Proceedings of the 1998 IEEE
International Conference on Robotics & Automation
Leuven, Belgium • May 1998

Understanding Mechanisms: from Images to Behaviors

Tzachi Dar
Technion, Israel Institute of Technology, Haifa, Israel
tzachi@tx.technion.ac.il

Leo Joskowicz
Institute of Computer Science
The Hebrew University Jerusalem, Israel
josko@cs.huji.ac.il

Ehud Rivlin
Technion, Israel Institute of Technology, Haifa, Israel
ehudr@cs.technion.ac.il

Abstract

This paper presents a new method for recognizing mechanisms and describing their behaviors from image sequences showing their relations. It uses a simple and expressive language for describing the behavior of fixed-axes mechanisms. The language symbolically captures the important aspects of the kinematics and the simple dynamics of the mechanism. We show how this language combined with a vision system can automatically identify mechanisms and their behaviors from a sequence of images.

1. Introduction

Visual classification is a major task in robotics and artificial intelligence. One aspect of this classification is understanding how a mechanism is organized and how it functions. To operate autonomously, a machine must be able to watch its environment, 'understand' what it sees and respond appropriately. One important aspect of this ability is the understanding of how action – usually motion – is originated, constrained, and how this determines what will happen in the immediate future.

Our goal is to build a system that inputs one or more sequences of images showing parts of a mechanism in motion, recognizes the mechanism and produces a description of the mechanism behavior. The behavior is determined by the motion of its parts and their relations.

Mechanisms perform their function by transforming motions via part interactions. The input motions, the part shapes, and the part contacts determine the output motions. The behavior of a mechanical assembly is a description of how its parts move and interact with each other to achieve a desired function. For example, a gearbox achieves its function by changing the transmission ratio between the input and output shafts.

In this paper, we use a simple and expressive language for describing the behavior of an important class of mechanisms: fixed-axes mechanisms. Parts in a fixed axes mechanism rotate and translate along axes that are fixed in space and interact via intermittent or permanent contacts. This class includes many common mechanisms such as door locks, staplers, and transmissions. The language symbolically captures the important aspects of the kinematics and simple dynamics of the mechanism. It uses predicate and algebraic relations to describe the configurations (positions and orientations) and motions of each part of the mechanism and the relationship between them. It allows partial, multi level abstract descriptions.

General part motions can be described by six parameters and their derivatives with respect to time. Motion relations are described by algebraic equations relating these parameters. However, such complex representations are seldom necessary to describe the behavior of most common mechanisms. Important classes of mechanisms can be described by simpler and more semantically meaningful representations. In fixed-axes mechanisms, parts have a single degree of freedom (rotation, translation, or helical) along an axis that is fixed in space. Contacts between parts can change, giving raise to different behaviors. In linkages, part motions can be arbitrary, but parts are permanently connected by joints (revolute, prismatic, etc) which restrict their degrees of freedom. In a study of 2,500 mechanisms from a mechanisms encyclopedia Joskowicz and Sacks determined that 60% of mechanisms are fixed axes, while 30% percent are linkages [12]. We thus concentrate initially on describing the behavior of fixed-axes mechanisms.

Previous work reports various methods to understand how mechanisms work and reason about their functioning. They can be classified into two categories, depending on the input they use. The first category assumes that full mechanism information is available, e.g. that the geometry of its parts, their degrees of freedom, and their motions is provided by the user. Several algorithms are described to interpret this information to recognize classify, and compare the mechanisms (see [10][11] and the cited references). For example, Gelsey [5] describes algorithms which create short and long term behavioral models from kinematic and dynamical analysis for a class of mechanisms. The second category assumes that the input is gathered from sensory data (e.g. still images or video sequences) and attempts to extract high-level behavioral information from it. For example Brand et al. [1][2] describe a computationally inexpensive method for recovering causal structure from image. In their method, a scene model is built incrementally through interleaved sensing and analysis. Our work belongs to this second category: we use long sequences of video images as input for the reasoning process. We use the type of behavior description languages developed in the first category to describe mechanisms.

