Kalman Filtering applied to Truck Tracking

Image Processing and Interpretation Center, Steve Mills and Tony Pridmore, The University of Nottingham
http://www.cs.nott.ac.uk/~tpp/G5BVIS/lectures.html

Application - Traffic Tracking

We want to track vehicles on a road

- Eg: The truck in the images to the left
- They are moving with a (fairly) constant velocity
- In each frame we can measure the position of a feature on the vehicle we want to track

Image Processing and Interpretation at The University of Nottingham

track lower left corner of truck.

State Update Equation

- We assume the truck is moving with a constant velocity
- Our state is the truck position (x,y) and velocity (u,v) $s = \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix}$
- At each time the velocity adds on to the position

time between steps

$$x_t = x_{t-1} + u_{t-1} \cdot 1$$
$$y_t = y_{t-1} + v_{t-1} \cdot 1$$
$$u_t = u_{t-1}$$
$$v_t = v_{t-1}$$
$$\begin{bmatrix} x_t \\ y_t \\ u_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ u_{t-1} \\ v_{t-1} \end{bmatrix}$$
$$s_t = As_{t-1}$$ ← state update equation

Measurement Equation

- At each time we can detect features in the image
- These make our measurements, m_t
- We can directly measure the position of the truck, but not the velocity

$$m_t = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ u_{t-1} \\ v_{t-1} \end{bmatrix}$$

$m_t = Hs_t$ measurement equation.

An Initial Estimate

- The initial estimate of the state
  - We give a rough value of x and y to say which feature we are tracking
  - We probably won't have any idea about u and v
  - So we will use

$$s_0 = \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix}$$

initial estimate of corner

- We also need to give the (un) certainty
  - Our estimate of the position is good to within a few pixels
  - Our motion estimate is not good, but we expect the motion to be small
  - We represent this as a covariance matrix

Covariance Matrices

- So what is a covariance matrix?
  - It gives the relationships between sets of variables
  - The variance of a variable, *x*, is
    $var(x) = E((x-x)^2)$
  - The covariance of two variables, *x* and *y*, is
    $cov(x,y) = E((x-\bar{x})(y-y))$
- Given a vector of variables
  $x=[x1,x2,\ldots,xk]$
  - The covariance, *C*, is a $k \times k$ matrix
  - The i,j^{th} entry of *C* is:
    $C_{i,j}=cov(x,y)$
  - A diagonal entry, $C_{i,i}$, gives the variance in the variable x_i
  - *C* is symmetric

$$
\begin{array}{c|c}
 & \begin{array}{cccc} x & y & u & v \end{array} \\
\hline
\begin{array}{c} x \\ y \\ u \\ v \end{array} & \begin{bmatrix} C_{xx} & C_{xy} & & \\ & & & \\ & & & \\ & & & C_{vv} \end{bmatrix}
\end{array}
$$

for example, $C_{xy} = E\left[(x-\bar{x})(y-\bar{y})\right]$.

which must be calculated for a set of measurements (data)

Covariance in Noise

- The noise terms v (the measurement noise or error) and w (the process noise) need to be estimated.
  - They have zero mean, and covariance matrices R and Q respectively.
  - We need an estimate of these matrices. Q and R say how certain we are about our model equations.
- To estimate Q (the process noise)
  - Our initial estimate will be within a few pixels, say $\sigma = 3$
  - The velocity is a bit less certain, but won't be large, say $\sigma=5$
  - There is no reason to think that the errors are related, so the covariance terms will be zero

W - process noise
white

V - measurement error

$Q = \begin{bmatrix} \quad \end{bmatrix}$ 4x4 since four state variables

$R = \begin{bmatrix} \quad \end{bmatrix}$ 2x2 since only two measurements

a priori Estimate Covariance Matrix

- The variances of x and y are $3^2=9$
- The variances of u and v are $5^2=25$

these are assumptions

- Since we assume independence the off-diagonal entries are all 0

$$P_0 = \begin{array}{c} \\ x \\ y \\ u \\ v \end{array} \begin{array}{c} \begin{array}{cccc} x & y & u & v \end{array} \\ \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix} \end{array}$$

initial variance of the estimates

Uncertainty in the model

- Our model equations have noise terms
  - *v* represents the fact that our state update model may not be accurate
  - *w* represents the fact that measurements will always be noisy
  - We need to estimate their covariances
- In general
  - Often the terms will be independent. If this is the case the off-diagonal entries will be zero
  - Choosing the diagonal entries (varainces) is often more difficult

Process Noise Covariance Q

- The state update equation is not perfect
  - It assumes that the motion is constant but *u* and *v* might change over time
  - It assumes that all the motion is represented by *u* and *v* but other factors might affect *x* and *y*
- These errors will probably be small
  - The motion is slow and quite smooth
  - So the variance in these terms is probably a pixel or less, say $\sigma = \frac{1}{2}$

$$Q = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix}$$

where we assummed the same variance $\sigma = \frac{1}{2}$ pixel, for all variables.

