Kalman Filter[1]

Lecture 26

See Section 9.3.4 in
Reinhard Klette: Concise Computer Vision
Springer-Verlag, London, 2014

[1] See last slide for copyright information.

Agenda

Motivation

Control of noisy systems: "Noisy data in, and, hopefully, less noisy data out."

Applications of Kalman filters:

1. tracking objects (e.g., balls, faces, heads, hands)
2. fitting Bezier patches to point data
3. economics
4. navigation
5. ...
6. many computer vision applications (e.g. stabilizing depth measurements, feature tracking, cluster tracking, fusing data from radar, laser scanner, and stereo-cameras)

Agenda

Continuous Equation of a Linear Dynamic System

A continuous *linear dynamic system* is defined by

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

nD vector $\mathbf{x} \in \mathbb{R}^n$: specifies the *state* of the process

$\mathbf{A}$ is the constant $n \times n$ *system matrix*

Notation $\dot{\mathbf{x}}$ is short for the derivative of $\mathbf{x}$ with respect to time t

Signs and magnitudes of the roots of the eigenvalues of $\mathbf{A}$ determine the *stability* of the dynamic system

Observability and *controllability* are further properties of dynamic systems

Example

Moving Object with Constant Acceleration

Video camera captures an object moving along a straight line

Object's centroid is described by coordinate x on this line and

its motion by speed v and *constant* acceleration a

Process state $\mathbf{x} = [x, v, a]^\top$; thus $\dot{\mathbf{x}} = [v, a, 0]^\top$ and

$$\dot{\mathbf{x}} = \begin{bmatrix} v \\ a \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ v \\ a \end{bmatrix}$$

Eigenvalues of 3×3 system matrix $\mathbf{A}$:

$$\det(\mathbf{A} - \lambda \mathbf{I}) = -\lambda^3$$

specifies identical eigenvalues $\lambda_{1,2,3} = 0$; system is "very stable"

Discrete Equations of a Linear Dynamic System

Continuous system on Page 5 mapped into a time-discrete system

Δt is the time difference between time slots t and $t+1$

For Euler number e, for any argument x:

$$e^x = 1 + \sum_{i=1}^{\infty} \frac{x^i}{i!}$$

The *state transition matrix* for Δt equals

$$\mathbf{F}_{\Delta t} = e^{\Delta t \mathbf{A}} = \mathbf{I} + \sum_{i=1}^{\infty} \frac{\Delta t^i \mathbf{A}^i}{i!}$$

with an $i_0 > 0$ such that $\mathbf{A}^i$ is zero everywhere, for all $i \geq i_0$

Equation defines a finite sum for a discrete system (we leave out Δt):

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1}$$

Initial state $\mathbf{x}_0$ at time slot $t = 0$

Discrete Linear System with Control and Noise

Consider noise and system control; the previous equation is replaced by

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \\ \mathbf{y}_t &= \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \end{aligned}$$

with a *control matrix* $\mathbf{B}$, applied to a *control vector* $\mathbf{u}_t$, *system noise* vectors $\mathbf{w}_t$, *observation matrix* $\mathbf{H}$, noisy *observations* $\mathbf{y}_t$, and *observation noise* vectors $\mathbf{v}_t$

System noise and observation noise vectors are assumed to be mutually independent

Control defines some type of system influence at time t which is not inherent to the process itself

Example

Continuation: Moving Object with Constant Acceleration

System vectors $\mathbf{x}_t = [x_t, v_t, a_t]^\top$, with $a_t = a$

State transition matrix $\mathbf{F}$ is defined by

$$\mathbf{x}_{t+1} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{x}_t = \begin{bmatrix} x_t + \Delta t \cdot v_t + \frac{1}{2}\Delta t^2 a \\ v_t + \Delta t \cdot a \\ a \end{bmatrix}$$

Verify this by applying the equation given above for $\mathbf{F}_{\Delta t}$.

