MOTION SEGMENTATION THROUGH FACTORIZATION. APPLICATION TO NIGHT DRIVING ASSISTANCE

Carme Julià, Joan Serrat, Antonio López, Felipe Lumbreras and Dani Ponsa
Computer Vision Center and Computer Science Department.
Universitat Autònoma de Barcelona.
08193 Bellaterra, Spain.
{*cjulia, joans, antonio, felipe, daniel*}*@cvc.uab.es*

Thorsten Graf
Volkswagen AG Group Research, Electronics.
Wolfsburg, D-38436
thorsten.graf@volkswagen.de

Keywords: driving assistance, motion segmentation, factorization.

Abstract: Intelligent vehicles are those equipped with sensors and information control systems that can assist human driving. In this context, we address the problem of detecting vehicles at night. The aim is to distinguish vehicles from lamp posts and traffic sign reflections by grouping the blob trajectories according to their apparent motion. We have adapted two factorization techniques, originally designed to estimate the scene structure from motion: the Costeira–Kanade and the Han–Kanade, named after their authors. Results on both vehicle existence in the field of view and motion segmentation are reported.

1 INTRODUCTION

We focus our work in detecting vehicles at night from video sequences when they are far away or at a mid-range distance. One possible application would be to define an intelligent lighting system, that includes automatic high and low beam switching. The aim is to distinguish vehicles from lamp posts and traffic sign reflections. To this end, we try to check whether there are two or more motions in a sequence, or just one. In the first case, at least one of the motions must be a vehicle, since the rest correspond to static features and/or other vehicles. In the second case, we can not decide just on the ground of motion, but other cues should be used, like size change over time or location in the image.

Our approach is to group trajectories of feature points according to their apparent motion. In (Megret and DeMenthon, 2002), a taxonomy of spatio-temporal techniques is presented. One of these techniques is *factorization*. This is a theoretically sound method addressing the structure from motion problem: recovering both 3D scene shape and camera motion from trajectories of tracked features. Since the camera motion can be made relative to each of the scene objects, it can be employed for motion segmentation, by grouping those features whose 3D motion is similar.

Factorization has some distinct advantages over other structure from motion approaches: it requires just one camera, not necessary calibrated, and there is no need to know the number of objects to reconstruct, nor to a priori group the features according to the object they belong to.

However, this technique has some restrictions: a certain number m of features have to be tracked along the same n frames, giving rise to *m complete* and *simultaneous* trajectories of length n. There is a minimum number of features and frames required, which depend on the specific factorization technique, and the camera is usually approximated either by an orthographic or an affine projection.

The central idea of all factorization techniques is the ability to express a matrix W containing the feature trajectories as the product of two unknown matrices $W = MS$, namely, the objects 3D shape S and the relative camera pose at each frame M. Although unknown, they can be estimated thanks to the key result that their rank is small and to constraints derived from the orthonormality of the camera axes. The exact rank of M and S and the specific constraints depend on the particular factorization technique at hand, as we will see in short.

The camera model used is the basis of all factorization methods. Kanatani and Sugaya rightly pointed out in a review paper (Kanatani and Sugaya, 2004) that there is still a wide–spread misunderstanding that factorization is a method for 3D reconstruction using SVD, whereas the underlying principle is only the affine approximation to camera imaging. SVD is just a mean for numerically computing the least–squares solution. Actually, factorization is a family of methods, the original one based on the simplest affine camera model —orthographic projection—, and subsequent methods on increasingly more accurate models: scaled orthographic projection, weak perspective, para–perspective and the full perspective camera.

Other techniques applied to vehicle detection from a moving observer have been recently published (Woelk, 2004; Hu, 1999). Both are based on the estimation of the focus of expansion (FOE) due to the translation component of the camera motion. They can not be applied to our night sequences, because very few or no static feature pairs are present in order to calculate the FOE. Besides, due to the pitch and yaw motion of the car, the trajectories of the points are not straight and that makes the FOE estimation difficult.

This paper is organized as follows. In sections 2 and 3 we introduce two factorization formulations. In section 4 we show how to adapt them to the vehicle detection problem. Section 5 contains the results, and we end with the conclusions and future work in section 6.