Measurement Error Covariance R

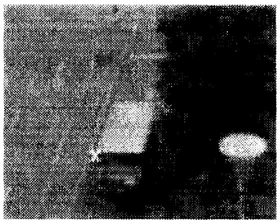
- The measurements we make will be noisy
  - The features are located only to the nearest pixel
  - Because of image noise, aliasing, etc, they might be off by a pixel or so
- These errors are a bit easier to estimate
  - The feature is probably in the right place, or a pixel off
  - So the variance in these terms is probably $\sigma^2 = 1$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

1. Predict the State

A matrix

- We can now run the filter
  - First we make a prediction of the state at t=1 based on our initial estimate at t=0



$$s_1^- = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix}$$

initial position

$$s_1^- = \begin{bmatrix} 100 \\ 170 \\ 0 \\ 0 \end{bmatrix}$$

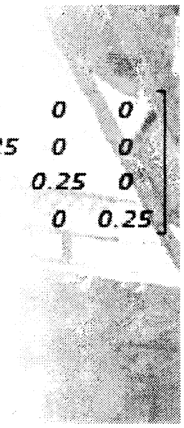
prediction of state
i.e. the next point
this is where we will look
for next state

2. Update the *a priori* Prediction Covariance

$$P_1^- = AP_0A^T + Q$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix}$$

$$= \begin{bmatrix} 34.25 & 0 & 25 & 0 \\ 0 & 34.35 & 0 & 25 \\ 25 & 0 & 35.25 & 0 \\ 0 & 25 & 0 & 25.25 \end{bmatrix}$$



Now update the estimate (prediction) covariance

a priori – before measurement

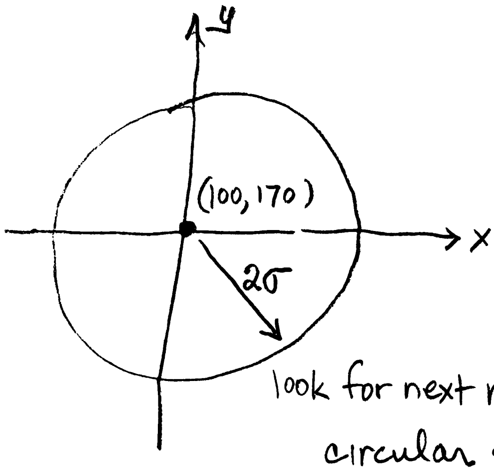
This tells us where and how far to look.

3a. Look for the Next Point

- The state prediction gives us a guide to where the feature will be
  - We expect it to be near (100,170)
  - The variance in the x position is 34.25
  - The variance in the y position is 34.25 also
- We can use this to restrict our search for a feature
  - We are 95% certain that the feature lies in a circle of radius 2σ of the prediction

$$\sigma = \sqrt{34.25} \approx 5.85$$

  - We look for a feature in this region



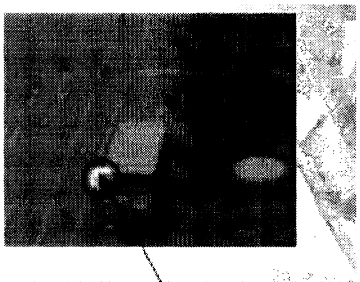
look for next measurement of feature in this circle

circular since $\sigma_x^2 = \sigma_y^2 = 34.25$

find the corner using image processing

3b. Making the Actual Measurement

- Within the search region
- We compute a value that tells us how likely each point is to be a feature (Harris interest operator)
- We find the point with the largest value within this region
- This is $m_1 = \begin{bmatrix} 103 \\ 163 \end{bmatrix}$



We look for a feature near our predicted value, and the covariances tell us how widely to search

4. Compute the Kalman Gain

- We now combine the prediction and measurement
  - We compute the Kalman gain matrix
  - This takes into account the relative certainty of the two pieces of information

$$K_1 = P_1^- H^T \left(H P_1^- H^T + R \right)^{-1}$$

$$\approx \begin{bmatrix} 0.972 & 0 \\ 0 & 0.972 \\ 0.709 & 0 \\ 0 & 0.709 \end{bmatrix}$$

- The first components are close to 1, which will give more trust to the measurement

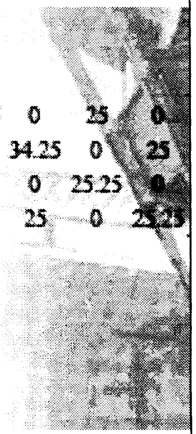
This begins the measurement update ("correct")

