# A Combined RANSAC-Hough Transform Algorithm for Fundamental Matrix Estimation

Richard J.M. den Hollander[†] Alan Hanjalic[‡]

[†] TNO Defence, Security and Safety
Oude Waalsdorperweg 63
P.O. Box 96864, 2509 JG Den Haag, The Netherlands
richard.denhollander@tno.nl

[‡] Information and Communication Theory Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
P.O. Box 5031, 2600 GA Delft, The Netherlands

**Abstract**

In this paper we will consider a combination of the RANSAC algorithm and the Hough transform for fast model estimation under the presence of outliers. The model will be computed by sampling a smaller than minimal subset, followed by a voting process of the remaining data. In order to use the combined method for this purpose, an adequate parameterization of the model in the Hough space is required. We will show that in case of hyperplane and fundamental matrix estimation, there is a similar and very general parameterization possible. It will allow these models to be estimated in a very efficient manner.

## 1 Introduction

The Hough transform determines for every data point the parameter subspace of models it supports, and increases the votes in the Hough space for all these models. An extension of this principle is to vote for sets of data points instead of single points. The subspace of supported models is then smaller, while the number of different point sets is larger. For example, a hyperplane in $\mathbb{R}^N$ is specified by $N$ points, and a single point imposes a $N-1$ dimensional subspace of supported models in the Hough space. When a pair of points is considered, the set of supported models is a $N-2$ dimensional subspace. The voting process is then faster, but we have to consider $\binom{n}{2}$ different pairs instead of only $n$ points. The limiting case is when precisely sets of $N$ points are selected, which then results in a single point in the Hough space. This is the principle of the randomized Hough transform [11]. Instead of the total number of possible sets $\binom{n}{N}$, only a small number of random sets is selected which is sufficient to find the best model.

In contrast to the Hough transform, the RANSAC algorithm [3] samples $N$ points and verifies the amount of support for the corresponding model. In view of the above, it is also possible to sample less than $N$ points and verify the support for each supported model in the parameter subspace. This use of RANSAC in combination with the Hough transform has been proposed in [8, 9] to improve the efficiency and quality of model estimation. It was argued that using sets of $N-1$ points is probably the best choice in terms of efficiency. This results in a one-dimensional subspace of models, which may be parameterized by a single quantity. Then there is no need to accumulate the large $N$ dimensional Hough space.

The number of iterations $J$ needed in the RANSAC algorithm is determined from the required probability of success, i.e. the probability that at least one all-inlier sample is found in $J$ iterations [3]. Let $\varepsilon$ denote the outlier ratio in the data, and $d$ the number of points needed to hypothesize a model. If $p$ is the probability of success, e.g. 0.99, then we have the relation

$$p = 1 - (1 - (1 - \varepsilon)^d)^J \tag{1}$$

The necessary number of iterations of the combined RANSAC and Hough method is clearly lower than for standard RANSAC, since only sets of $d = N-1$ instead of $d = N$ points are sampled for forming model hypotheses.

In general, an explicit parameterization of the fundamental matrix in the Hough space is impractical. Its estimation requires a 7-dimensional voting array (due to the 7 degrees of freedom [4]), which becomes unmanageable even for a moderate number of quantization levels. To be able to use the method in [8], we propose a new parameterization for hyperplanes which can also be applied to the fundamental matrix. The parameterization is based on the nullspace of a sample, where the sample will contain one point less than the minimally required number. For hyperplane estimation, we can include the threshold for the support set directly into the voting process. As a result, the whole range of models supported by the remaining data is taken into account. For fundamental matrix estimation, the correspondences will vote for single models. The resulting estimation by 6-point samples will be very efficient due to the reduced number of iterations. In [8] the quality of the model was also improved by using an error propagation mechanism for the data. Error propagation is not incorporated in our method, since no explicit parameterization of the model is used. Note that the standard RANSAC algorithm also neglects noise effects of points in the sample [2].

Several other modifications of RANSAC have been proposed to speed up the algorithm; the most directed to homography or fundamental matrix estimation. For example, in [10] the feature matching score is used in the selection probabilities of the correspondences in order to sample inliers more often. In [2], hypothesized models are optimized to compensate for noisy inliers and the resulting loss of support points. A faster support set evaluation has been proposed in [1], where a small number of randomly selected points is initially evaluated for support. Only when the hypothesized model has sufficient support points among this number, the remaining data is tested for support.

