Visual Motion and Structure Estimation Using Sliding Mode Observers

by

Mustafa Unel, munel@sabanciuniv.edu

Asif Sabanovic, asif@sabanciuniv.edu

Burak Yilmaz, burakyilmaz@su.sabanciuniv.edu

Eray Dogan, eraydogan@su.sabanciuniv.edu

Authors are affiliated with Faculty of Engineering and Natural Sciences
Sabanci University, Istanbul, Turkey

Contact Author: Mustafa Unel (munel@sabanciuniv.edu)
Faculty of Engineering and Natural Sciences
Sabanci University
Orhanli-Tuzla 34956, ISTANBUL, TURKEY
Phone: 90 (216) 483 9549
Fax: 90 (216) 483 9550

Abstract

The problem of estimating motion and structure from a sequence of images has been a major research theme in machine vision for many years and remains one of the most challenging ones. In this work, we use sliding mode observers to estimate the motion and the structure of a moving body with the aid of a CCD camera. We consider a variety of dynamical systems which arise in machine vision applications and develop a novel identification procedure for the estimation of both constant and time varying parameters. The basic procedure introduced for parameter estimation is to recast image feature dynamics linearly in terms of unknown parameters and construct a sliding mode observer to produce asymptotically correct estimates of the observed image features, and then use “equivalent control” to explicitly compute parameters. Much of our analysis has been substantiated by computer simulations and real experiments.

KEYWORDS: Machine Vision, Perspective, Motion and Structure Estimation, Camera Models, Sliding Mode Observers
Visual Motion and Structure Estimation Using Sliding Mode Observers

Mustafa Unel, Asif Sabanovic, Burak Yilmaz and Eray Dogan *

January 30, 2006

Abstract

The problem of estimating motion and structure from a sequence of images has been a major research theme in machine vision for many years and remains one of the most challenging ones. In this work, we use sliding mode observers to estimate the motion and the structure of a moving body with the aid of a CCD camera. We consider a variety of dynamical systems which arise in machine vision applications and develop a novel identification procedure for the estimation of both constant and time varying parameters. The basic procedure introduced for parameter estimation is to recast image feature dynamics linearly in terms of unknown parameters and construct a sliding mode observer to produce asymptotically correct estimates of the observed image features, and then use “equivalent control” to explicitly compute parameters. Much of our analysis has been substantiated by computer simulations and real experiments.

KEYWORDS: Machine Vision, Perspective, Motion and Structure Estimation, Camera Models, Sliding Mode Observers

1 Introduction

Estimating rigid scene structure and the relative 3D motion of a camera from image sequences has been one of the most challenging research issues in vision and it continues to be so [1]. Several solutions have been proposed, based on different measurements and different estimation algorithms. These solutions can be classified into two categories depending upon what is measured from the scene. If the brightness pattern is the data observed from images, a well known approach is based on analyzing the optical flow (see [1], [2], [3], [4]). On the other hand if the data observed are the discontinuity-curves in the image brightness pattern, a possible approach is to identify the correspondence of various features such as points, lines and curves between consecutive frames (see [5], [6], [7], [8], [10], [12], [14], [15]). The former approach assumes that the image intensity is a smooth function and considers only the smooth part of the image. The latter approach assumes that

*Authors are with the Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli-Tuzla 34956, ISTANBUL, TURKEY (e-mail: {munel, asif}@sabanciuniv.edu and {burakyilmaz, eraydogan}@su.sabanciuniv.edu)
the image intensity is a piecewise smooth function and concentrates onto the image discontinuity curves.

Motion parameter estimation in machine vision has been initiated by the early works of Ullman [11]. Roach and Aggarwal [18] tested the estimation problem with real images. Nagel [19] reduced the problem to the solution of a nonlinear equation. An analytical solution for eight feature points was given independently by Longuet-Higgins [20], and Tsai and Huang [21]. Zhuang [22], [23] presented a simplified eight point algorithm and considered the uniqueness issue. In the uniqueness context, Netravali et al., [24] introduced the homotopy method and showed the existence of ten solutions. Faugeras and Maybank [9] showed that at most ten solutions can be obtained from five feature points using projective geometry. Using the quaternions, Jerian and Jain [25] reduced the problem to solving the resultant of a pair of quartic binary polynomials. They also compared known algorithms based on their performances with noisy data [26].