<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>	::=	(<Part>,<Motion-type>,<Axis>,<Motion-parameter =Amount>,<interval>)
<Part>	::=	Part-Name
<Motion-type>	::=	<Motion> | <Motion> <Motion-modifier>
<Motion>	::=	**Translation | Rotation | Helical | Translation-and-rotation | No-motion**
<Motion-modifier>	::=	**Alternate | With-Dwell | Alternate With-Dwell**
<Axis>	::=	Axis-name
<Motion parameter>	::=	Motion-parameter-name (velocity)
<interval>	::=	<t_start> < t < <t_end>
<t_start>	::=	<Amount>
<t_end>	::=	<Amount>
<Amount>	::=	Real-value | Constant | Variable

Table 1: A BNF description of the improved language for mechanism's behaviors.

2. Method overview and paper outline

To automatically recognize and understand mechanisms behavior language developed by Joskowicz and Neville [9]. We found that the original language has several drawbacks for the task at hand, and have developed a revised BNF of it (Section 3). The algorithm parses the image sequence and extracts behavioral descriptions from it, which it expresses as sentences in the behavioral description language. It starts by extracting the motion parameters for each moving part of the mechanism by computing the normal optical flow (Section 4). It then derives motion profiles and finds the behavior that best matches it (Section 5). Motion profile computation is based on motion detection and segmentation of the uniform motion segments of the mechanisms individual parts. We have implemented this method and successfully processed a dozen image sequences (Section 6).

3. Describing mechanisms behavior

Joskowicz and Neville [9] present a simple yet comprehensive language for describing the behavior of fixed axes mechanisms. The advantages of the language are that it distinguishes between structural and behavioral information. It allows behavioral descriptions of subset of parts. Desired behaviors and mechanism functions are frequently described as an input/output relation between the configurations and motion parts. It allows partial descriptions of a subset of all possible behaviors.

The language is not directly suitable for our purposes because parts can only be recognized if they move. As a consequence, causal relations (e.g. which part is the input driving part and which part is the output driven part) cannot be distinguished. Also, static parts cannot be detected. We must modify the language to eliminate causality, allowing only sentences of the form <motion-sequence>+. In the revised language, part names will be arbitrary variables, not meaningful names.

After having processed the behavior of the mechanism, the recognition of its different parts follows a top-down approach. In the processing the algorithm will use memory to overcome the last problem mentioned. It tracks a part while it is moving and remembers its existence while it is not in motion. This way, the algorithm can not handle some of the situations where no motion is present. It can not be able to distinguish between the two types of no-motion (hold or stationary) and can not recognize objects that don't move during the entire sequence.

Table 1 describes the formal BNF specification of the language that we use for describing mechanisms. The language is based on motion descriptions. The sentences of this language are written automatically by a system that accepts a sequence of images as input. We informally describe 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 parts. 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 part. It consists of a unique part name, motion type, axis, the motion parameter, and the interval of the motion. The motion type can be a continuous motion along the axis: **Translation, Rotation, Helical** (coupled translation and rotation), simultaneous uncoupled **Translation-and-Rotation**, or **No-Motion**.

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.

3.1 Using the language

To show the expressiblty and coverage of the language we will present two simple examples of mechanisms and their behavioral description.

3.1.1 Two parallel gears of different diameters

Consider the mechanism formed by two meshed parallel gears (Figure 1(a)). The number of the teeth in one of them is twice their number in the other. Its behavioral description (as it appears in the database) is:

(Wheel 1, Rotation, O1, $C1 = \alpha$, $0<t<\infty$)

(Wheel 2, Rotation, O2, $C2 = -2\alpha$, $0<t<\infty$)

where C1 and C2 are the rotation parameters. Assuming the driving gear turns at constant speed ($C1 = \alpha$) then C2 must be -2α, which means that it is rotating in the opposite direction and at twice the speed. The derivation of this description follows the BNF nodes <Simultaneous-motions>. For the first <Motion- sequence> the moving <part> is Wheel1, the motion type is **<rotation>**, the <Axis> is O1, the <motion parameter> is C1, the amount is α, and the interval is from 0 to ∞.

