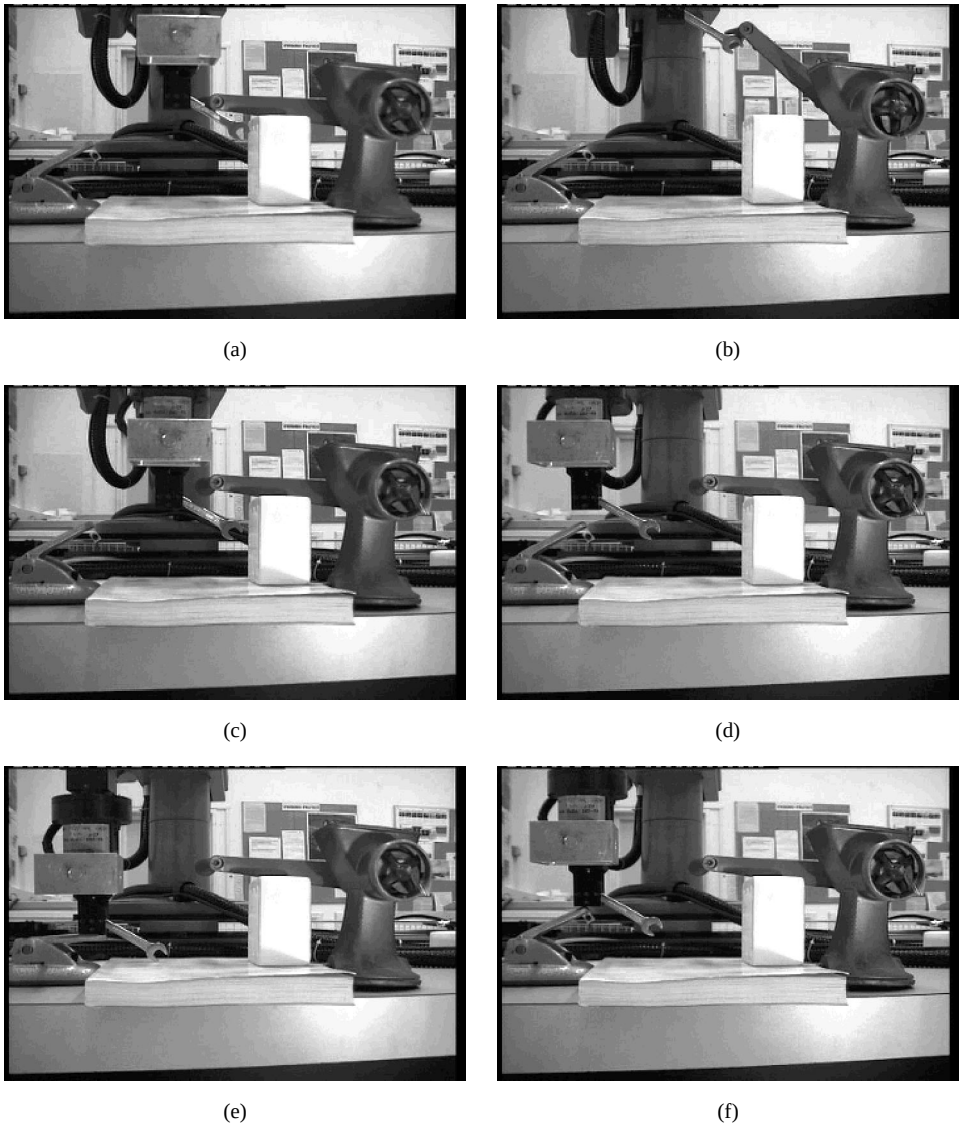


Fig. 14. Robot sequence. (a) Initial configuration. (b) Raised handle. (c) Lowered handle. (d) Tip above the lever. (e) Pushing down the lever. (f) Releasing the lever.

the robot tip keeps moving down after it leaves the handle (c), stops, and moves left until it is above the stapler lever (d). It then moves down and up, pushing (e) and releasing the lever (f). The robot tip keeps moving up after it releases the lever, stops, and moves right to its home position. The robot makes short pauses between motions. This sequence is repeated twice. The video segment last for about 60 seconds (1,500 images). We worked

