LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-200271

Block Matching for Object Tracking

A. Gyaourova, C. Kamath, S.-C. Cheung

October 14, 2003

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Block matching for object tracking

Aglika Gyaourova

Department of Computer Science, University of Nevada, Reno

aglika@cs.unr.edu

Chandrika Kamath and Sen-ching Cheung

Center for Applied and Scientific Computing

Lawrence Livermore National Laboratory

kamath2@llnl.gov and cheung11@llnl.gov

Abstract

Models which describe road traffic patterns can be helpful in detection and/or prevention of uncommon and dangerous situations. Such models can be built by the use of motion detection algorithms applied to video data. Block matching is a standard technique for encoding motion in video compression algorithms. We explored the capabilities of the block matching algorithm when applied for object tracking. The goal of our experiments is two-fold: (1) to explore the abilities of the block matching algorithm on low resolution and low frame rate video and (2) to improve the motion detection performance by the use of different search techniques during the process of block matching. Our experiments showed that the block matching algorithm yields good object tracking results and can be used with high success on low resolution and low frame rate video data. We observed that different searching methods have small effect on the final results. In addition, we proposed a technique based on frame history, which successfully overcame false motion caused by small camera movements.

1 Introduction

The growth of computer memory and processor speed has caused the expansion of automated systems for everyday life applications. Computer vision systems is one example of such intensive computations systems, whose use becomes more feasible with the advances in computing power. The area of computer vision aims at processing visual information, which in general, is characterized by large size and complex structure. Another reason that hinders the development of computer vision is the lack of complete understanding of the way human beings process visual information. However, computer vision systems have been incorporated in many real life applications (e.g. surveillance systems, medical imaging, robot navigation and identity verification systems).

Object tracking is a key computer vision topic, which aims at detecting the position of a moving object from a video sequence. Motion detection was initially developed for video encoding. Motion vectors can be used to predict changes in the scene between two or more video frames. The size of video data is reduced by encoding only the current frame and the motion vectors, from which several future frames can be recovered. Motion detection for object tracking has distinctly different requirements. In object tracking, there is a need to interpret the information given by the motion vectors. In compression algorithms all background changes, moving objects, and camera motion need to be encoded, but the meaning of the motion vectors is not important. Object tracking, on the other hand, must detect only the moving objects and filter out the noise, small camera motions, and insignificant other motions (e.g. rain drops or falling leaves). Grouping or some other kind of association of the motion vectors into a meaningful scene representation is the primary goal of motion detection for object tracking.

Typical problems in scene analysis algorithms are: (1) Noise with characteristics similar to actual scene objects (for example, falling snow). (2) Objects that look as background or noise. This might happen when an object enters a shadowed region or a cloud of fog. (3) Two or more objects, which appear as a single object and vice versa. The former occurs when the objects are, or appear to be, close in space relative to the viewing point and the latter, when only parts of a single object can be detected. The correct interpretation of a visual scene often requires prior knowledge about the scene content. This knowledge can be utilized by creating a model or making certain assumptions.

In this work we investigated the capabilities of the block matching algorithm (BMA) applied for the problem of vehicle tracking for the purposes of road monitoring. Our assumption for the process of video capturing is a motionless airborne camera. Although, our experiments were made with tracking road vehicles (i.e. cars, trucks, motorcycles and bicycles) the algorithm is general enough to be applied for tracking people or other moving objects.

The block matching algorithm and its parameters are outlined in Section 2. The results from the general BMA and its application on low resolution and low frame rate video data are in Section 3. More complicated block matching techniques and their performance on object tracking are discussed in Section 4. Finally, a discussion of the results and some conclusions are presented in Section 5.

2 The Block Matching Algorithm (BMA)

The block matching algorithm is a standard technique for encoding motion in video sequences [8]. It aims at detecting the motion between two images in a block-wise sense. The blocks are usually defined by dividing the image frame into non-overlapping square parts. Each block from the current frame is matched into a block in the destination frame by shifting the current block over a predefined neighborhood of pixels in the destination frame. At each shift, the sum of the distances between the gray values of the two blocks is computed. The shift which gives the smallest total distance is considered the best match.