3.1.2 A wheel and a slider pair

Consider the mechanism formed by a wheel that is rotating in a constant motion, and a slider (figure 1(g)). The slider translates back and forth as the wheel rotates according to a sine relationship. The pair's behavioral description (as it appears in the database) is given by the following sentence:

(Wheel 1, Rotation, O1, $C1 = \alpha$, $0<t<\infty$)

(Slider 1, Translation, O2, $U2 = \sin(\alpha \cdot t)$, $0<t<\infty$)

The wheel is turning in a constant velocity ($C1 = \alpha$). The slider translates back and forth as the whell rotates according to a sine relationship. Its x coordinate velocity parameter is U2. Similarly, other fixed axes mechanisms can be added to the database for later matching and retrieval.

4. Recovering motion

Motion parameter identification is an essential part of the parsing process. The parameters describe which parts are moving and their degrees of freedom. There are two main methods for motion parameter recovery: feature-based, which is a discrete approach, and optical-flow based, which is a differential approach. The feature-based method consists of two steps: 1) finding correspondences between points in the different frames, and 2) estimating the motion parameters and the structure of the scene from correspondences. Many works follow this approach (e.g.[13][3][8]) but have a common drawback: they require a large number of point feature correspondances to achieve robustness and require tracking these points for a long time, which is difficult. The optical flow method has two related stages. The first stage involves the computation of the optic flow field from monocular image sequences. The second stage is to meaningfully interpret the computed optic flow field regarding the underlying object structures and motion parameters. This approach, described in [4][7], is the one we will follow here. Its advantage is that with a few constrains on the nature of the motion, it permits the derivation of the motion parameters using linear equations. We describe this derivation next.

4.1 Rigid body motion

To facilitate the derivation of the motion equations of a rigid body we use two rectangular coordinate frames, one (O_{xyz}) fixed in space, the other $(C_{x_1y_1z_1})$ fixed in the body and moving with it. The coordinates x_1, y_1, z_1 of any point *P* of the body with respect to the moving frame are constant with respect to time *t*, while the coordinates X, Y, Z of the same point *P* with respect to the fixed frame are functions of *t*. The position of the moving frame at any instant is given by the position $\vec{d}_c = (X_c Y_c Z_c)^T$ of the origin *C*, and by the nine direction cosines of the axes of the moving frame with respect to the fixed frame. Let $\vec{i}, \vec{j}$ and $\vec{k}$ be the unit vectors in the directions of the Ox, Oy and Oz axes, respectively; and let $\vec{i}_1, \vec{j}_1$ and $\vec{k}_1$ be the unit vectors in the directions of the Cx_1, Cy_1 and Cz_1 axes, respectively. For a given position $\vec{p}$ of *P* in $C_{x_1y_1z_1}$ we have the position $\vec{r}_p$ of *P* in $Oxyz$

$$\vec{r}_p \equiv (X\ Y\ Z)^{(T)} \equiv R\vec{p} + \vec{d}_c$$

The velocity of $\vec{r}_p$ is then given by

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

where $\vec{\omega} = (A\ B\ C)^T$ is the rotational velocity of the moving frame; $\dot{\vec{d}}_c = (\dot{X}_c \dot{Y}_c \dot{Z}_c)^T \equiv (UVW)^T \equiv \vec{T}$ is the translational velocity of the point *C*. This can be written as

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} 0 & -C & B \\ C & 0 & -A \\ -B & A & 0 \end{pmatrix} \begin{pmatrix} X - X_c \\ Y - Y_c \\ Z - Z_c \end{pmatrix} + \begin{pmatrix} U \\ V \\ W \end{pmatrix} \tag{1}$$

Let the rotational velocity in the moving frame be $\vec{\omega}_1 = (A_1\ B_1\ C_1)^T$; we can write $\vec{\omega} = R\vec{\omega}_1$ and $\vec{\omega}_1 = R^T\vec{\omega}$.