2 THE MULTI–BODY CASE

The seminal factorization work for structure from motion by Tomasi and Kanade (Tomasi and Kanade, 1992) was formulated for a single static rigid object viewed by a moving camera. Instead, the Costeira and Kanade formulation (Costeira and Kanade, 1998) of the factorization method assumes one or more moving objects viewed by a static camera. According to the later, let be $\mathbf{p}_j, j = 1 \ldots m$ the 3D object points expressed in some arbitrary coordinate system *fixed* to the object. In homogeneous coordinates, $\mathbf{s}_j = \begin{bmatrix}\mathbf{p}_j & 1\end{bmatrix}^t$. At each time $i = 1 \ldots n$, these points are projected into the image according to the simple orthographic camera model:

$$u_{ij} = \mathbf{i}_i^t \mathbf{p}_j + t_{xi} \quad v_{ij} = \mathbf{j}_i^t \mathbf{p}_j + t_{yi} \tag{1}$$

being $\mathbf{i}_i, \mathbf{j}_i$ the camera axes and (t_{xi}, t_{yi}) its translation.

All image trajectories (u_{ij}, v_{ij}) are stacked into a $2n \times m$ measurement matrix W, which, according to equation (1), is factored as the product of the shape and motion matrices we want to calculate:

$$\begin{bmatrix} u_{11} & \ldots & u_{1m} \\ \vdots & & \vdots \\ u_{n1} & \ldots & u_{nm} \\ v_{11} & \ldots & v_{1m} \\ \vdots & & \vdots \\ v_{n1} & \ldots & v_{nm} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^t & t_{x1} \\ \vdots & \vdots \\ \mathbf{i}_n^t & t_{xn} \\ \mathbf{j}_1^t & t_{y1} \\ \vdots & \vdots \\ \mathbf{j}_n^t & t_{yn} \end{bmatrix} [\mathbf{s}_1 \; \mathbf{s}_2 \; \ldots \; \mathbf{s}_m]$$

$$W = MS \tag{2}$$

In the absence of noise, the rank of W is at most 4. The SVD decomposition $W = U_{2n\times 4}\Sigma_{4\times 4}V^t_{4\times m}$ yields the true M and S but for an unknown invertible matrix $A_{4\times 4}$: $W = MS = \hat{M}AA^{-1}\hat{S}$, with $\hat{M} = U\Sigma^{\frac{1}{2}}$, $\hat{S} = \Sigma^{\frac{1}{2}}S$. Fortunately, A can be computed through a so called *normalization* process by using the fact that the rows of M represent the camera rotation axes and satisfy, for $i = 1 \ldots n$, the following conditions:

$$\begin{aligned} \mathbf{i}_i^t \cdot \mathbf{i}_i = \mathbf{j}_i^t \cdot \mathbf{j}_i = 1 \\ \mathbf{i}_i^t \cdot \mathbf{j}_i = 0 \end{aligned} \tag{3}$$

If the scene contains p objects moving independently, each with a certain number of features, the matrix of all trajectories *sorted by object* W^* factorizes as before into the product of p block matrices M_k and S_k, $k = 1 \ldots p$. But, of course, we do not know W^*, because features are not sorted by object, which is precisely what we want to obtain from the motion segmentation. Instead, W is equal to W^* but having permuted some columns.

The main finding of the Costeira–Kanade's method is the so called shape interaction matrix:

$$Q^* = V^* V^{*t} \tag{4}$$

This matrix has an interesting block–diagonal structure and, if features k and l belong to different objects, Q^*_{kl} is zero. The key point is that this is also true even though the trajectories are not sorted by object: since W is equal to W^* but having permuted some columns, so V^t is equal to V^{*t} only that permuting the same set of columns. Therefore, Q^* will result by permuting rows and columns of Q in the same way.

The shape from motion problem for multiple moving objects has been reduced to that of finding out the right row and column permutations of Q such that it becomes block–diagonal, because then, the single body algorithm can be applied. However, recovering Q^* from Q, and thus performing feature grouping (motion segmentation), is still a tough problem because noise makes the entries of Q corresponding to features of different objects not exactly zero.

