Noise Reduction for Tiny Contours in Image Sequence

Zhai Ming
School of Electronic, Information and Electrical Engineering
Shanghai Jiao Tong University
Shanhai, P.R.China.
totoogo@hotmail.com, tootoogo@sjtu.edu.cn

Abstract - The contours of remote objects in the image sequence obtained by an optical sensor in space were often tiny and inclined to be stained by noise. It would affect the results of some contour based algorithms. In order to solve the problem a noise reduction technique for the tiny contour in image sequence was presented in this paper. In the technique every contour of the sequence were described in frequency domain by the Fourier Descriptors (FD’s). A set of Interacting Multiple Model (IMM) estimators were used to filter the noise in every frequency point of the contours FD’s. The contour after noise reduction was retrieved by inverting the filtered contour FD’s. The experimental results show the technique was useful and effective.

Keywords: Noise reduction, Interacting Multiple Model (IMM), Fourier Descriptors (FD’s), Kalman filter, image contour.

1 Introduction

It was significant to use remote optical sensors to monitor space objects such as satellites, debris etc. to obtain the informations like rotation angle, maneuvering time, 3D structure of these objects. The areas of objects in the image sequence obtained by remote optical sensors were often tiny for the sake of the long distance from the sensor. Given the remote optical sensor with angle resolution of 5 μradius, and then the space resolution was 0.45 meter when the optical sensor was 100 km. far away form its target. That meant that if the object area was 3 m × 3 m, it only took 7 × 7 pixels in the image sequence. In order to get useful informations from such tiny image area some contour based methods could be taken into account [1-4]. However due to all kinds of noise, the tiny contour would change unstably as time varying. It would affect the results of these contour based methods. Therefore, contour noise reduction was a necessary process in the contour based applications. Nevertheless the noise reduction for tiny contour in image sequence was seldom concerned.

In space, noise arose from many factors[5]. Radiation was one of the factors. Radiation would affect the imaging process of optical sensor and added salt noise into the image. Light diffusion would blur the edge of objects so it could be regarded as a kind of noise. Image digitization also affected the contour. It would make the evolution of the contour uncontinuous as time was varying. The effect of digitization could be modeled as noise. Although some smoothing method like B-spline interpolation would make the contour smoother, these methods did not take account of the relationship in the image sequence. Furthermore the noise of the sensor’s own such as electronic noise, temperature would bring noise into the contour [6].

For the stationary objects, some statistic information on noise computed from the contour in image sequence could be used to reduce noise, however, the objects may maneuvered and these methods were not suitable for moving objects. So a noise reduction method for the contour of both stationary and moving objects was needed urgently. For moving objects it seemed that the methods based on active contour tracking [7-9] could be used, but these were not suitable for this scenario. Most active contour based tracking methods were designed to track targets. Therefore these methods were not suitable for noise reduction for tiny contour. In [10], object contour was abstract from motion parameters, however, the motion parameters were estimated by optical flow which led to much more computation and moreover this method aimed at tracking so the contour detail was omitted. Some methods based on lowpass filter in frequency domain or smoothing operator in space domain could reduce noise, but these methods cut off contour detail and the noise in lower frequency still remained.

Our contribution was that we presented a noise reduction technique for tiny contours in image sequence. The technique was suitable for both stationary and moving objects. It would be needed as preproessing by some applications which needed accurate contour and it could extend some contour based applications. Moreover it could partly solve the uncontinuous evolutin of the contour sequence due to image digitization. In the technique when the contour was abstracted from current frame in the image sequence, some morphology operating was used to cut off branches on the contour and made the contour closed. After that the contour was described in frequency domain by the Fourier Descriptors (FD’s) [11]. The noise in the contour was also turned into frequency domain by FD’s. Assuming the noise was Gaussian distributed with zero mean, a set of adaptive estimators were used to filter the noise of every frequency point in the contour FD’s. Here the Interacting Multiple Model (IMM) [12] estimator was used. The same
operation was carried out on the following frames of the image sequence.

This paper is organised as follows: Section 1 is introduction; in Section 2 some basic notions such as FD’s, IMM, Kalman filter are reviewed; the implementation of our technique is in Section 3; Section 4 is experiments results; Section 5 is conclusion and discussion.

2 Preliminaries

In this section, we would review some basic notions from the theory of FD’s, Kalman filter, as well as the IMM techniques which we would need in the sequel.