5. Update the *a posteriori* error Covariance

$$P_1 = P_1^- + K_1 H P_1^-$$

$$\approx \begin{bmatrix} 34.25 & 0 & 25 & 0 \\ 0 & 34.25 & 0 & 25 \\ 25 & 0 & 25.25 & 0 \\ 0 & 25 & 0 & 25.25 \end{bmatrix} + \begin{bmatrix} 0.972 & 0 \\ 0 & 0.972 \\ 0.709 & 0 \\ 0 & 0.709 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 34.25 & 0 & 25 & 0 \\ 0 & 34.25 & 0 & 25 \\ 25 & 0 & 25.25 & 0 \\ 0 & 25 & 0 & 25.25 \end{bmatrix}$$

$$\approx \begin{bmatrix} 0.971 & 0 & 0.71 & 0 \\ 0 & 0.971 & 0 & 0.71 \\ 0.71 & 0 & 7.52 & 0 \\ 0 & 0.71 & 0 & 7.52 \end{bmatrix}$$



This is the covariance including the measurement just made.

6. Update estimate with measurement m_k

- The new (a posteriori) state estimate based upon measurement m_k is then

$$\hat{s}_k = \hat{s}_k^- + K_k\left(m_k - H\hat{s}_k^-\right)$$

Iteration

- We compute the next state

$$\hat{s}_k^- = \hat{s}_{k-1} + Bu_{k-1}$$

- And project the error covariance ahead.

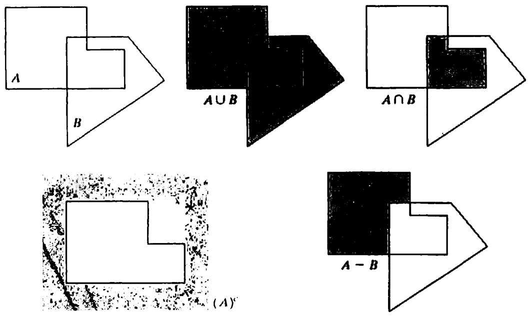
$$P_k^- = AP_{k-1}A^T + Q$$

This tells us where to look next.

Iteration

- We repeat this cycle for each frame
- Over time the state predictions become more accurate
- The Kalman gain takes this into account and places more weight on the predictions
- To implement the Kalman filter
- We need a lot of matrix subroutines
- These are tiresome to code by hand, but there are several libraries available
- Only need basic operations: +, -,x, transpose, and inverse

Chapter 9
Morphological Image Processing



a b c
d e

FIGURE 9.1 (a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

$A^c = \{w \mid w \notin A\}$ $\qquad$ $A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$

morphology — mathematical morphology uses set theory to extract and process image components such as boundaries, skeletons, etc.
The outputs are now attributes rather than a conventional image.

Z^2 set of binary images specified by (x,y) locations

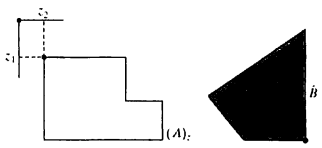
$$C = \{w \mid w = -d, \text{ for } d \in D\}$$

C is the set of elements w, such that w is formed by multiplying each of the two coordinates of all the elements of set D by −1

Digital Image Processing, 2nd ed.

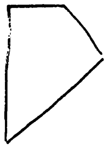
www.imageprocessingbook.com

Chapter 9
Morphological Image Processing



a b

FIGURE 9.2 (a) Translation of A by z. (b) Reflection of B. The sets A and B are from Fig. 9.1.



original B

Image morphology uses two definitions not normally used in set theory.

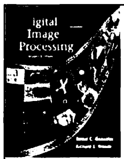
$$\hat{B} = \{ w \mid w = -b, \text{ for } b \in B \}$$

This is the reflection of B about the origin

$$(A)_z = \{ c \mid c = a + z, \text{ for } a \in A \}$$

translate coordinates by $z = (z_1, z_2)$

This is the translation of B.