Observation Matrix for this Example

We only observe the current location $\mathbf{y}_t = [x_t, 0, 0]^\top$

This defines observation matrix **H** as used in the following equation:

$$\mathbf{y}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{x}_t$$

Noise vectors $\mathbf{w}_t$ and $\mathbf{v}_t$ would be zero vectors under ideal assumptions

Control vector and control matrix are not used in the example

Agenda

Time-Discrete Prediction

Given: sequence $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{t-1}$ of noisy observations for a linear dynamic system

Goal: estimate internal state $\mathbf{x}_t = [x_{1,t}, x_{2,t}, \ldots, x_{n,t}]^\top$, which is of the system at time slot t

Minimize the estimation error

$\hat{\mathbf{x}}_{t_1|t_2}$ is the *estimate* of state $\mathbf{x}_{t_1}$ based on knowledge available at t_2

$\mathbf{P}_{t_1|t_2}$ is the variance matrix of the *prediction error* $\mathbf{x}_{t_1} - \hat{\mathbf{x}}_{t_1|t_2}$

Goal: minimize $\mathbf{P}_{t|t}$ in some defined way

Available Knowledge at Time of Prediction

Available knowledge at time t:

1. Estimate of state transition matrix $\mathbf{F}$ which is applied to the ("fairly known") previous state $\mathbf{x}_{t-1}$
2. Control matrix $\mathbf{B}$ which is applied to control vector $\mathbf{u}_t$, if there is a control mechanism at all in the system
3. Understanding about system noise $\mathbf{w}_t$ (e.g. modeled as a multivariate Gaussian distribution) by specifying a variance matrix $\mathbf{Q}_t$ and expected values $\mu_{i,t} = E[w_{i,t}] = 0$, for $i = 1, 2, \ldots, n$
4. Observation vector $\mathbf{y}_t$ for state $\mathbf{x}_t$
5. Observation matrix $\mathbf{H}$ ("how to observe $\mathbf{y}_t$"?)
6. Understanding about observation noise $\mathbf{v}_t$ (e.g. modeled as a multivariate Gaussian distribution) by specifying a variance matrix $\mathbf{R}_t$ and expected values $\mu_{i,t} = E[v_{i,t}] = 0$, for $i = 1, 2, \ldots, n$

Prediction and Filter

Key idea: not just one prediction after the other by applying available knowledge; we define a *filter* which aims at updating our knowledge about the system noise, based on experienced prediction errors and observations so far, and we want to use the improved knowledge about the system noise for reducing the prediction error

Basic issues, such as assuming an incorrect state transition matrix or an incorrect control matrix, are *not* solved by the filter

Predict Phase of the Filter = first phase of the filter

Calculate the predicted state and predicted variance matrix, using assumed state transition matrix $\mathbf{F}$ and control matrix $\mathbf{B}$; also apply the system noise variance matrix $\mathbf{Q}_t$:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}\mathbf{u}_t \\ \mathbf{P}_{t|t-1} &= \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q}_t \end{aligned}$$

Update Phase of the Filter

= second phase of the filter

Calculate the *measurement residual vector* $\tilde{\mathbf{z}}_t$ and the *residual variance matrix* $\mathbf{S}_t$:

$$\begin{aligned} \tilde{\mathbf{z}}_t &= \mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1} \\ \mathbf{S}_t &= \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R}_t \end{aligned}$$

using observation matrix $\mathbf{H}$ of the assumed model and observation noise variance matrix $\mathbf{R}_t$.

We aim at improving these noise matrices

Updated state-estimation vector (i.e., prediction at time t) by an *innovation step* of the *filter* at time t:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{z}}_t$$

Goal: matrix $\mathbf{K}_t$ such that innovation step is optimal

Agenda

History

R. E. Kalman (born 1930 in Hungary) defined and published in [R. E. Kalman. A new approach to linear filtering and prediction problems. J. Basic Engineering, volume 82, pages 35–45, 1960] a recursive solution to the linear filtering problem for discrete signals, today known as the *linear Kalman filter*

Related ideas were also studied at that time by the US-American radar theoretician P. Swerling (1929 – 2000)

The Danish astronomer T. N. Thiele (1838 – 1910) is also cited for historic origins of involved ideas

Apollo 8 (December 1968), the first human spaceflight from Earth to an orbit around the moon, would certainly not have been possible without the linear Kalman filter

Optimal Kalman Gain

Matrix

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^\top \mathbf{S}_t^{-1}$$

minimizes the mean square error $E[(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^2]$, which is equivalent to minimizing the trace ($=$ sum of elements on the main diagonal) of $\mathbf{P}_{t|t}$

Matrix $\mathbf{K}_t$ is known as the *optimal Kalman gain*; it defines the *linear Kalman filter*

