Research on Parallel Particle Filtering Target Tracking Algorithm Based on Hadoop

Fu Sun[1,a], JianXin Song[1,b]

[1]Nanjing University of Posts and Telecommunications

[a]hoffman@163.com,[b] songjx@njupt.edu.cn

Keywords: Cloud computing, Hadoop, MapReduce, Particle filter, Target tracking

Abstract. In order to achieve high efficiency and low cost moving target tracking in video sequences, we proposed parallel particle filtering target tracking algorithm based on Hadoop cloud platform, the algorithm using the open source calculation model to realize the parallel calculation of all particles in the particle filter.The experiments show that the parallel particle filter target tracking algorithm based on Hadoop improve the calculation efficiency when compared with the existing algorithms improve the calculation efficiency.

1 Introduction

Cloud Computing [1] proposed in 2007 is a kind of Information Technology new concept.Cloud computing is an emerging computing model, which is parallel computing, distributed computing, the integrated development of grid computing, in the form of a simple, transparent, service unlimited computing resources[2].Hadoop [3-4] is one of the most popular parallel computing framework existed, which provides the parallel computing model MapReduce, HDFS distributed file system.

Video sequence of movement target tracking is a core problem of computer vision in video surveillance, which has been widely used in such fields as human computer interaction, robot vision navigation [5-7], its purpose is according to the target initialized information automatically find tracking target in the current image.After more than 20 years of research, it has been existed many tracking algorithms, one of the important methods is based on particle filter tracking. At present, the research on target tracking based on particle filter more focus on improve the tracking accuracy of the algorithm, those algorithms include based on particle swarm optimization particle filter(PSOPF)[8],adaptive particle filter[9],based on the extended Kalman particle filter(EKFPF)[10].etc.All these methods are study how to solve the problem of particle degradation result in tracking low accuracy in one machine mode, but the particle filter algorithm need to sampling a large number of particles to calculate estimate, it leads to consume a large amount of computing time. Therefore making use of cloud computing parallel computing ability to improve the execution efficiency of the algorithmand system real-time performanceis the current and future research trend.Due to cloud computing adopt[11] the new computing model such as MapReduce, it need to parallel design for existing moving target tracking algorithm, let it can be directly applied to the cloud computing platform. In this paper, we designed parallel strategy of based on particle filter moving target tracking, using MapReduce calculation model realized the parallel model of the algorithm.

2 Related Work

2.1 MapReduce calculation model

MapReduce[11-12] is a simplified distributed programming model and efficient a task scheduling model emerging in recent years.MapReduce parallel process mass data through the two steps include the Map and Reduce.The advantage of MapReduce framework mainly include two aspects[13]which one is the distributed framework not only processing mass data but also hiding complicated computing details; the other one is framework having a good scalability, so with increasing a server every time ,the computing source of the server will be accessing to the cluster.

2.2Particle filtering target tracking algorithm

Particle filter[14] is by looking for the random sample which spread in a group of state space to approximate probability density function ,which is a process to use sample mean instead of integral operation for obtaining minimum variance estimation and these samples is called “particle”[15].

2.2.1 The target model

The paper uses color distribution probability function to describe the target model. Through the way of manual annotation in the initial frame determine which area contains tracking target and then get RGB color histogram by discrete statistics for the selected areas. The target pixel located in the edge of may belong to the background or obscured, so in order to enhance the reliability of color distribution, the target pixels which are far away from the center of tracking area is given less weight. Using the kernel function: $\mathrm{k}=\begin{cases}1-\mathrm{r}^2, & \mathrm{r}<1\\ 0, & \text{others}\end{cases}$, The r is the distance from the pixel to center of traking area.The color distribution of probability density function $\mathrm{p_y}=\{\mathrm{p_y(u)}\}$, $\mathrm{u}=1, \ldots, \mathrm{m}$ of the pixel located y distance is: $\mathrm{p_y^{(u)}}=\mathrm{f}\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{k}\left(\frac{\|\mathrm{y_i}-\mathrm{x_0}\|}{\mathrm{a}}\right)\delta[\mathrm{h(y_i)}-\mathrm{u}]$,
$\mathrm{f}=\frac{1}{\sum_{i=1}^{n}k\left(\frac{\|y_i-x_0\|}{a}\right)}$, $\mathrm{a}=\sqrt{H_x^2+H_y^2}$.

