Object Tracking Using SIFT Features in a Particle Filter

Abstract

This paper adds sift matching features into the particle filter tracking framework base on color histogram feature, and proposes a dual character tracking algorithm, in which the particle weights are calculated considering both the sift matching features and the color histogram feature. Experimental results shows this approach is working better especially in cases that illumination changes.

Intro

Tracking target in video is important application of CV.

- Color feature
- Color histogram (Runs well when no changes on scene)
- Fails on illumation etc.
- What to do ? (Combine the color histogram features with SIFT)

What was SIFT?

SIFT: Scale Invariant Feature Transform

The algortihm is a multi-scale feature extraction algorithm based on gradient information. The core idea is extracting feature points in different scales of the image. It makes full use of scale space features and has good robustness of scale invariant.

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{1}$$

The Algorithm

1) Detection of extreme-value points in scale space.Each sampling point is compared with all the adjacent points in image filed or scale field to see if its value is the biggest or the smallest.

2) Accurate determination of location of extreme-value points.We can accurately determine the key points' location and scale by fitting a 3D quadratic function. Meanwhile remove the key points of low contrast and instable edge response point to enhance the matching stability and improve resistance to noise.

3) Determine the main direction of extreme-value points

The Algorithm (Continue)

SIFT algorithm uses gradient direction's distribution characteristic of adjacent pixels of the key points to specify

direction parameters, as shown in (2).

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$$
$$\theta(x,y)=\tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \tag{2}$$

The Algorithm (Continue)

4) Generate feature vectors.Calculating gradient histogram in 8 directions and plotting accumulated value of each gradient direction to form a seed point. 16 seed points are used to describe each feature point, and 128D feature vector is formed.

5) Match feature points.Euclidean distance between feature vectors is used asthe similarity measure of key points in two images.

Particle Filter Principles

For a nonlinear dynamic stochastic system, let $\{x_t : t \in N\}, x_t \in R^{n_x}$ represents the hidden state (non-observation data) of the system at different times, and x_0 stands for the initial state of the system; $\{y_t : t \in N\}, y_t \in R^{n_z}$ represents the observation of the system at different times. Assume that the prior distribution of the initial state x_0 is $p(x_0)$, and then the system's motion equation and observation equation can be expressed as (3) and (4).

