

Improvement on Mean Shift based Tracking Using Second-Order Information

Lijuan Xiao, Peihua Li

*School of Computer Science and Technology, Heilongjiang University
Harbin, Heilongjiang Province, 150080, China
peihualj@hotmail.com*

Abstract

Object tracking based on Mean Shift (MS) algorithm has been very successful and thus receives significant research interests. Unfortunately, traditional MS based tracking only utilizes the gradient of the similarity function (SF), neglecting completely higher-order information of SF. The paper regards MS based tracking as an optimization problem, and proposes to make use of both the Gradient and Hessian of SF. Specifically, we introduce Newton algorithm with constant, unit step and Newton with varying step lengths, and Trust region algorithm. The advantage of exploiting higher-order information is that higher convergence rate and better performance are achieved. Diverse experiments are made to compare traditional MS based tracking with the proposed algorithms, showing that the proposed algorithms have better performance at comparable computational cost.

1. Introduction

Object tracking based on Mean Shift (MS) algorithm has been very popular in recent years, thanks to its real time response, robustness and easy implementation. A lot of related work has been presented, following the idea of MS based tracking originating from the seminal paper of Comaniciu et al. [3]. In that paper, color histograms are combined with spatial kernels, particularly Epanechnikov kernel, to characterize probability density function of objects, and Bhattacharyya coefficient is used as similarity function (SF) that measures likeness between target and candidate models.

MS based tracking is essentially a gradient descent algorithm, which makes only use of the gradient information of SF and thus has linear convergent rate. Yang et al. [7] made first attempt to introduce curvature information and adopt quasi-Newton method to avoid expensive computation of Hessian matrix. Their work focuses on optimization of unknown, non-parametric density for segmentation, in contrast to MS based tracking that aims at optimization of SF between a known,

target density and candidate density. Fashing et al. [4] also discuss optimization of a unknown, non-parametric density, rather than that of a SF, proving that mean shift is a Newton method for piecewise constant kernels and is a quadratic bound maximization for all kernels.

Liu et al. [5] use bivariate Gaussian as weighting function, whose covariance ellipse represents the position and pose of the object. They combined weighted color and edge histograms, and use K-L divergence as a similarity function that is optimized via Trust region approach. However, they failed to provide quantitatively comparison with traditional MS. In [2], Bajramovic et al. presented a weighted combination of several different histograms (CHT), instead of a single histogram, and either traditional MS algorithm or Trust region approach can be used for this framework.

The contributions of our paper are that we present MS tracking algorithm based on Newton method and Trust region method, and make qualitative and quantitative analysis of these algorithms, compared to traditional MS tracking algorithm. Thus, we have a deeper insight into traditional MS tracking in the framework of optimization theory and have clear understanding of their respective benefits and disadvantages for facility of appropriate choice in real-world applications.

Our paper is organized as follows. The next section formulates tracking problem as optimization of objective function. A brief review of traditional MS tracking follows in section 3. Two approaches that use both the Gradient and Hessian–Newton method and Trust region method, are introduced respectively in section 4 and section 5. Section 6 gives the experimental results and conclusion is given finally in section 7.

2. Formulation of tracking problem

Suppose the target is represented by a rectangle centered at the image origin, consisting of pixels of spatial coordinates (x_i^*, y_i^*) , $i = 1, \dots, n$. If the color space is divided into m bins, the target model can be represented by the histogram $\mathbf{p} = [p_1, \dots, p_m]^T$, and each individual p_u is computed by

$$p_u = C \sum_{i=1}^n g \left(\frac{x_i^*}{h_x^2} + \frac{y_i^*}{h_y^2} \right) \delta_{ui}, \quad u = 1, \dots, m$$

where $g(\cdot)$ is a symmetric kernel [3] with bandwidth (h_x, h_y) , the normalization constant C is derived by imposing the condition $\sum_{i=1}^m p_u = 1$, and δ_{ui} is the Kronecker delta function that is 1 if the pixel falls into u th bin and 0 otherwise.