In the ideal case, two matching blocks have their corresponding pixels exactly equal. This is rarely true because moving objects change their shape in respect to the observer's point of view, the light reflected from objects' surface also changes, and finally in the real world there is always noise. Furthermore, from semantic point view, in scenes containing motion there are occlusions among the objects, as well as disappearing of objects and appearing of new ones. Despite the problems of pixel by pixel correspondence, it is fast to compute and is used extensively for finding matching regions. Some of the most often used matching criteria based on pixel differencing are mean absolute distance (MAD), mean squared distance (MSD), and normalized cross-correlation (NCC) [7].

2.1 Block size

Choosing the right block size is not a trivial task. In general, bigger blocks are less sensitive to noise, while smaller blocks produce better contours. Certainly, the leading factor for choosing the block size is the size of the objects that need to be tracked. The next two factors are the amount of noise in the video frames and the texture of the objects and the background. The texture of the objects leads to the so called *aperture problem*.

The *aperture problem* appears in situations where the objects of interest have uniform color. The blocks which are inside the objects do not appear as moving because all of the blocks around them have the same color. When the uniform color regions consist of fewer blocks, there is a greater chance that their motion will be detected because some overlapping with non-uniform color regions is likely. A bigger block size can be used to overcome the *aperture problem*.

2.2 Search region

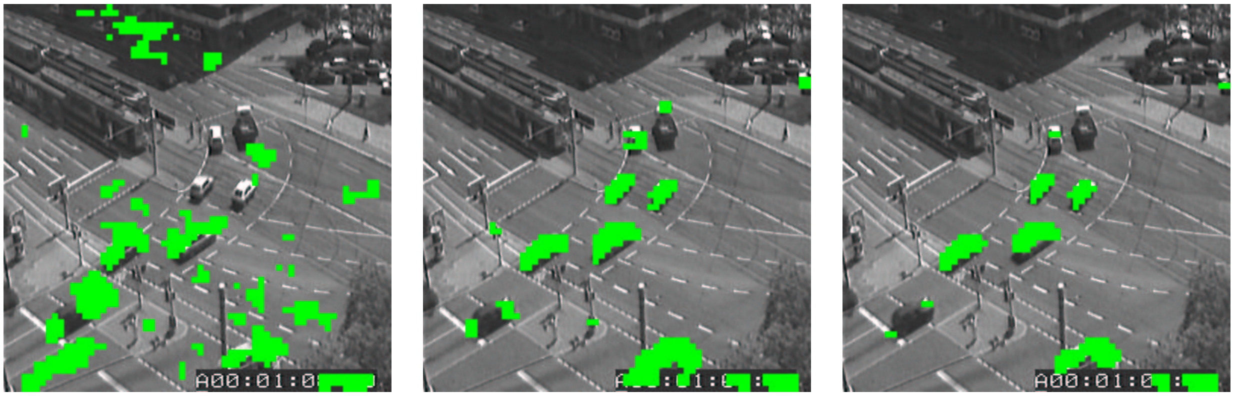
The size of the search region is important for finding the right match. Unfortunately the computational load grows fast (as a power of two) with the growth of the search area. Several sub-optimal search algorithms exist, which search far from the center of the block but only in a direction, which is predicted and adapted during the search itself (see Sect. 4).

3 Experiments

In our experiments we used the mean absolute difference (MAD) similarity measure:

$$MAD = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} |A(i,j) - B(i.j)|$$

We made one modification to the conventional block matching algorithm in order to attenuate the effect of noise – a threshold which defines the maximum distance between two corresponding blocks under which they



(a) (b) (c)

Figure 1: The effect of the *zero motion bias* threshold (T). (a) Motion detection without using *zero motion bias* threshold (T=0). (b) After applying the threshold (T=14). (c) After applying the threshold and using a larger value (T-18).

are considered matching. We name this threshold - the *zero motion bias* threshold. Finding the best match for a block is computed as follows. The MAD between the current block and the corresponding block (without displacement) in the target frame is computed first. If this MAD is under the *zero motion bias* threshold then the current block is considered as motionless and no more matches are evaluated. This technique results in a significant improvement in tracking moving objects - big amount of false motion is cleared away, Fig. 1. An implied consequence of this threshold is that for all blocks for which the *zero motion bias* does not hold, a non zero motion vector is forced by choosing the minimum of the blocks in the search area (excluding the block without displacement).

In our experiments, we used the video data collected by KOGS/IAKS Universität Karlsruhe, Germany [5].