Many of the existing algorithms in the literature perform poorly under noisy data. A robust algorithm was introduced by Weng et al., [27] and by Spetsakis and Aloimonos [16], [17]. They used optimization based methods to compute epipolar equations. Grzywacz and Hildreth [28] have also shown that the effects of image noise on reconstruction from image velocities are severe in some cases. Jerian and Jain [25], Murray and Buxton [29] proposed various schemes towards a stable reconstruction algorithm. The particular estimation problem has been summarized in two books by Maybank [30] and by Kanatani [13]. For some other references we would like to refer to [37], [40], [38]. Ghosh et al., [35, 36] has shown that the problem of motion and shape estimation for a moving textured surface can always be analyzed as a specific parameter estimation problem of a perspective system. A perspective system is a linear system with a homogeneous observation function (see [34]). The specific form of the perspective system depends on how the surface and the motion field have been parameterized.

In this paper, we consider a variety of dynamical systems which arise in machine vision applications and develop a novel identification procedure for the estimation of both constant and time varying parameters. We will be dealing with the ‘feature based analysis’ (see [32], [31], [39], [41], [14]). The basic procedure introduced for parameter estimation is to recast image feature dynamics linearly in terms of unknown parameters and construct a sliding mode observer [42, 43] to produce asymptotically correct estimates of the observed image features, and then use “equivalent control” to explicitly compute parameters.

The paper is organized as follows: Some motivational problems are presented in the next section.
The camera, motion and structure models are described in Section 3. Composite image features and sliding mode observers are introduced in Section 4 and 5 respectively. Simulation and experimental results are presented in Sections 6 and 7. Final section is on concluding remarks.

2 Motivational Problems

Let us consider the following sample problems and example applications provided beneath each problem. In the next sections, we will be focusing on these problems more formally and systematically.

Problem 1: Suppose that an object undergoes some kind of rigid and/or affine motion with possibly time varying parameters in a plane perpendicular to the optical axis of a CCD camera. Estimate the shape and motion parameters of the object from the observed time varying images produced by the camera.

We can cite several examples which fall into this problem category: a mobile robot maneuvering on a flat horizontal surface and being viewed from a camera whose optical axis is down to the surface; a robot arm which picks a free-form moving part on a conveyor belt using images taken from a camera mounted to the ceiling; tracking the motion of a microorganism whose shape deforms during its route under a composite vision system which consists of an optical microscope plus a CCD camera, and lip tracking or lip reading for speech recognition in noisy environments.

The first two examples are related to the identification of rigid motion parameters whereas the last two examples are related to the affine motion estimation. However, in all of these examples, the perspective projection of the CCD camera reduces to a scaled orthographic projection due to a constant depth.

Problem 2: Suppose that a camera produces a perfect image of a textured surface which is moving in space following a certain vector field where both the shape of the surface and the vector field are unknown. The problem of interest is to estimate the shape and motion parameters of the surface from the observed time varying image produced by the camera.

Here we assume that the surface is constantly under focus i.e. there is no blurring of the image as a result of imperfect focusing. We also assume that the photometric effects on the image due to the light sources and the physical properties of the surface is negligible and can be ignored.

Virtually all examples can be put into this problem category. However, we will be dealing in
detail with the rigid motion of a planar surface since most of the complex objects have planar faces.

3 Camera, Motion and Structure Models

In this section, we describe models for the camera imaging geometry, the motion of the scene relative to the camera, and the structure of rigid surfaces in the scene. We assume that a stationary camera is viewing a rigid surface which is moving smoothly in 3D. The same analysis applies to the case of a moving camera viewing a stationary surface.

3.1 Camera Projection Model

Let 3D points on a rigid surface be defined with respect to the camera reference frame, with the depth axis is aligned with the optical axis and perpendicular to the image plane. The image plane is at a distance \( f \), the focal length, from the center of projection (COP). The perspective projection of a 3D surface point \((X, Y, Z)\) onto an image point \((x, y)\) is given by

\[
\begin{align*}
x &= f \frac{X}{Z}, \\
y &= f \frac{Y}{Z}.
\end{align*}
\]

(3.1)

Note that when the focal length becomes quite large, i.e. \( f \to \infty \), the perspective projection is reduced to an orthographic projection given by

\[
\begin{align*}
x &= X, \\
y &= Y
\end{align*}
\]

(3.2)

