Online Appendix to: Progressive Motion Vector Clustering for Motion Estimation and Auxiliary Tracking

KE CHEN, ZHONG ZHOU, and WEI WU, Beihang University

This online appendix contains examples for the progressive motion vector clustering algorithm and the parallel matching process as well as additional results for the motion estimation experiment and tracking results for different methods.

A. AN EXAMPLE FOR THE PROGRESSIVE MOTION VECTOR CLUSTERING ALGORITHM

The clustering procedure is illustrated by an example in Figure A.1. Suppose C_1 is the existing cluster and $\{q, s, t, u, \mathrm{v}, w\}$ are the new motion vectors. Initially, C_1 has only one member, that is, $C_1 = \{p\}$ and $r_{C_1} = p$ (Figure A.1(a)). The clustering algorithm assigns the new motion vectors to C_1 and then creates new clusters for the unassigned vectors (Figure A.1(b)). Since $q \rightarrow p$, q satisfies the Rule of Directly Reachable Clustering and it is assigned to C_1. Since $s \rightarrow q$ and $w \rightarrow q$, $s > p$ and $w > p$ through q. According to the Rule of Indirectly Reachable Clustering, s and w are also assigned to C_1. $\{t, u, \mathrm{v}\}$ cannot be assigned to C_1. The algorithm chooses t to create C_2 and assign u to C_2, that is, $C_2 = \{t, u\}$. Then, it create C_3 for v, that is, $C_3 = \{v\}$. Because $s > p$ and $s \rightarrow v$, s becomes a member of C_3 under the Rule of Membership Change (Figure A.1(c)). Supposing N_s is 8 and N_v is 2, $cost(C_3, s) = 0.2$ and $cost(C_3, v) = 0.8$. Under the Rule of Representative Vector Selection, s become the representative vector of C_3 (Figure A.1(d)) because $cost(C_3, s) < cost(C_3, v)$. Finally, the algorithm stops and gets the clustering results.

B. AN EXAMPLE FOR THE PARALLEL MATCHING PROCESS

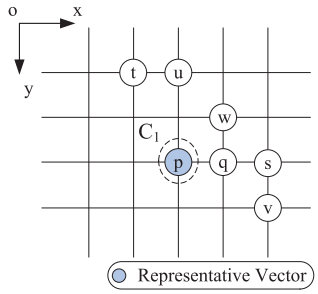
Figure B.1 illustrates the procedure of the parallel matching process for a block. As shown in Figure B.1(a), with the clustering based predictors, all the candidates in the three predicted search areas are tested by $3 \cdot s^2$ CUDA threads. Assume the distortion of the local MBD point is larger than the threshold T_2. The parallel line search algorithm is performed to find the motion vector, as shown in Figure B.1(b). First, all the candidates in line -1, line 0 and line 1 are tested by w CUDA threads ((1) in Figure B.1(b)). The MBD point in this step is in line 1, which is the boundary of the searched lines. Then, the candidates in line 2 are tested in parallel ((2) in Figure B.1(b)). After that, the MBD point is in line 2, and the candidates in line 3 are test ((3) in Figure B.1(b)). Finally, because no candidates in line 3 have lower distortion than the MBD point in line 2, the parallel line search stops and finds the motion vector.

C. ADDITIONAL MOTION ESTIMATION EXPERIMENT RESULTS

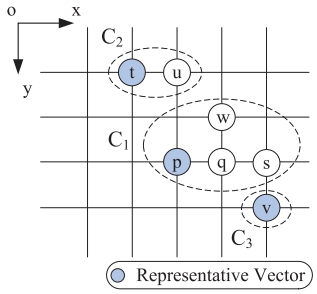
We test our clustering based search algorithm under different block distortion thresholds (i.e., $T_2 = 1024$, 1536 and 2048). Table C.I shows the MSE results as well as the time cost and the searched candidates. The time cost in Table C.I is the sum of the block matching time and the clustering time.