3.1 Basic block matching

The basic block matching technique gives very good results when the motionless camera assumption hold, Fig. 1.(b). However, small camera motions and snowfall cause a lot of noisy motion to appear, Fig. 2.

In these experiments we have used noncontinuous search step S. During the matching process the blocks are shifted only with the number of pixels of the search step. This reduces the number of searched positions to only nine, compared to $(2S+1)^2$ for the exhaustive search.

A desired feature of object tracking systems is to be able to work with small size of stored data. The size of the data can be reduced by the use of low resolution images and by reducing the frame rate, which are the focus of the next two sections.

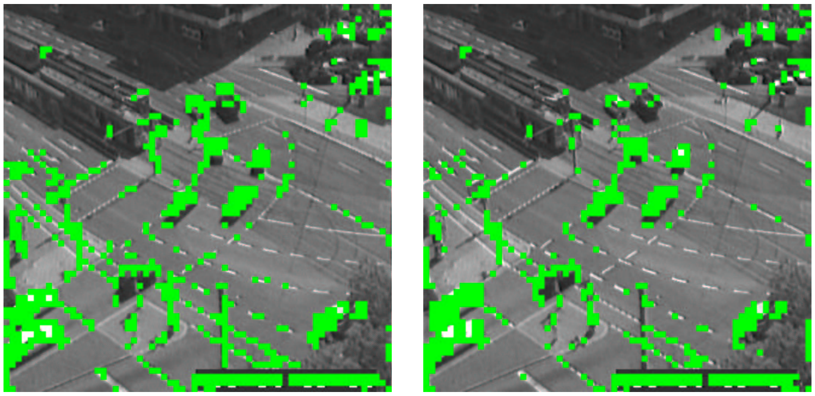


Figure 2: Two frames showing the effect of small camera movements.

3.2 Block matching in low resolution video

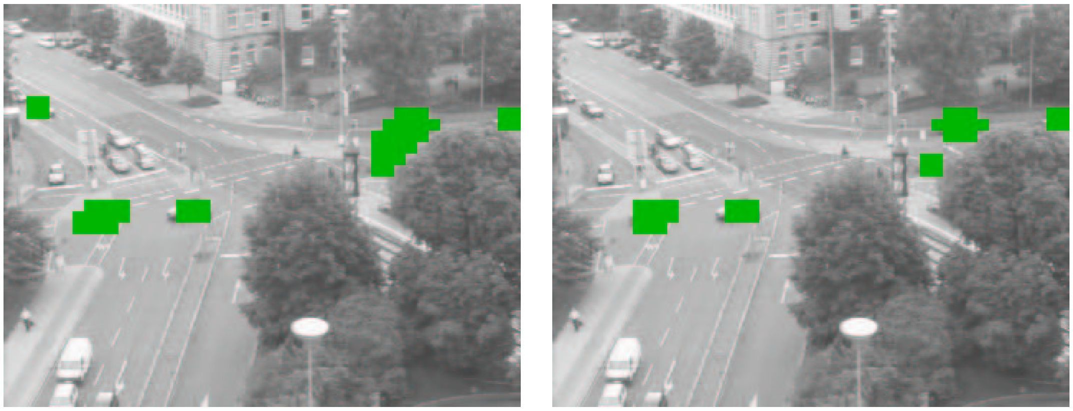
We investigated the behavior of the block matching algorithm when applied on low resolution images. Three types of low resolution images were compared – subsampled, smoothed by using a mean filter followed by subsampling, and images produced by applying Daubechies' quadrature mirror filter (QMF) [2].

Subsampling produced the worst results of all - it was often unable to detect some of the cars, or it would merge two or more cars into one object, Fig. 3. The mean filtering and the QMF filtering produced comparable results, Fig. 4. The only aspect in which the mean filtering yielded worse results, was that it omitted moving objects with small sizes, Fig. 5. The motion detection gave poor results when the image size was reduced by a factor of 16 in each dimension. We also observed that motion detection from lower resolution image frames were more sensitive to camera motions.

3.3 Block matching in low frame rate video

In general, a lower frame rate is equivalent to an increase in the speed of the moving objects. BMA based on low frame rate data resulted in motion vectors exhibiting long traces behind the cars, Fig. 6. The reason for this effect is that both the new position and the old positions of the cars yield a non-zero motion vectors. The new positions are detected because in the place of background in the current frame there are cars in the destination frame. Similarly, the old positions of the cars are detected because background in the destination frame appear in the places of the cars in the current frame.