Among of the above function ,the n is number of pixels in the selected the area,the $\mathrm{x_0}$ is the area center coordinates,f is the normalized factor, $\mathrm{H_x}$and $\mathrm{H_y}$is the width and height of area respectively. $\delta[\mathrm{h(y_i)}-\mathrm{u}]$ is the function of judge the color values of the pixels in the target area whether belongs to the u histogram unit, if it belong,so its value is 1, otherwise is 0.

2.2.2 The particle samples collection description

The particle sample is defined as s = { x, y, $\mathrm{v_x}$, $\mathrm{v_y}$, $\mathrm{H_x}$, $\mathrm{H_y}$, a} ,the x and y is denoted center position , $\mathrm{v_x}$ and $\mathrm{v_y}$ is denoted speed of partical in direction of the x and y, $\mathrm{H_x}$ and $\mathrm{H_y}$ is the width and height of area which include particles respectively,a is Scale factor .Through the system state equation $\mathrm{s_t}=\mathrm{As_{t-1}}+\mathrm{w_{t-1}}$ to update the particle samples collection, the A is denoted the state transition matrix, $\mathrm{w_{t-1}}$ is the Gaussian noise.

2.2.3 The particle filter algorithm based on color histogram

(1) The initialization of particles: In the initial frame of target tracking , an area is selected as a interested target template T, then the color histogram $\mathrm{H_0}$ of this area is calculated. Random drawing N particles from the prior probability distribution as the initial state of the particle collecton.

(2) The prediction of particle statement: Accoring to the state transition equation $\mathrm{s_t}=\mathrm{As_{t-1}}+\mathrm{w_{t-1}}$and particle state of the last moment, the particle state including position and speed of current time been predicited .

(3)The calculation of particle weight: According to each sample in the statement collection $\mathrm{S_t}$ and color histogram to update the new statement of particle, then obtain the new weight probability. Specific as follows:

(a)The calculation of color histogram of each particle in the statement collection $\mathrm{S_t}$:
$\mathrm{p}_{\mathrm{s_t^{(n)}}}^{(\mathrm{u})}=\mathrm{f}\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{k}\left(\frac{\left\|\mathrm{s_t^{(n)}}-\mathrm{x_0}\right\|}{\mathrm{a}}\right)\delta\left[\mathrm{h}\left(\mathrm{s_t^{(n)}}\right)-\mathrm{u}\right]$; (b) The calculation of the Bhattacharyya coefficient of each praticle color histogram and target model color histogram:P[$p_{s_t^{(n)}}$, q] = $\sum_{u=1}^{m}\sqrt{p_{s_t^{(n)}}^{(u)}q^{(u)}}$; (c)The calculation of each particle weight: $\pi_t^{(\mathrm{n})}=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(1-\rho[p_{s_t^{(n)}}^{(u)}, q])}{2\sigma^2}}$;

(4)The estimation of tracking:The normalization of the new particle weight $\pi_t^{(n)}$ to calculate the estimation value of tracking: $\mathrm{E}[S_t]=\sum_{n=1}^{N}\pi_t^{(n)}s_t^{(n)}$

(5)Re-sampling:The process of re-sampling which replace some particles is to resolve the degeneration problem of particle .In this process each particle will be judged whether degenerated ,then the degenerated particles be replaced.The result is N particles selected from statement collection S_{t-1} according to weight $\pi_{t-1}^{(n)}$ and each particle weight deassign $1/_N$.