The MSE value only rises by 1.7 on average when the block distortion threshold increases from 1024 to 2048. However, the time cost and the searched candidates

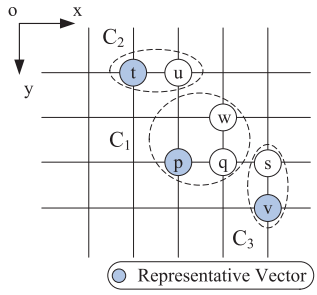
DOI: http://dx.doi.org/10.1145/2700296



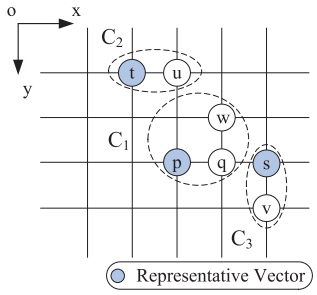
(a) The existing cluster C_1 and the new motion vectors $\{q, s, t, u, v, w\}$.



(b) Assign the new motion vectors to C_1 and create new clusters for the unassigned vector.

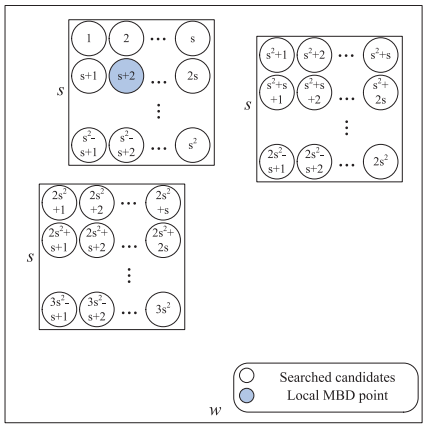


(c) Modify the membership of s from C_1 to C_3.

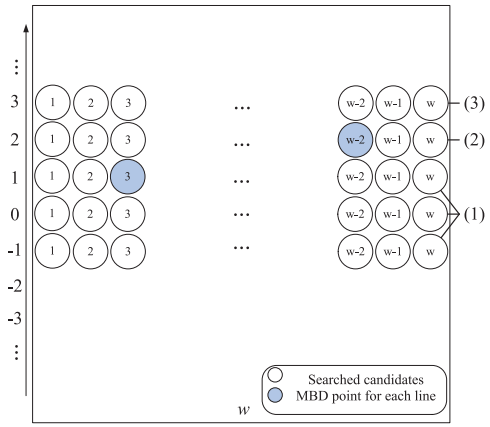


(d) Select s as the representative vector of C_3.

Fig. A.1. The procedure of the progressive motion vector clustering algorithm.



(a) The candidates in the predicted search areas are tested by 3 s^2 CUDA threads.



(b) The candidates in the same search line are tested by w CUDA threads in the parallel line search.

Fig. B.1. The procedure of the parallel matching process.

greatly decrease, when the block distortion threshold becomes larger. The block distortion threshold affects the number of the blocks to be searched only in the three predicted areas and the possibility to use the line search (the block matching process in Algorithm 2). The searched candidates will be fewer with a bigger threshold. We choose 2048 as the block distortion threshold (T_2) of our search algorithm, because it saves nearly 60% of the time cost only with a negligible MSE increase.

Table C.I. Performance of Our Search Algorithm under Different Block Distortion Thresholds

	Block Distortion Threshold								
	1024			1536			2048		
Video Sequence	MSE[a]	TC[b]	SC[c]	MSE[a]	TC[b]	SC[c]	MSE[a]	TC[b]	SC[c]
Aspen	19.4	156.33	44	19.9	99.39	27.4	21.1	77.78	21.1
Blue Sky	29.7	177.91	50.5	30.1	118.98	33.3	32.16	77.87	21.3
Park Joy	282	148.04	99	282.6	122.24	81.5	283.58	89.96	59.6
Ducks Take Off	102.3	163.81	109.1	102.57	126.03	83.8	102.83	47.49	31.2
In To Tree	31.9	117.95	76.5	33.1	42.96	27.3	35.64	29.89	18.7
Station 2	11.96	166.45	47.4	12.04	97.1	25.2	12.67	74.93	18.2
Rush Hour	31.65	173.26	49.1	31.87	128.28	36	32.1	104.36	27.1
Tractor	59.5	339.74	99.3	61.31	224.12	63.5	62.27	121.03	32.4
Average	71.05	180.43	71.9	71.69	119.88	47.3	72.79	77.91	28.7

