EXTRACTING SPATIAL INFORMATION FROM DIGITAL VIDEO IMAGES USING MULTIPLE STEREO FRAMES

Martin Braess, Geodätisches Institut der RWTH Aachen

Commission II/5

KEY WORDS: Acquisition, Automation, Stereoscopic, Three-dimensional, Close_Range, Scene Reconstruction, Feature Based Matching.

ABSTRACT:

This paper describes a feature-based approach for the reconstruction of 3D scene geometry. Digital images are taken from a moving surveying vehicle. Straight lines are matched. We minimize a cost function that incorporates feature attributes and relations between features using a branch-and-bound algorithm. An example for the line matching in a short sequence with two image pairs is presented.

1 Introduction

1.1 Problem Description

A surveying vehicle that collects various data for GIS databases is given. The vehicle is equipped with a GPS receiver and wheel sensors to determine its absolute position. The wheel sensors furnish 2500 impulses per turn which are registered with a hardware counter. The counter values and GPS data are recorded once per second.

In addition, the vehicle contains a digital video camera pair. We use two PULNiX TM 9700 cameras. They synchronously capture grey scale images with standard video resolution on a PC. The cameras are mounted on a stable aluminium profile, thus their relative orientation remains constant. Whenever an object of interest appears, an operator records one or more stereo pairs on a fixed disk. The counter values are stored with every image pair.

The images have to be evaluated in postprocessing by an operator who is mainly interested in the position of distinct points and lines in 3D-space. Most objects of interest in the traffic environment are man-made and contain straight lines. The object reconstruction is possible if the *correspondence problem* is solved. In this paper we want to support the operator by suggesting correctly matched objects or features.

If a feature is captured in two images, its position can be determined. In our approach, however, an object is followed in the image sequence in order to get both a higher accuracy and a higher reliability of the object position. Here, we confine ourselves to the use of interest points and straight lines only to describe objects in space.

The outline of this paper is as follows: first, the camera calibration and feature extraction is briefly described. The matching process consists of three stages: initial line matching, orientation of the camera pair and final matching. The paper is concluded with a practical example and a short discussion.

1.2 Algorithm Outline

The matching problem is often described as depending on three items:

- feature attributes
- relations between features
- geometric constraints

Feature *attributes* are used to determine the similarity of features to be matched. Typical attributes are grey values, line parameters or operator values.

Relations between features, however, tell us whether a match is consistent or not: if two features in one image have a certain relation, the corresponding features in another image are likely to have the same relation. Typical relations are angles and distances between features. In some applications the assignment is based on relations only, this is the case of structural matching [8]. In principle, ternary or even higher order relations can be used but in this paper it is assumed that binary relations provide a sufficient describtion [2].

When the camera orientation is known, this information can be introduced in form of *geometric constraints*. In our case, the relative orientation of the camera pair is known. Then we can impose the coplanarity constraint for points using two images: two points can only be matched if they are coplanar.

Lines in space are completely described with 4 parameters, lines in the image are described with 2 parameters [7]. That is why any two lines in two images furnish exactly one line in space, there is no redundancy (the degenerated case where the 3D-line lies in the epipolar plane is excluded). A third orientated image is needed to impose a geometric constraint. The only escape from this situation is to claim that the line segments in space have to overlap [6].

The assignment problem is a multi-dimensional decision problem. A multidimensional optimisation can, in general, not be achieved. That is why we transform the problem into a onedimensional one. For this purpose, we introduce a cost function that has to be minimized.

The assignment problem is a problem of exponentional complexity. It would be too time-consuming to check the cost function for all possible sets of matches. Therefore, we reduce the number of possible matches considerably by heuristic means. The optimisation procedure is performed afterwords.

2 Camera Calibration

Before the surveying drive is carried out, the camera pair is calibrated in a three dimensional test field. The test field consists of 92 circular targets that are captured. The positions of the targets in the images are automatically determined using least squares matching. Then the interior orientation and the exterior orientation in a local co-ordinate system can be derived. We use the well known photogrammetic collinearity equations, where the exterior orientation is described with 6 parameters: the position of the centre of projection and 3 spatial angles. The interior orientation is modeled using the principal distance c, the position of the principal point (x_0, y_0), the lens distortion which is modeled with a circular distortion A_1, A_2, and a linear-affine distortion B_1:

$$\begin{aligned} dr(x,y) &= A_1(r^2 - r_0^2)r + A_2(r^4 - r_0^4)r, \quad \text{with} \\ r^2 &= (x - x_0)^2 + (y - y_0)^2. \\ dx &= xB_1 \end{aligned} \tag{1}$$

Tests have shown that this model for the interior orientation is appropriate for the used camera-lens system.

