ROBUST 3D-OBJECT REPRESENTATION BY LINEAR FEATURES

Anja Wilkin
Institute of Photogrammetry and Remote Sensing
Helsinki University of Technology
Otakaari 1, SF-02150 Espoo, Finland

Abstract:

The use of linear features - instead of points - in photogrammetry allows a 3D-object reconstruction without corresponding points in the different images. The goal of this paper is the reconstruction of a straight line in space from contaminated image data. A new algorithm, based on the Random Sample method combined with the least squares adjustment, is able to perform the robust 3D-reconstruction of the line only by image observations from calibrated cameras. In numerous simulation experiments the algorithm is tested.

Key words: Robust, object reconstruction, linear feature, straight line, random sampling consensus.

0. INTRODUCTION

In order to improve our existing photogrammetric station for close range applications we aim at an automatic object recognition for robot vision and at precise measurements on the object (e.g. a car carosse). For these purposes the 3D-object is represented by different types of linear features in the space like lines, circles or ellipses. The problem consists in reconstructing the 3D-structure from several 2D-images. The feature-based approach has the advantage that no point correspondence is required in the different images. So signalization of points on the object can be avoided and no image matching is necessary.
The projection of a three dimensional linear feature into image space produces a two dimensional linear feature, which can be detected in the image as arbitrarily measured feature pixels. After extraction of these pixels from the images the determination of the 3D-feature parameters will be carried out with robust estimation methods.

The paper is organized as follows: Chapter 1 introduces a model for the straight line representation, chapter 2 deals with robust methods for the 3D-reconstruction, chapter 3 treats the details of the presented algorithm and chapter 4 finally reports about the simulation results.

1. MODEL FOR STRAIGHT LINE REPRESENTATION

1.1 Representation of a straight line in space

A straight line in the 3D-Euclidean space has 4 degrees of freedom in its parametric representation (Roberts, 1988). Defining the straight line in terms of an arbitrary point $C = \{C_x, C_y, C_z\}$ and an orientation $\beta = \{\beta_x, \beta_y, \beta_z\}$ is a representation which is not unique and uses more parameters than necessary. For that reason two constraint equations are imposed: First, the direction vector β is forced to be a unit vector. Second, C is chosen as the line's nearest point to the origin, the line center point.

constraint 1: $|\beta| = \beta_x^2 + \beta_y^2 + \beta_z^2 = 1$ (1-1)

constraint 2: $\beta \bullet C = \beta_x C_x + \beta_y C_y + \beta_z C_z = 0$ (1-2)

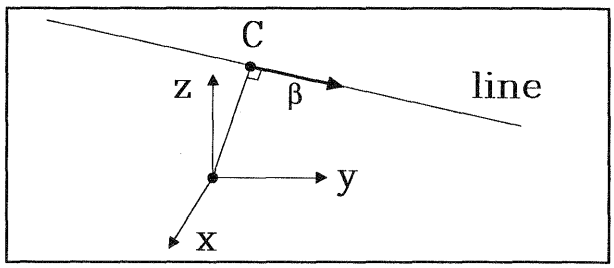

Fig. 1: Straight line representation, using two constraints

The only remaining weakness of this representation is the undetermined sign of the vector β. Because this plays no role in the calculations later on, no conventions are made concerning the sign.

1.2 Photogrammetric treatment of space lines

Usually the relationship between object- and image space is expressed by the collinearity equation. The weak point of this pointwise representation is the need of many nuisance parameters in addition to the line parameters. A better way is a feature description based on the object space geometry (Mulawa, 1988).
The image ray ρ is the vector from the perspective center **L** to the observed image point (x,y). Its direction in space is calculated from the orientation data of the corresponding perspective center. The image coordinates are assumed to be corrected from systematic errors.

$$\rho = \mathbf{R} \begin{bmatrix} x \\ y \\ -c \end{bmatrix} \qquad (1\text{-}3)$$

R = rotation matrix; **c** = camera constant

In case of errorfree observations the object line, the observed image ray ρ from the perspective center **L** to any point **P** on the line and the vector between the perspective center **L** and the line center point **C** form a plane.

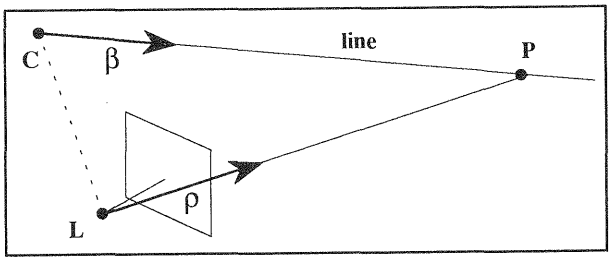

Fig. 2: Line representation in object and image space

This coplanarity relationship can be expressed by the scalar triple product

$$[\rho\ \beta\ (C-L)] = 0 \qquad (1\text{-}4)$$

The expression is free of any nuisance parameters. One equation allows the description of one observed point (x,y). The linearization of (1-4) with respect to the parameters **C** and β to be determined in a space intersection gives

$$\frac{\partial F}{\partial \beta} = [(C-L) \times \rho]^T, \quad \frac{\partial F}{\partial C} = [\rho \times \beta]^T \qquad (1\text{-}5)$$

According to (Mulawa, 1988) the coplanarity relationship is very stable with respect to the initial approximations. Line determination is not possible, when two cameras are used and the line falls on an epipolar plane of the cameras.