$$x_t = f_x(x_t - 1) + w_t \tag{3}$$

$$y_t = g_y(x_t) + v_t \tag{4}$$

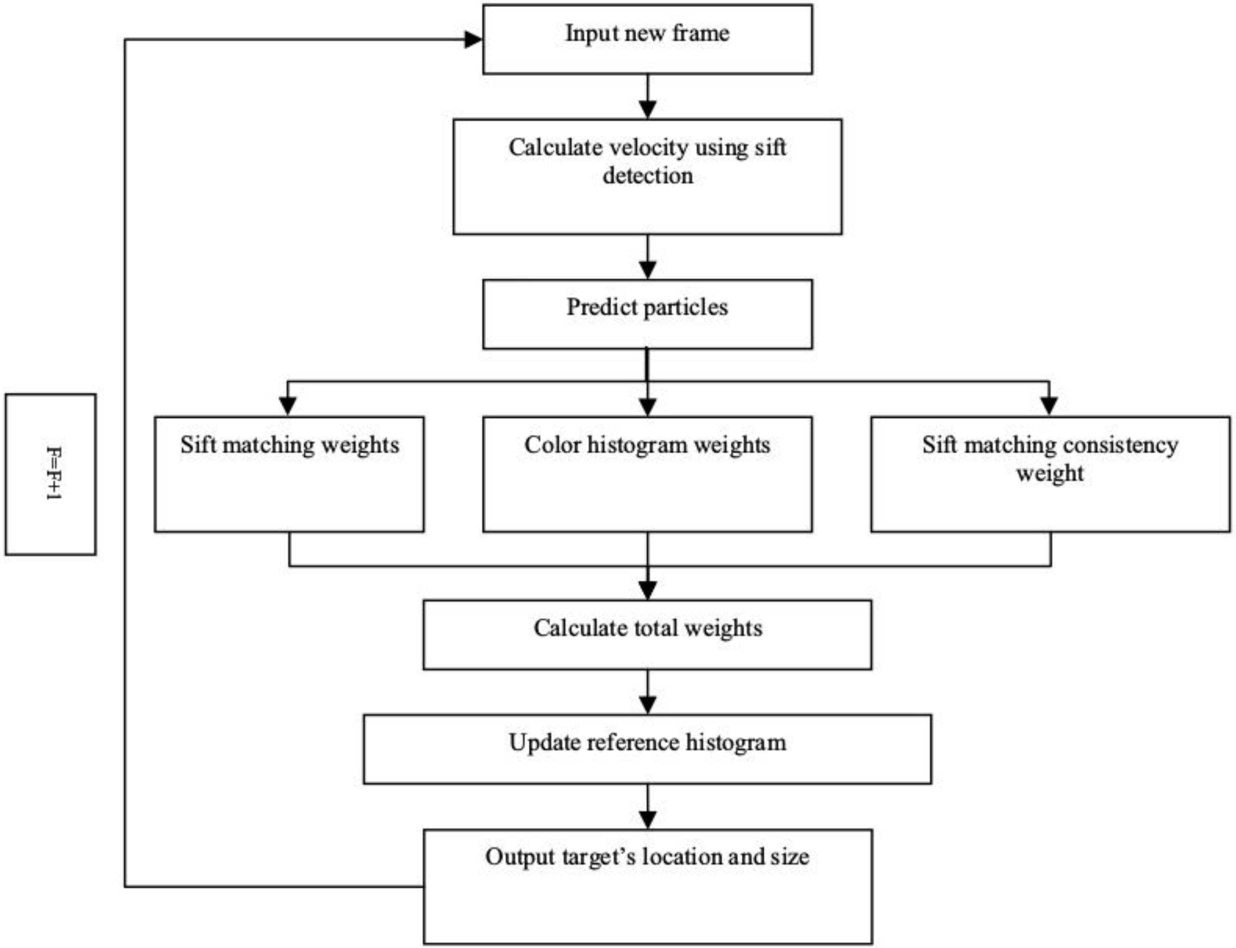
Here w_t and v_t are independent noise vectors with a certain probability distribution to represent process noise and observation noise, respectively. f_x and h_x are the state transition function and observation function, respectively.

Calculate posterior probability density recursively by Bayesian equation, as shown in (5).

$$p(x_t/y_{1:t}) \propto p(y_t/x_t) \int p(x_t/x_{t-1}) p(x_{t-1}/y_{1:t-1}) dx_{t-1} \tag{5}$$

Here $p(y_t/x_t)$ is likelihood function, which represents observation model; while $p(x_t/x_{t-1})$ is state prediction, which stands for motion model.

The basic idea of the particle filter is to use a set of particles with weights to approximate the target state posterior probability density $p(x_t/y_{1:t})$. When the number of particles is large enough that tends to infinity, then these particles can fully describe the posterior probability density. To achieve the recursive posterior density estimation above, at each time particle filter algorithm includes particle prediction, particle balance, particle transfer and particle re-sampling as well as other steps.



Input new frame
Calculate velocity using sift detection
Predict particles
Sift matching weights
Color histogram weights
Sift matching consistency weight
Calculate total weights
Update reference histogram
Output target's location and size
F=F+1

Calculate Velocity Using SIFT

Calculate the center coordinates of the best sift matching points between current frame and templates, and then calculate the Euclidean distance between the center coordinates and tracking results of the previous frame as target velocity. As shown in (6).

$$v_t = sqrt\left(\left(\frac{1}{n}\sum_{m=1}^{n} x_m - x_{t-1}\right)^2 + \left(\frac{1}{n}\sum_{m=1}^{n} y_m - y_{t-1}\right)^2\right) \quad (6)$$

Predicting Particles

Particle state is defined as (x, y, w, h), which represent the center coordinates of particle, width and height, respectively. Predict particles in accordance with uniform distribution of random walk, as shown in (7).