Filter also requires an updated variance matrix

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}$$

of the system noise for predict phase at time $t+1$

$\mathbf{P}_{0|0}$ needs to be initialized at the begin of the filter process

Example

Continuation: Moving Object now with Random Acceleration

The object is still assumed to move along a straight line

Now with *random* acceleration a_t between $t-1$ and time t

For modeling randomness, we assume a Gauss distribution with zero mean and variance σ_a^2; measurements of positions of the object are assumed to be noisy; again we assume Gaussian noise with zero mean and variance σ_y^2

State vector given by $\mathbf{x}_t = [x_t, \dot{x}_t]^\top$ where $\dot{x}_t$ equals the speed v_t

We have that

$$\mathbf{x}_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ v_{t-1} \end{bmatrix} + a_t \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t$$

with variance matrix $\mathbf{Q}_t = \mathrm{var}(\mathbf{w}_t)$

Example Continued

Using $\mathbf{G}_t = [\frac{\Delta t^2}{2}, \Delta t]^\top$ we have that

$$\mathbf{Q}_t = E[\mathbf{w}_t \mathbf{w}_t^\top] = \mathbf{G}_t E[a_t^2] \mathbf{G}_t^\top = \sigma_a^2 \mathbf{G}_t \mathbf{G}_t^\top = \sigma_a^2 \left[\begin{array}{cc} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{array} \right]$$

$\mathbf{Q}_t$ and $\mathbf{G}_t$ are also independent of t, thus just denoted by $\mathbf{Q}$ and $\mathbf{G}$

At time t we measure the position of the object:

$$\mathbf{y}_t = \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \end{array} \right] \mathbf{x}_t + \left[\begin{array}{c} v_t \\ 0 \end{array} \right] = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

Observation noise $\mathbf{v}_t$ has the variance matrix

$$\mathbf{R} = E[\mathbf{v_t} \mathbf{v}_t^\top] = \left[\begin{array}{cc} \sigma_y^2 & 0 \\ 0 & 0 \end{array} \right]$$

Example Continued

If initial position $\hat{\mathbf{x}}_{0|0} = [0,0]^\top$ accurately known then use matrix

$$\mathbf{P}_{0|0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

otherwise (with a suitably large real $c > 0$)

$$\mathbf{P}_{0|0} = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix}$$

$t = 1$: Predict $\hat{\mathbf{x}}_{1|0}$ and calculate variance matrix $\mathbf{P}_{1|0}$ by

$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} &= \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}^\top + \mathbf{Q} \end{aligned}$$

Calculate auxiliary data $\tilde{\mathbf{z}}_1$ and $\mathbf{S}_1$ by update equations

$$\begin{aligned} \tilde{\mathbf{z}}_t &= \mathbf{y}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1} \\ \mathbf{S}_t &= \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R} \end{aligned}$$

Example Continued

Calculate the optimal Kalman gain $\mathbf{K}_1$ and update $\hat{\mathbf{x}}_{1|1}$:

$$\begin{aligned} \mathbf{K}_t &= \mathbf{P}_{t|t-1}\mathbf{H}^\top \mathbf{S}_t^{-1} \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \tilde{\mathbf{z}}_t \end{aligned}$$

Calculate $\mathbf{P}_{1|1}$ to prepare for $t = 2$:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}$$

Those calculations are basic matrix or vector algebra operations, easy to implement, but numerically already rather complex

Agenda

Tuning the Kalman Filter

Specifications of variance matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$, or of constant $c \geq 0$ in $\mathbf{P}_{0|0}$, influences the number of time steps of the Kalman filter such that the predicted states converge to true states

Assuming a higher uncertainty (i.e., larger $c \geq 0$, or larger values in $\mathbf{Q}_t$ and $\mathbf{R}_t$), increases values in $\mathbf{P_{t|t-1}}$ or $\mathbf{S}_t$; due to the use of the inverse $\mathbf{S}_t^{-1}$ in the definition of the optimal Kalman gain, this decreases values in $\mathbf{K}_t$ and the contribution of the measurement residual vector in the update equation

If we are totally sure about the correctness of the initial state $\mathbf{z}_{0|0}$ (i.e., $c = 0$), and that we do not have to assume any noise in the system and in the measurement processes, then matrices $\mathbf{P}_{t|t-1}$ and $\mathbf{S}_t$ degenerate to zero matrices; the inverse $\mathbf{S}_t^{-1}$ does not exist, and $\mathbf{K}_t$ remains undefined: The predicted state is equal to the updated state; this is the fastest possible convergence of the filter