All these methods apply different speed-up mechanisms than our algorithm, and can therefore be combined with our algorithm to achieve even faster fundamental matrix estimation.

In Section 2 the proposed parameterization technique is discussed for hyperplane estimation. Section 3 describes the application of the method to fundamental matrix estimation. In Section 4, hyperplane and fundamental matrix estimation are evaluated on range data and real image pairs, respectively. Section 5 will conclude the paper.

## 2 Hyperplane estimation

The data points $\mathbf{x}_i$ for $i = 1,\ldots,n$ in $\mathbb{R}^N$ will be denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_N)^\top$. A hyperplane with normal vector $\mathbf{n} = (a_1, a_2, \ldots, a_N)^\top$ and offset $b$ is given by $a_1x_1 + a_2x_2 + \ldots + a_Nx_N + b = 0$. In short, the parameters of the hyperplane will be indicated by $\mathbf{h} = (\mathbf{n}^\top\ b)^\top$. The random samples that will be drawn consist of $N-1$ points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{N-1}\}$, and solving for the hyperplane

$$\begin{bmatrix} \tilde{\mathbf{x}}_1^\top & 1 \\ \tilde{\mathbf{x}}_2^\top & 1 \\ \vdots & \vdots \\ \tilde{\mathbf{x}}_{N-1}^\top & 1 \end{bmatrix} \mathbf{h} = \mathbf{0} \tag{2}$$

yields a two-dimensional space $\{\mathbf{h_1}, \mathbf{h_2}\}$ for $\mathbf{h}$. This nullspace can in practice be computed by a singular value decomposition of the lefthand-side matrix. If the sample $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{N-1}\}$ contains only inliers, then the true hyperplane can be given by a linear combination of the nullspace vectors as

$$\mathbf{h} = \alpha\mathbf{h}_1 + (1-\alpha)\mathbf{h}_2 \tag{3}$$

The value of $\alpha$ can be found by solving $(\mathbf{x}^\top\ 1)\mathbf{h} = 0$ for another inlying point $\mathbf{x}$, and should be the same for all other inliers. The outliers will produce different values for $\alpha$.

To find the true value of $\alpha$ we use a Hough-based voting mechanism for the remaining $n - N + 1$ data points [8]. We could use the projections of $\mathbf{x}$ onto $\mathbf{h}_1$ and $\mathbf{h}_2$ directly for computing $\alpha$, but this may result in $\alpha$ values which are difficult to quantize. In particular, the nullspace vector with the largest singular value, say $\mathbf{h_1}$, is likely to constitute the largest part of $\mathbf{h}$ and therefore $\alpha \approx 1$. The binning of many values close to 1 and possibly some values far from 1 is impractical. It would be more convenient to have an $\alpha$ with equiprobable values over a large range.

For this purpose, we will make use of an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ for the space spanned by $\mathbf{n}_1$ and $\mathbf{n}_2$, which are the normals in $\mathbf{h}_1$ and $\mathbf{h}_2$ from (3). We will take a point $\tilde{\mathbf{x}}_1$ from the sample, and project all vectors $\mathbf{x}_i - \tilde{\mathbf{x}}_1$ for $i = 1,\ldots,n$ (except those from the sample) onto this basis. The point $\tilde{\mathbf{x}}_1$ can be seen as the origin for the space spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, which is shown in Fig. 1 for a line in 2D. From (3) we have that $\mathbf{n} = \alpha\mathbf{n}_1 + (1-\alpha)\mathbf{n}_2$, and since $\mathbf{n}_1$ and $\mathbf{n}_2$ are linear combinations of $\{\mathbf{u}_1, \mathbf{u}_2\}$ we can write

$$\mathbf{n} = c_1\mathbf{u}_1 + c_2\mathbf{u}_2 \tag{4}$$

for certain values $c_1$ and $c_2$. It then follows, that for the inliers the ratio of projections onto $\mathbf{u}_2$ and $\mathbf{u}_1$ becomes

$$\begin{aligned} \frac{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_2}{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1} &= \frac{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top(\mathbf{n}-c_1\mathbf{u}_1)\frac{1}{c_2}}{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1} \\ &= \frac{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1\frac{-c_1}{c_2}}{(\mathbf{x}-\tilde{\mathbf{x}}_1)^\top\mathbf{u}_1} \\ &= \frac{-c_1}{c_2} \end{aligned} \tag{5}$$