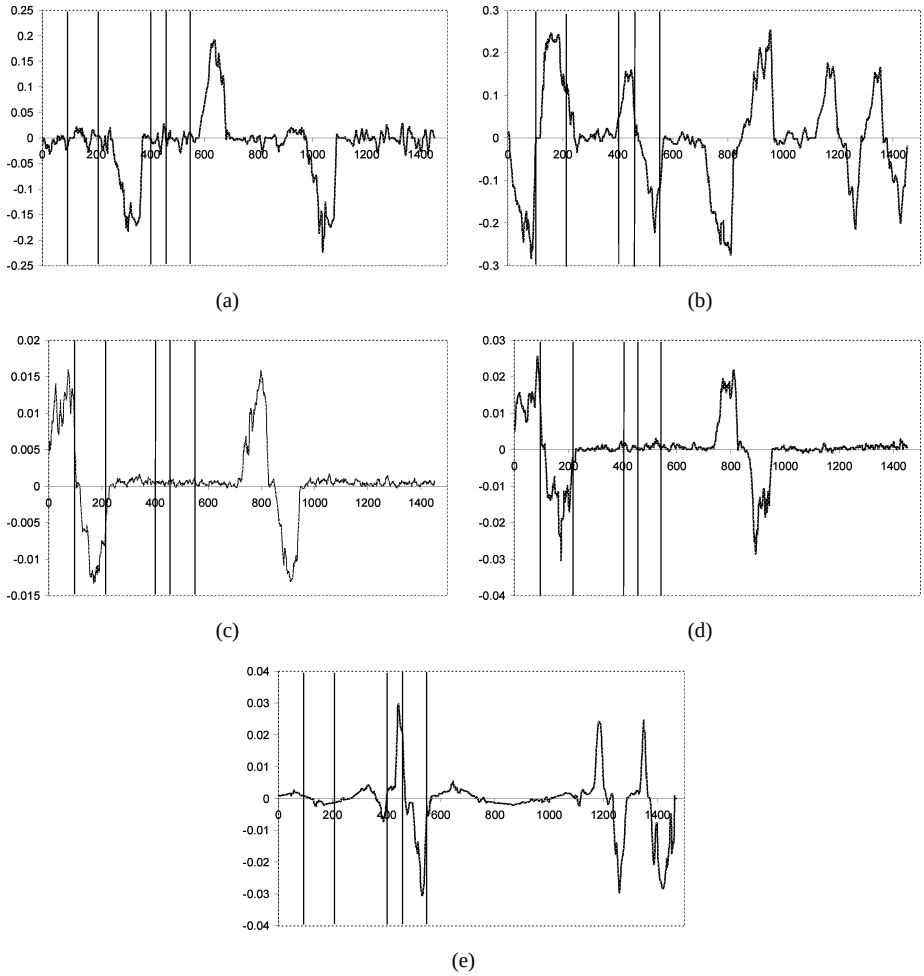


Fig. 15. Motion graphs. The horizontal axes is the frame number, the vertical axis is the magnitude of the velocity. (a) Robot tip horizontal translation. (b) Robot tip vertical translation. (c) Grinder handle rotation. (d) Grinder knife rotation. (e) Stapler lever rotation.

with an image resolution of 320×256 because of disk space limitations (no difference in the outcome was detected).

The program recognizes four moving objects and 30 uniform motion events, including the pauses between motions. Fig. 15 shows the five relevant motion graphs. The top two graphs show the horizontal and vertical translations of the robot tip. The two middle graphs show the simultaneous clockwise and counterclockwise rotations of the grinder handle and knife. The bottom graph shows the stapler lever turning clockwise and counterclockwise, followed by a rest period. The vertical lines show five significant simultaneous motion events (out of 32) corresponding to the snapshots in Fig. 14. There are short, 2 second pauses between motions. The final behavior description is shown in Table 4.