To be able to obtain distinct objects during tracking, the frame rate should be coordinated with the speed of the objects. However, small reductions of the frame rate can be done in almost any case without any adverse effects.



(a) (b)

Figure 3: Motion detection based on low resolution image frames obtained by subsampling. The images were subsampled by a factor of two in each direction. The resulted images have size equal to 1/4 of the original frame size. The results based on two different values for the *zero motion bias* threshold T are shown. (a) T=6 and (b) T=7.

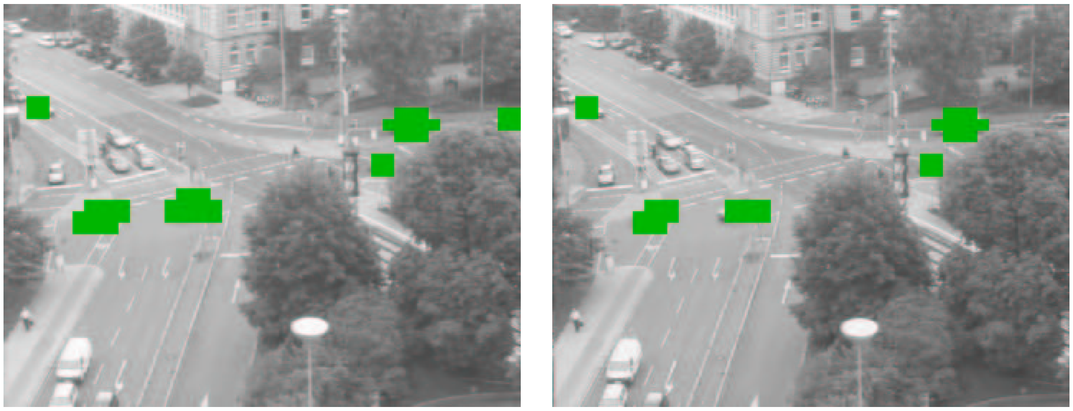
4 Extending the block matching algorithm

Some of the problems that cannot be handled by the basic block matching algorithm are the *aperture problem* (see Sect. 2.1), global or local changes in lighting, and camera motions. However, simple extensions to the algorithm and postprocessing help to reduce the effect of such problems. Two general directions for improvement in the matching results are: (1) enlarging the number of frames used to predict the motion in one frame (i.e. having longer history) and (2) more sophisticated search and matching schemes while using just two frames to do the prediction.

4.1 Top-down approach

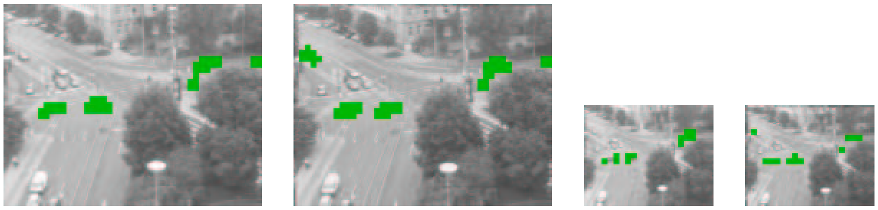
The top-down approach uses different block sizes in a manner similar to image pyramids [1]. The basic BMA is applied using a large block size. The goal is to have the entire object covered only with few blocks. Then the blocks size is reduced by half in each dimension and the block matching is applied again but only for those blocks for which motion was detected during the first step. This can be repeated as many times as desired or until the block size becomes 1×1 pixels.

This approach aims at filtering the noise by using big initial value for the block size followed by improvement of the car contours achieved by using small final values for the block size, Fig. 7.



(a) (b)

Figure 4: Motion detection in low resolution images. (a) Obtained by mean filtering and subsampling. (b) Obtained by QMF filtering. The size of these images is 1/4 of the original images size (half the original size in each direction).



(a) (b) (c) (d)

Figure 5: Motion detection in very low resolution images – the small cars in the upper left corner are omitted in the mean filtered images (a)&(c) but not in the QMF filtered ones (b)&(d). The filtering is applied two times in the (a)&(b) and three times in the (c)&(d) cases.

4.2 Three step search (TSS)

An efficient way to reduce the number of searches and at the same time to reach over a larger distance is an interesting topic because it might improve the matching results while saving computations. There are several sub-optimal search methods used in the image coding area: three step search (TSS) [4], four step search (FSS) [8], 2D logarithmic search [3], orthogonal search algorithms [6], etc. We chose to test the behavior of the TSS method, mainly because it is one of the algorithms adopted in the MPEG video standard.

