ROBUST ESTIMATION WITH NON LINEAR REPULSIVE PARTICLE SWARM OPTIMIZATION

Onay Urfalıoḡlu

urfaliog@lfi.uni-hannover.de
University of Hannover
Information Technology Laboratory
Hannover, Germany

ABSTRACT

The RPSO algorithm was introduced in a previous paper as an application for robust estimation of epipolar geometry and focal length from point correspondence sets with high outlier rates. It was shown that using RPSO and an appropriate cost function the number of required iterations was significantly reduced at high outlier rates, compared to RANSAC based robust estimation. In this paper a new global optimization algorithm called Non Linear Repulsive Particle Swarm Optimization (NLRPSO) is proposed showing significant improvements compared to RPSO. NLRPSO is tested on robust estimation of epipolar geometry and focal length where the form of the utilized cost function are derived analytically.

1. INTRODUCTION

Estimation algorithms are based on a model which describes the parameters of the entity which is supposed to be estimated. Having some kind of data, it is the task of the algorithm to find a set of parameters which fits to the model optimally. However, often the data set is contaminated with data, also called outlier, which does not conform to the model at all. An estimation which takes into account the contamination of the data set is called *Robust Estimation* (RE). Many conventional approaches for RE are based on the RANSAC [3] algorithm. There have been several enhancements and modifications of RANSAC [11, 1, 2, 10, 7, 6] which are all based on the common random sampling scheme. One disadvantage of random sampling based algorithms is the computational expense at high outlier rates because the number of required iterations increases very rapidly with the rate of outliers. This increase makes the use of RANSAC unpractical at high outlier rates.

This paper presents a global optimization algorithm called Non Linear Repulsive Particle Swarm Optimization (NLRPSO). It is tested as a robust estimator for epipolar geometry and focal length.

In the following section the impact of outliers on an error model is discussed. Section 3 introduces an approach for the robust estimation of epipolar geometry and focal length with data sets containing outliers. Section 4 presents the proposed NLRPSO algorithm. In section 5 results of the experiments are shown and in the last section the paper is concluded.

2. ERROR MODELS

Often, there is a given data set which has to be processed somehow in order to estimate any entity. As an example, line estimation in 2D is considered. We assume that some points are detected which are known to lie on a certain line. Furthermore, we assume that the points are noisy and the error probability distribution is known enabling *Maximum Likelihood Estimation* (MLE).

2.1. The error model for inliers

With $p_{\hat{\mathbf{x}}_i}$ being the error *Probability Density Function* (PDF) for the position of the estimated point $\hat{\mathbf{x}}_i$ we have to optimize the following term:

$$L = \Pi_i \; p_{\hat{\mathbf{x}}_i}(\hat{\mathbf{x}}_i|\mathbf{x}_i) \qquad (1)$$

where $\mathbf{x}_i$ is the corresponding measured point. The likelihood L can be interpreted as a cost function and the task of the estimation is to find the global optimum. In our example we assume that the PDF of the points is Gaussian with variance σ^2:

$$p_{\hat{\mathbf{x}}_i}(\hat{\mathbf{x}}_i|\mathbf{x}_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{|\hat{\mathbf{x}}_i - \mathbf{x}_i|^2}{2\sigma^2}) \qquad (2)$$

so the likelihood is:

$$L = \Pi_i \; \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{|\hat{\mathbf{x}}_i - \mathbf{x}_i|^2}{2\sigma^2}). \qquad (3)$$

Estimating the line $\hat{\mathbf{l}}$, we need to express the PDF using the distance to $\hat{\mathbf{l}}$:

$$p_d = \frac{2}{\sqrt{2\pi}\sigma} \exp(-\frac{\mathrm{d}(\hat{\mathbf{l}}, \mathbf{x}_i)^2}{2\sigma^2}). \tag{4}$$

The likelihood becomes

$$L = \Pi_i \frac{2}{\sqrt{2\pi}\sigma} \exp(-\frac{\mathrm{d}(\hat{\mathbf{l}}, \mathbf{x}_i)^2}{2\sigma^2}) \tag{5}$$

utilizing the distance $\mathrm{d}(\hat{\mathbf{l}}, \mathbf{x}_\mathrm{i})$ of the point $\mathbf{x}_i$ to the estimated line. It should be mentioned that the variance changes:

$$\begin{aligned} E[d] &= \int_0^\infty \frac{2e^{-\frac{t^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} t\, dt = \frac{\sqrt{2}}{\pi}\sigma \\ E[(d - E[d])^2] &= \int_0^\infty \frac{2e^{-\frac{t^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} (t - E[d])^2\, dt \\ &= \sigma^2 \frac{\pi - 2}{\pi} \end{aligned} \tag{6}$$

This means that by describing the PDF by the distance the variance becomes

$$\sigma^2 \rightarrow \sigma_d^2 = \sigma^2 \frac{\pi - 2}{\pi}. \tag{7}$$

What happens when we encounter data which is not compatible to our model? In this case any point $\mathbf{x}_o$ which suffices the condition

$$\mathrm{distance}(\mathbf{l}, \mathbf{x}_o) \gg \sigma \tag{8}$$

may be considered as an outlier and the estimation of the line will fail with the cost function (3) because it is based on a wrong PDF in that particular case.