2.1 Fourier Descriptors (FD’s)

It was more than two decades since FD’s was introduced [11]. There were many types of FD, such as 2D FD’s [13], angle based FD’s [14] etc. In this paper, the most popular type of FD’s was used. Given a closed contour \( S_t \) at time \( t \) in a 2D Cartesian plane, let \( X_t = \{ (x_{t1}, y_{t1}), (x_{t2}, y_{t2}), \ldots, (x_{tk}, y_{tk}) \} \) be the points sampled from \( S_t \) with equivalent interval in clockwise, and \( K \) was the number of the points. Under this condition, the contour \( S_t \) could be expressed as a complex number sequence:

\[
  c_i(u) = x_u + y_u j, \quad u=1,2,3,\ldots,K; \quad (1)
\]

That was the \( x \)-axis was treated as the real axis while the \( y \)-axis as the imaginary axis. For a closed contour, it could be seen as a period signal, that was

\[
  c_i(u+M) = c_i(u)
\]

where \( M \) was the multiple of \( K \). For the sake of periodical property \( c_i(u) \) could be transformed by the Discrete Fourier Transform (DFT). The DFT of \( c_i(u) \) was

\[
  C_i(v) = \sum_{k=1}^{K} c_i(k) e^{-2\pi jk/k}, \quad v=1,2,\ldots,K \quad (2)
\]

The complex sequence \( C_i(u) \) was called Fourier Descriptor (FD’s) of the contour \( S_t \). The noise sequence added to contour \( S_t \) was denoted as \( X_t' = \{ (x'_{t1}, y'_{t1}), (x'_{t2}, y'_{t2}), \ldots, (x'_{tk}, y'_{tk}) \} \) and the corresponding complex sequence denoted as \( c_i'(u) \), \( u=1,2,\ldots,K \), where \( x'_{t1}, y'_{t1}, x'_{t2}, y'_{t2}, \ldots, x'_{tk}, y'_{tk} \) were Gaussian distributed with zero-mean and mutually independent. Let \( X_t'' = \{ (x''_{t1}, y''_{t1}), (x''_{t2}, y''_{t2}), \ldots, (x''_{tk}, y''_{tk}) \} \) be the point sequence of contour \( S_t \) with noise and \( c_i''(u) \), \( u=1,2,\ldots,K \), be the corresponding complex sequence, and then we had

\[
  X_t'' = X_t + X_t' \quad (3)
\]

where \( X_t' \) was without noise. Equation (3) in the complex number sequence form was

\[
  c_i''(u) = c_i(u) + c_i'(u)
\]

\[
  = (x_u + y_u j) + (x'_u + y'_u j)
\]

\[
  = (x_u + x'_u) + (y_u + y'_u) j \quad (4)
\]

Impose DFT on the equation (4) yielding:

\[
  C_i''(v) = C_i(v) + C_i'(v)
\]

\[
  = C_i(v) + \frac{1}{K} \sum_{s=1}^{K} c_i'(s) \cos(-2\pi sv/K) +
\]

\[
  \frac{j}{K} \sum_{s=1}^{K} c_i'(s) \sin(-2\pi sv/K)
\]

\[
  = C_i(v) + \text{real}(C_i'(v)) + \text{imag}(C_i'(v)) \quad (5)
\]

Where \( C_i''(u) \), \( C_i'(v) \) were the DFT of \( c_i''(u) \) and \( c_i'(u) \) respectively. The Equation (5) meant that the noise of every frequency point was the weighted sum of noise sequence \( c_i'(u) \) and could be divided into real part and imaginary part. The weighted sum of Gaussian distributed data was also Gaussian distributed, so the real (or imaginary) part of the noise at every frequency point was Gaussian distributed and therefore in this condition the Kalman filter could be used to filter the noise in the FD’s.

2.2 The Kalman Filter and the Interacting Multiple Model

The Kalman filter was a linear estimator based on MMSE (Minimum Mean Squared Error) [12, 15, 16], for the sake of less computation and better performance it had been widely used since it was introduced. Here was a brief review of Kalman filter.

Given a system equation and measurement equation as follows:

\[
  x(k+1) = F(k)x(k) + G(k)u(k) + v(k)
\]

\[
  z(k+1) = H(k+1)x(k+1) + w(k+1) \quad (6)
\]

Where \( x \) and \( z \) were state vector and measurement vector respectively; \( v \) and \( w \) were zero mean Gaussian noise with covariance \( Q \) and \( R \); \( u \) was known input vector; \( F \) and \( H \) were state matrix and measurement matrix. \( F, G, H \) and \( Q \) were assumed known and possibly time-varying. The two noise sequences and the initial state were assumed mutually independent. The above constituted the linear Gaussian (LG) assumption.