The three step search algorithm explores large search area but performs block matching only at a few positions. TSS proceeds as follows. The first step is the same as in the basic block matching having search step bigger than four pixels. At the next step the center of current block is moved to the position of the best match from the previous step and matching is performed again but with smaller search step (usually half the original size). This step can be repeated as many times as desired.

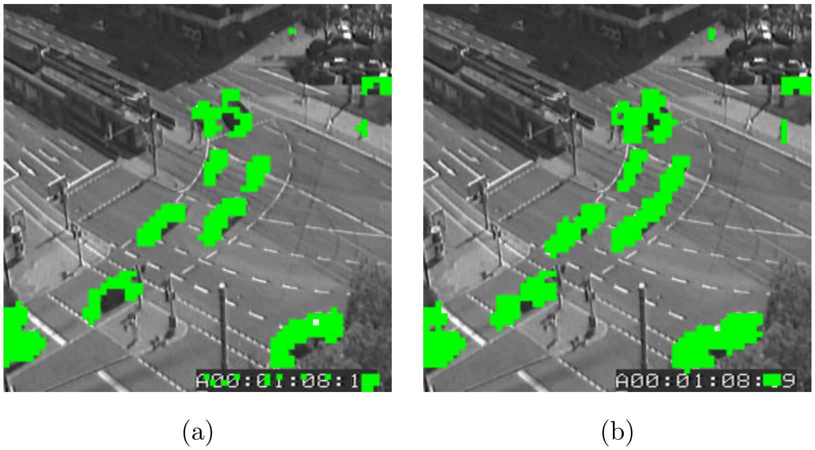


Figure 6: BMA from low frame rate yields trailing-like motion. (a) five skipped frames, which is equivalent to 0.2 frames/second video rate. (b) Ten skipped frames - equivalent to 0.1 frames/second.

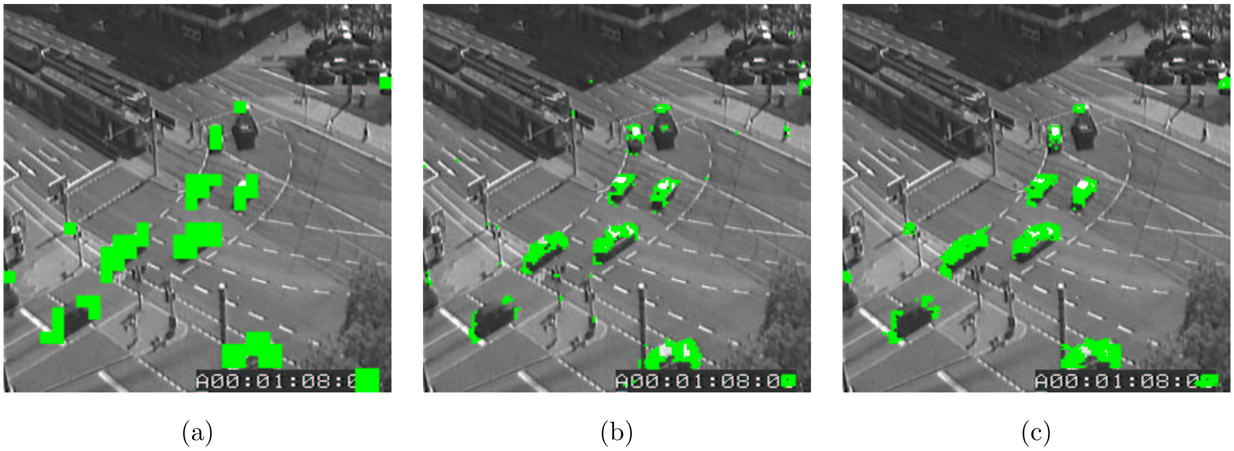
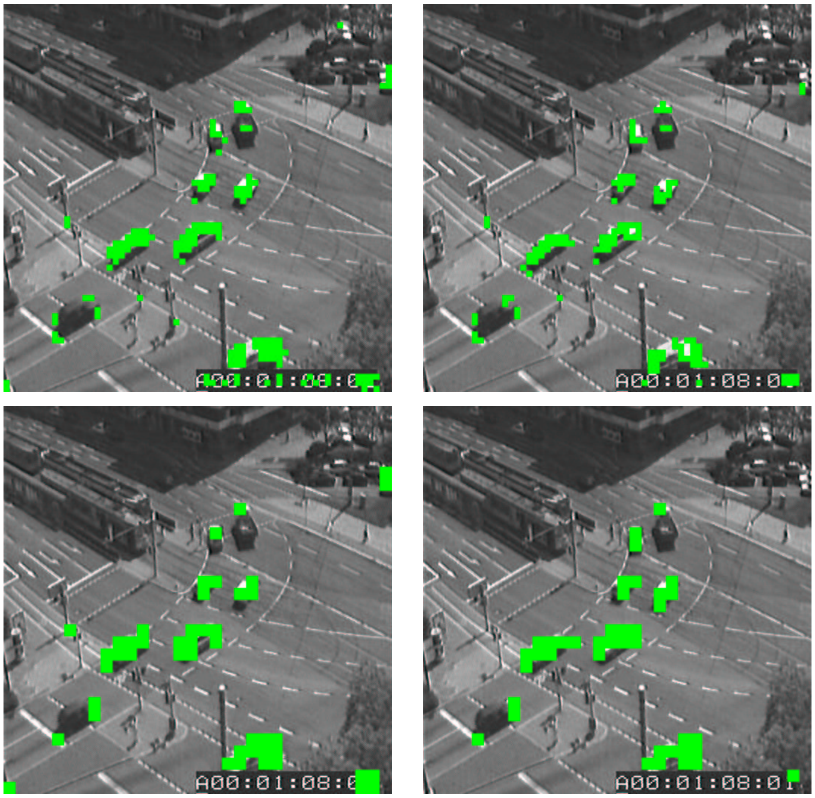