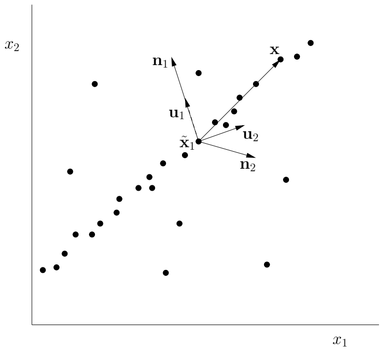


Figure 1: The sampled point $\tilde{\mathbf{x}}_1$ will serve as the origin for the space spanned by $\{\mathbf{n}_1, \mathbf{n}_2\}$. Each point $\mathbf{x}$ is projected onto this space by projecting the vector $\mathbf{x} - \tilde{\mathbf{x}}_1$ onto the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$.

since $(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{n} = -b - (-b) = 0$. The outliers will produce different values for the projection ratio since in that case $\mathbf{x}^\top \mathbf{n} \neq -b$. The projection ratio in (5) will cover a relatively large range of values, and the angle $\gamma$ of the projected vector $\mathbf{x} - \tilde{\mathbf{x}}_1$ with respect to the basis $\{\mathbf{u}_1, \mathbf{u}_2\}$

$$\gamma = \arctan\left(\frac{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2}{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1}\right) \tag{6}$$

offers a quantity which can conveniently be used in a voting space.

In principle, not only the hyperplane which crosses a point $\mathbf{x}$ should receive a vote, but all possible hyperplanes that are within allowable distance from the point. A data point will support a hyperplane if its orthogonal distance to the hyperplane is smaller than a threshold $T$ (which is usually chosen heuristically in the RANSAC algorithm), see Fig. 2. Here the angle $\beta$ determines for which models the indicated point can possibly be a support point, and we have $\sin(\beta) = \frac{T}{\delta}$ where $\delta = \sqrt{((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1)^2 + ((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2)^2}$. A data point will vote for all angles in the range $[\gamma - \beta, \gamma + \beta]$. We note that the distance

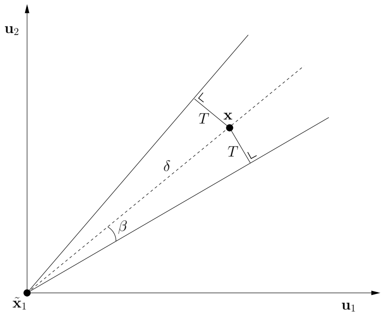


Figure 2: The projection of the point $\mathbf{x}$ in the frame $\{\mathbf{u}_1, \mathbf{u}_2\}$. There is a range of hyperplanes which the point supports. The maximum angle $\beta$ for the range depends on threshold $T$ and the length $\delta$ of the projection of the vector $\mathbf{x} - \tilde{\mathbf{x}}_1$.

from point $\mathbf{x}$ to the hyperplane is equal to the projected distance in the space spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, since the component of $\mathbf{x}$ that lies outside this space is orthogonal to it.

The angle $\gamma$ will be measured in degrees and we choose to use a voting space of 180 bins; one bin for each degree from -90 to 89. After calculating this angle for all points, it should result in a large number of votes in the bin of the true angle. A drawback of using all data points for voting, is that the voting operation may become quite complex for large data sets. Following the concept of the probabilistic Hough transform [6], we can also examine a subset of randomly sampled data points and calculate the best angle for this subset. This should give a sufficiently accurate estimate of the angle $\gamma$ while making the voting process much faster. In the experiments we have chosen for a total of 100 randomly sampled data points, and only in case $n \leq 100$ we use all data.

The bin containing most votes determines the angle $\gamma^*$ for which the final hyperplane is calculated according to

$$\mathbf{h} = \begin{pmatrix} \mathbf{u}_1 + \tan(\gamma^* + \frac{1}{2}\pi)\mathbf{u}_2 \\ -\tilde{\mathbf{x}}_1^\top(\mathbf{u}_1 + \tan(\gamma^* + \frac{1}{2}\pi)\mathbf{u}_2) \end{pmatrix} \tag{7}$$

where point $\tilde{\mathbf{x}}_1$ is taken from the sample.