The main equations of Kalman filter were as follows:

\[
  x(k+1|k) = F(k)x(k|k) + G(k)u(k)
\]

\[
  z(k+1) = H(k+1)x(k+1) \quad |k)
\]

\[
  \tau(k+1) = z(k+1) - z(k+1 \quad |k)
\]

\[
  P(k+1|k) = F(k)P(k|k)F^T(k) + Q(k)
\]

\[
  \frac{j}{K} \sum_{s=1}^{K} c_i'(s) \sin(-2\pi sv/K)
\]

\[
  = C_i(v) + \text{real}(C_i'(v)) + \text{imag}(C_i'(v)) \quad (5)
\]

Where \( C_i''(u) \), \( C_i'(v) \) were the DFT of \( c_i''(u) \) and \( c_i'(u) \) respectively. The Equation (5) meant that the noise of every frequency point was the weighted sum of noise sequence \( c_i'(u) \) and could be divided into real part and imaginary part. The weighted sum of Gaussian distributed data was also Gaussian distributed, so the real (or imaginary) part of the noise at every frequency point was Gaussian distributed and therefore in this condition the Kalman filter could be used to filter the noise in the FD’s.
\[ S(k+1) = R(k+1) + H(k+1)P(k+1|k)H^T(k+1) \]
\[ x(k+1|k+1) = x(k+1|k) + W(k+1)\tau(k+1) \]
\[ W(k+1) = P(k+1|k)H^T(k+1)S(k+1)^{-1} \]
\[ P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W^T(k+1) \]

Where the expression \( A(k+1|k) \) denoted the estimate of \( A \) at time \( k+1 \) estimated at time \( k \); \( P \) and \( S \) were the state covariance and the innovation covariance; \( W \) and \( \tau \) were the filter gain and the measurement residual.

Considering that the maneuvering model might change as time varying, a single Kalman filter would perform badly, so an adaptive estimator – the Interacting Multiple Model (IMM) estimator [12] was used to offset the limitation of the single Kalman filter. IMM was widely used in target tracking and here we will review it briefly.

The main idea of IMM was that at time \( k \) the state estimate was computed under each possible current model using \( r \) filters, with each filter using a different combination of the previous model-conditioned estimated-mixed initial condition. The notation \( M(k) \) denoted the model at time \( k \), and the model at time \( k \) was assumed to be among the possible \( r \) models:

\[ M(k) \in \{ M_j \}_{j=1}^r \quad (8) \]

With the total probability theorem, \( r \) filters running in parallel was as follows:

\[ p(x(k)\mid Z^k) \]
\[ = \sum_{j=1}^r p(x(k)\mid M_j(k), Z^k)P(M_j(k)\mid Z^k) \]
\[ = \sum_{j=1}^r p(x(k)\mid M_j(k), z(k), Z^{k-1})\mu_j(k) \]

Where \( M_j(k) \equiv \{ M(k) = M_j \} \). The probability of model \( j \) at time \( k \) was \( \mu_j(k) \equiv P(M_j(k)\mid Z^k) \) and the measurement sequence was \( Z^k = \{ z(1), z(2), ..., z(k) \} \). The model-conditioned posterior pdf of the state was

\[ p(x(k)\mid M_j(k), z(k), Z^{k-1}) \]
\[ = \frac{p(z(k)\mid M_j(k), x(k))}{p(z(k)\mid M_j(k), Z^{k-1})} \]
\[ \cdot p(x(k)\mid M_j(k), Z^{k-1}) \quad (10) \]

Applying the total probability theorem to the last term above, yielding:

\[ p(x(k)\mid M_j(k), Z^{k-1}) \]
\[ = \sum_{j=1}^r p(x(k)\mid M_j(k), M_j(k-1), Z^{k-1}) \]
\[ \cdot P(M_j(k-1)\mid M_j(k), Z^{k-1}) \quad (11) \]

\[ \approx \sum_{j=1}^r p(x(k)\mid M_j(k), M_j(k-1), {x'}(k-1|k-1)) \cdot P'(k-1|k-1) \cdot \mu_j(k-1|k-1) \]
\[ = \sum_{j=1}^r p(x(k)\mid M_j(k), M_j(k-1), x'(k-1|k-1)) \cdot P'(k-1|k-1) \cdot \mu_j(k-1|k-1) \]

where \( \mu_j(k-1|k-1) \equiv P(M_j(k-1)\mid M_j(k), Z^{k-1}) \) is called the mixing probability and \( x' \) was the state of the \( i \)th model. With Gaussian assumption \( x(k\mid k) \) could be estimated from Equation (9) and (11).

3 Implementation

Given the contour abstracted from the current frame of the image sequence, before the noise reducing process some preprocessings were necessary. Some morphologic operating was needed to prune little branches in the contour and made the contour closed. Like most FD’s based algorithms, the premise of the equal number of points of every contour in the image sequence should be satisfied. In order to made every contour in the image sequence have the same number of points, interpolation was needed. Re-sampling the B-spline interpolated contour was a proper way to make the number of points of every contour equal. After these preprocessings the contour sequence was described by FD’s using Equation (2) and then the noise reduction process was carried out.