Figure 7: Top-down approach and basic BMA comparison. (a) Basic block matching using 16×16 pixels block size. (b) Basic block matching with blocks size of 4×4 pixels. (c) Top-down approach with initial block size of 16×16 and final block size equal to 4×4 pixels

Our experiments showed that the three step search algorithm yields similar motion detection results as the basic BMA, Fig. 8. The computational load added by TSS is not worthwhile in our case.



(a) (b)

Figure 8: TSS algorithm. The basic BMA (a) and the TSS algorithm results (b). Block size of 8×8 is used for the first row and 16×16 pixels for the second row.

4.3 Frame buffering

The idea for frame buffering was motivated by the fact that the noise at a certain position usually "lives" more than one or two frames. The same behavior is valid for the noise created by small camera motions, which usually persist only in few frames. However, in contrast to noise, which appears in single pixels scattered throughout the image, a small amount of camera motions generates false motion for many image pixels. Blocks matching by looking into several frames together can help in overcoming the noise presence.

We did block matching with frame buffering in three steps:

(1.) At the first step the motion vectors between the current frame and the first frame in the buffer were computed using the basic BMA.

(2.) The centers of the blocks in the current frame were moved according to the motion vectors found in the previous step and the blocks around these new centers are matched with the next frame in the buffer. Step (2) was repeated until no more frames are left in the buffer.

(3.) The final motion vectors were computed by applying logical “and” to the motion vectors computed from the separate steps.

An implied assumption in this approach is that when a block moves to a new position, the block at that new position must have also moved in order to free the space for the first block. It is obvious, that such assumption does not always hold because blocks representing background do not move but just disappear being hidden by the objects in the foreground. This technique, however, works quite well, Fig. 9. In general, there is a certain lower limit on the number of buffered frames needed for the removal of the noise caused by camera motion. This number should be at least one more than the duration of the camera motion. For example, if the camera moves during in two consecutive frames Fig. 9(a), two buffered frames plus the current frame can compensate for noise in the first frame but not in the second one, Fig. 9(b). On the other hand, four or more buffered frames can compensate for this same motion, Fig. 9(c) and (d). When using this technique, the *zero motion bias* threshold can be substantially lower than its usual value because of the logical “and” operation.

A slightly different approach using the same buffering technique is instead of displacing and comparing blocks from the current frame to the blocks of the next frame in the buffer, to do the comparison between each two consecutive frames. In this manner the current frame will be matched with the first frame in the buffer and then each frame in buffer will be matched with the next frame in the buffer. The main idea in this technique is to follow the trajectory of the objects through the frames in the buffer. In contrast, the first technique described aims at using the future frames to remove the noise from the current frame. Although the two different techniques give very similar results, the first version produces less noise, Fig. 10.