1.3 Model with conditions and constraints for least squares adjustment

The model with conditions and constraints is chosen for a least squares adjustment of **C** and β. The model is applied in order to handle the implicit observation equation and the two constraints. This paper will present the model only in a rough way, because it is already detailed described in (Mikhail, 1976) and (Mulawa, 1988).

The covariance matrix Σ of observations is usually scaled by the a priori reference variance σ^2. In the adjustment the scaled version **Q** is used.

$$Q = \frac{1}{\sigma^2}\Sigma \qquad (1\text{-}6)$$

The linearization of the non linear condition equation **F** is done by the Taylor Series expansion up to the first order.

$$F(l,\hat{x}) \approx F(l,x_o) + \frac{\partial F}{\partial l}\Delta_l + \frac{\partial F}{\partial x}\Delta_x \approx -f + Av + B\Delta \qquad (1\text{-}7)$$

$l = [x_1,y_1,...,x_n,y_n]$ = vector of image observations
v = residuals, approximation that $v = \Delta_l$
x = unknown parameters $C_x, C_y, C_z, \beta_x, \beta_y, \beta_z$
$\hat{}$ = superscript referring to estimated values
x_o = current approximations for parameters
A = matrix of derivates with respect observations
B = matrix of derivates with respect unknowns
Δ_x = corrections to parameters
f = coplanarity value = current value of condition equation
n = number of image observations
c = number of condition equations = ½**n**
u = number of parameters = 6

The linearized form of the condition equation is written

$$A_{c,n}\, v_{n,1} + B_{c,u}\, \Delta_{u,1} = f_{c,1} \qquad (1\text{-}8)$$

As weights for the condition equations F_i - including both observations x_i and y_i - the matrix W_e is introduced as

$$Q_e = A\,Q\,A^T,\ W_e = Q_e^{-1} \qquad (1\text{-}9)$$

The matrix Q_e has diagonal structure and its elements are here called 'pseudo weights'. In the value of a pseudo weight the individual weights of the coordinates and the local geometry described by the matrix **A** are involved. The linearized form of the two constraint equations is

$$C_{s,u}\,\Delta_u = g_{s,1} \qquad (1\text{-}10)$$

$$\begin{bmatrix} 0 & 0 & 0 & 2\beta_x & 2\beta_y & 2\beta_z \\ \beta_x & \beta_y & \beta_z & C_x & C_y & C_z \end{bmatrix} \Delta = \begin{bmatrix} 1-g_1 \\ 0-g_2 \end{bmatrix}$$

where **s** = number of constraint equations = 2

The least squares technique is based on minimizing a quadratic form. It leads to the normal equations

$$\begin{bmatrix} B^T W_e B & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ k_c \end{bmatrix} = \begin{bmatrix} B\,W_e\,f \\ g \end{bmatrix} \qquad (1\text{-}11)$$

For building up the matrices B^TW_eB and B^TW_ef in a computational efficient way the summation accumulation algorithm suggested by (Mulawa, 1988) is used. It is based on the diagonal structure of the **A** and **Q** matrices, so that a pointwise partitioning of the data and the matrices is possible. One update step consists of calculating the matrices or vectors **A**, **We**, **B** and **f** only due to one condition equation. The complete normal equation is the sum of all those pointwise calculated values.

The redundancy of this model is **r** = **c** - (**u** - **s**). Then the a posteriori reference variance can be computed by

$$\hat{\sigma}^2 = \frac{v^T W v}{r} = \frac{f^T W_e f}{r} \qquad (1\text{-}12)$$

Here the second term of this expression was used because the residuals **v** are not computed. All considerations about quality and outlier detection are done with the coplanarity values **f**. The coordinates x,y are not any longer handled as single observations. The coplanarity values offer the treatment of a point as 'pseudo observation' with a standard deviation expressed in the inverse pseudo weight and the coplanarity value as residual.