When the depth of objects, \( \delta z \), is much smaller than their average distance, \( \bar{Z} \), from the camera along the optical axis, i.e. \( |\delta z| \ll \bar{Z} \), the full perspective camera model can be replaced by the weak-perspective or so called scaled orthographic camera model. An example for this is the camera located at a sufficiently large distance, \( d \), above a plane of a conveyor belt, with moving piece-parts over it as in Figure 1. In this case \( \bar{Z} = d \). The perspective projection given in (3.1) takes the following form:

\[
\begin{align*}
x &= f \frac{X}{\bar{Z}} = \frac{f X}{d}, \\
y &= f \frac{Y}{\bar{Z}} = \frac{f Y}{d}
\end{align*}
\]

(3.3)

3.2 Motion Models

If a rigid body is moving with instantaneous translational velocity, \( T \), and rotational velocity, \( \Omega \), then the 3D instantaneous velocity of points on the surface is given by
\[
\begin{pmatrix}
\dot{X} \\
\dot{Y} \\
\dot{Z}
\end{pmatrix} = \Omega \times \begin{pmatrix} X \\
Y \\
Z \end{pmatrix} + T \Rightarrow \begin{pmatrix}
\dot{X} \\
\dot{Y} \\
\dot{Z}
\end{pmatrix} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\
\omega_3 & 0 & -\omega_1 \\
-\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} X \\
Y \\
Z \end{pmatrix} + \begin{pmatrix} t_1 \\
t_2 \\
t_3 \end{pmatrix} \tag{3.4}
\]

where \( \Omega = (\omega_1, \omega_2, \omega_3)^T \) and \( T = (t_1, t_2, t_3)^T \). Note that \([\Omega]_x\) is a skew-symmetric, i.e. \([\Omega]_x + [\Omega]_x^T = 0\), obtained from \(\Omega\).

The motion of the projected points in the image plane, \((\dot{x}, \dot{y})\), can be obtained from (3.1):
\[
\begin{pmatrix}
\dot{x} \\
\dot{y}
\end{pmatrix} = f \begin{pmatrix} \frac{X-xZ}{Z} \\
\frac{Y-yZ}{Z} \end{pmatrix}, \tag{3.5}
\]
and using (3.4) gives the required 2D motion equations:
\[
\dot{x} = f\omega_2 x^2 - f\omega_1 xy - f\omega_3 y + f\omega_2 + f(t_1 - xt_3)/Z, \tag{3.6}
\]
\[
\dot{y} = -f\omega_1 y^2 + f\omega_2 xy + f\omega_3 x - f\omega_1 + f(t_2 - yt_3)/Z. \tag{3.7}
\]

An affine motion in 2D, on the other hand, can be described as
\[
\frac{d}{dt} \begin{pmatrix} X \\
Y \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\
a_3 & a_4 \end{pmatrix} \begin{pmatrix} X \\
Y \end{pmatrix} + \begin{pmatrix} b_1 \\
b_2 \end{pmatrix} \tag{3.8}
\]
In case the \(2 \times 2\) matrix
\[
M \triangleq \begin{pmatrix} a_1 & a_2 \\
a_3 & a_4 \end{pmatrix}
\]
happens to be a skew-symmetric matrix, i.e. \(M + M^T = 0 \iff M = \begin{pmatrix} 0 & -\omega \\
\omega & 0 \end{pmatrix}\), the motion will be termed as a rigid motion. Note that in this case \(e^{Mt} \in SO(2)\), the rotation group of \(2 \times 2\) matrices.

### 3.3 Planar Structure

Note the existence of depth variable, \(Z\), in the motion field described in (3.6)-(3.7). Depth affects the image dynamics. If we assume a smooth surface structure, locally a linear depth variation can be used. This means that locally about a point \((X, Y, Z)\), the surface is approximately planar, i.e. \(Z \approx pX + qY + r\), with the orientation defined by the surface normal, \(N = (p, q)^T\).

Now the depth variable \(Z\) can be eliminated from motion fields by plugging
\[
Z \approx pX + qY + r \Rightarrow 1 \approx \frac{pX + qY + r}{Z} = px + qy + r/Z \Rightarrow 1/Z \approx \frac{1 - (px + qy)}{r}
\]
into (3.6)-(3.7). This substitution yields so called Riccati dynamics:

$$\frac{d}{dt} \left( \begin{array}{c} x \\ y \end{array} \right) \approx \left( \begin{array}{c} d_1 \\ d_2 \end{array} \right) + \left( \begin{array}{c} d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} \right) \left( \begin{array}{c} x \\ y \end{array} \right) + \left( \begin{array}{c} d_7x^2 + d_8xy \\ d_9y^2 + d_7xy \end{array} \right)$$

(3.9)

Image dynamics are governed by differential equations with quadratic right hand sides. Previous work on motion estimation suggests that, in practice, the second order terms play a minor role, to the extent that they often end up fitting the noise or deviations from the planar model [1],[33].