Costeira and Kanade present a two–steps method to determine the diagonal blocks of Q^* from Q. Firstly, Q needs to be sorted so that its structure resembles as much as possible a block diagonal matrix. They propose a simple greedy algorithm for this search. Secondly, the bounds of each block have to be determined. This is a necessary step to decide how many independent motions are there and which features belong to each one. An energy function is defined both to find potential block limits and to choose the motion segmentation solution:

$$\varepsilon(l) = \sum_{i=1}^{l} \sum_{j=1}^{l} (Q_{ij}^*)^2, \quad l = 1 \dots m \tag{5}$$

3 LINEARLY MOVING MULTIPLE OBJECTS

The algorithm of Han and Kanade (Han and Kanade, 2004) deals with a moving camera and multiple static or dynamic objects, but now constrained to move in linear trajectories and at constant speeds. Recall that this is a reasonable assumption for small intervals of time.

In a certain world coordinate system, any point $\mathbf{p}_{ij}$, $j = 1 \dots m$ is represented by

$$\mathbf{p}_{ij} = \mathbf{s}_j + i\mathbf{v}_j \tag{6}$$

where s_j is the position of the j-th feature at frame 0, v_j is its motion velocity and $i = 1 \dots n$ is the frame number. Again, the image trajectories obtained with the orthographic camera model are stacked in the matrix W:

$$W = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1m} \\ v_{11} & v_{12} & \dots & v_{1m} \\ \vdots & \vdots & & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nm} \\ v_{n1} & v_{n2} & \dots & v_{nm} \end{bmatrix} \tag{7}$$

As before, this matrix is factored as the product of the shape and motion matrices. Now, the rank is 6 since we assume $m, n \geq 6$ and S has 6 rows:

$$W_{2n\times m} = M_{2n\times 6} S_{6\times m} + T_{2n\times 1} \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \tag{8}$$

where

$$M = \begin{bmatrix} \mathbf{i}_1^t & 1\mathbf{i}_1^t \\ \mathbf{j}_1^t & 1\mathbf{j}_1^t \\ \vdots & \vdots \\ \mathbf{i}_n^t & n\mathbf{i}_n^t \\ \mathbf{j}_n^t & n\mathbf{j}_n^t \end{bmatrix} \tag{9}$$

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \dots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_m \end{bmatrix} \tag{10}$$

$$T = \begin{bmatrix} t_{x1} & t_{y1} & \dots & t_{xn} & t_{yn} \end{bmatrix}^t \tag{11}$$

By moving the world coordinate system to the centroid of all the $\mathbf{p}_{ij}$ at each frame, it is possible to obtain T and subtract it from the matrix W, obtaining $\hat{W}_{2n\times m} = \hat{M}_{2n\times 6}\hat{S}_{6\times m}$. The coordinate system moves as its origin $\mathbf{p}_{ij}$ linearly with constant speed $\mathbf{v}_c$ and the static points have the same velocity, but with changed sign. Therefore, in order to obtain $\mathbf{v}_j$ in a fixed coordinate system, we must subtract $\mathbf{v}_c$ to $\mathbf{v}_j$.

In the absence of noise, the rank of $\hat{W}$ is at most six and M and S are obtained from its SVD: $\hat{W} = U_{2n\times 6}\Sigma_{6\times 6}V_{6\times m}^t$. Again, this decomposition is up to an affine transformation $A_{6\times 6}$: $\hat{W} = \hat{M}AA^{-1}\hat{S} = MS$, which can be computed with a normalization process similar to that of section 2.

4 VEHICLE DETECTION THROUGH FACTORIZATION

4.1 Trajectory grouping through multibody factorization

Two scenarios are possible with regard the content of a sequence:

1. Only one type of motion is present, that is, all features belong to a unique group or object. In this case we can not yet decide whether they are static or dynamic.
2. At least two types of motion can be differentiated. Then, it is clear that at least one of them is a moving vehicle.