Table 4
Fragment of the symbolic behavior description of the robotic cell motions

$t \in [0, 96)$
robot-tip: translation, v-axis, $v = -$
knife: rotation, k-axis, $c = +$
handle: rotation, h-axis, $c = +$
lever: no-motion
then $t \in [96, 118)$
robot-tip, knife, handle, lever: no-motion
then
$t \in [118, 213)$
robot-tip: translation, v-axis, $v = +$
knife: rotation, k-axis, $c = -$
handle: rotation, h-axis, $c = -$
lever: no-motion
then
$t \in [213, 235)$
robot-tip: translation, v-axis, $v = +$
knife, handle, lever: no-motion
then
$t \in [235, 268)$
robot-tip, knife, handle, lever: no-motion
then
$t \in [268, 365)$
robot-tip: translation, u-axis, $u = -$
knife, handle, lever: no-motion
then
$t \in [365, 401)$
robot-tip, knife, handle, lever: no-motion
then
$t \in [401, 431)$
robot-tip: translation, v-axis, $v = +$
knife: no motion
handle: no-motion
lever: no-motion
then
$t \in [431, 458)$
robot-tip: translation, v-axis, $v = +$
lever: rotation, l-axis, $c = +$
knife, handle, no-motion
then
$t \in [458, 497)$
robot-tip, knife, handle, lever: no-motion
then
$t \in [497, 542)$
robot-tip: translation, v-axis, $v = -$
lever: rotation, l-axis, $c = -$
knife, handle, no-motion
then
$t \in [542, 557)$
robot-tip: translation, v-axis, $v = -$
knife, handle, lever, no-motion
then
$t \in [557, 591)$
robot-tip, knife, handle, lever: no-motion
then
$t \in [591, 674)$
robot-tip: translation, u-axis, $u = +$
knife, handle, lever: no-motion
then
$t \in [674, 718)$
robot-tip, knife, handle, lever: no-motion
then
$t \in [718, 738)$
robot-tip: translation, v-axis, $v = -$
knife: no motion
handle: no-motion
lever: no-motion
then
$t \in [738, 818)$
robot-tip: translation, v-axis, $v = -$
knife: rotation, k-axis, $c = +$
handle: rotation, h-axis, $c = +$
lever: no-motion
then
$t \in [818, 870)$
robot-tip, knife, handle, lever: no-motion
...

9. Conclusion

We have presented a new algorithm for producing behavior descriptions of planar fixed axes mechanical motions from image sequences. The algorithm uses a formal behavior language that symbolically captures the qualitative aspects of objects that translate and rotate along an axis that is fixed in space. The language covers the most important class of mechanical motions and is based on the first-principles theory of configuration spaces. The algorithm follows a multi-step process whose aim is to recover

the internal structure of the object motions and their relations based on the language. It starts by identifying the independently moving objects, their motion parameters, and their variation with respect to time using normal optical flow analysis, iterative motion segmentation, and motion parameter estimation. It isolates individual moving objects by finding rectangular image regions containing their motion envelopes. It then produces a formal description of their behavior by identifying individual uniform motion events and simultaneous motion changes, and parsing them with a motion grammar. These formal behavior descriptions can be used as input to programs that automate other tasks, such explanation generation, and automatic comparison and classification of image sequences.

The distinguishing characteristics of our method are that it performs *all* the process, from low-level image processing to high-level behavior description, that it exploits the constrained structure of fixed-axes mechanical motions, and that it provides a generative approach to motion event identification and behavior description based on a formal motion language. Because the algorithm works directly with object motions and their changes, it does not rely on object shapes and thus does not require shape modeling, segmentation, or recognition. It uses an iterative scheme for motion segmentation and motion parameter estimation to identify and classify individual object motions based on velocity profiles. This scheme makes no a-priori assumption on the number of moving objects and their size. It is capable of identifying and keeping track of objects with stop and go motions, correctly identifying them as the same object. It segments the image sequence by adaptively identifying individual motion events and simultaneous motion changes, which occur frequently and convey meaningful behavioral information. It uses a small number of predefined parameter thresholds for comparing optical flows and determining when a motion is present based on velocity ratios.

Algorithmic improvements

Various improvements are possible in each one of the algorithm steps. The most important issue is the thresholds determination for normal optical flow comparison, uniform motion event identification, and simultaneous event identification. Currently, these thresholds are predefined or established by simple averaging schemes. Robust statistical methods based on histogramming can significantly improve the stability and reliability of the parsing process, and better automate the process.

When deriving motion parameters from image sequence, the most sensitive process is motion segmentation. We plan to extend the scope of the segmentation process by allowing moving regions with various shapes, not just axis-aligned rectangles. This will yield better, more accurate region division, and will enhance the segmentation results. It will also allow us to remove the current restriction on non-overlapping motion envelopes. Another subject of current research is to develop better methods for simultaneous event detection, which will become a critical issue as the scope of motions is extended. Other research subjects include better parsing strategies, curve fitting for motion graphs for richer, more detailed motion description, and noise reduction.