2. ROBUST METHODS FOR OBJECT RECONSTRUCTION

A general problem when dealing with real measurements is the influence of data contamination such as noise and outliers on the result. The aim of the present research is a reliable 3D-reconstruction of features from images. In a first step the feature pixels are extracted from the images with the help of any edge detection routines, whose robustness is not discussed in this paper. The extracted pixels are the input to the presented algorithm. Pixels from other features or errorpixels are outliers related to our line. The term 'robust' here means that we want to have a result which takes only those pixels into account which really belong to our line.
A least squares adjustment leads to wrong results when the data contains observations which do not belong to the model (Huber, 1981), (Rousseeuw/Leroy, 1987), (Förstner, 1989). Even a single outlying observation may affect the result severely.
Methods of hypothesis testing using test statistics are able to detect outliers, but they don't work very effective in presence of a high contamination.
In image processing, however, we sometimes have to deal with highly contaminated data. For that reason different attempts have been made to develop robust methods. Some of the methods are the M-estimation (Huber, 1981), the Random Sampling Consensus procedure of (Fischler/Bolles, 1981) and the Least Median Squares estimation by (Rouseeuw/Leroy, 1987). They all have in common that they try to find out how well an observation fits into the model. This is expressed by an individual weight for each observation.

2.1 The random sample consensus procedure

Random sample consensus (RANSAC) is a hypothesis-verify technique which can be used for determining model parameters, even if the amount of contamination exceeds 50% ($\varepsilon > 0.5$). It requires no initial values. A closed form solution for the parameters, based on a minimum set of u observations, must be known. For some problems these solutions are complicated or impossible. Basically the technique works in the following heuristic way:
1) Determine parameters from a random subset of u observations.
2) Compute residuals for all observations. A tolerance threshold, based on the expected noise level, has to decide if an observation will be accepted.
3) Count accepted observations.
4) Stop procedure when enough observations are accepted, otherwise repeat it with a new random subset.

Remark: Residuals in this procedure shouldn't be mixed up with the residuals v in chapter 1.3. The computation of the residuals meant here is explained in 3.2.1.

The algorithm parameters are the threshold for data acceptance in step 2), the threshold for model acceptance in step 4) and the number of subsets needed. The latter corresponds to the computation time. More about the number of subsets in chapter 2.1.1. The thresholds are discussed for the special case of a space line in chapter 3.1.1. The determination of model parameters in step 1) by a closed form solution is treated in chapter 3.1.

The described method is a powerful tool to detect outliers and to find approximate values for model parameters.
The method is not able to find the best possible solution when not all subsets are used or when the non-outlying observations are noisy. How to procede in case of noisy data is treated for the space line example in chapter 3.3.

2.1.1 The number of subsets in RANSAC

According to step 4) in the RANSAC procedure the computations are finished when enough observations are accepted. In general, however, the correct solution, the number and size of outliers and the optimum threshold for data acceptance are unknown. Under these circumstances the best matching subset cannot be determined. For very small data sets all possible subsets are taken, but this is not possible for large data sets, because the number of subsets as well as the computation time increases rapidly.

Under the assumption that the data contains **n-o** 'good' and **o** 'bad' points, the expected number **m** of random subsets to find at least one subset containing **u** good observations can be calculated due to probability laws. The probability to have at least one subset is usually set to 95% or 99%. The probability for one subset to contain **u** good points is

$$P_u = \left(1-\frac{o}{n}\right)\left(1-\frac{o}{n-1}\right)\ldots\left(1-\frac{o}{n-(u+1)}\right) \qquad (2\text{-}1)$$

The probability of at least one uncontaminated subset out of m can be calculated by the binomial distribution:

$$P(m_{good} \geq 1) = \sum_{i=1}^{m-1} \binom{m}{i} (1-P_u)^i \, P_u^{m-i} \qquad (2\text{-}2)$$

Instead of this exact formula an approximation is given for large numbers **n** of observations and a contamination of ε% (Rousseeuw/Leroy, 1987):

$$1-(1-(1-\varepsilon)^u)^m) = 95\%$$

$$\Rightarrow m = \frac{\log 0.05}{\log (1-(1-\varepsilon)^u)} \qquad (2\text{-}3)$$

Unfortunately the formula requires the input of ε which is usually not known exactly. Some experimental results on a suitable number of subsets are presented in chapter 4.

2.2 Least median squares method

Another heuristic, robust method is the least median squares estimation (Rousseeuw/Leroy, 1987). Here the squared residuals of observations according to different random models are computed. The model with the smallest median of squared residuals will be the solution.

$$\text{med}\ (\mathbf{residuals}_i^2) \quad \rightarrow \quad \text{minimum}$$

This method is able to cope with an amount of outliers up to 49.9%. In this study the method is not applied.

2.3 M-estimation, a robust adjustment technique

In contrast to the methods mentioned before the maximum-likelihood-type or M-estimation is a deterministic procedure. In the M-estimation one ends up at minimizing the sum of a function of squared residuals (Huber, 1981), (Förstner, 1989). That function is called weight function with the values w_i.

$$\sum_i w_i \, r^2_i \rightarrow \min. \qquad (2\text{-}4)$$

The minimization procedure is an iteratively reweighted least squares adjustment. A difficult problem is the proper choice of the weights w_i. The weight function has to fulfil certain mathematical properties, described in (Huber, 1981), to guarantee the adjustment converge to a solution.
For the method the calculation time can be predicted.