Alternative Model for Predict Phase

An estimate of the continuous model matrix $\mathbf{A}$ in $\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$ supports the use of equations

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_{\mathbf{t}|\mathbf{t}-\mathbf{1}} &= \mathbf{A}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t \mathbf{u}_t \\ \mathbf{P}_{t|t-1} &= \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^\top + \mathbf{Q}_t \end{aligned}$$

and defines a modified matrix $\mathbf{B}$, now for the impact of control on the derivatives of state vectors

This modification in the predict phase does not have formal consequence on the update phase

Agenda

1. Understanding the Situation

We go through the sequence of recommended steps when implementing a Kalman-filter-based solution

Task: Design a Kalman filter for improving disparities calculated when operating a stereo-vision system in a driving vehicle in a static environment

1. *Understanding the situation*

We assume an ego-vehicle driving in a static environment

Due to ego-motion, we experience some changes in disparities

Assume that every pixel is independent; we set up *iconic* Kalman filters
    i.e., one Kalman filter at a pixel location p of the image

2. Model the State Process

Disparity is constant in a totally static world (i.e. if also no ego-motion):
We have a scalar system state $x_t = d_t = d_{t-1}$ and $\mathbf{F}_{\Delta t} = 1$

But: We have a moving platform

This disparity (i.e. the state) *will* change when the car is moving

The $p = (x, y)$ pixel position will also change where we follow the changing disparity

The car moves forward: corresponding pixels move outward from the *focus of expansion*:

in human vision the retinal point where lines defined by translatory motion meet, also assuming a corresponding direction of gaze

Focus of Expansion

Overlay of two subsequently recorded frames. Five pairs of corresponding points define translatory motion; extended into straight lines those meet at the focus of expansion. The arrow illustrates outward motion. The length of the arrow depends on the depth (or disparity) of the tracked surface point.

Use of Control Variables

This is where we can use our control variables $\mathbf{B}$ and $\mathbf{u}$

State (i.e. disparity) at time t defined by a disparity at time $t-1$
at a pixel sliding along the line defined by translatory motion

Assume that ego-motion is given (e.g.) by inertial sensors
(providing velocity v and yaw $\psi =$ amount of angle turned through)
This helps to derive a pixel's translatory motion

Control Variables: We use $\mathbf{B}$ and $\mathbf{u}$
We know the vehicle movement in real-world coordinates

Triangulate and backproject
pixel coordinates $p = (x, y, d)$ plus disparity,
and real-world coordinates $\mathbf{P} = [X, Y, Z]^\top$ (in vector format)
w.r.t. the ego-vehicle

Specify Control Parameters

For each measurement, we apply the following process:

1. Transform coordinates (x_{t-1}, y_{t-1}) at time $t-1$ into real-world coordinates $\mathbf{P}_{t-1}$, as being a standard in stereo vision
2. Predict the new position of $\mathbf{P}_{t-1}$ at time t in real-world coordinates using velocity v during Δt and total yaw ψ at time t:

$$\mathbf{R}_t = \begin{bmatrix} \cos(\psi) & 0 & -\sin(\psi) \\ 0 & 1 & 0 \\ \sin(\psi) & 0 & \cos(\psi) \end{bmatrix} \quad \text{and} \quad \mathbf{T}_t = \frac{v \cdot \Delta t}{\psi} \begin{bmatrix} 1 - \cos(\psi) \\ 0 \\ -\sin(\psi) \end{bmatrix}$$

3. Transform the new real-world coordinates $\mathbf{P}_t = \mathbf{R}_t \mathbf{P}_{t-1} + \mathbf{T}_t$ back to pixel coordinates (x_t, y_t) using backprojection

Here, $\mathbf{R}_t$ is the rotation matrix in the XZ-plane, due to the total yaw, and $\mathbf{T}_t$ is the translation matrix between times t and $t-1$