4.2 The imaging model

Let (X, Y, Z) denote the Cartesian coordinates of a scene point with respect to the fixed camera frame, 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 we fix a reference point (X_c, Y_c, Z_c). A scene point (X, Y, Z) is first projected onto the point (X, Y, Z_c) then, through plane perspective projection the point (X, Y, Z_c) is projected onto the image point (x, y). The projection equations are then given by $x = \frac{X}{Z_c} f, y = \frac{Y}{Z_c} f.$ (2)

4.3 The motion field and the optical flow field

The instantaneous velocity of the image point (x, y) under weak perspective projection can be obtained by taking derivatives of (2) with respect to time and using (1):

$$\dot{x} = f\frac{\dot{X}Z_c - X\dot{Z}_c}{Z_c^2} = f\frac{[-C(Y-Y_c)+B(Z-Z_c)+U]Z_c - XW}{Z_c^2} = \frac{Uf - xW}{Z_c} - C(y - y_c) + fB\left(\frac{Z}{Z_c} - 1\right) \quad (3)$$

$$\dot{y} = f\frac{\dot{Y}Z_c - Y\dot{Z}_c}{Z_c^2} = f\frac{[C(X-X_c)-A(Z-Z_c)+V]Z_c - YW}{Z_c^2} = \frac{Vf - yW}{Z_c} + C(x - x_c) + fA\left(\frac{Z}{Z_c} - 1\right) \quad (4)$$

where $(x_c, y_c) = (f\,x_c/z_c, f\,y_c/z_c)$ is the image of the point C. Let $\vec{i}$ and $\vec{j}$ be the unit vectors in the x and y directions, respectively; $\dot{\vec{r}} = \dot{x}\vec{i} + \dot{y}\vec{j}$ is the projected motion field at the point $\vec{r} = x\vec{i} + y\vec{j}$.

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

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = (I_x i + I_y j)\cdot(\dot{x}\vec{i} + \dot{y}\vec{j}) + I_t = \nabla I \cdot \dot{\vec{r}} + I_t$$

where ∇I is the image gradient. If we assume $dI/dt = 0$, i.e. that the image intensity does not vary with time [6], we obtain $\nabla I \cdot \vec{u} + I_t = 0$. 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 then obtain:

$$\vec{u}_n = (\vec{u}\cdot\vec{n}_r)\vec{n}_r = \frac{-I_t \nabla I}{\|\nabla I\|^2} \quad (5)$$

where $\vec{u}_n$ is called the *normal flow*.

It was shown in [14] that the magnitude of the difference between $\vec{u}_n$ and the normal motion field $\dot{\vec{r}}_n$ is inversely proportional to the magnitude of the image gradient. Hence $\dot{\vec{r}}_n \approx \vec{u}_n$ when $\|\nabla I\|$ is large. Equation (5) thus provides an approximate relationship between the 3-D motion and the image derivatives.

4.4 Recovering the motion parameters

Following [4] and adopting their assumption that the transnational velocity of the object is constrained mostly to the fronto-parallel plane, that is $T_1 = (U_1\ V_1\ 0)^T$ and $\vec{T} = R\vec{T}_1$, we define $\boldsymbol{a}$ and $\boldsymbol{c}$ (following [4]) to be:

$$a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} \equiv \begin{pmatrix} n_x f \\ -n_x x \\ -n_x(y - y_c) \\ n_y f \\ -n_y y \\ n_y(x - x_c) \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix} \equiv \begin{pmatrix} U/Z_c + (x_c/f)C_1 N_x N_y/N_z \\ W/Z_c + C_1 N_x N_y/N_z \\ C_1(N_z + N_y^2/N_z) \\ V/Z_c - (y_c/f)C_1 N_x N_y/N_z \\ W/Z_c + C_1 N_x N_y/N_z \\ C_1(N_z + N_x^2/N_z) \end{pmatrix} \quad (6)$$

where N is the normal to the motion plane of the mechanism and n ($\equiv n_x\vec{i} + \vec{n}_y j$) is the normal direction (image gradient direction), then we can write the equation:

$$\dot{\vec{r}}_n \cdot \vec{n} = a^T c \quad (7)$$

The column vector $\boldsymbol{a}$ is formed of the observable quantities only, while each element of the column vector $\boldsymbol{c}$ contains quantities which are not directly observable from images. To estimate $\boldsymbol{c}$ we need estimates of $\dot{\vec{r}}_n \cdot \vec{n}$ at six or more image points.