3 Feature extraction

The feature extraction, the first step of the image evaluation, is performed using standard operators known from image processing. Interest points are extracted using the FÖRSTNER interest operator [4]. These points show a high significance and can be matched with high accuracy.

Straight lines are found in two steps: first grey scale edges are found by using a gradient operator [3] and then straight lines which are longer than a threshold length are extracted from the edge image.

Due to image noise and other influences, the end points of the line segments are very unstable features.

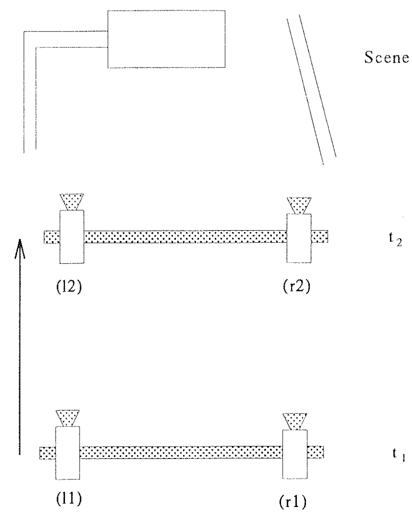

Figure 1: A scene is recorded in two stereo frames. The displacement of the frames is between 3 and 10m.

Therefore we only deal with lines, which can be described by

$$y = mx + b \qquad \text{or} \qquad x = m'y + b' \tag{2}$$

The first form is used if the angle θ between the line and the x-axis is between -45^o and 45^o, otherwise the second form is used.

For every extracted feature some *attributes* are calculated. The attributes of points are the position in the image, a small square grey value matrix with the point in the centre. The attributes of lines are again the position, the line length and orientation, and the mean grey values g_1 and g_2 on both sides of the line.

Relations are calculated between *pairs* of features. Two types of line relations are considered. Every line pair has an enclosed angle in the domain $0^o \ldots 180^o$. Further on for each line pair the perpendicular distance of the second's line midpoint from the first line is calculated.

Points are related if their distance is below a threshold distance. In that case their relative position is determined. All features and relations are saved in a database.

4 Finding Initial Line Matches

In order to restrict the total number of possible line matches, we determine initial matches by heuristic means. For this purpose geometric properties and the extracted features are used. The procedure is described for the case of 4 images, it can easily be generalized for more than 4 images. The imagining situation is shown in figure 1.

First all possible matches of lines in both images $(l1)$ and $(r1)$ of time t_1 are regarded. We only admit a match if the squared difference of grey values is below a threshold Γ_g:

$$(g_1^{(l1)} - g_1^{(r1)})^2 + (g_2^{(l1)} - g_2^{(r1)})^2 \leq \Gamma_g. \quad (3)$$

From these two line segments a line in 3D–space can exactly be determined. The line segments of $(r1)$ and $(l1)$ are projected onto the line in space and furnish line segments in space. A match is only assumed to be true if the overlap in space is higher than a certain threshold Γ_o. Currently we claim a relative overlap in space of 50% minimum.

In the next step possible matches in the consequent image pairs are searched. If a line in the left image $(l2)$ has a similar length and orientation as in the left image $(l1)$, i.e. the absolute differences are less than some thresholds Γ_l and Γ_θ, this line is a possible candidate. Now we check whether the grey value difference is the smallest one for all lines in image $(l2)$. If this condition is fulfilled as well we make a geometrical test: the projection of the reconstructed line in 3D–space into the image $(l2)$ is compared with the candidate line. For the orientation of image $(l2)$ we use the estimates from the wheel sensors. If their differences again lie beneeth thresholds Γ_m and Γ_b, we finally accept the candidate line as a possible match.

The same procedure is performed with all lines of the right image $(r2)$. An initial correspondence is only kept if a corresponding line in $(l2)$ or $(r2)$ is found.