5 Discussion

Block matching proved to be a reasonably successfully technique for object tracking, taken its computational simplicity. The application of block matching for low frame rate video, as well as for low resolution video was successful too. These observations are valid when the assumptions about motionless camera and not changing illumination hold.

We came to the conclusion that the size of the search area does not have big influence on the motion detection results. On the other hand, the block size can make a big difference and should be synchronized with the size of the tracked objects. The use of multiple block sizes during the process of motion detection was helpful for finding better contours of the objects but could not overcome the effect of noise.

Besides the general *aperture problem* in object tracking other problems, which occurred in our experiments were noise caused by small camera motions and noise caused by light reflection from metal surfaces (e.g. the pole of the street lights). The second problem can be overcome by the use of postprocessing techniques. We proposed a simple technique using frame buffering which overcome the problem caused by small camera movements.

Frame buffering proved to be able to bring valuable improvements in the motion detection results. It adds

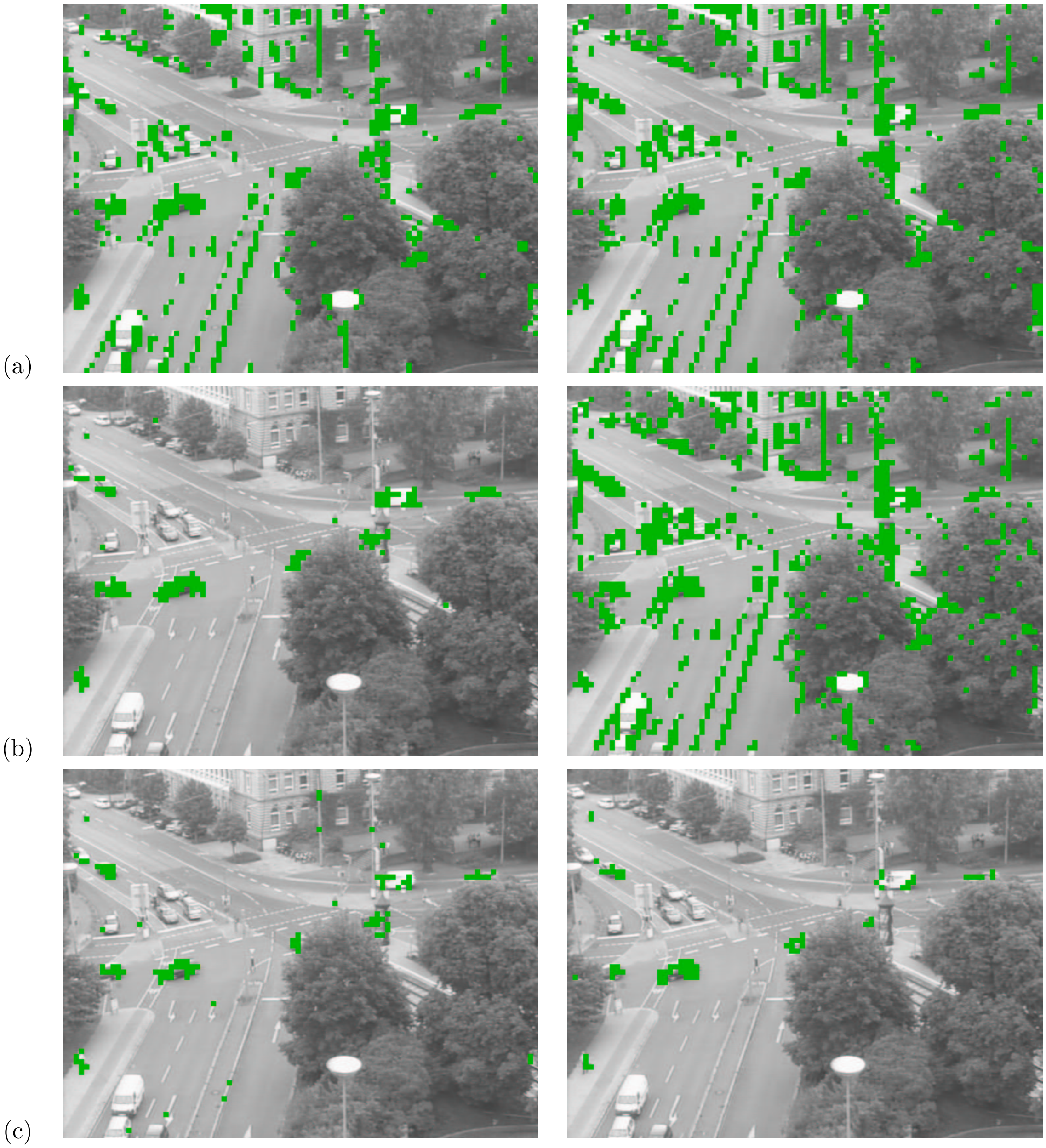
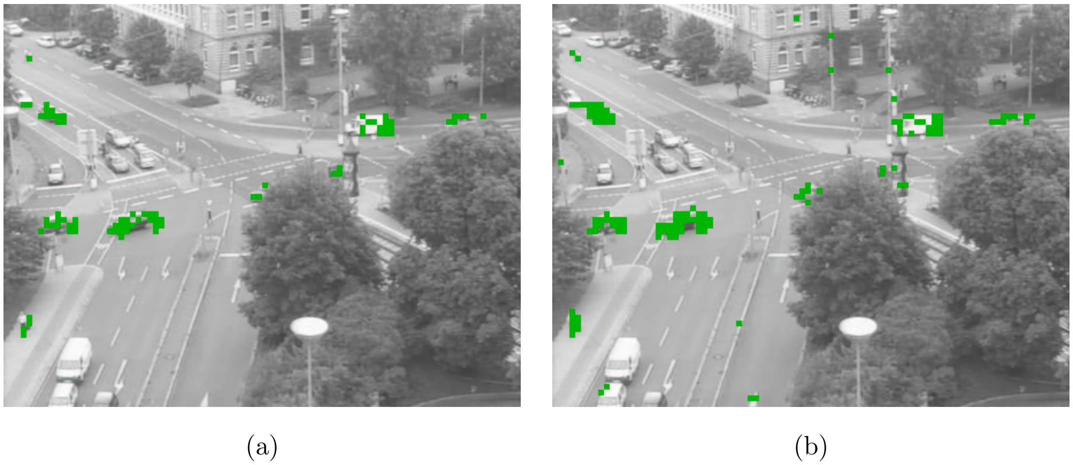