4.5 Estimating motion from normal flow

If we use the spatial image gradient as the normal direction $\vec{n}_r \equiv \nabla I/\|\nabla I\| = n_x\vec{i} + \vec{n}_y j$ and $\dot{\vec{r}}_n = \vec{u}_n$ we can obtain an approximate equation by replacing the left hand side of (7) by normal flow $-I_t/\|\nabla I\|$. In this way we obtain one approximate equation in the six unknown elements of $\boldsymbol{c}$. For each point $(x_i, y_i), i = 1, \ldots, m$ of the image at which $\|\nabla I(x_i, y_i, t)\|$ is large we can write one equation. If we have more than six points we have an over-determined system of equations $Ac \approx b$; the rows of the $m \times 6$ matrix A are the vectors a_i and the elements of the $m-$vector $\boldsymbol{b}$ are: $-(\partial I(x_i, y_i, t)/\partial t)/\|\nabla I(x_i, y_i, t)\|$. We seek the solution for which $\|b - Ac\|$ is minimal. This solution is the same as the solution of the system $A^T Ac = A^T b \equiv d$. The system $A^T Ac = d$ can be solved using the Cholesky decomposition.

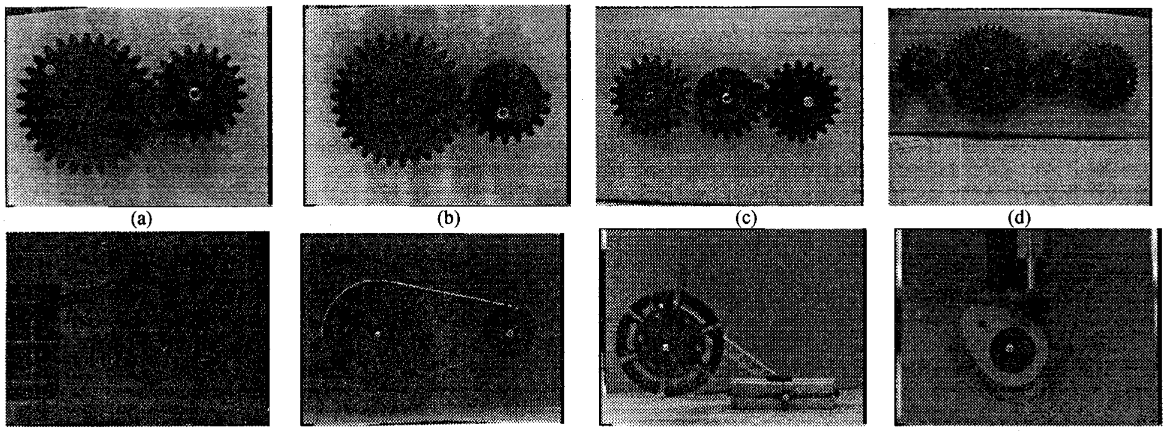


Figure 1: Mechanisms from the database: (a) two gears pair; (b) half gear pair; (c) two gears and half a gear; (d) four gears pair; (e) a door lock mechanism (f) two gears connected with a chain; (g) a wheel slider pair; (h) en ellipse and a slider pair.

After estimating $\boldsymbol{c}$ we can use (6) to obtain $\vec{T}/Z_c$ and C_1:

We let $c_7 = (c_2 - c_5)/2$, and have

$$\frac{U}{Z_c} = c_1 - \frac{x_c c_7}{f}, \quad \frac{V}{Z_c} = c_4 - \frac{x_c c_7}{f}$$

$C_1 = \operatorname{sgn}(c_6)\sqrt{c_3 c_6 - c_7^2}$ where sgn is the sign function.

In the next Section we will show how these results are used to detect and classify mechanisms.