This quite complicate procedure gives a number of possible matches that are most probable linked to really corresponding line segments. Typically the matches are not unique: some lines are matched more than once.

5 Camera Orientation

An orientation algorithm which estimates the orientation of the camera pair in t_2 is presented. The absolute orientation is determined in the local coordinate system. All measurements are performed in a local coordinate system defined by the first image pair: the camera $(l1)$ lies in the co-ordinate origin. In a later processing the measurement results obtained in this local coordinate system can be transformed into a global co-ordinate system using GPS observations.

As already mentioned, the relative orientation of the camera pair remains constant. The orientation of both cameras is therefore completely described by the known orientation of one (in our case the left) camera.

The orientation algorithm uses straight lines to determine the exterior orientation. It is based on the method presented in [7] and only uses the line parameters of the observed lines. At least three lines in at least three images are needed to calculate the exterior orientation. They must not be parallel in space. The unknown parameters of the lines in space and the unknown exterior orientation are determined in a least square approach. The parameters of the image lines are compared with those calculated from the projection of the lines in space into the image with the assumed orientation. The weighted square differences of the line parameters in the images is minimized.

The orientation starts with the lines in the images with the highest radiometric similarity. The Schwermann-algorithm [7] is performed once. If there are false matches, the algorithm will not converge to the correct values. There are two possibilities to detect this case. A low accuracy of the calculated lines in space indicates false matchings. These are eliminated and the process is repeated until the algorithm converges to reasonable values. If orientation values are near to the approximation determined from the wheel sensors, this orientation is accepted.

6 Finding Optimal Line Correspondences

6.1 Optimisation Algorithm

It is assumed that N possible assignments are given. We denote the assignments with $\mathcal{C}_i$, $i \in (1 \ldots N)$. Every assignment contains a list of matched elements $x_i^{(j)}$, j is the image number ($j = 1, \ldots, M$) and M is the number of available images.

$$\mathcal{C}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(M)}),$$

'Null matches' where no object in one image is matched are allowed as well. The task is now to find a set of optimal assignments. We are looking for a set Ω being a subset of $(1, 2, \ldots, N)$, the set of all possible assignments. The number of assignments in Ω is denoted with $|\Omega|$. Ω has to fulfill a compatibility restriction. The compatibility or uniqueness restriction requires that every object must only be matched once:

$$x_i^{(j)} \neq x_k^{(j)} \quad \forall j \in (1, \ldots, M), \text{ and } i, k \in \Omega, i \neq k$$

Further on we claim that the selected assignment set is 'better' than every other assignment set. For this purpose we construct a cost function $K(\Omega)$ that describes the quality of an assignment set. The cost function $K(\Omega)$ has to be minimized. A possible cost function is presented in the next paragraph. We assume here that the cost function depends on two factors: attribute costs K_A and relational costs K_R. Then we get the following cost function:

$$K(\Omega) = \sum_{i \in \Omega} K_A(\mathcal{C}_i) + \sum_{i \in \Omega} \sum_{j \in \Omega, j > i} K_R(\mathcal{C}_i, \mathcal{C}_j). \quad (4)$$

For ervery $|\Omega|$ there exists an optimal solution Ω. To find the optimal soultion we use a branch-and-bound algorithm.

for $i = 1, \ldots, N$: Optimal $(i) = \infty$

for $i = 1, \ldots, N$
  for all Ω_j in the list
    if Ω_j and $\mathcal{C}_i$ are incompatible
      continue
    $K_1 = \sum_{k \in \Omega_j} K_R(\mathcal{C}_k, \mathcal{C}_i) + K_A(\mathcal{C}_i)$
    $K_2 = K_1 + K(\Omega_j)$
    if $K_2 > \rho \cdot \mathrm{Optimal}(|\Omega_j| + 1)$
      continue
    if $\mathrm{Optimal}(|\Omega_j| + 1) > K_2$
      $\mathrm{Optimal}(|\Omega_j| + 1) = K_2$
    create new list element $\Omega_l = \Omega_j + \mathcal{C}_i$
    $K(\Omega_l) = K_2$, insert Ω_l into list

Figure 2: Branch-and-bound algorithm for the determination of optimal assignment sets.