Accordingly, two strategies are possible following the Costeira–Kanade method:

1. Inspect the obtained Q^* and the list of possible block limits just to decide whether there is just one group formed by all the features or instead two or more groups are more likely.
2. Perform motion segmentation in order to group all features into independently moving objects.

The first one answers if there are two or more vehicles, whereas the second one tells which of the features correspond to vehicles and which ones to static lights. Of course, the later is a much more interesting question but also more difficult to answer reliably.

In both cases, it is necessary to know a priori, or else estimate, the true rank r of W in order to compute $Q^* = V^*V^{*t}$. We are thus going to select the r for which the resulting Q^* achieves the most block-like structure. Hopefully, if there is just one moving object, any value of r will not achieve a sufficient block–diagonal matrix, according to a certain *blockiness* measure we have to design. Otherwise, we want this measure to attain its maximum for the right r. However, we must first bound the range of values for the rank of W.

One key assumption of the Costeira–Kanade method is not often satisfied in practice in traffic sequences: that objects move independently. The motion matrix M_k for each object k is of the form:

$$M_k = \begin{bmatrix} \mathbf{i}^t_{1,k} & t_{x1} \\ \vdots & \vdots \\ \mathbf{i}^t_{n,k} & t_{xn} \\ \mathbf{j}^t_{1,k} & t_{y1} \\ \vdots & \vdots \\ \mathbf{j}^t_{n,k} & t_{yn} \end{bmatrix} \quad (12)$$

Our road sequences are around just one or two seconds. During these short time intervals, vehicles do not rotate by themselves but for lane changes and road curves. However, it can hardly be appreciated due to the usually large distance to the camera. Nevertheless, there is always a continuous and oscillating relative rotation with respect the camera, caused by the pitch motion of the car to which it is attached. Since this rotation is the same for all objects, $\mathbf{I}_k = [\mathbf{i_{1,k}} \ldots \mathbf{i_{n,k}}]$ and $\mathbf{J}_k = [\mathbf{j_{1,k}} \ldots \mathbf{j_{n,k}}]$ are equal for each $k = 1 \ldots p$. Thus, the rank of M is reduced to 3 plus the number of linearly independent translation vectors $\mathbf{t}_k$. At this point, we introduce the following simplifying assumptions: every vehicle appears, at least, as two blobs (features), and in any sequence at least $m = 4$ features are tracked. Consequently, the maximum number of objects is $p = \lceil \frac{m}{2} \rceil$ and, in theory, the rank of W lies within the range

$$4 \le r \le \min(m, 3 + \left\lceil \frac{m}{2} \right\rceil) \quad (13)$$

Translation vectors $\mathbf{t}_k$ contribute too to the motion degeneracy. Trajectories along such short intervals are almost always straight lines and relative vehicle velocities mostly constant. Therefore translation vectors tend also to be linearly dependent, being related by a constant equal to the ratio of their speeds.

In sum, our trajectories are motion degenerate and the rank of W is, in practice, almost always 4, sometimes a little bit greater. Hence, only values $r = 4, 5, 6$ are usually worth to try.

Now, we turn to the problem of assessing each possible value of r according to the *blockiness* of its sorted interaction matrix Q^*. The bad news is that even the noiseless interaction matrix Q^* is no more block–diagonal. However, for the correct rank of W and even in presence of noise, the interaction matrix in our sequences is still quite block–diagonal and the energy function of equation (5) can be again used to find out possible block limits. Figure 1 shows an example of rank r determination for the sequence 3. Note that for $r = 5$ the block–diagonal aspect of the computed Q^* is maximum. In fact, $r = 5$ provides the right motion segmentation: the second 3×3 block corresponds to three lamp posts, and each of the other blocks is made of a couple of features which belong to the same vehicle.

Let be $l_1, l_2 \ldots l_r$ the list of computed possible block limits taken as those columns of $Q^*_r = V^*_r V^{*t}_r$ where ε shows a sharp increase. According to equation (5), a normalized blockiness measure is defined as:

$$b(r) = \frac{1}{r} \sum_{k=2}^{r} \mathrm{sign}(l_k - l_{k-1} - 1)(\varepsilon(l_k) - \varepsilon(l_{k-1})) \quad (14)$$

We select the right r as the one for which the normalized energy within all the possible computed blocks is maximum. However, blocks of size 1 feature are not taken into account, because for $r = m$, Q^*_m is perfectly diagonal with 1×1 blocks and we do not want to interpret this situation as a case of maximum blockiness. Figure 1 (top) shows an example of the blockiness measure of Q^*_r for the possible values of r according to equation (13).