5. Image sequence parsing

The goal of the image sequence parsing is to retrieve from the mechanism database the description that best matches the given image sequence. The algorithm starts by segmenting the sequence to identify the different parts that are in motion (see Section 6.1). Next, it finds the motion parameters of each object and translates this information into sentences that represent each behavior of the mechanism.

To find the sentences that represent each behavior in the database, we hold a list of all the behaviors that exist in the database. For each behavior the algorithm compares the difference between the behavior in the database and the behavior of the mechanism as it is understood by the vision system. Each mechanism behavior is described by three graphs: one for each motion in the linear directions U,V, and one for the rotation C1. To decide which sentence in the database of the mechanisms represents the observed behavior, the algorithm minimizes the difference between the observed behavior of the mechanism and the behaviors in the database. The difference is formally given by $e_i = \int(\left|S_i - S_v\right|)$. The algorithm then chooses the behavior from the database that brings to minimum as the one that best fits the observable behavior. The output is a sentence in the language that represent the behavior of the mechanism. This process is repeated for each behavior. The algorithm queries the database with the resulting sentence and retrieves the matching mechanism.

We thus obtain the sentence in the language that represent the behavior of the mechanism. The process is repeated for each behavior. The algorithm queries the database with the resulting sentence and retrieves the matching mechanism

As an example let us assume that we have a sequence of images showing two meshed gears spinning together, one of them has 20 teeth and the other 30 teeth. The vision system will segment the scene into two moving objects, both rotate continuously and their velocities. An example for such an output is:

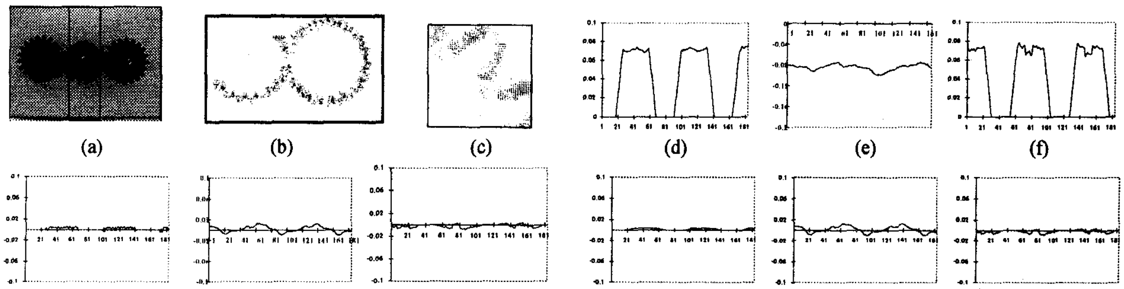
(Object 1, Rotation, O1, C1 = 0.05 rad/sec, $0<t<10$)

(Object 2, Rotation, O2, C1 =-0.075 rad/sec, $0<t<10$)

Our system will search all the mechanisms in the database and come to a conclusion that this is a two spinning object pair with a velocity ratio of 2:3. Once this information is known it will vote for a two gear pair.

6. Experimental results

This Section presents results of experiments with several mechanisms. Our database contains over ten different mechanisms. For each mechanism images were taken from three angles (one fronto-parallel, and the others from 20° and 40°). Each sequence contains more than 100 images. Each sequence was segmented in order to find the independently moving objects (The technique of the segmentation is explained in the next subsection). For each object we calculated three graphs: The rotational velocity C1, the motion in the x direction (U) and the motion in the y direction (V). The complete calculation process took about 2 hours on a SGI IP25 Computer. This time was achieved without any optimizations. The search in the database is done in a linear way, but in the future we intend to use a more sophisticated way [10].