In the noise reduction process the IMM estimator was used. In the IMM estimator, several kinetic models were used, that is, the constant velocity model (CV) and the constant acceleration model (CA) [12]. When the target was in a stationary state it was suitable to use the CV model. While the target was maneuvering it was suitable to use the CA model. In the CV model, let \( \chi_{tj} = (\alpha_{tj}, \beta_{tj})^T \) denote the state vector of the contour FD’s real part at frequency \( j \) at time \( t \). \( \alpha \) and \( \beta \) could be looked on as displacement and velocity respectively of the CV model. Let the corresponding measurement be the real part of the contour FD’s at frequency \( j \) and the measurement was regarded as the velocity with noise of the CV model. In the CA model, \( \chi'_{tj} = (\alpha'_{tj}, \beta'_{tj}, \gamma_{tj})^T \) denoted the state vector of the contour FD’s real part at frequency \( j \) at time \( t \). \( \alpha, \beta \) and \( \gamma \) were looked on as displacement, velocity and acceleration respectively. The corresponding measurement was the contour FD’s real part at frequency \( j \). The measurement was regarded as the velocity with noise. For the imaginary part the similar denotations were used. And then the real part and imaginary part of the FD’s at every frequency point were filtered by two sets of IMM estimators under the possible CV model and CA model respectively. The contour with less noise was retrieved by the inverse transform of the filtered contour FD’s. In order to analyze the result of the noise reduction, the Euclidean
metric in the space of FD’s [17] was used to compute the distance between the real contour (without noise) and the filtered (after noise reduction). If $C_a$ and $C_b$ are the FD’s of two contour, the Euclidean metric in the space of FD’s between $C_a$ and $C_b$ is defined:

$$d(C_a, C_b) = \frac{1}{K} \left[ \sum_{v=1}^{K} |C_a(v) - C_b(v)|^2 \right]^{1/2} \quad (12)$$

The smaller distance meant more similarity. The flow chart (Fig. 1) illustrated the noise reduction process briefly.

4 Experiment

In order to test our technique many experiments were carried out. Here the two of them were presented. In Experiment I, a plane contour drawn by hand was used. The image sequence were computed out assuming the plane was stationary and the camera moves along a predefined locus in 3D space. The scenario that the plane and the camera move simultaneously could be turned into the equivalence scenario that only the camera moves, so the assumption was proper. The first frame without noise was shown in Fig. 2. The Gaussian noise with the variance 0.5 was added to the contour points of every frame directly. In Experiment I, in the first 30 frames the camera kept stationary, after that the plane rotated along one center axis. Some results were shown in Fig. 3. The distance in FD’s space between the filtered and the real and the distance between the noisy and the real were shown in Fig. 4. Fig 4 showed that the filtered results were more similar to the real than the noisy. In the first several frames there was an unstable state. This was caused by adaptively adjusting the parameters of the estimators. After that the state kept stable.
In Experiment II, an AutoCAD model which imitated a satellite was used. The image sequence was obtained by AutoCAD assuming the model keeping stationary and the camera moving along a predefined ellipse locus in 3D space. The model was scaled properly to imitate the remote condition. The first frame without noise of the image sequence was shown in Fig. 5. The Gaussian noise with the variance 0.5 was added to the contour points in every frame directly. Some results were shown in Fig. 6. The distance in FD’s space between the filtered FD’s and the real FD’s and the distance between the noisy and the real was shown in Fig. 7.

Similar to Experiment I, the filtered results were more similar to the real than the noisy. In the first several frames there was also an unstable state. This was caused by adaptively adjusting the parameters of the estimators. After that the state kept stable. However the results of Experiment II were not as good as Experiment I. This was caused by two factors. The first one was that the contours in Experiment II was smaller than the one in Experiment I. With the same noise mean and variance, the smaller contour was more prone to be disturbed by noise than the larger contour. The second was that when obtaining the image sequence by AutoCAD the noise of digitization was added. The Gaussian noise was only the approximation of the noise of image digitization. So it would affect the results when using Gaussian noise substituting it while in Experiment I there was no the noise of image digitization.
5 Conclusions and Discussion

The experiments results show that our technique is useful and effective for both stationary and moving objects. The technique will get better result when using large contour than using small contour. Moreover the technique can partly solve the unstable evolution of the contour sequence due to image digitization. However the technique needs some time to be stable, that is in the first several frames the results are useless. Furthermore the technique is more suitable for the Gaussian noise condition than the non-Gaussian condition. However the non-Gaussian noise like the noise of image digitalization is often met so the noise reduction in a non-Gaussian condition is our future work.

References