It is planned to use the M-estimation for the line reconstruction with the Danish reweighting method (Krarup/Kubik, 1981). The residuals should be derived from the coplanarity values **f**. Some very first experiments showed that the method seems to be able to cope with noise and outliers up to a certain amount and size. The required approximate values might be taken from the results of the RANSAC technique.

3. ALGORITHM FOR ROBUST 3D-LINE RECONSTRUCTION

In this chapter a method is explained how to reconstruct a 3D-line from errorous observations in several images. Up to now the algorithm is specialized on lines as used in the simulation experiments. Some routines and thresholds still have to be tested for more general cases. A method is presented of combining the RANSAC method with the least squares adjustment. In the experiments this turned out to be a suitable way to cope with the noise and the outliers.

3.1 A closed form solution for the RANSAC procedure

The underdetermined perspective view from 2D- to 3D-space is an inverse problem, which possesses no direct analytical solution.
A closed form solution for the 6 line parameters can be obtained from space geometry. No linearizations or initial values are needed, but knowledge is required about the camera constant **c**, the position **L** and the orientation **R** of two perspective centers. The presented solution requires the input of two different image points out of two images.

3.1.1 Closed form solution for the line center point C and the line direction vector β

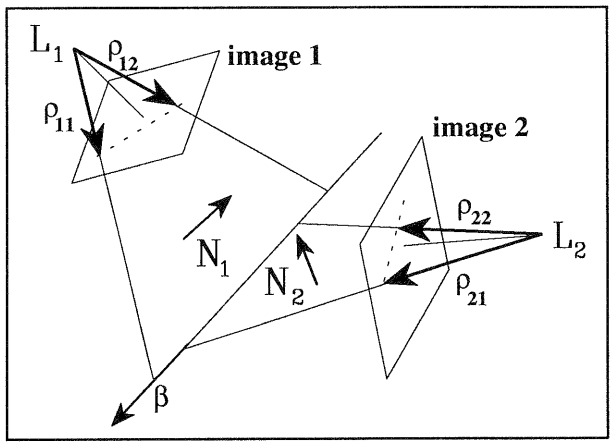

Fig. 3: Space geometry in the closed form solution

In a first step the four image rays ρ_{ij} from the perspective centers to the image points are constructed. The observed ray $(x_{ij}, y_{ij}, -c_i)$ of a point j in image i is transformed into space with the help of the rotation matrix $\mathbf{R}_i$:

$$\rho_{ij} = \mathbf{R}_i \begin{bmatrix} x_{ij} \\ y_{ij} \\ -c_i \end{bmatrix} \qquad (3\text{-}1)$$

Two rays belonging to one image construct a 'projection' plane, which is described by its normal vector

$$\mathbf{N}_i = \rho_{i1} \times \rho_{i2} \qquad (3\text{-}2)$$

The space line is the intersection of the two projection planes. Thus the direction vector β of the line is perpendicular to both normal vectors $\mathbf{N}_1$, $\mathbf{N}_2$.

$$\beta = \frac{\mathbf{N}_1 \times \mathbf{N}_2}{\| \mathbf{N}_1 \times \mathbf{N}_2 \|} \qquad (3\text{-}3)$$

To calculate the center point **C** of the line, a linear equation system is designed. It contains the equations of two projection planes in space and the second model constraint.

The position of a projection plane in space can be described by its normal vector **N** and a point lying in the plane, the perspective center **L**.

$$\begin{aligned} 0 &= N_{1x} L_{1x} + N_{1y} L_{1y} + N_{1z} L_{1z} - d_1 \\ 0 &= N_{2x} L_{2x} + N_{2y} L_{2y} + N_{2z} L_{2z} - d_2 \end{aligned} \qquad (3\text{-}4)$$

Knowing $\mathbf{N}_1$, $\mathbf{N}_2$, $\mathbf{L}_1$ and $\mathbf{L}_2$ we get $\mathbf{d}_1$ and $\mathbf{d}_2$ out of these equations. Then the 3×3 linear equation system can be solved for the line center point **C**.

$$\begin{aligned} &\textbf{constraint:} && \beta_x C_x + \beta_y C_y + \beta_z C_z = 0 \\ &\textbf{plane1:} && N_{1x} C_x + N_{1y} C_y + N_{1z} C_z = d_1 \\ &\textbf{plane2:} && N_{2x} C_x + N_{2y} C_y + N_{2z} C_z = d_2 \end{aligned} \qquad (3\text{-}5)$$