## 3 Fundamental matrix estimation

The fundamental matrix can be estimated by following roughly the same technique as for hyperplane estimation. However, there are two major differences with the preceding scenario.

First, the 7-point algorithm uses the singularity constraint to determine the fundamental matrix. After seven correspondences are selected, solving for the fundamental matrix yields a two-dimensional nullspace [4]. Then the singularity constraint of the fundamental matrix needs to be used to find the solution. If we sample six points, the resulting nullspace is three-dimensional. We would like to use the singularity constraint for removing one dimension and use voting to find the final solution. Unfortunately, the singularity constraint on the three-dimensional nullspace is a cubic polynomial in two variables, which does not allow voting with respect to a fixed pair of nullspace vectors. As a result, we have to solve the singularity constraint for each correspondence individually. The complexity of the algorithm will therefore increase, but as we have already indicated, a subset of the data points will suffice in the voting process.

Second, since there is no fixed two-dimensional nullspace during voting, we can not calculate the range of allowable models as in Fig. 2. The number of fundamental matrices consistent with a seventh correspondence will be either one or three, just like for the 7-point algorithm. Therefore, there is no range of matrices for which e.g. the Sampson distance can be evaluated, and votes are cast for either one or three separate angles.

To start the estimation process we sample 6 correspondences
$\{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$, and solve

$$\begin{bmatrix} \tilde{x}'_1\tilde{x}_1 & \tilde{x}'_1\tilde{y}_1 & \tilde{x}'_1 & \tilde{y}'_1\tilde{x}_1 & \tilde{y}'_1\tilde{y}_1 & \tilde{y}'_1 & \tilde{x}_1 & \tilde{y}_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{x}'_6\tilde{x}_6 & \tilde{x}'_6\tilde{y}_6 & \tilde{x}'_6 & \tilde{y}'_6\tilde{x}_6 & \tilde{y}'_6\tilde{y}_6 & \tilde{y}'_6 & \tilde{x}_6 & \tilde{y}_6 & 1 \end{bmatrix} \mathbf{f} = \mathbf{0} \tag{8}$$

which results in a three-dimensional space of solutions

$$\mathbf{f} = \alpha \mathbf{f}_1 + \beta \mathbf{f}_2 + (1 - \alpha - \beta)\mathbf{f}_3 \tag{9}$$

If we take a single correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ and solve

$$\begin{pmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{pmatrix} \mathbf{f} = 0 \tag{10}$$

for $\mathbf{f}$ from (9) we get a linear constraint in $\alpha$ and $\beta$. When the correspondence is an inlier, the true values for $\alpha$ and $\beta$ will satisfy this constraint. Let the resulting linear relation be $\beta = r\alpha + g$. Then we use the singularity constraint

$$\det(\alpha F_1 + (r\alpha + g)F_2 + (1 - \alpha - (r\alpha + g))F_3) = 0 \tag{11}$$

where $F_1$, $F_2$, and $F_3$ are the $3 \times 3$ matrices containing the elements of $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$, respectively. This will result in either one or three real solutions for $\alpha$ and thus for $\mathbf{f}$. Now, writing the vectors $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$ in (9) as $\mathbf{f}_1 = (\mathbf{n}_1^\top \; b_1)^\top$, $\mathbf{f}_2 = (\mathbf{n}_2^\top \; b_2)^\top$ and $\mathbf{f}_3 = (\mathbf{n}_3^\top \; b_3)^\top$, we construct an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ from $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$. This basis is used for the projection of the solutions for $\mathbf{f}$. In particular, we calculate the angles

$$\gamma_1 = \arctan\left(\frac{(f_1 \cdots f_8)\mathbf{u}_2}{(f_1 \cdots f_8)\mathbf{u}_1}\right) \qquad \gamma_2 = \arctan\left(\frac{(f_1 \cdots f_8)\mathbf{u}_3}{(f_1 \cdots f_8)\mathbf{u}_2}\right) \tag{12}$$

and use them to cast a vote in a two-dimensional array. The angles will be rounded towards full degrees in the range -90 to 89.

As in hyperplane estimation, we do not use all data points during voting. When the data set contains more than 100 correspondences, only 100 randomly selected correspondences are considered. Examples of vote distributions are given in [5].