Regarding strategy 1, the blocks for the best r yield the motion segmentation. As for strategy 2, we use the value of the blockiness measure to make a decision concerning the existence of more than one block. A simple threshold (set at 0.7) can differentiate the two cases in our experiments. In figure 2 the obtained motion segmentation is shown.

4.2 3D velocity computation

The Han-Kanade (Han and Kanade, 2004) algorithm allows to recover the velocity ratios of the features, i.e. $\|\mathbf{v}_k\| / \|\mathbf{v}_l\|$ for each $k, l = 1 \ldots m$. The velocity values are useful not only to group the features in objects but to distinguish between the objects that are approaching and the ones that are moving farther away from the camera, since their velocities have opposite sign.

The problem is that we would need to know which of the points are static in order to obtain the real velocity ratios, as it can be seen in section 3. But that is, precisely, our final goal: to find out which of the points correspond to vehicles and which to static features (such as lamp posts or traffic signals).

The method described in section 3 solves the case when the scene is 3D and the velocities of moving objects span a $3D$ space ($rank(\hat{W}) = 6$). Unfortunately, degenerate cases can arise due to degenerate shape and/or motion. Specifically, in the traffic sequences we work with, the most common situation is the existence of one or multiple moving objects in the same direction or perhaps opposite sense. As before, the static structure of the objects is 3D, but 3D velocities $\mathbf{v}_k$ span only a one dimensional space, since they differ in module or sign but not in direction.

Therefore, we have to deal with a degenerate case

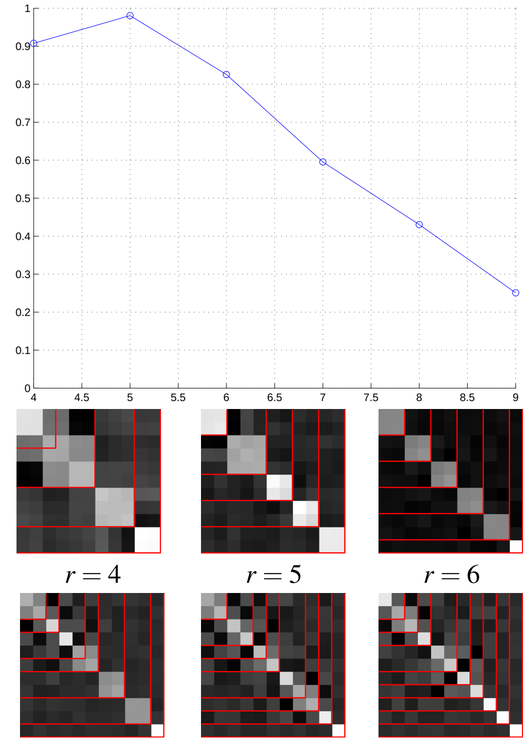

Figure 1: Sequence with $m = 11$ trajectories manually tracked over $n = 28$ frames, showing four cars and three lamp posts. Top : Measure of *blockiness*, sum of the normalized energy of Q^* within the possible blocks larger than 1×1 features. The maximum is attained at $r = 5$. Bottom: computed Q^* for $r = 4\ldots9$.

in which $rank(\hat{W}) = 4$ and the motion and shape matrices are:

$$M_{2n\times4} = \begin{bmatrix} \mathbf{i}_1 & i_{x1} \\ \mathbf{i}_2 & 2i_{x2} \\ \vdots & \vdots \\ \mathbf{i}_n & ni_{xn} \\ \mathbf{j}_1 & j_{x1} \\ \mathbf{j}_2 & 2j_{x2} \\ \vdots & \vdots \\ \mathbf{j}_n & nj_{yn} \end{bmatrix} \tag{15}$$

$$S_{4\times2n} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \ldots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \ldots & v_{xm} \end{bmatrix} \tag{16}$$

There is a *rank*-4 variant of the linear motion factorization, through which we can compute S. Now, we try to group features which share a similar velocity module: v_{xj}.