3.1.2 Stability of the closed form solution

Stability means that small changes in the data produce small changes in the results. In case the geometric situation is not stable, singularities in the analytical formulas and unreliable results may occur.
The input to the closed form solution described above consists of 4 randomly chosen image points. In order to avoid those combinations of image points, which lead to an ill-conditioned geometry, a measure for the stability of the solution is developped.
Weak points with respect to the numerical sensitivity are the three cross products used to compute the line direction vector β. A cross product is computed in a stable way if the angle between the two input vectors is close to 90°. In case the angle comes close to 0°, the result is a random outcome according to the noise in the input vector components. Image points very close to each other as well as line/camera formations with similar projection planes should be avoided.
The stability of a cross product $\mathbf{a}\times\mathbf{b}$ is measured by the sinus of the angle φ between two vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\sin\varphi = \frac{\|\mathbf{a}\times\mathbf{b}\|}{\|\mathbf{a}\|\,\|\mathbf{b}\|} \qquad (3\text{-}6)$$

Random points for which the value of $\sin\varphi$ doesn't exceed an upper bound are considered to have a good geometry. The thresholds can be derived from error propagation of the interior camera geometry and the noise level.
In this study the thresholds were found by experiments. The value 0.05 for the cross products of the image rays and 0.2 for the cross product of the normal vectors were sufficient to avoid bad cases. Because of the nearly symmetric geometry in the experiments the impact of the stability of $\mathbf{N}_1\times\mathbf{N}_2$ is very strong and so the threshold very low.

3.2 Application of the RANSAC method

The straight line model, described by (1-4), allows the calculation of the RANSAC algorithm parameters based on the individual geometry of that line representation.

3.2.1 Thresholds for outlier detection and model acceptance

An important item in the RANSAC procedure is the decision in algorithm step 2) if an observation fits or not into the model of a certain random subset. The decision is drawn with the help of a threshold for the residuals.
As residual the coplanarity value $\mathbf{f}$ is taken. It can be seen as 'pseudo residual' - compared to the residuals v of the original observations. It is computed by the scalar triple product, where the observation data is the original one and the line center and line direction come from the random model. The coplanarity value $\mathbf{f}$, expressing the volume of the error tetrahedon, grows strongest along the normal of the projection plane.
The coplanarity values are scaled by their pseudo weights $\mathbf{Q}_{e\,i}$ to make them comparable to each other. The pseudo residual $\mathbf{d}_i$ for one point is

$$d_i = f_i \sqrt{Q_{e_i}} \qquad (3\text{-}7)$$

A point is called outlier when its pseudo residual $\mathbf{d}_i$ is larger than a threshold t = 3 * noiselevel. The noiselevel is the expected noise of the image observations. Thus the comparison is done in image space. For the further processing the observation weights are set to 0 in order to mark an outlier and to 1 for an accepted point. The number of accepted points for one random subset is counted.

The threshold for the model acceptance in algorithm step 4) has to decide which one of the models fits the data best. The primary best fit criterion is the number of inliers, the secondary is the a posteriori reference variance.

$$\hat{\sigma}^2 = \mathbf{f}\,\mathbf{W}_e\,\mathbf{f} \qquad (3\text{-}8)$$

Of course the search for the best model is limited to those models provided by the used random subsets. The chances for better results grow with the number of subsets.

3.3 RANSAC in the presence of noise

What happens to the RANSAC solution, when the data is as well noisy as contaminated with outliers? - Up to now the RANSAC distinguishes between good and bad observations. The procedure is designed to find a subset containing only good points, even if the noise in these good points might be so heavy that the solution becomes wrong. One method against that is to get rid of those random points which are very close to each other and easily produce an incorrect solution.
But even points at different ends of a line segment may in some case produce a solution deviating from the correct line more than 3*noiselevel for the most image points. The consequence is the detection of more outliers than really exist and the rejection of the model.

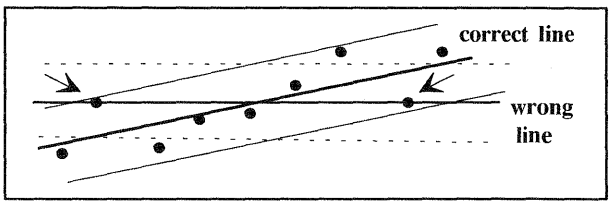

Fig. 4: RANSAC-solution for a line, based on a random subset with noisy observations

The basic idea to cope with the noise in the observations of the random subset is to use a least squares adjustment, after an initial solution is found by the closed form solution. Because the observation weights are set to 1 and 0, only the reduced set of inlying points is used for the adjustment. The normal distributed noise in the points will affect the least squares solution come closer to the correct line than the RANSAC solution and find more inlying points. After some iterations this process is able to find the best possible solution of a certain subset. The iterations are stopped when no more inliers are found. If the same data is used, all random subsets will end up in exactly - with respect to the computing precision - the same least squares solution.