Suppose the candidate model comprises pixels of spatial coordinates (x_i, y_i) , $i = 1, \dots, n_h$. Its histogram is denoted by the $\mathbf{q} = [q_1, \dots, q_m]^T$ and each individual bin $q_u(\mathbf{z})$ is given by

$$q_u(\mathbf{z}) = C_h \sum_{i=1}^{n_h} g \left(\frac{(x_i - x)^2}{h_x^2} + \frac{(y_i - y)^2}{h_y^2} \right) \delta_{ui}$$

where $\mathbf{z} = (x, y)$ is the center coordinate, C_h is the normalization constant.

The similarity function of Bhattacharyya coefficient $S(\mathbf{z}) = \sum_{u=1}^m \sqrt{p_u q_u(\mathbf{z})}$ is used to measure the likeness between the target and candidate models. Hence, object tracking can be formulated as optimization of the following objective function:

$$\operatorname{argmin}_{\mathbf{z}} F(\mathbf{z}) \triangleq -S(\mathbf{z}) \quad (1)$$

3. Traditional MS tracking algorithm

Suppose \mathbf{z}_k is object center after k th iteration in current frame. Let's consider the derivative of $F(\mathbf{z})$ in Eq. (1) with respect to \mathbf{z} . The gradient $\nabla F(\mathbf{z})$ of the object function has the following form:

$$\nabla F(\mathbf{z}) = C_h \begin{bmatrix} \sum_{i=1}^{n_h} w_i g'(\cdot) \frac{x_i - x}{h_x^2} \\ \sum_{i=1}^{n_h} w_i g'(\cdot) \frac{y_i - y}{h_y^2} \end{bmatrix}$$

where the weight w_i is given by $w_i = \sum_{u=1}^m \sqrt{\frac{p_u}{q_u(\mathbf{z})}} \delta_{ui}$. Let $\nabla F(\mathbf{z}) = 0$, we can get the well-known MS iteration equation for tracking

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \frac{\sum_{i=1}^{n_h} g'(\cdot) w_i (\mathbf{x}_i - \mathbf{z}_k)}{\sum_{i=1}^{n_h} g'(\cdot) w_i} \quad (2)$$

with $\mathbf{x}_i = (x_i, y_i)$.

It can be clearly seen that traditional MS tracking uses only gradient information, while neglecting completely the Hessian, of similarity function (objective function), and is essentially a gradient descent method with linear convergence rate.

4. The Newton method

By the second-order Taylor expansion, we can get an approximation \hat{F} of F of the form

$$\hat{F}(\mathbf{z}_k + \mathbf{p}) \approx F_k + \mathbf{p}^T \nabla F_k + \frac{1}{2} \mathbf{p}^T \nabla^2 F_k \mathbf{p} \quad (3)$$

where F_k , ∇F_k , and $\nabla^2 F_k$ are respectively abbreviated forms of $F(\mathbf{z}_k)$, the Gradient $\nabla F(\mathbf{z}_k)$ and the Hessian $\nabla^2 F(\mathbf{z}_k)$. \hat{F} is in agreement with F up to the second-order, and the error between them is of $O(\|\mathbf{p}_k\|^3)$.

Minimization of Eq. (3) leads to one of the most important method—*Newton method*

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_k \mathbf{p}_k \quad (4)$$

where $\alpha_k \equiv 1$ and $\mathbf{p}_k = -\nabla^2 F_k^{-1} \nabla F_k$ is the Newton direction. Note that constant, unit step length $\alpha_k \equiv 1$ may give unsatisfactory results, and so in many applications we need to choose appropriate step lengths.

Note that in our case, elements of $\frac{1}{C_h}$ times the Hessian, $\frac{1}{C_h} \nabla^2 F(\mathbf{z}_k)$, take the following forms:

$$\left\{ \begin{array}{l} \frac{1}{C_h} \frac{\partial^2 F(\mathbf{z})}{\partial x^2} = \frac{\left(\sum_{i=1}^{n_h} g'(\cdot) \omega_i \frac{x_i - x}{h_x^2} \right)^2}{\sum_{i=1}^{n_h} g(\cdot) \omega_i} - \sum_{i=1}^{n_h} g'(\cdot) \omega_i \frac{1}{h_x^2} \\ \quad - \sum_{i=1}^{n_h} g''(\cdot) \omega_i \frac{2(x_i - x)^2}{h_x^4} \\ \frac{1}{C_h} \frac{\partial^2 F(\mathbf{z})}{\partial y^2} = \frac{\left(\sum_{i=1}^{n_h} g'(\cdot) \omega_i \frac{y_i - y}{h_y^2} \right)^2}{\sum_{i=1}^{n_h} g(\cdot) \omega_i} - \sum_{i=1}^{n_h} g'(\cdot) \omega_i \frac{1}{h_y^2} \\ \quad - \sum_{i=1}^{n_h} g''(\cdot) \omega_i \frac{2(y_i - y)^2}{h_y^4} \\ \frac{1}{C_h} \frac{\partial^2 F(\mathbf{z})}{\partial x \partial y} = \frac{\left(\sum_{i=1}^{n_h} g'(\cdot) \omega_i \frac{x_i - x}{h_x^2} \right) \left(\sum_{i=1}^{n_h} g'(\cdot) \omega_i \frac{y_i - y}{h_y^2} \right)}{\sum_{i=1}^{n_h} g(\cdot) \omega_i} \\ \quad - \sum_{i=1}^{n_h} g''(\cdot) \omega_i \frac{2(x_i - x)(y_i - y)}{h_x^2 h_y^2} \end{array} \right.$$

If Epanechnikov kernel is used, $g'(\cdot) = \text{const}$, $g''(\cdot) = 0$, and the Hessian $\nabla^2 F(\mathbf{z})$ is *positive definite* because all its leading principle minors are larger than zero.

Choice of the step length Given Newton direction \mathbf{p}_k , the desired choice of the step length is the minimizer of the following univariate function

$$\phi(\alpha) = \hat{F}(\mathbf{z}_k + \alpha \mathbf{p}_k)$$

Practically a series of step lengths α_k are tried out, and we stop if a particular α_k is achieved satisfying certain termination conditions. The search algorithm in principle comprises two stages—one stage is to find an appropriate interval containing desired step length, and the other concerns seeking a suitable one on the interval. In our paper, two termination conditions are used: Armijo-Goldstein conditions and Wolfe-Powell conditions.

The Armijo-Goldstein conditions require that α_k achieve sufficient decrease of the objective function $F(\mathbf{z})$ while precluding α_k from being very small, which are given by

$$\left\{ \begin{array}{l} F(\mathbf{z}_k + \alpha_k \mathbf{p}_k) \leq F(\mathbf{z}_k) + c_0 \alpha_k \nabla F_k^T \mathbf{p}_k \\ F(\mathbf{z}_k + \alpha_k \mathbf{p}_k) \geq F(\mathbf{z}_k) + (1 - c_0) \alpha_k \nabla F_k^T \mathbf{p}_k \end{array} \right.$$

where $c_0 \in (0, \frac{1}{2})$ is a constant and is set to 10^{-5} .

Wolfe-Powell conditions also ensure that α_k obtains sufficient decrease. Besides, a curvature condition is enforced so that α_k is in at least a neighborhood of a minimizer of ϕ . Concretely, the step length satisfies the following two inequalities:

$$\begin{cases} F(\mathbf{z}_k + \alpha_k \mathbf{p}_k) \leq F(x_k) + c_1 \alpha_k \nabla F_k^T \mathbf{p}_k \\ \|\nabla F(\mathbf{z}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k\| \leq c_2 \|\nabla F_k^T \mathbf{p}_k\| \end{cases}$$

where $0 < c_1 < c_2 < 1$, and we set $c_1 = 10^{-4}$ and $c_2 = 0.9$ in our experiment.