(6)Updating model:Calculating target color histogram distribution in the tracking outputposition as the next frame color histogram distribution. Re-calculation Bhattacharyya coefficient between the current target color histogram distribution and the previous then updating the next frame weight $\pi_t^{(n)}$.

3 The parallel particle filtering target tracking algorithm

3.1 The parallel idea of the particle filtering target tracking algorithm

The particle filtering target tracking algorithm include six process :the initialization of the particle,re-sampling, the prediction of particle statement, the calculation of particle weigh, the estimation of tracking, updating model.Because of the data inheritance between these processes ,each process is not independent on the whole,so the whole process can not use MapReduce model to parallel.The each of particle in each process is independent and have not data dependence ,so this principle provide the possibility for MapRedunce paralleling the the particle filtering target tracking algorithm.

3.2 The parallel design for the particle filtering target tracking algorithm

The particle filter target tracking algorithm parallelization process is shown in Fig.1, the following will detail every Job. Attention should be paid to the every Job that is executed serially .

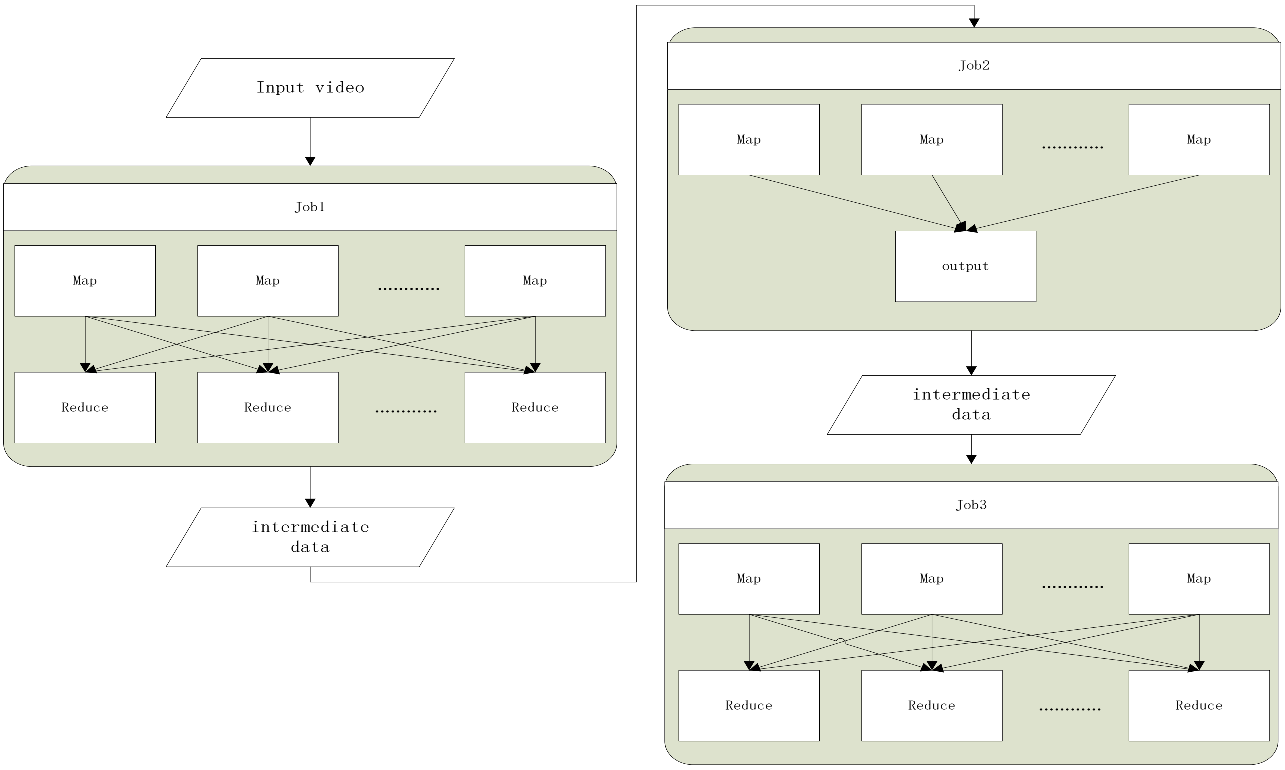


Fig.1 The parallel particle filter target tracking algorithm flow chart

(1)The parallel initialization of the particle : The initialization process of the particle initialization is to initialise for every particle. Each of particle is independent which provide a condition for parallel initialization. In the Hadoop cloud platform using the MapReduce parallel framwork to parallel initialise each particle in the Map function of the Job1.The input of the Map function is key-value pair(key, value),the key is denoted the name of particle and the value is denoted the corresponding initialization value include of the position and speed and weight of the particle.The function of Map is initializing the particle and the output of Map is collecting the value based the same key which is the Map input,so the output of the Map is $\left(\mathrm{key},\mathrm{list}(\mathrm{value})\right)$and the key is denoted the ID of particle and the $\mathrm{list}(\mathrm{value})$ is all the particls initialization.