Scope extensions

The first extension is to objects with general mechanical planar motions, where the angular and translational velocities are coupled. Planar motions usually occur in linkage mechanisms, as discussed in Section 3. The most difficult issue is finding an appropriate symbolic language, possibly along the lines of [34]. The first step of the algorithm remains unchanged, as it assumes planar motion. For the second step, event identification requires to extend the event identification step to detect motion events in the three velocity graphs simultaneously because general planar motions couple the angular and translational velocities in specific patterns.

The second extension is to fixed-axes spatial motions, such as meshed bevel gears with orthogonal axes. The motion language can be easily extended to deal with two new types of spatial motion: helical motion and independent translation and rotation (cylindrical joint) while all the rest remains the same. The easiest way to deal with this issue is to simultaneously record the moving objects from different camera positions so that each object moves on a plane roughly perpendicular to the camera normal. We believe that specialized motion estimation parameter techniques are best to identify this type of motion. Note that the individual motion events and simultaneous motion changes identification method described in Section 6 remains the same. Identifying general spatial motions is a hard problem, as discussed in the introduction. Identifying human motion is of particular interest [44]. In our framework, it requires first the development of a motion language for the domain (classical dance, soccer, exercising). Another avenue is to attempt to predict motion by simulating the physical laws of motion.

Applications

We envisage several applications to our algorithm. One is the production of natural language explanations. Automating this task is relatively straightforward, since the structure of the natural language sentences can closely follow the structure of the formal sentences. Another one is the segmentation of video sequences of machines for database retrieval. Mechanical behavior can be classified and retrieved according to partial formal descriptions stated in the motion language [21]. A third application is in inspection and repair of machines. We also plan to use the qualitative partitioning as a guide for more refined analysis.

Acknowledgement

Leo Joskowicz is supported in part by grant 98/536 from the Israeli Academy of Science, by a grant from the Authority for Research and Development, The Hebrew University of Jerusalem, Israel, and by a Ford University Research Grant, the Ford ADAPT2000 project.

References

[1] G. Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, IEEE Trans. Pattern Analysis and Machine Intelligence (1985) 384–401.