The algorithm is described in detail, a pseudo code version is shown in figure 2. An array called 'Optimal' always contains the minimal costs for a given number of chosen assignments. A chained list contains all possible assignments regarded. The list is ordered from small to high costs, at the beginning the list ist empty. In the outer loop all assignments $\mathcal{C}_i$ are regarded. We try to add them to every assignment set Ω_j yet in the list. If they are not compatible the procedure is continued with the next Ω_j. We do the same if the new generated $\Omega_l = \Omega_k + \mathcal{C}_i$ produces costs much higher than the optimal costs for that number of assignments. The generated new list element is inserted into the ordered list.

When all assignments $\mathcal{C}_i$ are treated, we can search the optimal assignment sets in the list. As the list is ordered by the costs, we go through the list and search for the first occurence of assignment sets Ω_j with $|\Omega_j| = k,\ k = 1, \ldots, N$.

The presented algorithm examines only the best and only compatible assignment sets. A new assignment set is created if it is compatible and not too far from the best assignment set known at that time. The parameter ρ determines which solutions are considered. For $\rho = 1$ only solutions better than the actually best solution are accepted. For $\rho = \infty$ all solutions are accapted. Currently we use the parameter value $\rho = 3$ which has proved to be appropriate.

The algorithm furnishes a list of optimal assignment sets. For every number of matches we get the optimal assignment set. The last element is the combination with the maximal number of compatible matches possible.

6.2 Spezification of attribute and relational costs

The attributes and relations contain information that should be used for the matching. It is assumed that all attributes contain the same amount of informa-

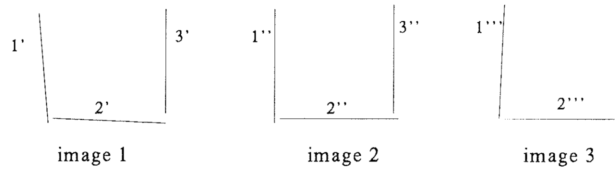

Figure 3: 3 images: 3 matches are found, but one match (3',3") is only found in two images

tion. Let's assume to have an attribute in two images $a^{(1)}$ and $a^{(2)}$. The attributes a are equally distributed over a certain domain. We assume that the attribute difference $a^{(1)} - a^{(2)}$, however, is normally distributed with a certain variance σ_a^2 and zero mean if two features correspond.

A possible cost function $K_A(\mathcal{C}_i)$ is the squared sum of attribute differences. When there is more than one attribute the squared differences have to be weighted. Deviding the squared differences by the variances results in a sum of normally distributed random variables with unary variance — provided that the match is correct. We denote the ith attribute in the nth image $a_i^{(m)}$ and the mean attribute $\bar{a}_i$. Thus for n attributes in M images we get:

$$\bar{a}_i = \frac{1}{M} \sum_{m=1}^{M} a_i^{(m)} \qquad i = 1, \ldots n \qquad (5)$$

$$K_A(\mathcal{C}) = \sum_{i=1}^{n} \sum_{m=1}^{M} \frac{(a_i^{(m)} - \bar{a}_i)^2}{\sigma_i^2} \qquad (6)$$

Here σ_i^2 is the variance of the ith attribute difference. As $K_A(\mathcal{C})$ is a sum of squared normal distributed random variables, it is $\chi^2(nM)$-distributed. Its mean values $\mathrm{E}(K_A(\mathcal{C}))$ is nM.

Often there is the case that one match does not exist in all images (see figure 3). In (6) we sum only over the images where the regarded feature exists and add n to compensate for this. The consequence of this approach is that we do not punish nor honour missing correspondences in one image.

To determine the relational costs $K_R(\mathcal{C}_i, \mathcal{C}_j)$ we proceed in a similar way. Again we assume that the differences of relation values are normally distributed.

The formula for $K_R(\mathcal{C}_i, \mathcal{C}_j)$ is shown for only one relation, the case of more relations is straight forward. We denote $r_{ij}^{(m)}$ the relational value between feature i and feature j in the mth image. Again we calculate the mean value $\bar{r}_{ij}$ and from this $K_R(\mathcal{C}_i, \mathcal{C}_j)$:

$$\bar{r}_{ij} = \frac{1}{M} \sum_{m=1}^{M} r_{ij}^{(m)} \qquad i = 1, \ldots n \qquad (7)$$

$$K_R(\mathcal{C}_i, \mathcal{C}_j) = \sum_{m=1}^{M} \frac{(r_{ij}^{(m)} - \bar{r}_{ij})^2}{\sigma_{ij}^2} \qquad (8)$$

6.3 Matching lines and points

The forementioned selection algorithm can easily be generalised for the case of line and point matching. We have to define new cost functions for the point attributes, the relations point–point and point–line. We present the case of two images to be matched, the generalisation for more than one image pair is possible.