(2) The parallel prediction of particle statement: Because the currency prediction statement value of each particle only relate to itself previous statement, on a moment with the other particles and this time there is no correlation, so between particles are independent of each other.In this process, the input$\left(\mathrm{key},\mathrm{list}(\mathrm{value})\right)$ of the function reduce in the Reduce of MapReduce is the last step output of the map function in the Map of MapReduce and collecting the same key value ,then parallel calculating the particle statement prediction based the statement transfer equation in the reduce function.

(3) The parallel claculation of particle weight: Although in the process of calculating Bhattacharyya coefficient each particle is compared with the color histogram of the target model,each particle is only read operation on the target model color histogram and no write operations at the same time.So this process can parallel process.In the last reduce step,based on above formulas to parallel calculate each particle weight in the Reduce Function,and output the Reduce Function result in the step .The output is $(\mathrm{key},\mathrm{value})$, the key is denoted the ID of particle and the value is the particle weight.

(4)The parallel estimation of tracking: In this parallel step ,the value of the last step Redcue Function output is used to calculate the sum of the particle weight $\sum_{i=1}^{N}\pi_i$.The result as a middle value which is used to parallel normalize and weight the particle collection for each particle in the Map Function of the Job2 map step .The result of Map Function output is the tracking estimation,so there is no reduce step in the Job2 process.

(5)The parallel Re-sampling: In this parallel setp,the parallel re-sampling in the Map Function of the Job3 map step ,in the Map Function,every particle is parallel processed,and the middle value$\sum_{i=1}^{N}\pi_i$ saved in last two steps is used to calculate the normalized cumulative probability c_{t-1}^{k}.The output is key-value pair$(\mathrm{key},\mathrm{value})$, the key is denoted the ID of particle and the value is the $(s_t,\ \pi)$ data structure ,the π is deassigned $1/_{N}$.