After having located the values of $\gamma_1^*$ and $\gamma_2^*$ for the bin containing most votes, we can find the first eight elements of the corresponding $\mathbf{f}$ by

$$\begin{pmatrix} f_1 \\ \vdots \\ f_8 \end{pmatrix} = \mathbf{u}_1 + \tan(\gamma_1^*)\mathbf{u}_2 + \tan(\gamma_1^*)\tan(\gamma_2^*)\mathbf{u}_3 \tag{13}$$

and the last element by

$$f_9 = -\begin{pmatrix} \tilde{x}_1'\tilde{x}_1 & \tilde{x}_1'\tilde{y}_1 & \tilde{x}_1' & \tilde{y}_1'\tilde{x}_1 & \tilde{y}_1'\tilde{y}_1 & \tilde{y}_1' & \tilde{x}_1 & \tilde{y}_1 \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_8 \end{pmatrix} \tag{14}$$

The correspondence $\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}_1'$ is part of the 6-point sample, and therefore lies on the final $\mathbf{f}$.

The fundamental matrix that is found this way does not automatically satisfy the singularity constraint. Due to the rounding effect in the voting array, the matrix will slightly deviate from a singular one. We can solve this by applying the SVD to this matrix, and setting the smallest singular value to zero [4]. A prerequisite for this to work properly is a normalization of the correspondences. This entails a translation which results in zero means for the $(x, y)$ coordinates, followed by a scaling which makes their average distance to the origin equal to $\sqrt{2}$. The transformation is applied to both images' correspondences independently. Before the support of the fundamental matrix is evaluated, the coordinates are transformed back again to their original values.

The whole sequence of steps in the estimation process is listed in Fig. 3. Note that the number of iterations $J$ is determined adaptively as in [4]. When the largest support set so far is found, i.e. $|S_j| > |S_{max}|$, the outlier ratio $\varepsilon$ is updated accordingly. The number of iterations $J$ is then recomputed according to (1).

```
• $j = 1, \quad J = \infty, \quad S_{max} = \emptyset$
• Normalize the image correspondences.
• **while** $j < J$ **do**
    • Randomly select 6 correspondences $\{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$ and use them to compute their nullspace $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$ by solving (8).
    • Determine the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ for the space spanned by the normals in $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$.
    • **if** $n > 100$ **then**
        • form the set $C$ by randomly selecting 100 correspondences from $\{\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1, \ldots, \mathbf{x}_n \leftrightarrow \mathbf{x}'_n\} \setminus \{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$
    • **else**
        • form $C = \{\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1, \ldots, \mathbf{x}_n \leftrightarrow \mathbf{x}'_n\} \setminus \{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \ldots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$
    • **for each** $\mathbf{x} \leftrightarrow \mathbf{x}'$ **in** $C$ **do**
        • Find the possible solutions for $\mathbf{x} \leftrightarrow \mathbf{x}'$ by solving (10) and (11).
        • Determine $\gamma_1$ and $\gamma_2$ according to (12) for each solution, and round the angles to the nearest degree.
        • Add one vote for each pair of angles $(\gamma_1, \gamma_2)$ in the voting array.
    • Determine the pair $(\gamma_1^*, \gamma_2^*)$ with the maximum number of votes.
    • Construct $\mathbf{f}$ from (13) and (14) for $\gamma_1^*$ and $\gamma_2^*$.
    • Find the closest approximation $\hat{\mathbf{f}}$ to $\mathbf{f}$ with $\det(\hat{F}) = 0$ using the SVD.
    • Determine the set of support points $S_j$ for the denormalized $\hat{\mathbf{f}}$, by verifying which points are within distance $T$.
    • **if** $|S_j| > |S_{max}|$ **then**
        • $J = \log(1-p) \cdot \log^{-1}\left(1 - \left(\frac{|S_j|}{n}\right)^6\right)$
        • $S_{max} = S_j$
    • $j = j + 1$
• Re-estimate the fundamental matrix based on the largest support set $S_{max}$.
```

Figure 3: The RANSAC-Hough algorithm for fundamental matrix estimation using a two-dimensional voting space.