3.4 Algorithm for a combined procedure of RANSAC and 1/0-weighted least squares adjustment

The main steps of the algorithm tested in the simulation experiments are the following:

- determine number of subsets
for i=0 up to the number of subsets
{
  - Choose 2 different random points in 2 random images.
  - Compute the line direction for the actual subset.
  - Check the stability; if geometry not stable reject subset.
  - Compute the line center for the actual subset.
  - Set all observation weights to 1.
  - Compute the residuals $\mathbf{d}_i$ for all points with respect to the actual solution and detect outliers.
  - Count the number of inliers.
  if (more than 4 inliers (⇒ random points))
  while (number of inliers changes)
  {
    - Do a least squares adjustment with the weights 1 or 0.
    - Reset all observation weights to 1.
    - Compute residuals, detect outliers and count inliers.
    - Compute the reference variance of the actual fit.
  } end of while-loop
} end of for-loop
- Find the best subset as the solution with the maximum number of inliers and the smallest reference variance.

4. SIMULATION EXPERIMENTS

A straight line reconstruction was simulated with the help of a generated space line.

4.1 The simulation data

The space line is observed by 4 cameras and represented by 96 image points equally distributed over the 4 images. The image format is 512 × 512 pixels. For the generation a line center point C (0,0,0) and a normalized line direction vector β ($\sqrt{1/3}$,$\sqrt{1/3}$,$\sqrt{1/3}$) is used.
In this experiment the 24 points per image correspond to the points in the other images. In general, however, this correspondence is not required for the chosen line representation.

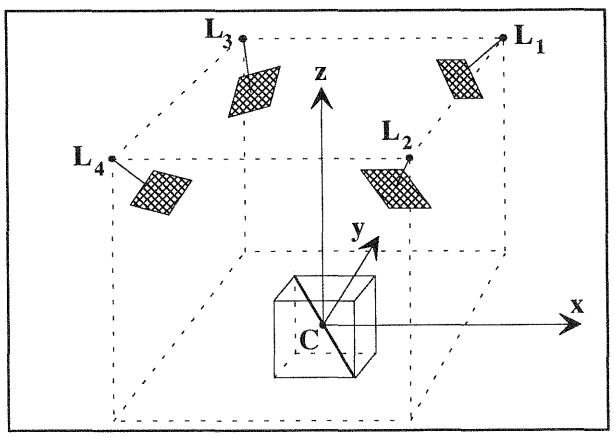

Fig. 5: Camera positions in the simulation experiments

The camera constant is 950 pixel for each camera. The positions and orientation angles of the cameras are given.
In addition to the basic data the simulation parameters are:
1) noiselevel σ = standard deviation of normal distribution,
2) number of outliers = **o**,
3) size of outliers, size = **s***σ

4.2 Noise and outlier generation

In all experiments Gaussian noise is used, distributed like N(mean, noiselevel) = N(0,1). The noisy point is generated by shifting the original point away from its position in perpendicular direction to the correct line. The perpendicular distance of the shift equals the value of the normal distribution. All points are contaminated with noise.

After the contamination with noise the outlying points are randomly selected. Their number is as equally as possible distributed over the 4 images.
Two variations of outlier generation are used.
The first variation produces outliers which all possess the same perpendicular distance from the correct line. Their position is reached after a shift with random sign and absolute value of size **s***σ. In the following text this kind of outliers is called 'constant' outliers.
The second variation produces outliers, whose absolute value of the perpendicular shift is uniformly distributed in the interval [-s*σ; +s*σ]. That kind of outliers is called 'variable' outliers.

4.3 Experiments

The experiments are carried out by reading the original data, then contaminating the data with normal distributed noise and outliers, marking the outlying points and performing the algorithm listed in 3.4. In the end some values for the table statistics are counted.

For the experiments with outliers which possess a constant distance from the correct line, following values are calculated and listed in the table columns:
[1] percentage of outliers (among 96 points)
[2] constant distance of outliers from line [pixel]
[3] number of subsets for RANSAC procedure
[4] probability [%] for at least 1 good subset (see 2-3)
[5] number of experiments carried out for the results in one line of the table
[6] percentage of experiments which find the correct line, detect all outliers and reject not more than 3 good points = success rate = convergence rate
[7] percentage of experiments which don't converge to the correct line = failure rate
[8] percentage of successful experiments with solutions based on a random subset containing outliers
[9] probability [%] for detection of a good point as outlier = false alarms = ε_1

The probability of ε_1 equals the probability to generate normal noise exceeding 3σ, which is

$$\varepsilon_1 = 2 \int_3^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{x^2}{2\sigma^2}\right) dx = 0.26998\% \quad (4\text{-}1)$$

In an experiment converging to the correct line 0.259% of the good points should be detected as outliers. The practical value to compare this to, is calculated as the (number of false alarms in all successful experiments) divided by the (inlying points of all successful experiments).