[a]Mean-Square Error. [b]Time Cost per frame (ms/frame). [c]Searched Candidates per block.

Table C.II. Performance of Our Search Algorithm with Different Clustering Algorithms

	Clustering Algorithm								
	Our clustering algorithm			K-means			DBSCAN		
Video Sequence	MSE[a]	TC[b]	SC[c]	MSE[a]	TC[b]	SC[c]	MSE[a]	TC[b]	SC[c]
Aspen	21.1	77.78	21.1	19.64	340.2	97.6	19.35	387.68	112.9
Blue Sky	32.16	77.87	21.3	33.06	324.53	93.3	33.24	363.85	107.7
Park Joy	283.58	89.96	59.6	329.19	163.37	109.4	338.31	175.11	119.4
Ducks Take Off	102.83	47.49	31.2	103.38	120.16	79.8	103.49	131.69	89.6
In To Tree	35.64	29.89	18.7	33.01	128.01	83.1	32.48	143.81	96.1
Station 2	12.67	74.93	18.2	13.51	328.74	93.5	13.23	352.73	103.2
Rush Hour	32.1	104.36	27.1	32.9	367.58	107.2	33.12	397.32	119.3
Tractor	62.27	121.03	32.4	63.45	345.73	98.5	63.87	376.55	109.4
Average	72.79	77.91	28.7	78.52	264.79	95.3	79.63	291.09	107.2

[a]Mean-Square Error. [b]Time Cost per frame (ms/frame). [c]Searched Candidates per block.

For additional comparison, we substitute the progressive motion vector clustering algorithm with K-means [MacQueen 1967] and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [Ester et al. 1996]. In the experiment, Davies-Bouldin index [Davies and Bouldin 1979] is used as a guideline to the parameter k of K-means, that is, the number of clusters. Table C.II shows the performance of our search algorithm with different clustering algorithms. When we use K-means and DBSCAN in our search algorithm, the time cost and the number of the searched candidates rise significantly. The clustering results of K-means greatly rely on the parameter k. Although Davies-Bouldin index helps to determine the parameter k, it is still very difficult for K-means to discover the appropriate clusters of motion vectors. DBSCAN is designed to discover the clusters with arbitrary shapes. It discovers the clusters based on the density in the neighborhood. As Table C.II shows, the clustering results of DBSCAN are not suitable for blocking matching. K-means and DBSCAN cannot provide efficient predictors. Therefore, most of the blocks need to use the line search (the block matching process in Algorithm 2) to find their motion vectors, and the speed performance of our search algorithm significantly degrades.

D. ADDITIONAL TRACKING RESULTS OF DIFFERENT METHODS

Figure D.1 and Figure D.2 respectively illustrate the tracking results of road sign #2 and road sign #3 with different tracking methods, including our tracking method, the original mean-shift tracking [Comaniciu et al. 2000], the adaptive color-based particle