|  | $\sum J_R$ $(\cdot 10^3)$ | $\sum J_{RH}$ $(\cdot 10^3)$ | $t_R$ | $t_{RH}$ | $\sum \lvert S_{max} \rvert_R$ $(\cdot 10^4)$ | $\sum \lvert S_{max} \rvert_{RH}$ $(\cdot 10^4)$ | #$\mathbf{h}_R$ | #$\mathbf{h}_{RH}$ |
|---|---|---|---|---|---|---|---|---|
| image 0 | $9.50 \pm 1.42$ | $1.63 \pm 0.17$ | $19.9 \pm 2.37$ | $3.84 \pm 0.31$ | $6.29 \pm 0.032$ | $6.29 \pm 0.033$ | $14.0 \pm 0.7$ | $13.8 \pm 0.6$ |
| image 1 | $3.41 \pm 0.48$ | $0.72 \pm 0.09$ | $5.47 \pm 0.64$ | $1.39 \pm 0.13$ | $6.37 \pm 0.029$ | $6.37 \pm 0.028$ | $10.8 \pm 0.6$ | $10.4 \pm 0.6$ |
| image 2 | $5.36 \pm 0.75$ | $1.05 \pm 0.11$ | $9.58 \pm 1.53$ | $2.06 \pm 0.24$ | $6.41 \pm 0.034$ | $6.41 \pm 0.034$ | $13.0 \pm 0.7$ | $12.7 \pm 0.6$ |
| image 3 | $2.12 \pm 0.37$ | $0.51 \pm 0.07$ | $4.13 \pm 0.70$ | $1.13 \pm 0.14$ | $6.41 \pm 0.028$ | $6.41 \pm 0.026$ | $10.0 \pm 0.6$ | $9.7 \pm 0.6$ |
| image 4 | $2.88 \pm 0.50$ | $0.63 \pm 0.08$ | $5.71 \pm 0.73$ | $1.45 \pm 0.14$ | $6.41 \pm 0.029$ | $6.40 \pm 0.028$ | $11.1 \pm 0.7$ | $10.6 \pm 0.6$ |
| image 5 | $2.55 \pm 0.34$ | $0.58 \pm 0.07$ | $4.32 \pm 0.57$ | $1.14 \pm 0.13$ | $6.39 \pm 0.029$ | $6.39 \pm 0.030$ | $10.0 \pm 0.6$ | $9.7 \pm 0.6$ |
| image 6 | $5.03 \pm 0.68$ | $1.00 \pm 0.11$ | $9.80 \pm 1.18$ | $2.18 \pm 0.20$ | $6.36 \pm 0.029$ | $6.36 \pm 0.031$ | $12.3 \pm 0.6$ | $12.0 \pm 0.6$ |
| image 7 | $1.62 \pm 0.29$ | $0.41 \pm 0.05$ | $3.74 \pm 0.39$ | $1.07 \pm 0.08$ | $6.44 \pm 0.026$ | $6.44 \pm 0.024$ | $9.7 \pm 0.6$ | $9.3 \pm 0.5$ |
| image 8 | $2.00 \pm 0.36$ | $0.47 \pm 0.07$ | $3.61 \pm 0.57$ | $1.00 \pm 0.11$ | $6.40 \pm 0.028$ | $6.40 \pm 0.028$ | $9.2 \pm 0.6$ | $8.9 \pm 0.6$ |
| image 9 | $1.91 \pm 0.37$ | $0.46 \pm 0.07$ | $4.09 \pm 0.52$ | $1.14 \pm 0.10$ | $6.44 \pm 0.029$ | $6.44 \pm 0.029$ | $10.2 \pm 0.7$ | $9.7 \pm 0.7$ |

Table 1: The results for finding all planes using RANSAC (R) and RANSAC-Hough (RH) in the ABW range images. Indicated are the averages and standard deviations (±) for the total number of iterations $\sum J$ per image, the running time $t$ in seconds, the total size of the maximum support sets $\sum |S_{max}|$ and the number of planes #**h** found.

# 4 Experimental results

We will compare the proposed RANSAC-Hough method with the standard RANSAC algorithm for plane fitting and fundamental matrix estimation. For all experiments we report the averages and standard deviations over a number of runs of both the executed number of iterations $J$ and the size of the maximum support set $|S_{max}|$. Furthermore, the average running time for a single run is listed. The final re-estimation step in RANSAC will be omitted. The algorithms were implemented in C and ran on Intel Xeon 3.07 GHz / 3.2 GHz computers. For implementation details see [5].