[1] cont.	[2] size	[3] sets	[4] prob.	[5] exp.	[6] succ.	[7] fail	[8] out	[9] ε_1
0.0	0	1	100	10000	100.0	0.0	-	0.214
4.2	10	2	95	10000	99.8	0.2	-	0.208
4.2	10	3	99	1000	100.0	0.0	9	0.176
4.2	20	2	95	1000	97.3	2.7	1	0.215
4.2	20	3	99	1000	99.6	0.4	1	0.192
4.2	50	2	95	1000	95.7	4.3	1	0.198
4.2	50	3	99	1000	98.8	1.2	1	0.188
4.2	50	4	99.9	1000	99.9	0.1	1	0.171
4.2	100	2	95	10000	95.9	4.1	-	0.211
4.2	100	3	99	10000	99.1	0.9	-	0.204
4.2	100	4	99.9	10000	99.8	0.2	-	0.198
8.3	10	3	95	3000	99.7	0.3	17	0.205
8.3	10	4	99	2000	99.9	0.1	18	0.183
8.3	10	5	99.5	10000	99.9	0.1	18	0.202
8.3	50	3	95	3000	95.0	5.0	2	0.202
8.3	50	4	99	7000	98.3	1.7	1	0.206
8.3	50	5	99.5	10000	99.4	0.6	2	0.193
8.3	100	3	95	3000	94.5	5.5	2	0.187
8.3	100	4	99	3000	98.5	1.5	1	0.185
8.3	100	5	99.5	10000	99.4	0.6	2	0.198
25.0	10	8	95	10000	98.3	1.7	-	0.197
25.0	10	13	99	10000	99.8	0.2	23	0.182
25.0	50	8	95	1000	88.4	11.6	3	0.178
25.0	50	13	99	1000	96.4	3.6	4	0.212
25.0	100	8	95	10000	86.4	13.6	-	0.213
25.0	100	10	97	10000	90.7	9.3	-	0.203
25.0	100	13	99	10000	94.2	5.8	-	0.198
49.0	10	43	95	1000	93.9	6.1	-	0.176
49.0	10	66	99	1000	98.4	1.6	11	0.224
49.0	50	43	95	1000	81.2	18.8	5	0.231
49.0	50	66	99	1000	91.8	8.2	5	0.213
49.0	100	43	95	1000	64.7	35.3	-	0.161
49.0	100	66	99	1000	72.0	28.0	7	0.161
62.5	10	150	95	1000	79.8	20.2	4	0.237
62.5	100	150	95	1000	50.8	49.2	7	0.252

Table 1: Experimental results for detection of outliers with constant distance from the correct line

Results from table 1:
In a first experiment the algorithm is tested without outliers in the data, but with Gaussian noise N(0,1) in the image coordinates. The table shows that the algorithm converges in 100% to the correct line, so it is able to deal with noise.

The values of [6] + [7] have to equal [5]. This means that the probability not to detect an outlier in a properly converging experiment can be assumed to be zero. The reason of that are the non overlapping noise distributions of the outliers and the line points - in case the outlier distance from the line is at least 10 pixels.

The probability ε_1 to detect good points as outliers is around 0.20% in the experiments. For unknown reasons this rate is better than the theoretical value for false alarms of 0.26%.

The frequency of successful experiments in [6] is compared to the value in [4] for a RANSAC procedure. Because of the additional iterative least squares adjustment the values are different. Even if there is an outlier in the subset, the line may converge to its correct position. In that case the RANSAC solution has provided enough inlying points as input to the adjustment. Column [8] shows the dependency from the size of outliers. The smaller the outlier distance, the easier a subset containing bad points will converge. The increase of the convergence rate for small outlier sizes is an advantage of the algorithm in contrast to RANSAC.
On the other hand the convergence rate is worse than in theory if there are large outliers and a high amount of contamination. In these cases the number of subsets derived from (2-3) does not seem to be sufficient. It still has to be examined if a number of subsets, derived from the exact expression of probability (2-2), brings better results.

A second group of experiments examines the behaviour of the algorithm for uniformly distributed outliers. Here some outliers have a small size so that it is impossible to detect them. The probability to detect all outliers in one experiment is based on ε_2, the probability not to detect one outlier in the interval $[-3*\sigma,+3*\sigma]$.

$$\varepsilon_2 = 2*3*\sigma/s*\sigma \quad (4\text{-}2)$$

So the probability to detect all outliers o in one experiment is

$$\mathbf{P}(\text{detect o outliers}) = (1-\varepsilon_2)^{o} \quad (4\text{-}3)$$

Following values are listed in the table:
[1] percentage of outliers (among 96 points)
[2] maximum size for uniformly distributed outliers [pel]
[3] number of subsets for RANSAC procedure
[4] number of experiments carried out for the results in one line of the table
[5] percentage of experiments which find the correct line, detect all outliers and not more than 3 good points = success rate != convergence rate
[6] percentage of experiments which don't converge to the correct line = failure rate
[7] percentage of successful experiments out of all experiments which converge to the line = [5]*100 / (100 - [6])
[8] theoretical probability [%] for an experiment to find all outliers under the assumption that it has found the correct line

The percentage of experiments finding the correct line, detecting or not all outliers (= convergence rate), equals 100% - [6].