First, we run the Han-Kanade algorithm and obtain the velocities of the feature points: v_{xj}. In fact, only the ratios between the velocities are meaningful, not the actual value. Unfortunately, we can only recover the real ratios in case we know which are the static points and we subtract previously their velocity to the vector v_{xj}. Thus, we secondly compute the ratios v_{xk}/v_{xl} for all pairs of features (k,l) and put them into a matrix R. For pairs of features corresponding to the same object (or at least, those that move in the same way), a ratio near to 1 is obtained, since their velocity values are similar. We have experimentally fixed a tolerance of $tol = 0.3$ and if we define $q_{kl} = v_{xk}/v_{xl}$, we change the matrix of ratios according to:

$$R_{m\times m}(k,l) = \begin{cases} q_{kl} & \text{if } |q_{kl} - tol| \leq 1 \ , \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

The aim is to obtain a matrix whose non-zero elements give us the motion segmentation. In the best case, we obtain columns with at least 2 non-zero elements when the feature points correspond to a vehicle. And for the static points, we obtain columns of one or more non-zero elements.

For instance, taking again the sequence 3, we construct the matrix of the ratios between the velocities R (figure 3) and it results in the motion segmentation we can see in figure 4. There are three 2×2 blocks, that correspond to three of the vehicles. The third 2×2 block (the third vehicle) and the last 3×3 (the three lamp posts) are considered as a unique block. Thus, there are 4 different motions. This is due to the similar value of the velocity of one of the cars and the static points. As we mention before, the correct segmentation would be obtained subtracting the velocity of the static points. However, that is not possible, since we can not know which of the points are the static ones.

Figure 2: Motion segmentation, sequence 3, Costeira-Kanade method

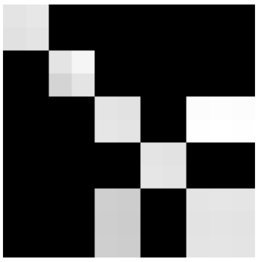

Figure 3: Matrix R of velocities ratios, Han-Kanade method

Figure 4: Motion segmentation for sequence 3, Han-Kanade method

5 RESULTS

We have tested both methods on 19 sequences ranging from 9 to 30 frames, at a frame rate of 10 frames per second. With these numbers, even oncoming vehicles stay in the desired mid to far range distance. All features are manually tracked to avoid gaps and errors in the trajectories. Working with a tracker it would be necessary to deal with occlusions, which is an additional problem not tackled in this paper. Additionally the assumption that every vehicle appears at least as two blobs could not be introduced. Hence, computational cost is added to the algorithm, since the range of possible ranks of W is extended (see inequality (13) in section 4.1).

Table 1 summarizes the characteristics of the sequences and the obtained results for the vehicle existence test (when two or more feature groups are detected, the answer is *yes*, otherwise we can not decide) and motion segmentation, and for both methods. We classify as *good* results the ones for which only few mistakes are obtained or those that can be, in some way, justified (for the characteristics of the sequence, the features position or movement, etc). Figures 5-7 show some examples of motion segmentation obtained with both methods.

For the case of Costeira-Kanade adaptation, motion segmentation fails in 8 out of 11 cases, where both static and moving features are assigned to the same group. But in two of them (sequences 5 and 6) the vehicle is so far away that it can not exhibit a different image motion regarding static points, also quite far away. However, vehicle existence fails only in 4 and can not decide in 2.

In the Han-Kanade adaptation, the results are better when the feature points are far from the camera or situated in the center of the image. This is necessary to achieve a good approximation of the perspective projection by an affine one. Besides, for this technique, it is important that the camera rotates. The vehicle existence, however, provides better results. In some cases, the velocities of features corresponding to the same object have opposite sign and our algorithm considers those features independent, so the motion segmentation is not achieved. In most cases, this problem would be solved subtracting the velocity of static points. Unfortunately, that information is not available, since at the moment it is impossible to distinguish between static and dynamic features.