## 4.1 Plane fitting

As application we consider the fitting of planes in range image data. We have used 10 images (“train” 0 to 9) from the ABW structured light scanner in the USF database[1]. The images contain several different planar objects, and the intensity values correspond to the measured depth by the scanner. An example of one of the images is shown in Fig. 4(a). We have subsampled the images with a factor 2 to obtain $256 \times 256$ sized images. We search with RANSAC for planes in the images, and subsequently delete the points from the data set which belong to a plane. The repeated application of RANSAC is stopped when a plane is returned with support smaller than 500 pixels. For the shown example image, the number of planes extracted this way will be about 12. The experiment is repeated 500 times for each image. The threshold for the orthogonal distance to the plane is set to $T = 2.5$, which is large enough to capture noisy variations of the inliers. Table 1 shows the results of the experiments.

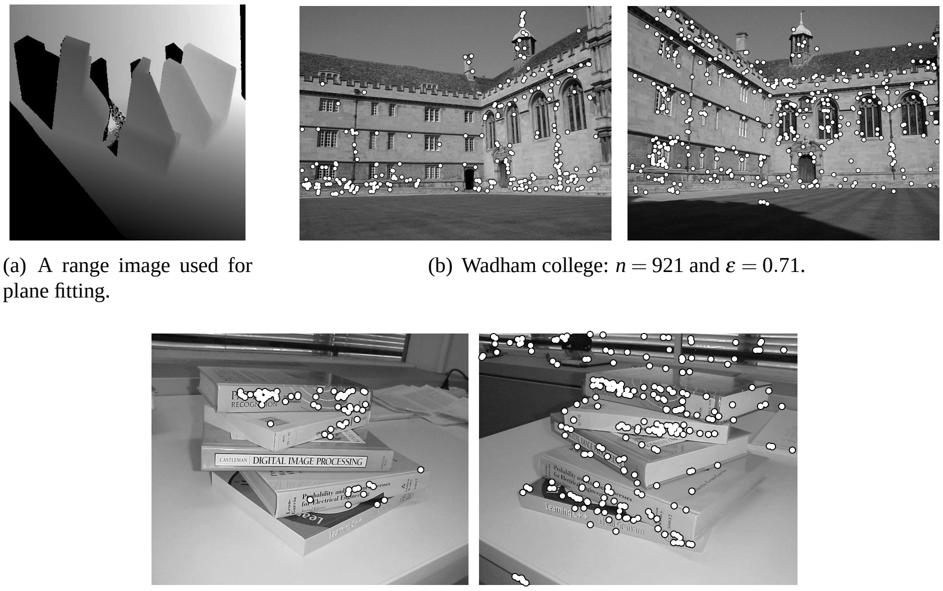
The RANSAC-Hough method outperforms RANSAC in all aspects; in some cases it is up to a factor 5 faster. The total number of points on the extracted planes is comparable, while the number of planes is a bit smaller. This means that the extracted planes are actually better fits, since they contain a larger part of the data.

## 4.2 Fundamental matrix estimation

Some of the real images we have used for testing are shown in Fig. 4(b) and 4(c). There are differences in viewpoint and/or zoom factor between the left and right images. The

[1] Available at http://marathon.csee.usf.edu/range/seg-comp/images.html.

SIFT keypoint detector[2] [7] has been applied for establishing correspondences between the image pairs. The left images in Fig. 4(b) to 4(c) show the final sets of inlying feature points, and the right images the outlying feature points. Also indicated for every image pair are the total number of correspondences $n$ and the outlier ratio $\varepsilon$. The Sampson distance is chosen as error measure, and we have set the square root of the threshold to $T = 1.5$ pixels.



(a) A range image used for plane fitting.

(b) Wadham college: $n = 921$ and $\varepsilon = 0.71$.

(c) Pile of books: $n = 548$ and $\varepsilon = 0.82$.

Figure 4: Some of the images used in the experiments.

The results of running RANSAC 500 times on the image pairs are shown in Table 2. The difference in running times is best noticeable for higher outlier ratios. The number of iterations is reduced here considerably and the additional complexity of the voting process does not prohibit a speedup anymore. The support sets found are slightly smaller than those for RANSAC. This is a result of the rank 2 enforcement which finds an approximation of the fundamental matrix computed from the data. Since the data is not considered in finding this approximation, some inliers are lost in the process.