[2] A. Bobick, J. Davis, An appearance-based representation of action, in: Proc. IEEE International Conference on Vision and Pattern Recognition, 1996, A7E.4.
[3] M. Brand, Physics-based visual understanding, Computer Vision and Image Understanding 65 (1997) 192–205.
[4] M. Brand, L. Birnbaum, P. Cooper, Sensible scenes: Visual understanding of complex structures through causal analysis, in: Proc. AAAI-93, Washington, DC, 1993, pp. 588–593.
[5] T. Broida, R. Chellapa, Physics-based visual understanding, IEEE Trans. Pattern Analysis and Machine Intelligence (1996) 90–99.
[6] A.M. Bruckstein, R.J. Holt, A.N. Netravali, How to catch a crook, J. Visual Communication and Image Representation 5 (1994) 273–281.
[7] A.M. Bruckstein, R.J. Holt, A.N. Netravali, How to track a flying saucer, J. Visual Communication and Image Representation 7 (1996) 196–204.
[8] H. Buxton, R. Howarth, Watching behaviour: The role of context and learning, in: Proc. International Conference on Image Processing, 1996, 18A2.
[9] C. Cedras, M. Shah, Motion-based recognition: A survey, Image and Vision Computing 13 (1995) 129–155.
[10] Z. Duric, J. Fayman, E. Rivlin, Function from motion, IEEE Trans. Pattern Analysis and Machine Intelligence (1996) 579–591.
[11] S. Engel, J. Rubin, Detecting visual motion boundaries, in: Proc. Workshop on Motion Analysis, 1986, pp. 107–111.
[12] B. Faltings, Qualitative kinematics in mechanisms, Artificial Intelligence 44 (1990) 89–120.
[13] F. Freudenstein, E.R. Maki, The creation of mechanisms according to kinematic structure and function, Environment and Planning B 6 (1979) 375–391.
[14] K. Gould, M. Shah, The trajectory primal sketch: A multi-scale scheme for representing motion characteristics, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1989, pp. 79–85.
[15] D. Heeger, Optical flow from spatiotemporal filters, in: Proc. First International Conference on Computer Vision, 1987, pp. 181–190.
[16] B. Horn, B. Schunck, Determining optical flow, Artificial Intelligence 17 (1981) 189–203.
[17] Y. Huang, K. Palaniappan, X. Zhuang et al., Optic flow field segmentation and motion estimation using a robust genetic partitioning algorithm, IEEE Trans. Pattern Analysis and Machine Intelligence (1995) 1177–1190.
[18] S. Intille, J. Davis, A. Bobick, Real time closed world tracking, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1997, A6: Tracking.
[19] C. Jerian, R. Jain, Determining motion parameters for scenes with translation and rotation, IEEE Trans. Pattern Analysis and Machine Intelligence 17 (1994) 523–530.
[20] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Image and Vision Computing 14 (1996) 609–615.
[21] L. Joskowicz, Mechanism comparison and classification for design, Research in Engineering Design 1 (1990) 149–166.
[22] L. Joskowicz, D. Neville, A representation language for mechanical behavior, Artificial Intelligence in Engineering 10 (1996) 109–116.
[23] L. Joskowicz, E. Sacks, Computational kinematics, Artificial Intelligence 51 (1991) 381–416.
[24] S. Kannapan, K. Marshek, An algebraic and predicate logic approach to representation and reasoning in machine design, Mechanism and Machine Theory 25 (1990) 335–353.
[25] D. Koller, K. Daniilidis, H. Nagel, Model-based object tracking in monocular image sequences of road traffic scenes, Internat. J. Computer Vision 10 (1993) 257–281.
[26] D. Koller, H. Heinze, H. Nagel, Algorithmic characterization of vehicle trajectories from image sequences by motion verbs, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1991, pp. 90–95.
[27] S. Kota, S. Chiou, Design representation and computational synthesis of mechanical motions, in: Proc. 4th International ASME Conference on Design Theory and Methodology, 1992, pp. 365–372.
[28] J.-C. Latombe, Robot Motion Planning, Kluwer Academic, Dordrecht, 1991.
[29] T. Murakami, N. Nakajima, Mechanism concept retrieval using configuration space, Research in Engineering Design 9 (1997) 99–111.
[30] H. Nagel, A vision of vision and language comprises action: An example from road traffic, J. Artificial Intelligence Res. 8 (1994) 189–214.

[31] E. Rivlin, S. Dickinson, A. Rosenfeld, Recognition by functional parts, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 133–140.
[32] A.M.C. Rogonone, A. Verry, Identifying multiple motions from optical flow, in: Proc. European Conference on Computer Vision, 1992, pp. 258–266.
[33] E. Sacks, L. Joskowicz, Automated modeling and kinematic simulation of mechanisms, Computer-Aided Design 25 (1993) 106–118.
[34] H. Shrobe, Understanding linkages, in: Proc. AAAI-93, Washington, DC, 1993, pp. 620–625.
[35] J. Siskind, Naive physics, event perception, lexical semantics, and language acquisition, Ph.D. Thesis, MIT, Cambridge, MA, 1992.
[36] J. Siskind, Q. Morris, A maximum likelihood approach to visual even classification, in: Proc. European Conference on Computer Vision, 1996, II:347–360.
[37] T.D.R. Stahovich, H. Shrobe, Qualitative rigid body mechanics, in: Proc. AAAI-97, Providence, RI, 1997, pp. 138–144.
[38] L. Stark, K. Bowyer, Achieving generalized object recognition through reasoning about association of function to structure, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1991, pp. 129–137.
[39] L. Stark, K. Bowyer, Generic recognition through qualitative reasoning about 3-D shape and object function, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1991.
[40] L. Stark, K. Bowyer, Indexing function-based categories for generic recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1992.
[41] D. Subramanian, C. Wang, Kinematic synthesis with configuration space, Research in Engineering Design 7 (1995) 192–213.
[42] A. Verry, T. Poggio, Against quantitative optical flow, in: Proc. First International Conference on Computer Vision, 1987, pp. 171–180.
[43] J. Weng, T. Huang, N. Ahuja, Motion and structure from two perspective views: Algorithms, error analysis and error estimation, IEEE Trans. Pattern Analysis and Machine Intelligence 65 (1989) 451–476.
[44] Y. Yacoob, L. Davis, Learned temporal models of image motion, in: Proc. International Conference on Computer Vision, 1998, pp. 90–95.