We would like to remark that considering an almost planar object, $Z \approx pX + qY + r$, moving in 3-D space according to (3.4) and assuming a weak perspective projection as

$$x = \frac{F}{\bar{Z}}X = \alpha X, \quad y = \frac{F}{\bar{Z}}Y = \alpha Y \quad (3.10)$$

where $\bar{Z}$ is the average scene depth, the resulting image dynamics in terms of $(x, y)$ are obtained as:

$$\left( \begin{array}{c} \dot{x} \\ \dot{y} \end{array} \right) = \alpha \left( \begin{array}{c} \dot{X} \\ \dot{Y} \end{array} \right) = \alpha \left( \begin{array}{c} -\omega_3 y + \omega_2 Z + b_1 \\ \omega_3 X - \omega_1 Z + b_2 \end{array} \right) = \left( \begin{array}{c} -\omega_3 \alpha Y + \omega_2(\alpha pX + \alpha qY + \alpha r) + \alpha b_1 \\ \omega_3 \alpha X - \omega_1(\alpha pX + \alpha qY + \alpha r) + \alpha b_2 \end{array} \right)$$

$$= \left( \begin{array}{c} -\omega_3 + \omega_2q \end{array} \right) y + \omega_2px + (b_1 + \omega r)\alpha \right)$$

(3.11)

which are clearly in the form of affine dynamics given in (3.8), and thus verify the practical importance of 2-D affine motion and related work.

Note also that under the scaled-orthographic projection described in (3.3), a 2D rigid motion

$$\left( \begin{array}{c} \dot{X} \\ \dot{Y} \end{array} \right) = \left( \begin{array}{cc} 0 & -\omega \\ \omega & 0 \end{array} \right) \left( \begin{array}{c} X \\ Y \end{array} \right) + \left( \begin{array}{c} b_1 \\ b_2 \end{array} \right) \quad (3.12)$$

will imply the following motion on the image plane:

$$\left( \begin{array}{c} \dot{x} \\ \dot{y} \end{array} \right) = \frac{f}{d} \left( \begin{array}{c} \dot{X} \\ \dot{Y} \end{array} \right) = \frac{f}{d} \left[ \left( \begin{array}{cc} 0 & -\omega \\ \omega & 0 \end{array} \right) \left( \begin{array}{c} X \\ Y \end{array} \right) + \left( \begin{array}{c} b_1 \\ b_2 \end{array} \right) \right] = \left( \begin{array}{cc} 0 & -\omega \\ \omega & 0 \end{array} \right) \left( \begin{array}{c} fX/d \\ fY/d \end{array} \right) + \frac{f}{d} \left( \begin{array}{c} b_1 \\ b_2 \end{array} \right)$$

$$= \left( \begin{array}{cc} 0 & -\omega \\ \omega & 0 \end{array} \right) \left( \begin{array}{c} x \\ y \end{array} \right) + \left( \begin{array}{c} \bar{b}_1 \\ \bar{b}_2 \end{array} \right) \quad (3.13)$$

where, clearly the only difference is in the translational parameters, thus making the image dynamics also of rigid motion type.

4 Composite Image Feature Dynamics

Previous sections have shown that image dynamics can be rigid, affine or Riccati. Note also that motion and shape parameters enter into these dynamics linearly. In other words, we can rewrite
these image dynamics as
\[ \dot{\xi} = f(\xi) \varphi \] (4.1)
where \( \xi \in \mathbb{R}^2 \) is the image feature vector, i.e. \( \xi = (x, y)^T \), \( f(\xi) \) is a linear (or nonlinear) mapping and \( \varphi \) is the vector of unknown parameters which can be either constant or time-varying. Suppose we have \( n \) parameters to be estimated, i.e. \( \varphi \in \mathbb{R}^n \) and \( f \in \mathbb{R}^{2 \times n} \). From (4.1) it is clear that each image feature provides 2 equations. For the estimation of \( n \) parameters, we will therefore require at least \( m \equiv n/2 \) independent image features. For example, rigid motion (3.12) in 2D is defined by 3 parameters \((\omega, t_1, t_2)\) and we need at least 1.5 image points. In the case of affine motion (3.8), 3 image features imply 6 equations which can be solved for 6 parameters, namely \( a_1, a_2, a_3, a_4, t_1 \) and \( t_2 \). In the case of Riccati dynamics (3.9), we need at least 4 image features since they will imply 8 equations which can be used to solve 8 parameters, \( d_1, d_2, \ldots, d_8 \).