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 2: Results for the two gears and half a gear mechanism: (a) the results of the segmentation; (b) flow results for the two gears and half a gear mechanism (c) zoomed flow results for the two gears and half a gear mechanism. (d) C1 for the left gear (e) C1 for the center gear; (f) C1 for the right gear; (g) U for the left gear; (h) U for the center gear; (i) U for the right gear; (j) V for the left gear; (k) V for the center gear; (l) V for the right gear;

6.1 Motion recovery and segmentation

As described in Section 4,5, the algorithm starts the recovery of the motion parameters using normal flow. The next step should have been to calculate the motion parameters. However, this technique was based on the assumption that the area contained only a single moving rigid object. As can be seen in the images, this assumption is not valid in our case. Our sequences contain an unknown number of objects that are occasionally moving and stopping. To overcome this problem we have used a two-stage method. In the first stage we define *a uniform motion criterion.* This criterion is used in the second stage to segment the different motion areas.

6.1.1 Uniform motion criterion

The uniform motion criterion is used to decide weather a sequence of images contains one or more moving objects. As described before there is a simple connection between the motion of a rigid body and the flow that it creates. If we assume that a sequence has only one rigid motion in it, then we can recover the parameters of this motion. Now that these parameters are calculated we can compute the normal optical flow for each point in the image. Each point has now two normal flows: one that is calculated from the sequence of images and the other that is calculated from the recovered motion parameters, under the assumptions that only a single rigid motion exists. If the sequence contains a single rigid motion the two flows should be identical (or very close, because of errors in the calculation of the flow). We now calculate the distance between the flows for each point and check if it's bigger then a pre-defined threshold. The ratio between the points that don't pass the threshold and the total number of points is the value of the uniform motion criterion for this region. In 6.2 we will use this method to segment the sequence of images.

6.2 Motion segmentation

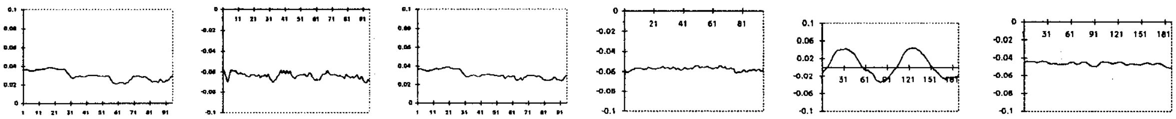
To segment the sequence of images into areas that contain a uniform motion of a single object we use the uniform motion criterion. The first stage is to do motion segmentation for the first pair of images in the sequence. Then, we start tracing the different parts that are moving (notice that after the first stage new objects can start moving or moving objects can stop). In the first stage, the algorithm segments the motion for the first two images. It starts by testing the uniform motion area of the entire region. If it's bigger than a predefined threshold, the algorithm halts. Otherwise it cuts the entire area into rectangular regions with at least a single motion area. It recursively repeats the splitting until a single motion is left in each area.

In the second stage, the algorithm uses the current segmentation as a basis for analyzing the next frames. It examines each area for one of three possible cases: 1) a new motion starts – the algorithm split the areas to hold the next motion; 2) an existing motion stops – the algorithm records the area that last contained this motion to keep trace of the object; and 3) objects change their motion through continued movement -- the algorithm updates the areas of the different motions.

6.3 Results

We now present the experimental results for three mechanisms from our database. For the first mechanism we show all the recovered motion parameters, but for the other two we show only the informative parameters (the non-informative parameters in our case are parameters that are always (close to) zero – such as U, V for a wheel).

The first mechanism consists of half a gear that is driven by a motor, and is rotating the two other gears (see figure 1(c)). Each time the half gear is meshed with another gear this gear is moving. The pair's behavioral description (as it appears in the database) is given by the



(a) (b) (c) (d) (e) (f)

Figure 3: Informative results for the four wheels mechanisms: (a) C1 for the left gear; (b) C1 for the center left gear; (c) C1 for the center right gear; (d) C1 for the right gear. Informative results for the wheel and slider pair: (e) U for the slider; (f) C1 for the wheel.

following two sentences:

B1:

(L-Wheel, No-motion, O1, C1 = 0, $t1<t<t1+\pi/\alpha$)

(half-Wheel, Rotation, O2, C2 = α, $t1<t<t1+\pi/\alpha$)

(R-Wheel, Rotation, O3, C3 = $-\alpha$, $t1<t<t1+\pi/\alpha$)

B2:

(L-Wheel, Rotation, O1, C1 = $-\alpha$, $t1++\pi/\alpha<t<t1+2\pi/\alpha$)

(half-Wheel, Rotation, O2, C2 = α, $t1++\pi/\alpha<t<t1+2\pi/\alpha$)

(R-Wheel, No-motion, O3, C3 = 0, $t1++\pi/\alpha<t<t1+2\pi/\alpha$)

In this example t1 serves as a parameter for the start time of the first cycle. Since we have assumed that for the half wheel the velocity is α, the time for half a cycle is π/α. The velocity of the wheels is in the same size but in the opposite direction than that of the half-wheel. This behavior is shown on the graphs(figure 2(d-l)): α=0.075 rad/frame therefore a complete cycle should end in $2\pi/0.075$=83 frames. The U, V values are close to zero since the gears have only rotational velocity.

The second mechanism consists of four gears that are meshed together (see figure 1 (d)). The four wheels are always moving together but each two of them are spinning in a different direction. The absolute value of the rotation parameter (C1) is inversely proportional to the number of teeth that each gear has. The results can be seen in figure 3(a)-3(d).

The third mechanism that we examine is a wheel and a slider pair (see figure 1(g)). The wheel is being driven by a motor and is rotating in a constant velocity. It is pushing and pulling the slider that is moving in (see figure 3 v,vi) a motion that is close to a sinus. The graphs attached show this behavior. The non-informative parts of the graphs were omitted.

7. Conclusions

In this paper we have presented a system that given a sequence of images describing a mechanism in action parses a movie into different behaviors and recognizes the mechanism. We have presented an appropriate language for representing these different behaviors of fixed axes mechanisms and have given an algorithm for extracting them from such a sequence of images. Our experiments have showed the ability to recognize a mechanism in motion from a sequence of images, and to successfully recover it's behaviors. In the future we plan to expand the family of mechanisms we are dealing with to include a wider sets of mechanisms.

References

[1] M. Brand. Physics-Based visual understanding. Computer vision and image understanding. vol 65 pp.192-205, February 1997.

[2] M. Brand, L. Birnbaum, P. Cooper. sensible scenes: Visual understanding of complex structures through causal analysis. Proceedings, National Conference on Artificial Intelligence, Washington D.C., 1993

[3] T.J. Broida and R. Chellapa. Estimation of object motion parameters from noisy images. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 90-99, January 96.

[4] Z. Duric, J. Fayman and E. Rivlin. Function from motion. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.579-591, June 1996,

[5] A. Gelsey. Automated reasoning about machines. Artificial Intelligence 74 (1995) pp.1-53

[6] B.K.P. Horn and B.G. Schunck. Determining optical flow. Artificial intelligence. Vol. 17, pp. 189-203, 1981.

[7] Y. Huang, K. Palaniappan, X. Zhuang and J.E. Cavanaugh. Optic flow field segmentation and motion estimation using a robust genetic partitioning algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 1177-1190 December 95.

[8] C. Jerian and R. Jain. Determining motion parameters for scenes with translation and rotation. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 523-530, July 1994.

[9] L. Joskowicz and D. Neville. A Representation Language for Mechanical Behavior. In Artificial Intelligence in engineering pp 109-116 10 1996.

[10] L. Joskowicz. Mechanism Comparison and classification for Design. Artificial Intelligence in Engineering, Vol 10, 1996.

[11] L. Joskowicz and E. Sacks. Unifying Kinematics and Dynamics for the automatic Analysis of Machines. Submitted to the 8'th National Conference on Artificial Intelligence, Boston, MA 1990.

[12] L. Joskowicz and E. Sacks. Computational kinematics. Artificial Intelligence, Vol. 51, Nos. 1-3,1991.

[13] J. Weng, T.S. Huang and N. Ahuja. Motion and structure from two perspective views: algorithms, error analysis and error estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence,pp, 451-476, May 1989.

[14] A. Verry and T. Poggio. Against quantitative optical flow. Proc. Int'l Conf. Comp Vision, pp. 171-180, London, England, June 1997.