Chapter 9
Morphological Image Processing

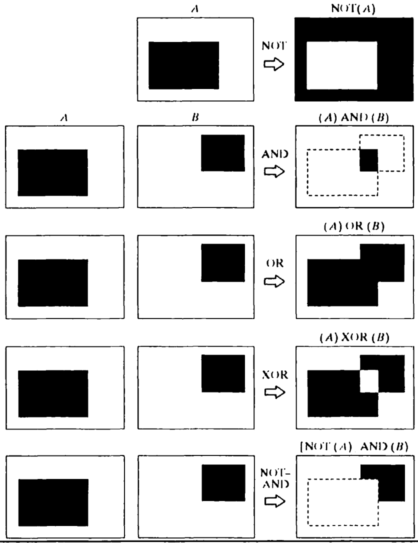


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

complement of A

AND gives the intersection of the sets

OR gives the union of the sets

This is the exclusive or and basically gives the complement of the OR

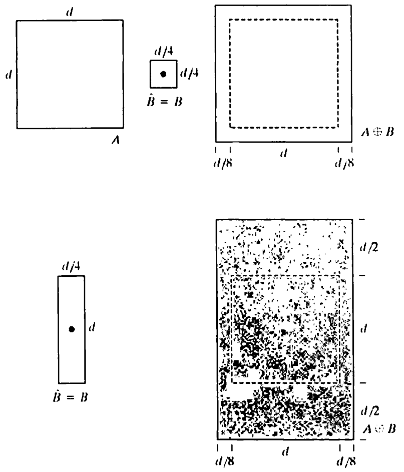
more complex

We can perform logic operations between images on a pixel by pixel basis.

Chapter 9
Morphological Image Processing

a b c
d e

FIGURE 9.4 (a) Set A. (b) Square structuring element (dot is the center). (c) Dilation of A by B, shown shaded. (d) Elongated structuring element. (e) Dilation of A using this element.



dilation of A by B

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}$$

B is called the structuring element.

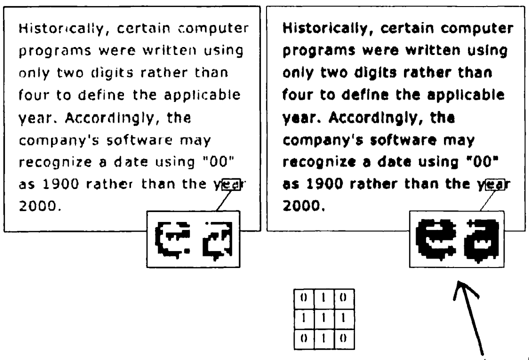
<u>Explanation.</u>

- Reflect B about the origin $\hat{B}$
- Shift it by z, $(\hat{B})_z$
- $A \oplus B$ is the set of all displacements z such that $\hat{B}$ and A overlap.

This is very similar to a convolution mask.

Other definitions are possible.

Chapter 9
Morphological Image Processing



a c
b

FIGURE 9.5 (a) Sample text of poor resolution with broken characters (magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

joined gaps

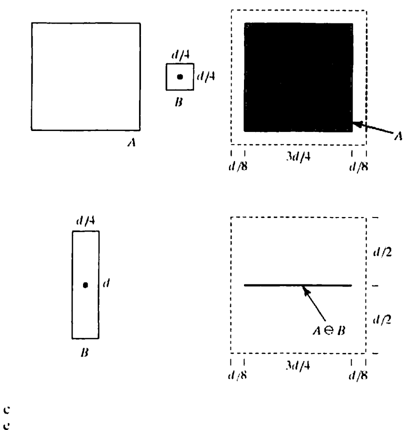
structuring element $\hat{B} = B$

Advantage of dilation over low-pass filtering is that result remains binary

Digital Image Processing, 2nd ed. www.imageprocessingbook.com

Chapter 9
Morphological Image Processing

not drawn to scale



no up/down translation is possible so this becomes just a straight line.

FIGURE 9.6 (a) Set A. (b) Square structuring element. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A using this element.

Erosion

$$A \ominus B = \{ z \mid \underbrace{(B)_z \subseteq A}_{\subseteq \text{ subset of}} \}$$

$A \ominus B =$ set of all points z such that the translation of B by z is contained in A

Note that dilation and erosion are dual processes.

$$(A \ominus B)^c = \{ z \mid (B)_z \subseteq A \}^c$$

$$= \{ z \mid (B)_z \cap A^c = \emptyset \}^c$$

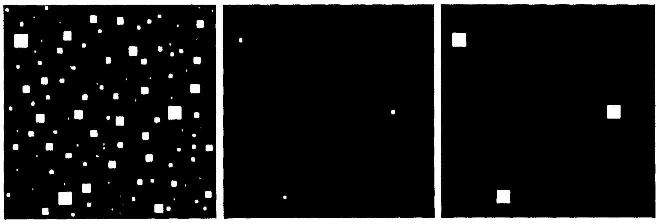
if $(B)_z$ is in A then $(B)_z \cap A^c = \emptyset$

$$= \{ z \mid (B)_z \cap A^c \neq \emptyset \}$$

The complement is just the set for which $(B_z) \cap A^c \neq \emptyset$

$$= A^c \oplus \hat{B}$$

Chapter 9
Morphological Image Processing



a b c

FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

result of erosion by a 13 x 13 image of 1's

result of dilating the eroded image with a 13x13 all 1's structuring element

Use erosion to eliminate irrelevant (small) detail

Select a structuring element slightly smaller than the details you want to keep.