6 CONCLUSION

In this paper we have addressed the problem of vehicle detection at night when they are far away or at a mid-range distance by adapting two different factorization algorithms. The first one is due to Costeira and Kanade and the second one to Han and Kanade. The Costeira–Kanade method gives better results in the motion segmentation. Besides, fewer feature points are necessary. However, the method of Han–Kanade is interesting because the matrix of trajectories does not need to be sort like in the previous approach. Moreover, it provides the ratios of the velocities of the feature points. The problem is that we would need to know at least one static point in order to have the real velocity ratios.

One of the main problems we have to deal with is the approximation of the perspective projection by an affine one. It is quite difficult to find a sequence where the points are far away (there the approximation is better) and at the same time they have enough image motion (then the factorization performs well).

In summary, the test of vehicle existence performs fairly well for both of the methods, whereas right motion segmentation is more difficult to achieve. In the future, the combination of the two factorization techniques we have explored could be useful. We also plan to work with longer sequences and address to the problem of incomplete trajectories.

Sequence	Number of frames	Number of features	Tracked vehicles	Tracked static features	Costeira–Kanade Vehicle existence	Costeira–Kanade Motion segmentation	Han–Kanade Vehicle existence	Han–Kanade Motion segmentation
1	16	10	4	2	right	right	right	fail
2	24	8	2	2	right	good	right	good
3	28	11	4	3	right	right	right	good
4	30	7	Any	7	don't know	right	fail	fail
5	24	4	1	2	fail	fail	not tested	
6	9	4	1	3	fail	fail	not tested	
7	33	5	Any	5	don't know	right	not tested	
8	13	8	3	2	fail	fail	right	fail
9	13	8	3	2	right	right	right	good
10	12	8	1	6	fail	fail	right	fail
11	14	10	3	4	right	fail	right	good
12	26	9	3	3	not tested		right	fail
13	27	8	2	4	not tested		right	good
14	10	7	3	1	not tested		right	good
15	21	16	3	10	right	fail	right	fail
16	20	14	3	8	right	fail	right	fail
17	20	14	4	2	right	fail	not tested	
18	21	7	2	3	right	good	not tested	
19	19	10	2	6	right	good	right	good

Table 1: Table of results for both methods.

REFERENCES

Costeira, J. and Kanade, T. (1998). A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3).

Han, M. and Kanade, T. (2004). Reconstruction of a scene with multiple linearly moving objects. *International Journal of Computer Vision*, 59(3):285–300.

Hu, U. (1999). Moving objects detection from time-varied background: an application on camera 3d motion analysis. *Second International conference of 3-D digital imaging and modeling*.

Kanatani, K. and Sugaya, Y. (2004). Factorization without factorization : complete recipe. *Memoirs of the Faculty of Engineering, Okayama University*, 38(2):61–72.

Megret, R. and DeMenthon, D. (2002). A survey of spatio–temporal grouping techniques. Technical Report LAMP–094, Language and Media Processing, University of Maryland.

Tomasi, C. and Kanade, T. (1992). Shape from motion from image streams under orthography : a factorization approach. *International Journal of Computer Vision*, 2(9).

Woelk, Gehrig, K. (2004). A monocular image based intersection assistant. *IEEE Intelligent Vehicles Symposium*.

Figure 5: Sequence 1: four cars (three approaching the camera, one in the same direction) and two lamp posts. Left: Costeira–Kanade method, the three cars of the left are considered a unique object. Right: Han–Kanade method, one lamp post is grouped with one of the cars.

Figure 6: Sequence 2: one bus approaching the camera (four feature points), one car in the same direction as the camera and two lamp posts. Left: Costeira–Kanade method. Right: Han–Kanade method. In both methods the features of the bus are not grouped as a unique object, due to the perspective effect (in the last frames the bus is very close to the camera).

Figure 7: Left: Costeira–Kanade method. Sequence 4: seven static points, segmented as one object. Right: Han–Kanade method. Sequence 14: three cars in the same direction as the camera and one lamp post. The features of the second car are segmented separately.