# 5 Discussion

The combination of RANSAC and the Hough transform, that has been advocated in the past, is made applicable to hyperplanes and the fundamental matrix by a new parameterization of the model. For hyperplanes, the result is an efficient one-dimensional voting space and a reduction of the sample size by one point. For the fundamental matrix, a two-dimensional voting space is applied because of the singularity constraint. Instead of sampling 7 correspondences per model, we now only need to take 6-point samples. This

[2] The code is obtained from http://www.cs.ubc.ca/∼lowe/keypoints/.

| image pair | $\varepsilon$ | # inliers | $J_R$ | $J_{RH}$ | $\lvert S_{max}\rvert_R$ | $\lvert S_{max}\rvert_{RH}$ | $t_R$ | $t_{RH}$ |
|---|---|---|---|---|---|---|---|---|
| books | 0.74 | 189 | $7.2 \pm 0.64 \cdot 10^4$ | $1.92 \pm 0.17 \cdot 10^4$ | $187 \pm 2.5$ | $185 \pm 2.9$ | $25.1 \pm 3.6$ | $11.7 \pm 1.9$ |
| pile of books | 0.82 | 97 | $4.03 \pm 0.71 \cdot 10^5$ | $0.93 \pm 0.18 \cdot 10^5$ | $109 \pm 2.9$ | $106 \pm 3.5$ | $114 \pm 24.2$ | $52.5 \pm 12.0$ |
| Wadham college | 0.71 | 264 | $7.08 \pm 2.93 \cdot 10^4$ | $1.96 \pm 0.73 \cdot 10^4$ | $241 \pm 13.8$ | $236 \pm 14.4$ | $29.0 \pm 12.4$ | $12.4 \pm 4.9$ |
| Univ. British Columbia | 0.56 | 399 | $2.65 \pm 0.56 \cdot 10^3$ | $1.14 \pm 0.25 \cdot 10^3$ | $372 \pm 11.0$ | $369 \pm 12.8$ | $1.13 \pm 0.28$ | $0.72 \pm 0.20$ |
| Corridor | 0.43 | 150 | $466 \pm 160$ | $261 \pm 95.5$ | $139 \pm 5.8$ | $138 \pm 6.8$ | $0.08 \pm 0.031$ | $0.13 \pm 0.055$ |
| Valbonne church | 0.58 | 127 | $2.56 \pm 0.63 \cdot 10^3$ | $1.23 \pm 0.40 \cdot 10^3$ | $123 \pm 3.7$ | $121 \pm 5.4$ | $0.49 \pm 0.13$ | $0.64 \pm 0.23$ |

Table 2: Fundamental matrix estimation using RANSAC (R) and RANSAC-Hough (RH) on real image pairs. Indicated are the averages and standard deviations (±) for the executed number of iterations $J$, the maximum number of support points $|S_{max}|$ and the running time $t$ in seconds.

makes it much easier to find an all-inlier sample by random trials. In addition, we use for both models randomly selected subsets of the data to speed up the voting stage.

The consecutive extraction of planes in range images took considerably less time using the RANSAC-Hough method. The quality of the solutions is either equal or better than standard RANSAC. In case of the fundamental matrix, a much faster estimation is achieved for high outlier ratios, with only a minor decrease in the size of the support.

A further improvement of the algorithm may be circumventing the loss of support points caused by enforcement of the singularity constraint.

# References

[1] O. Chum and J. Matas. Randomized ransac with $T_{d,d}$ test. In *Proc. British Machine Vision Conference*, 2002.

[2] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Proc. DAGM*. Springer-Verlag, 2003.

[3] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[4] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[5] R.J.M. den Hollander and A. Hanjalic. A six-point RANSAC algorithm for fundamental matrix estimation. Technical Report ICT-2007-02, Delft University of Technology, 2007.

[6] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[7] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[8] C.F. Olson. Constrained Hough transforms for curve detection. *Computer Vision and Image Understanding*, 73(3), 1999.

[9] C.F. Olson. A general method for geometric feature matching and model extraction. *Int. Journal of Computer Vision*, 45(1), 2001.

[10] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *Proc. European Conference on Computer Vision*, 2002.

[11] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough Transform (RHT). *Pattern Recognition Letters*, 11(5), 1990.