Let us consider \( m \) image features concatenated in the following dynamical system:
\[
\begin{pmatrix}
\dot{\xi}_1 \\
\dot{\xi}_2 \\
\vdots \\
\dot{\xi}_m
\end{pmatrix} =
\begin{pmatrix}
f_1(\xi_1) \\
f_2(\xi_2) \\
\vdots \\
f_m(\xi_m)
\end{pmatrix} \varphi \Rightarrow \dot{\Xi} = \mathcal{F} \varphi
\] (4.2)
where \( \xi_r = (x_r, y_r)^T \) and \( r = 1, 2, \ldots, k \). Note that \( \mathcal{F} \) is now a square matrix with \( m \times m \).

Since the image features are extracted from images, they are measurable and therefore we can also introduce an output (read-out) equation as
\[ Y = \Xi \] (4.3)

5 Sliding Mode Observers

In this section, we construct an observer whose state follows the state of the composite image feature dynamics given in 4.3 as closely as possible. We will employ sliding mode control (SMC) in the design of the observer to stabilize the state estimation error around zero. Basically we are copying our image feature dynamics and try to control this copied version by SMC. More precisely, let our observer be
\[ \dot{\hat{\Xi}} = u \] (5.1)
where \( u \) will be designed using SMC so that \( \hat{\Xi} = \Xi - \hat{\Xi} \to 0 \) as \( t \to \infty \).

Let us define the sliding mode manifold as
\[ \sigma = Y - \hat{\Xi} \]
which, in light of 4.2, then implies that

$$\dot{\sigma} = \dot{Y} - \dot{\hat{X}} = F \varphi - u. \quad (5.2)$$

Let us pick the following Lyapunov function

$$V = \frac{1}{2} \sigma^T \sigma$$

whose time derivative is

$$\dot{V} = \sigma^T \dot{\sigma}$$

which can be made negative definite by setting $\dot{\sigma}$ to either $-M \text{sgn}(\sigma)$, where $M > 0$ and $\text{sgn}(.)$ is the signum function, or $-D \sigma$ (asymptotic approach to $\sigma = 0$ but strictly speaking not sliding mode), where $D$ is a positive definite matrix. If $-M \text{sgn}(\sigma)$ is selected, all components of the control are switching between lower and upper bound of control. This may cause unnecessary chattering in the system especially in the discrete-time implementation of the control algorithm. Combination of the $\dot{\sigma} = -M \text{sgn}(\sigma)$ and $\dot{\sigma} = -D \sigma$ by selecting $\dot{\sigma} = -D \sigma - \rho(x,t) \text{sgn}(\sigma)$ yields a solution that may combine good properties of both solutions and allows selecting $\rho(x,t)$ small enough to minimize chattering and at the same time to guarantee the existence of sliding mode.

So by selecting $\dot{\sigma} = -D \sigma - \rho(x,t) \text{sgn}(\sigma)$, $\dot{V}$ then becomes

$$\dot{V} = -\sigma^T D \sigma - \rho(x,t) \sigma^T \text{sgn}(\sigma) = -\sigma^T D \sigma - \rho(x,t) ||\sigma||,$$

which is clearly negative definite since $D > 0$ and $\rho > 0$. Therefore for stability,

$$\dot{\sigma} = -D \sigma - \rho \text{sgn}(\sigma) \Rightarrow \dot{\sigma} + D \sigma + \rho \text{sgn}(\sigma) = 0 \quad (5.3)$$

must be satisfied.

As shown in [43], $u$ can be computed recursively using the algebraic distance $\dot{\sigma} + D \sigma$, namely

$$u(k) = u(k-1) + (\dot{\sigma} + D \sigma) = u(k-1) + \left\{ \frac{\sigma(kT) - \sigma((k-1)T)}{T} + D \sigma(kT) \right\} \quad (5.4)$$

or,

$$u(k) = u(k-1) + \frac{1}{T} \left\{ (I + TD) \sigma(kT) - \sigma((k-1)T) \right\} \quad (5.5)$$

where $I$ is the identity matrix and $T$ is the sampling time.