5. Trust region method

Trust region method defines a region around the current iterate and then chooses the step (including both the direction and the distance along the direction simultaneously). If the step is not acceptable, the algorithm reduces or increases the size of the region. The idea is that the object function $F(\mathbf{z}_k)$ is approximated by a quadratic model function $\hat{F}(\mathbf{z}_k)$ to be minimized over a region, i.e.,

$$\begin{aligned} \arg \min_{\mathbf{p}} \hat{F}(\mathbf{p}) &= F(\mathbf{z}_k) + \nabla F(\mathbf{z}_k)^T \mathbf{p} \\ &+ \frac{1}{2} \mathbf{p}^T \nabla^2 F(\mathbf{z}_k) \mathbf{p} \quad \text{s.t. } \|\mathbf{p}\| \leq r_k \end{aligned} \quad (5)$$

where r_k is the Trust region radius. Hence, this approach requires us to solve a sequence of subproblems (5) to search for the radius r_k and the step \mathbf{p} .

5.1 Trust region radius

The ratio of the actual reduction and predicted reduction is defined by

$$\rho_k = \frac{F(\mathbf{z}_k) - F(\mathbf{z}_k + \mathbf{p}_k)}{\hat{F}(\mathbf{0}) - \hat{F}(\mathbf{p}_k)}$$

The ratio ρ_k measures agreement between the model \hat{F} and objective function F . If ρ_k is close to 1, there is a good agreement and r_k can be expanded to allow more ambitiously longer step; if ρ_k is negative or close to 0, the agreement is poor, indicating \hat{F} is an inadequate representation of F over the current Trust region, and r_k should be decreased; otherwise, r_k should remain unchanged. These strategies are summarized as below:

$$r_{k+1} = \begin{cases} \frac{1}{4} \|\mathbf{p}_k\| & \rho_k < \beta_1 \\ \min(2r_k, \bar{r}) & \rho_k > \beta_2 \\ r_k & \text{otherwise} \end{cases} \quad (6)$$

Here \bar{r} is an overall bound on the step length. Practically we set $\beta_1 = 0.25$ and $\beta_2 = 0.75$. The initial radius r_0 is set to, when a new frame is available, 2-norm of the Gradient of F , computed in current image, at tracking position of previous frame.

5.2 Trust region step

We consider two methods for finding approximate solutions to subproblem (5): the Cauchy point and the Dogleg methods.

Cauchy point The direction in this method is along the steepest-descent and Cauchy step \mathbf{p}_k^C is given by

$$\mathbf{p}_k^C = -\tau r_k \frac{\nabla F_k}{\|\nabla F_k\|}$$

Then $\hat{F}(\mathbf{p}_k^C)$ becomes a quadratic function of τ , minimization of which leads to the solution of either $\tau = \|\nabla F_k\|^3 / (r_k \nabla F_k^T \nabla^2 F_k \nabla F_k)$ or the boundary value $\tau = 1$. The steepest-descent direction may perform poorly due to disregarding of curvature information.

Dogleg method Let \mathbf{p}_k^D and \mathbf{p}_k^N be unconstrained minimizers of Eq. (5) respectively in steepest-descent direction and Newton direction, i.e.,

$$\begin{aligned} \mathbf{p}_k^D &= -\frac{\nabla F_k^T \nabla F_k}{\nabla F_k^T \nabla^2 F_k \nabla F_k} \nabla F_k \\ \mathbf{p}_k^N &= -\nabla^2 F_k \nabla F_k \end{aligned}$$

Then the dogleg direction is defined by

$$\tilde{\mathbf{p}}_k(\tau) = \begin{cases} \tau \mathbf{p}_k^D & 0 \leq \tau \leq 1 \\ \mathbf{p}_k^D + (\tau - 1)(\mathbf{p}_k^N - \mathbf{p}_k^D) & 1 \leq \tau \leq 2 \end{cases}$$

The dogleg direction enjoys the properties that $\|\tilde{\mathbf{p}}_k(\tau)\|$ is an increasing function of τ , and $\hat{F}(\tilde{\mathbf{p}}_k(\tau))$ is decreasing function of τ [6, pp.71-72]. Hence, if $\|\mathbf{p}_k^N\| \geq r_k$ the dogleg direction intersects the Trust region boundary at only one point that can be computed by

$$\|\mathbf{p}_k^D + (\tau - 1)(\mathbf{p}_k^N - \mathbf{p}_k^D)\| = r_k$$

6. Experimental results

The program is written with C++ on a PC with 3.0GHz Intel Pentium(R) 4 CPU and 2G Memory. The RGB color is quantized into 16x16x16 bins. Comparisons are made among traditional MS, Newton with constant, unit step length (NCU), Newton with Armijo-Goldstein conditions (NSA) and with Wolfe-Powell conditions (NSW), and Trust region with dogleg (TRD).