Figure 9: Frame buffering. Two frames containing small camera motion are shown. The basic BMA approach (a), yields a big amount of false motion. Three frame buffering (b), removes the noise in the first frame but cannot completely remove the noise in the second frame. Four frame buffering (c), clears out the noise in both frames.



(a) (b)

Figure 10: Frame buffering. Matching between the current frame and each frame in the buffer (a) produces less noise than matching each two consecutive frames (b).

robustness to the matching algorithm by improving the reliability of the matches. In this way, frame buffering can be used to overcome big amounts of noise that appear in only few frames. Furthermore, it might help in reducing the effect of the *aperture problem* by making use of the fact that a moving object is usually represented by a cluster of blocks. Such clusters should be consistent in subsequent frames.

Acknowledgments: We would like to thank the KOGS/IAKS Universität Karlsruhe for providing their video data for public use.

UCRL-TR-200271. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] E. Alyson and P. Burt. Image data compression with the Laplacian pyramid. Proc. PRIP81, Dallas, Texas (1981) 218–223

[2] I. Daubechies. *Ten lectures on Wavelets.* SIAM, Philadelphia, PA (1992)

[3] J. R. Jain and A. K. Jain. Displacement measurement and its application in interframe image coding, IEEE Trans. Commun., vol. COM-29, Dec. (1981) 1799–1808

[4] T. Koga et al. Motion-compensated interframe coding for video conferencing. Proceedings NTC'81 (IEEE), G5.3.1-5, New Orleans, LA (1981)

[5] KOGS/IAKS Universität Karlsruhe. Available at: http://i21www.ira.uka.de/image_sequences/

[6] C. Manning. Motion Compensated Video Compression Overview.
Available at: http://www.newmediarepublic.com/dvideo/compression/adv08.html

[7] Y. Wang, J. Ostermann and Y. Zhang. *Video Processing and Communications*. Prentice Hall, Signal Processing Series (2002)

[8] J. Watkinson. *MPEG Handbook*. Focal Press (2001)