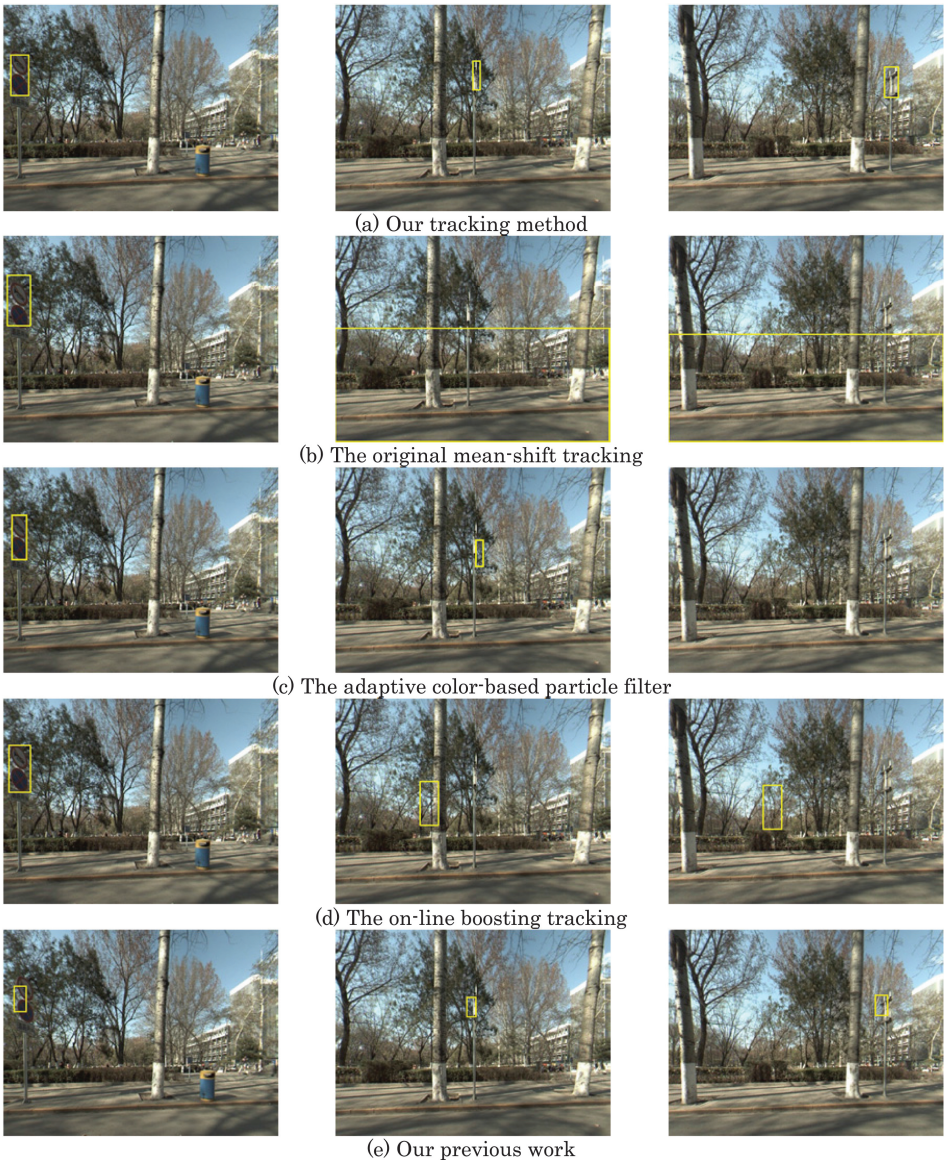


Fig. D.1. The tracking results of Road Sign #2 with different tracking methods. The first, second, and third columns show frame #4, frame #50, and frame #77, respectively.

filter [Nummiaro et al. 2003], the on-line boosting tracking [Grabner and Bischof 2006] and our previous work [Zhou et al. 2012]. Due to severe appearance changes and occasional disappearances, the original mean-shift tracking, the adaptive color-based particle filter and the on-line boosting tracking fail to track and lose the target road signs. Compared with our previous work, the proposed tracking method is shown to be more accurate in terms of target position.

(a) Our tracking method

(b) The original mean-shift tracking

(c) The adaptive color-based particle filter

(d) The on-line boosting tracking

(e) Our previous work

Fig. D.2. The tracking results of Road Sign #3 with different tracking methods. The first, second, and third columns show frame #2, frame #92, and frame #126, respectively.