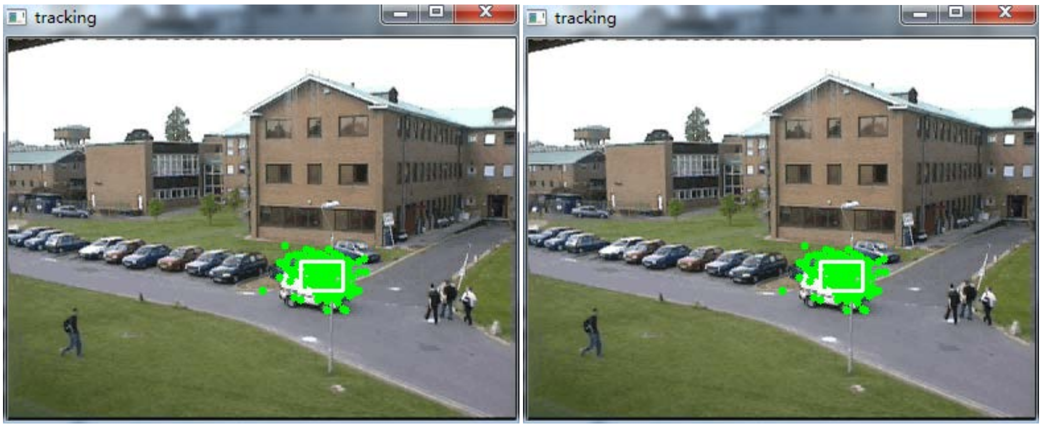
(6) The parallel updating model:In the Job3 reduce step ,the target model is parallel updated.The output of the Reduce Function is the key-value pair（key， value）, the key is denoted the ID of particle and the value is weight value π_t which have been updated already.

4.Experiments

4.1 The parallel particle filter target tracking algorithm experiment

In order to verify the effectiveness of the parallel strategy for the parallel particle filter target tracking algorithm, this paper will compare to the the traditional serial particle filter target tracking algorithm with the algorithm proposed in this paper for the simulation result and at the same time in order to verify the execution efficiency of the parallel algorithm proposed ,execution time will be compared for the both algorithm. Experiment is divided into five times, each time the selection on the number of particles, were 200, 400, 800, and in order to reduce experimental error, each experiment two algorithm's execution, 5 times the execution time averaged comparison. Test time starts from the initialization of particle, including the whole process of particle filter, until the end of the test video, test execution time, time is pure particle filter algorithm testing output including target tracking image output and execution time.

As shown in Fig.2, the picture is when the population is 200, the results of the particle number for other situations as well, the results of the two algorithms, it can be seen that the parallel particle filter target with algorithm can meet the target tracking, to verify the effectiveness of the proposed parallel strategy.



（a）The serial particle filter target tracking effect and（b）Based on the Hadoop cloud platform parallel particle filter target tracking effect
Fig.2　200 the tracking effect of the particles

Table1 recorded the serial and parallel algorithm proposed in this paper under different number of particles in the implementation of 5 this time and calculate the average and then calculate the parallel efficiency increase rate.

The result of the Table1 shows with the number of particle increasing the execution efficiency of the parallel algorithm can increase dramatically, the reason is that the proposed parallel algorithm is executed in the Hadoop cloud platform ,so it can use all resource of the platform to execute the parallel algorithm .When the number of particle is small for example the 200 , the communication time of each node is a big proportion in all execution time.While with the nmber of particle increasing the coummunication time is a　small proportion.

Table1 The execution time comparison for serial and parallel algorithm

		200 particles（test1）			400 particles（test2）			800 particles（test3）	
The times of experiment		The serial execution time	The parallel execution time		The serial execution time	The parallel execution time		The serial execution time	The parallel execution time
1		11.52	10.26		13.33	11.24		20.24	15.23
2		11.34	10.13		15.67	13.79		20.68	15.31
3		10.98	10.59		13.98	12.51		21.62	14.06
4		11.68	11.04		14.02	11.39		19.79	17.79
5		11.42	11.12		13.87	12.45		17.22	16.25
The mean value		11.388	10.628		14.174	12.276		19.91	15.728
The increasing rate		6.67%			13.39%			21.01%	

4.2 The cluster scale experiment

In order to survey the influence of the number of data node of the Hadoop cloud platform to the execution efficiency of the proposed parallel algorithm, the number of data node will be 2,4,6,8 and the number of particle is the same (400).The result of simulation is shown in Fig.3.