[1] outl.	[2] size	[3] sets	[4] exp.	[5] success	[6] fail	[7] conv.	[8] theory
4.2	10	2	1000	20.7	0.0	20.7	24.0
4.2	10	3	1000	22.3	0.0	22.3	24.0
4.2	20	2	1000	46.3	0.4	46.5	52.2
4.2	20	3	1000	49.8	0.4	50.0	52.2
4.2	50	2	1000	75.9	1.0	76.7	78.1
4.2	50	3	1000	76.9	0.2	77.1	78.1
4.2	100	2	1000	86.0	1.7	87.5	88.5
4.2	100	3	1000	87.9	0.3	88.2	88.5
8.3	10	3	1000	5.1	0.0	5.1	5.8
8.3	10	4	1000	5.0	0.0	5.0	5.0
8.3	20	3	1000	24.8	0.6	25.0	27.3
8.3	20	4	1000	25.3	0.0	25.3	27.3
8.3	50	3	1000	56.7	1.5	57.6	61.0
8.3	50	4	1000	59.3	0.7	59.7	61.0
8.3	100	3	1000	70.8	3.8	73.6	78.4
8.3	100	4	1000	76.3	0.9	77.0	78.4
25.0	10	8	1000	0.1	0.0	0.0	0.0
25.0	10	13	1000	0.0	0.3	0.0	0.0
25.0	20	8	1000	1.2	0.4	1.2	2.0
25.0	20	13	1000	1.8	0.1	1.8	2.0
25.0	50	8	1000	21.5	3.5	22.3	22.7
25.0	50	13	1000	19.8	0.6	19.9	22.7
25.0	100	8	1000	39.7	7.3	42.8	48.1
25.0	100	13	1000	44.6	3.1	46.0	48.1
49.0	10	43	1000	0.0	0.1	0.0	0.0
49.0	20	43	1000	0.0	1.3	0.0	0.1
49.0	50	43	1000	4.0	6.0	4.3	5.5
49.0	50	66	1000	4.0	5.1	4.2	5.5
49.0	100	43	1000	15.9	16.8	19.1	23.9
49.0	100	66	1000	17.3	12.3	19.7	23.9
62.5	100	150	1000	8.2	32.0	12.1	16.1
62.5	100	231	1000	8.0	26.6	10.9	16.1

Table 2: Experimental results for detection of outliers with variable distance from the correct line

Results from table 2:
The practical results in column [7] are quite close to the theoretical values of [8]. The expected detection rate of outliers [8] is slightly higher than the rate in the experiments [7]. The rate of converging experiments is higher than in the experiments with constant sized outliers.

5. CONCLUSIONS AND OUTLOOK

The combined RANSAC/least squares algorithm is an effective method for the robust reconstruction of a space line. The line parameters are derived directly from the observations without any approximate values. A success of the method is guaranteed with a probability that can be predicted under the knowledge of the algorithm parameters and some preknowledge on the data (number and size of outliers). This paper verifies theoretical probabilities by practical results. It also reveals the complexity of dependencies between the algorithm parameters and their effects on the results (success rate, convergence rate, failure rate, frequency of false alarms). This might help to develop a problem orientated parameter choice in order to get a high probability for a robust solution.
The application of the algorithm can be extended to other linear features such as circles or ellipses. In industrial applications we usually have a preknowledge about the approximate feature parameters. Then robust techniques like the M-estimation, which are not dependend on closed form solutions, may be used.

6. REFERENCES

Fischler / Bolles, 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Graphics and Image Processing, June 1981, p. 381-395.

Förstner W., 1989. Robust Methods for Computer Vision. IEEE Computer Society Conference on CV and Pattern Recognition, June 4-8, 1989.

Huber, Peter J., 1981. Robust Statistics. Wiley Series in probability and mathematical statistics.

Krarup, T. / Kubik, K., 1981. The Danish Method; Experience and Philosophy. Deutsche Geodätische Kommission, A/98.

Mikhail, Edward M., 1976. Observations and Least Squares. IEP, New York, pp. 213-217.

Mulawa, David. C., 1988. Photogrammetric Treatment of Linear Features. PhD thesis, Purdue University, USA.

Rousseuw / Leroy, 1987. Robust Regression & Outlier Detection. John Wiley & Sons, New York.