We use publicly available benchmark datasets of CAVIAR project [1] where ground truth has been labeled. One section of video clips concerns a view of the corridor of a shopping center, in which EnterExitCrossingPaths1cor.mpg (EECP), OneLeaveShopReenter1cor.mpg (OLSR) and OneShopOneWait1cor.mpg (OSOW) are used. In another section video clips are filmed with a wide angle camera

lens in an entrance lobby, among which three are used: Browse1.mpg (BROW), Fight_Chase.mpg (FICH) and Fight_OneManDown.mpg (FOMD). Each object in every video clip is labeled as ID0, ID1, ...

In Table 1, the 4th and 5th columns list distance errors of object center, and overlapping region errors [2] both in mean plus/minus standard variance format (mean \pm STD), between the ground truth and tracking results. It can be seen that, in almost all cases, NSA, NSW and TRD have better tracking accuracy than MS, while in most cases NCU performs better than MS.

The 6th column in Table 1 gives average tracking time of different algorithms. It is not surprising that NCU is invariably faster than MS, for it has super-linear convergence rate, and, above all, uses unit, constant step length. The reason that other Newton algorithms (NSA and NSW) are slower is that search of step lengths in each iteration introduces extra evaluation of SF.

The last column in Table 1 shows the number of successful tracking frames until lost as judged by the experimenter or over of the sequence, which shows all algorithms demonstrate very similar robustness.

Table 1: Comparisons among traditional MS, Newton with constant, unit step length (NCU), Newton with Armijo-Goldstein conditions (NSA) and with Wolfe-Powell conditions (NSW), and Trust region with dogleg (TRD)