In light of (5.3), we can modify the algebraic distance used in (5.4) as $\dot{\sigma} + D \sigma + \rho \text{sgn}(\sigma)$ and compute the control again recursively as:

$$\dot{u}(k) = \dot{u}(k-1) + (\dot{\sigma} + D \sigma + \rho \text{sgn}(\sigma))$$
\[ u(k) = u(k-1) + \frac{1}{T} \{ (I + TD)\sigma(kT) - \sigma((k-1)T) + \rho \text{sgn}(\sigma(kT)) \} \quad (5.6) \]

Since \( F \) in (4.2) is square and invertible, plugging (5.2) and (5.6) in (5.3) implies

\[ F \hat{\phi} - \hat{u} + D\sigma + \rho \text{sgn}(\sigma) = 0 \Rightarrow \hat{\phi} = F^{-1}(\hat{u} - D\sigma - \rho \text{sgn}(\sigma)). \quad (5.7) \]

Note that when \( \dot{\sigma} + D\sigma + \rho \text{sgn}(\sigma) \to 0 \) then \( \hat{u} \to \hat{u}_{eq} \), namely we have the “equivalent control”, \( \hat{u}_{eq} \), on the sliding manifold. In other words, we are estimating the parameters using the equivalent control.

6 Simulation Results

The proposed approach is simulated using Matlab7 and Simulink 6.0. FR denotes the frame rate of the camera in fps (frame per second). Since image data is available in every 1/FR seconds (which is larger than simulation sample time), we constructed a filter that interpolates the state values between two consecutive frames. We considered two cases: high frame rate (200 fps) and low frame rate (30 fps). At 30 fps, we also needed a filter to get non oscillatory results. To simulate the behavior of the camera, we constructed motion dynamics subsystem in Simulink, which generates state values, i.e. image coordinates of the object, at prescribed frame rates. This is achieved using zero-order hold at the output port of the motion dynamics subsystem with a sample time equal to 1/FR. Interpolator used is the well known \( G(s) = \frac{w_n^2}{s^2 + 2\varsigma w_n s + w_n^2} \) with \( w_n = 50, \varsigma = 1 \), which creates smooth values of the states between two consecutive frames. Sample time of the simulations is 0.0001 sec. Run time is 10 sec. Controller employed is exactly the one given by eqn (5.6), with \( D = 0.1 \) and \( \rho = 0.0001 \). Estimated parameters are filtered at low frame rate simulations, with a first order filter of time constant \( \tau = 0.05 \).

6.1 Rigid Motion Estimation

Rigid motion with time varying parameters was simulated with 200 fps. As time varying functions, step, sine and square wave were used for the parameters. Motion trajectory of the object is depicted in Fig. 2. Actual and estimated values are superimposed in Fig.3-5.

There are two major differences between actual and estimated values. First we have a very small time delay; estimated values follow the actual ones with a small lag. This is due to the second order interpolator we use, since the interpolator needs previous and 2nd previous data to
generate the current data. Also in the case of $b_1$ and $\omega$, we don’t see any sharp corners as in the actual ones, because interpolater kills these corners and produces a smoother response.

As can be seen from plotted graphs, our algorithm works fine, and fast estimation of the parameters with acceptable accuracy is achieved. Note that parameters converge to their actual values as soon as the sliding manifold is reached.

Simulation with 30 fps was also carried out. Low frame rate causes oscillations, so we needed to filter the results. Motion trajectory of the object is depicted in Fig. 6. Actual and estimated parameter values are superimposed in Fig.7-9.

### 6.2 Affine Motion Estimation

Time varying parameter affine motion was simulated with 200 fps. Motion trajectory of the object is depicted in Fig. 10. Actual and estimated values are superimposed in Fig.11-16.

Simulation with 30 fps was carried out next. Again low frame rate introduces oscillations, so we needed to filter the results. Actual and estimated values are superimposed in Fig.17-22.