Here v(t) is random velocity. Theta is the angle between prediction vector and the horizontal axis, generated equal-interval according to the number of particles. Sigma is uniformly distributed random noise, and alpha1, alpha2 is the variance of sigma.

$$
\begin{aligned}
x_t &= x_{t-1} + v_t \cdot \cos\theta \\
y_t &= y_{t-1} + v_t \cdot \sin\theta \\
w_t &= w_{t-1} + \alpha_1 \cdot \sigma \\
h_t &= h_{t-1} + \alpha_2 \cdot \sigma
\end{aligned} \tag{7}
$$

Calculating Particle Weights

$$w_c = \exp(B(h_t, h_r)) \qquad w_s = N_{bestmatch} \qquad w_p = N_{picturematch}$$

Then assign empirical constant λ 1， λ 2， λ 3，(λ 1+ λ 2+ λ 3=1)， as shown in (11).

$$w = w_c \times \lambda_1 + w_s \times \lambda_2 + w_p \times \lambda_3 \tag{11}$$

(B is Bhattacharyya distance)

Remove Error Detection Points

1) Feature points corresponding to two or more points in the templates are removed.

2) Estimate the rotation angle between matching image and templates, the points with larger error are ignored.

3) Estimate the size scale between matching image and templates, the points with larger error are ignored.

Results

C. Tracking results

Since the space is limited, here we only display 5 tracking results between the 30th frame and the 80th frame.

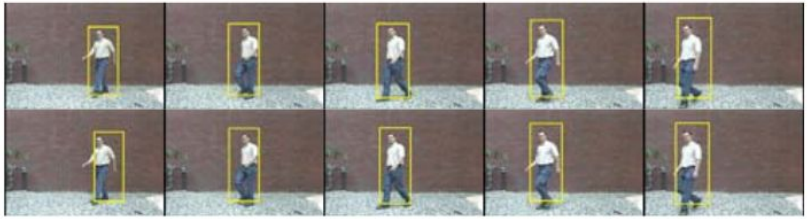


Figure 4. Tracking results

As shown in Fig.4, the upper row shows the combined-feature PF tracking results and the lower row shows PF tracking results only with color histogram feature. From left to right are Frame 33, Frame 47, Frame 58, Frame 66 and Frame 80, respectively.

In order to quantitatively compare the tracking performances, we extract the center coordinates that is considered accurate by human eyes as center-1 and calculated the Euclidean distance between center-1 and the tracking results of each method. As shown in Fig.5:

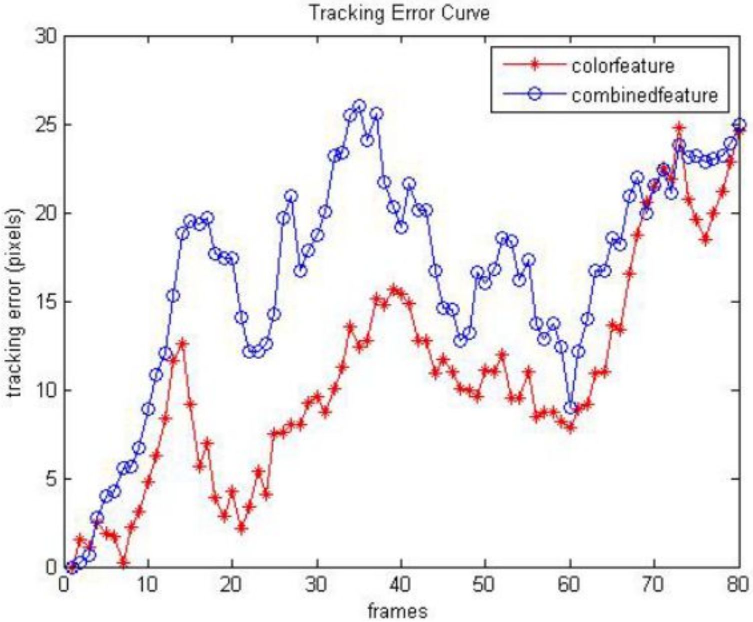


Figure 5. Tracking Error Curve