Summary for Control Parameters

Starting at pixel (x, y) and disparity d at time $t-1$

this provides an estimated disparity d' at a pixel (x', y') at time t

identified with being the value of

$$\mathbf{F}_{\Delta t}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t$$

at (x', y'), where $\mathbf{u}_t$ is defined by

> yaw rate $\dot{\psi}(t)$ between $t-1$ and t,
> and velocity v_t during Δt

and matrix $\mathbf{B}$ is defined by

> projection at time $t-1$, affine transform, and
> backprojection at time t

This may be implemented pixel by pixel, or for the whole image at once

3. Model the Measurement Process

We are filtering our measurements directly (i.e. calculating the disparity)

Therefore, for the individual pixel:

$$\mathbf{y} = y \quad \text{and} \quad \mathbf{H} = 1$$

4. Model the Noise

Disparity measurements have a Gaussian noise distribution in depth direction (i.e., for sub-pixel measurements), and these can fluctuate to either side (i.e. within an ellipsoidal region)

The main state error is in the depth direction (i.e. the above ellipsoid is elongated into depth direction); thus we assume that

$$\mathbf{P} = P = \sigma_d^2$$

For our model, both the process and measurement noise (at a single pixel) are scalars; therefore,

$$\mathbf{Q} = q_d \quad \text{and} \quad \mathbf{R} = r_d$$

(we could also assume that these values change between each iteration t)

5. Test the Filter

The equations simplify at a single pixel as follows:

Predict:

$$
\begin{aligned}
\hat{x}_{t|t-1} &= \text{as derived above using } \mathbf{u}_t \\
P_{t|t-1} &= P_{t-1|t-1} + q_d
\end{aligned}
$$

Update:

$$
\begin{aligned}
\hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t \left(y_t - \hat{x}_{t|t-1} \right) \\
K_t &= P_{t|t-1} \left(P_{t|t-1} + r_d \right)^{-1} \\
P_{t|t} &= \left(1 - K_t \right) P_{t|t-1}
\end{aligned}
$$

6. Choose Logical Noise Parameters

For example, for measurement noise we may take $r_d = 1$, considering to be up to 1 pixel out in our measurement

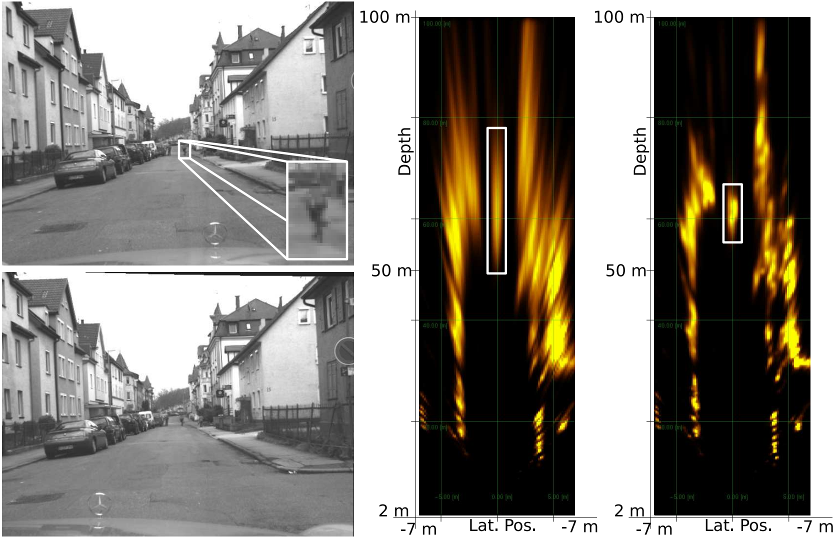
If we want to filter out all moving objects, then a logical process parameter is $q_d = 0.0001$ (i.e., some small value portraying that we assume the model is good)

This ends the description of an iconic Kalman filter approach
for disparity calculations

For testing you need stereo video data recorded in a driving car,
with known ego-motion parameters of the car

For example, check KITTI and EISATS for such data

An Illustration of a Possible Improvement



Caption to Figure on Page Before

The left images show original images of a cyclist on the road in front of the ego-vehicle at times $t - 1$ and t (left image of a stereo pair in both cases)

The middle and right hand grids show the birds-eye views of depth maps (i.e. object disparities projected into 3D coordinates), also known as *occupancy grids*

The left-hand grid shows the results using no Kalman integration; the right-hand grid shows results using the iconic filters and illustrates a "sharpening" in detected object locations

Because we assumed a static world we can expect that there will be errors on moving objects such as for the cyclist in this figure

Copyright Information