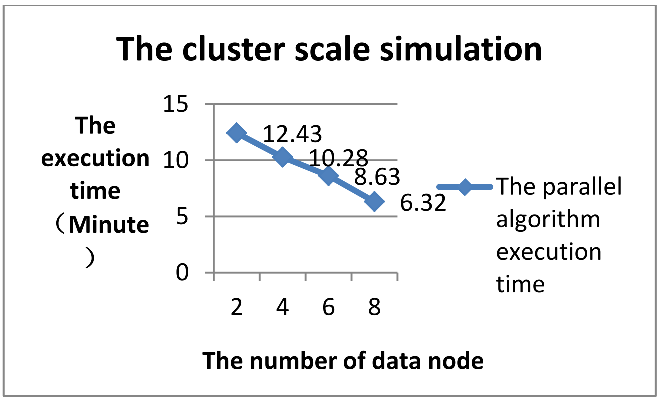


Fig.3 The cluster scale experiment

The result of the figure 3 shows with the number of data node increasing the execution time of the parallel algorithm can decrease dramatically, the reason is the resource of the cloud paltform increasing dramatically.

5. Conclusion

In this paper we proposed the Parallel Particle Filtering Target Tracking Algorithm Based on Hadoop. The algorithm by introducing the idea of particle parallel computing and cloud computing platform is able to handle huge amounts of data on large clusters.This solves the problem that the one server limited resource which including memory and cpu and I/O lead to expense too much time to execute the traditional serial algorithm .

The parallel particle filter target tracking algorithm has good parallelism and scalability, along with the rising of the cluster scale it is able to process massive amounts of surveillance video data in a given time while the existing serial algorithm can not achieve.

References

[1]Wikipedia. Cloud Computing [EB/OL]. http://en.wikipedia.org/wiki/Cloud_computing.

[2]Armbrust M, et al. 2009. Above the Clouds: A Berkeley View of Cloud Computing [R].Department of ElectricalEngineering and Computer Sciences, University of California atBerkeley, Report No.UCB/EECS-2009-28, CA, USA, 2009.

[3]Apache.Hadoop[EB/OL].http://hadoop.apache.org/.2012-10-20.

[4]White T. Hadoop: The definitive guide[M]. " O'Reilly Media, Inc.", 2012.

[5]Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system[C]//Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010: 1-10.

[6]Robert T. Collins, Alan J. Lipton, Takeo Kanade. A system for surveillance andmonitoring[R]. Tech. report CMU-RI-TR-00-12, Robotics Institute, Carnegie MellonUniversity, May, 2000.

[7]J. M. Ferryman, S. J. Maybank, A. D. Worrall. Visual Surveillance for MovingVehicles[C]. International Journal of Computer Vision, 2000,37(2): 187–197.

[8]Neal D, Rahman S. Video surveillance in the cloud?[J]. The International Journal of Cryptography and Information Security (IJCIS), 2012, 2(3).

[9]Bhandarkar S M ， Zheng Wen-long. Face detection and tracking using a Boosted Adaptive Partical Filter[J]. Journal of Visual Communication and Image Representation ， 2009,20（1）： 9-27.

[10]Dean J ， Ghemawat S. MapReduce : Simplified data processing on large clusters [J]. Communications of the ACM ， 2008,51（1）： 107-113

[11]Maskell S. and N. Gordon.A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking Target Tracking [J]. Algorithms and Applications,2001,174(2):1-15.

[12]Hossain M S, Hassan M M, Qurishi M A, et al. Resource allocation for service composition in cloud-based video surveillance platform[C]//Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on. IEEE, 2012: 408-412.

[13]Yu R, Zhang Y, Gjessing S, et al. Toward cloud-based vehicular networks with efficient resource management[J]. Network, IEEE, 2013, 27(5): 48-55.

[14]Geronimo L D, Ferrucci F, Murolo A, et al. A parallel genetic algorithm based on hadoop mapreduce for the automatic generation of junit test suites[C]//Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on. IEEE, 2012: 785-793.

[15]Ekanayake J, Li H, Zhang B, et al. Twister: a runtime for iterative mapreduce[C]//Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. ACM, 2010: 810-818.