For the attribute costs we propose to use the complanarity constraint and the correlation coefficient between points. In the ideal case the correlation coefficient ρ_{ij} between two points P_i and P_j equals 1.0. The spat product p_{ij} of the base vector and the two direction vectors from the centres of projection to the points equals 0. Thus we define

$$\mathcal{K}_A(P_i, P_j) = \frac{(\rho_{ij} - 1)^2}{\sigma_\rho^2} + \frac{p_{ij}^2}{\sigma_p^2} \tag{9}$$

For the costs of point–point relations we regard the distances $(dx^{(1)}, dy^{(1)})$ between the points in the first image and the distances $(dx^{(2)}, dy^{(2)})$ of the corresponding points in the other images. Then we get the relational costs for the point pair $P_1^{(1)}, P_2^{(1)}$ matched to $P_1^{(2)}, P_2^{(2)}$ respectively:

$$\mathcal{K}_R(\cdot) = \frac{(dx^{(1)} - dx^{(2)})^2}{\sigma_{dx}^2} + \frac{(dy^{(1)} - dy^{(2)})^2}{\sigma_{dy}^2} \tag{10}$$

For the relation point–line again we regard the distance of the point from a line. One can differentiate between the 'right' and the 'left' side of the line by using a signed distance.

7 Experimental Results

Practical tests with an image sequence taken on a German highway have been perfomed. Figure 4 shows four images with extracted straight lines. The line detector furnishes 32, 32, 31 and 37 lines respectively. The initial line matching furnishes 20 possible assignments. The orientation of the second image pair is performed using 5 assignments. The final optimisation furnishes assignment sets with up to 8 assignments. There is only one false match in the right part of the images: a neighbouring line is assigned. These lines are even for the human operator difficult to match as they are quite similar.

8 Performace Analysis

We have performed the tests on a DEC Alpha 3000/600 (175 MHz). The process time for the example described in the previous section is listed in table 1. The greatest part of the calculation time is consumed in the standard image processing part. This calculation could be performed with a special image processing hardware which is up to 100 times faster than a workstation. Thus, the construction of a system working under operational conditions is possible.

Table 1: Calculation time for the whole feature extraction and matching process with the four example images from section 7

Procedure	process time (s)
Interest point calculation	148
Lines calculation	116
Initial matches	4
Camera orientation	8
Optimal matching	4

9 Discussion

An algorithm for line matching based on the minimisation of a cost function was presented. The minimus is found by first applying heuristic means and a branch-and-bound optimisation afterwards. Currently only a small part of the possible line matches ($\approx$ 20%) are found. We hope to enhance the number of matches found in an aditional stage where the matches from the first stage are assumed to be correct ones. The matched lines are then of course discarded from the list of possible assignments.

References

[1] Aussems, Thomas, 1995. Fahrzeugortung mittels GPS und Koppelnavigation. In: Benning (Ed.), 125 Jahre Geodäsie an der RWTH Aachen, Veröffentlichung des Geodätischen Instituts Nr. 53, pp.71–80.

[2] Christmas, W., Kittler, J., Petrou, M., 1995. Structural Matching in Computer Vision Using Probabilistic Relaxation, IEEE–PAMI, Vol. 17, No.8, pp. 749–764.

[3] Deriche, R, 1985. Optical Edge Detection Using Recursive Filtering. In: First International Conference on Computer Vision.

[4] Förstner, W, 1991. Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung. DGK, Reihe C.

[5] Melsa, J., Cohn, D., 1978. Decision and estimation theory, McGraw-Hill, New York.

[6] Taylor, C., Kriegman, D., 1995. Structure and Motion from Line Segments in Multiple Images, IEEE–PAMI, Vol. 17, No.11, pp. 1021–1032.