Seq.	ID	Algorithm	Distance error (pixels)	Overlapping region error (%)	Tracking times(ms)	Length of frames
ID0		MS	4.14 \pm 4.36	0.088 \pm 0.091	7.7	383
		NCU	4.09 \pm 3.90	0.083 \pm 0.080	5.4	383
		NSA	3.95 \pm 4.21	0.081 \pm 0.089	9.8	383
		NSW	3.97 \pm 3.91	0.083 \pm 0.078	7.4	383
		TRD	3.98 \pm 3.95	0.078 \pm 0.078	9.1	383
EECP	ID1	MS	6.14 \pm 5.57	0.136 \pm 0.100	7.2	383
		NCU	5.84 \pm 5.15	0.143 \pm 0.099	4.7	383
		NSA	6.14 \pm 5.72	0.128 \pm 0.101	9.9	383
		NSW	6.11 \pm 5.51	0.131 \pm 0.102	8.3	383
		TRD	5.77 \pm 4.97	0.139 \pm 0.098	8.2	383
	ID3	MS	6.63 \pm 4.69	0.165 \pm 0.078	13.3	118
		NCU	6.08 \pm 4.87	0.145 \pm 0.084	11.5	118
		NSA	6.58 \pm 4.75	0.159 \pm 0.077	20.3	121
		NSW	6.21 \pm 4.87	0.159 \pm 0.080	21.6	121
		TRD	5.67 \pm 4.32	0.138 \pm 0.082	17	121
OLSR	ID2	MS	9.65 \pm 5.59	0.153 \pm 0.062	10.3	180
		NCU	8.48 \pm 5.42	0.165 \pm 0.077	7.8	183
		NSA	9.21 \pm 5.51	0.157 \pm 0.064	14.2	180
		NSW	8.47 \pm 5.65	0.165 \pm 0.074	14.8	180
		TRD	8.70 \pm 5.49	0.166 \pm 0.070	14.2	185
OSOW	ID2	MS	9.38 \pm 6.61	0.178 \pm 0.096	11.7	825
		NCU	8.88 \pm 6.85	0.170 \pm 0.092	8.5	825
		NSA	8.66 \pm 6.46	0.180 \pm 0.103	15.7	825
		NSW	9.07 \pm 6.66	0.180 \pm 0.095	15.6	825
		TRD	9.10 \pm 6.49	0.184 \pm 0.100	13.6	825
	ID3	MS	13.83 \pm 10.24	0.417 \pm 0.208	9.3	284
		NCU	15.63 \pm 12.53	0.332 \pm 0.179	7.1	284
		NSA	11.46 \pm 10.71	0.345 \pm 0.175	15.1	284
		NSW	13.55 \pm 10.94	0.342 \pm 0.152	17.1	284
		TRD	13.54 \pm 10.58	0.305 \pm 0.140	12.8	284
ID4	MS	6.84 \pm 4.12	0.228 \pm 0.115	8.8	576	
	NCU	6.60 \pm 3.83	0.221 \pm 0.114	5.7	576	
	NSA	6.74 \pm 4.18	0.222 \pm 0.110	12.9	576	
	NSW	6.46 \pm 4.02	0.222 \pm 0.115	13.2	576	
	TRD	6.60 \pm 3.88	0.225 \pm 0.119	9.7	576	
BROW	ID0	MS	9.22 \pm 5.37	0.476 \pm 0.180	7.7	171
		NCU	10.1 \pm 5.67	0.465 \pm 0.175	5.3	171
		NSA	7.95 \pm 5.62	0.527 \pm 0.209	11.7	171
		NSW	7.81 \pm 5.53	0.509 \pm 0.219	11.2	171
		TRD	10.0 \pm 5.65	0.465 \pm 0.177	9.52	171
FICH	ID0	MS	8.30 \pm 5.16	0.342 \pm 0.096	7.9	136
		NCU	8.55 \pm 5.41	0.346 \pm 0.077	5.5	136
		NSA	8.28 \pm 5.26	0.337 \pm 0.079	11.4	136
		NSW	7.90 \pm 5.24	0.400 \pm 0.131	10.6	136
		TRD	8.39 \pm 5.70	0.348 \pm 0.076	8.25	136
FOMD	ID5	MS	7.73 \pm 5.46	0.397 \pm 0.243	6.6	559
		NCU	7.43 \pm 4.78	0.401 \pm 0.222	4.3	559
		NSA	6.26 \pm 4.82	0.383 \pm 0.226	8.4	559
		NSW	6.11 \pm 4.68	0.382 \pm 0.232	7.8	559
		TRD	7.29 \pm 4.74	0.395 \pm 0.224	7.8	559

7. Conclusion

In this paper we regard the MS-based tracking as problem of the numerical optimization, and introduce Newton and Trust region methods for solving it. Newton and Trust region methods exploit both the Gradient and Hessian of SF, rather than only the Gradient in traditional MS tracking, which enjoy faster convergence rate and better performance. Quantitative analysis shows that, compared to traditional MS, the second-order methods achieve better tracking accuracy at comparable average tracking time. Future research concerns more extensive experiments and comparisons in diverse image sequences.

Acknowledgements

The work was supported by the National Natural Science Foundation of China under Grant 60673110 and Natural Science Foundation of Heilongjiang Province (F200512), supported in part by Program for New Century Excellent Talents of Heilongjiang Province (1153-NCET-002), Sci. & Tech. Research Project of Educational Bureau of Heilongjiang Province (1151G033), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry and Ministry of Personnel of China, Sci. and Tech. Innovation Research Project (2006RFLXG030) of Harbin Sci. & Tech. Bureau.

References

- [1] EC funded CAVIAR project/IST 2001 37540, 2004. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>.
- [2] F. Bajramovic, C. Gräbl, and J. Denzler. Efficient combination of histograms for real-time tracking using mean-shift and trust-region optimization. In *Proc. 27th DAGM Symp. on Patt. Recog.*, pages 254–261, 2005.
- [3] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conf. Comp. Vis. Patt. Recog.*, pages 142–149, 2000.
- [4] M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 27(3):471–474, 2005.
- [5] T. Liu and H. Chen. Real-time tracking using trust-region methods. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 26(3):397–402, 2004.
- [6] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.
- [7] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis. Mean-shift analysis using quasi-newton methods. In *IEEE Conf. Image Processing*, pages 448–450, 2003.