### 7 Experimental Results

#### 7.1 Experimental Setup of Vision System

Some initial real time experiments were conducted with the vision setup shown in Fig. 23. The system consists of Nikon SMZ 1500 Stereo Optical Microscope and a Firewire CCD camera with 30 fps on top of it. What is observed as a motion is created by NanoCube, which can provide nanometer range accuracy motion and positioning in XYZ directions. The actual control of Nanocube is done through dSPACE 1103 board and its control software. The points of interest to be tracked are chosen from micrometer, on which has a micron range checker board pattern, on top of NanoCube using some image processing operations in the OpenCV library. Each (1/30) second, image is taken, processed, and the points of interest are extracted from the image. To synchronize the sample time of capturing images and the sample time of the control, the same interpolation technique is used as in simulations.
7.2 Results for Rigid Motion

7.2.1 Linear Motion along x-axis

In this case NanoCube moves along x-axis with 6 \( \mu \) m/s for a while and stops, and then turns back to opposite direction again with a 6 \( \mu \) m/s. Since there is no motion along y-axis, the estimated linear velocity along y-axis is around zero. Furthermore, since there is no rotational motion, the estimated angular velocity is again around zero. (See Fig. 24-26.)

7.2.2 Pure Rotational Motion

In this case NanoCube undergoes a pure rotational motion with a constant angular velocity, \( \pi/5 \) rad/s in counterclockwise. The radius of the circular motion is 10 \( \mu \)m. Actual and estimated angular velocities are superimposed in Fig. 27.

Although, the reference motion is pure rotation, the angular velocity becomes very large for small time interval at very beginning. Then it converges to reference value very quickly. At the same time, the estimated trajectory seems to undergo linear motion (See Fig. 28). The reason is that the initial points, which are given as initial conditions for estimation, are different from the actual points on the circle. Therefore estimated points show some linear behavior until the dynamic of the system reaches the sliding manifold. When the system is on the sliding manifold, estimated values converge to the actual values and hence the motion becomes circular.

8 Conclusions

We have developed a new framework to solve motion and structure estimation problems in machine vision using sliding mode observers. We have shown that most of the dynamics encountered in machine vision applications can be recast as dynamical systems which are linear in terms of parameters to be estimated. We then constructed an appropriate sliding observer whose output follows the output of the original dynamics asymptotically. Control which drives the observer has been computed recursively based on the algebraic distance \( \dot{\sigma} + D\sigma + \rho Sgn(\sigma) \).

Simulation and initial experimental results are promising and much has to be done to assess the performance of our proposed algorithm with various data perturbations.
References


14


Figure 1: Rigid motion of an object on a conveyor belt under a scaled orthographic camera

Figure 2: Motion Trajectory
Figure 3: $\omega$ (dashed) and $\hat{\omega}$ (solid)

Figure 4: $b_1$ (dashed) and $\hat{b}_1$ (solid)

Figure 5: $b_2$ (dashed) and $\hat{b}_2$ (solid)

Figure 6: Motion Trajectory

Figure 7: $\omega$ (dashed) and $\hat{\omega}$ (solid)
Figure 8: $b_1$ (dashed) and $\hat{b}_1$ (solid)

Figure 9: $b_2$ (dashed) and $\hat{b}_2$ (solid)

Figure 10: Motion Trajectory

Figure 11: $a_1$ (dashed) and $\hat{a}_1$ (solid)

Figure 12: $a_2$ (dashed) and $\hat{a}_2$ (solid)

Figure 13: $a_3$ (dashed) and $\hat{a}_3$ (solid)
Figure 14: \( a_4 \) (dashed) and \( \hat{a}_4 \) (solid)

Figure 15: \( b_1 \) (dashed) and \( \hat{b}_1 \) (solid)

Figure 16: \( b_2 \) (dashed) and \( \hat{b}_2 \) (solid)

Figure 17: \( a_1 \) (dashed) and \( \hat{a}_1 \) (solid)

Figure 18: \( a_2 \) (dashed) and \( \hat{a}_2 \) (solid)
Figure 19: $a_3$ (dashed) and $\hat{a}_3$ (solid)

Figure 20: $a_4$ (dashed) and $\hat{a}_4$ (solid)

Figure 21: $b_1$ (dashed) and $\hat{b}_1$ (solid)

Figure 22: $b_2$ (dashed) and $\hat{b}_2$ (solid)
Figure 23: Experimental Vision Setup

Figure 24: Actual (dashed) and Estimated (solid) linear velocity, $b_1$ along $x$ axis

Figure 25: Actual (dashed) and Estimated (solid) linear velocity, $b_2$ along $y$ axis
Figure 26: Estimated angular velocity, $\omega$

Figure 27: Actual (dashed) and Estimated (solid) angular velocity, $\omega$

Figure 28: Estimated Motion Trajectory