[7] Schwermann, R., 1994. Automatic image orientation and object reconstruction using straight lines

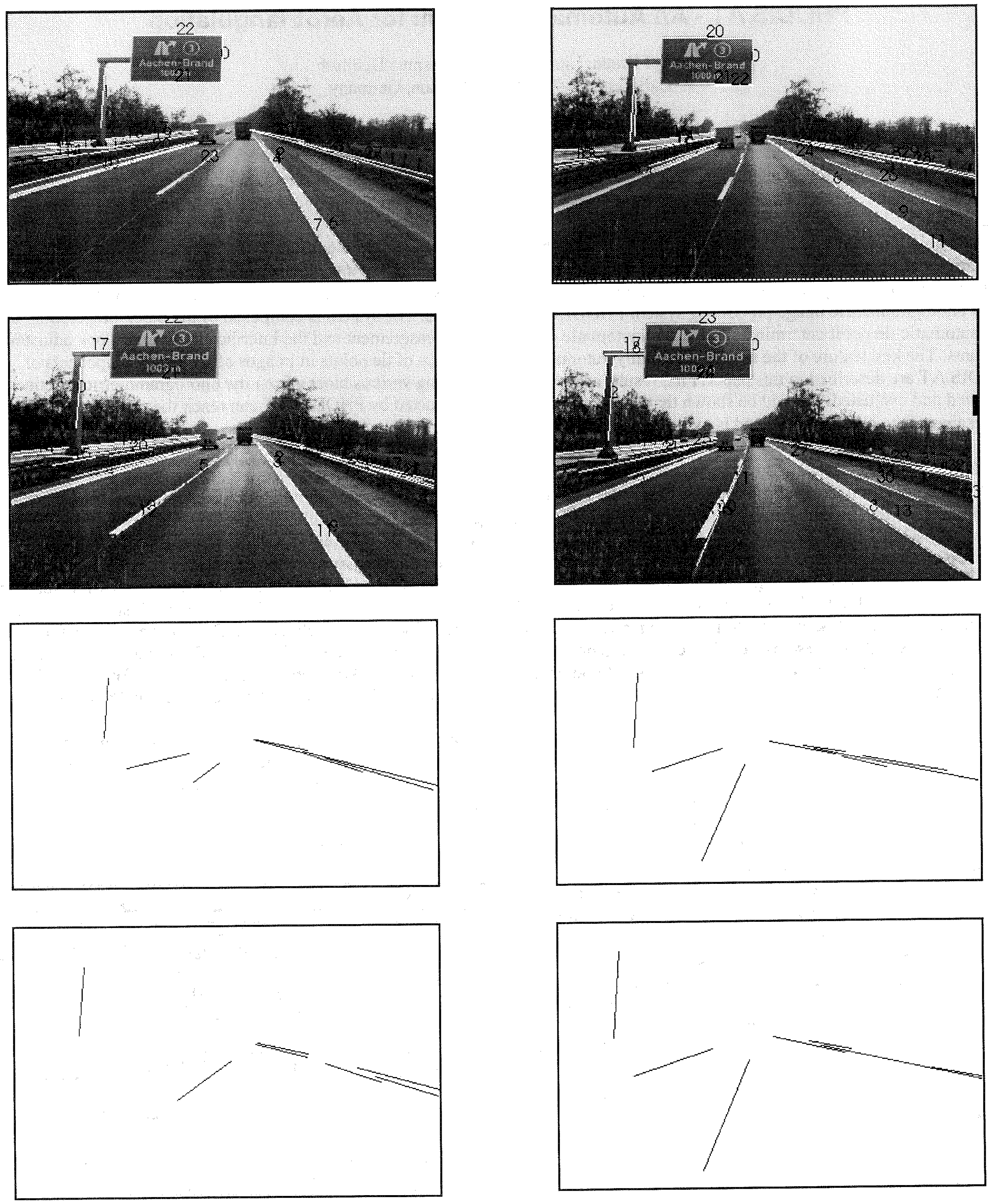

Figure 4: Result of a practical test: the left column shows the left images, the right column shows the right images. The matched lines are shown in the two bottom rows.

in close range photogrammetry. In: Close range techniques and Machine Vision, ISPRS Commission V, pp. 349–356.

[8] Vosselman, G., 1992. Relational MAtching. Lecture Notes in Computer Science. Springer Verlag Berlin–Heidelberg–New York.