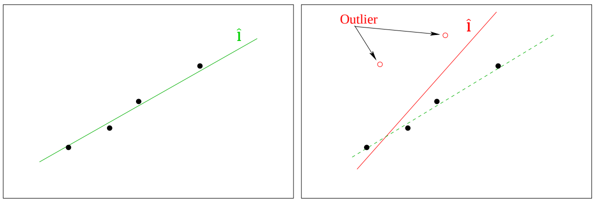

Figure 1: *Left: estimation succeeds without outliers. Right: estimation fails due to outliers.*

The next section describes how to make cost functions robust for a data set which is contaminated with outliers.

2.2. The error model for outliers

Data sets containing outliers indicate that the error model is not general enough. In order to generalize our model in the case of line estimation it is required to determine the PDF for the outliers. We assume that an outlier point has constant PDF regarding its *position*:

$$q_{\mathbf{x}_o}(\mathbf{x}_o) = \frac{1}{A} \tag{9}$$

where A is the area within the outlier point may be located. In order to determine the PDF represented by the distance we need to analyse all constellations of line - outlier point locations. We assume for symmetry reasons that the area where all points may occur is bounded by a circle with radius R.

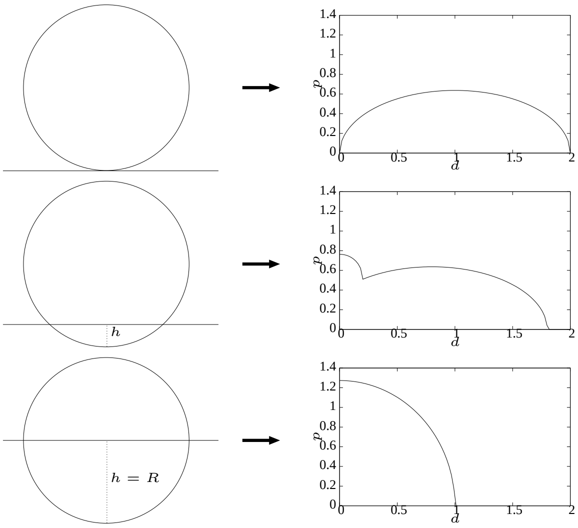

Figure 2: *Left column: location of the line relative to the circle. Right column: the corresponding outlier point distance PDF.*

In fig. 2 three examples for line locations and the corresponding PDF are shown. The PDF for a tangential line is given by:

$$q_d(d) = \frac{2}{\pi R^2} \sqrt{R^2 - (d - R)^2}\, \mathrm{rect}(2R, d) \quad d \in [0, 2R] \tag{10}$$

where

$$\mathrm{rect}(a, x) = \left\{ \begin{array}{ll} 1 & \text{for} \quad x \in [0, a] \\ 0 & \text{else} \end{array} \right. . \tag{11}$$

Displacing the line parallelly by h towards the center of the circle results in:

$$q_h(d) = q_d(d + h) + q_d(-d + h). \tag{12}$$

We need to determine the mean PDF considering all possible line locations due to the fact that we do not know the real line:

$$\begin{aligned}\overline{q}_d(d) &= \frac{1}{2R}\int_0^{2R} q_h(d)dh \\ &= \frac{1}{2R^3\pi}\left[(\pi+2C)R^2+2S(R\sqrt{d}-\sqrt{d^3})\right]\end{aligned} \tag{13}$$

where

$$\begin{aligned}S &= \sqrt{2R-d} \\ C &= \arcsin(\frac{R-d}{R}).\end{aligned} \tag{14}$$

Fig. 3 shows the plot of $\overline{q}_d(d)$.

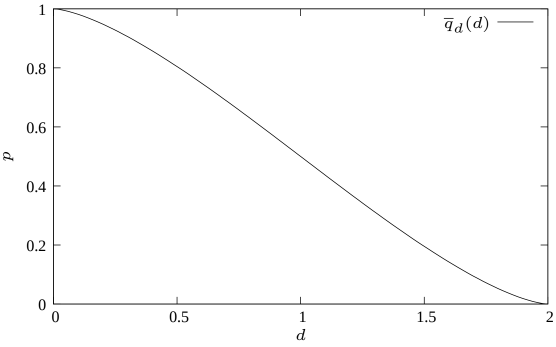

Figure 3: *Mean PDF* $\overline{q}_d(d)$ *for* $R = 1$.

2.3. The generalized error model

The knowledge about the error model for inliers and outliers, respectively, enables to determine the error model for the general case. With the inlier rate:

$$\alpha = \frac{I}{N} \tag{15}$$

where I is the number of inliers and N is the number of points, the generalized PDF is defined by [11]:

$$p_g(d) = \alpha p_d(d) + (1-\alpha)\overline{q}_d(d). \tag{16}$$

In the next section we derive the generalized PDF for robust estimation of epipolar geometry and focal length in a similar way.

3. ROBUST ESTIMATION OF EPIPOLAR GEOMETRY AND FOCAL LENGTH

As in the case of line estimation, the estimation of epipolar geometry also may have to deal with data containing outliers due to a correspondence set C with wrong point correspondences between two camera perspectives. The outliers may be divided in two categories: random-outliers and relative-motion-outliers. Random outliers are a result of total mismatches of the correspondence generation algorithm where the feature points do not correspond to each other. Relative motion outliers are a result of moving objects in the scene. Though the correspondence generation succeeds for that kind of feature points the correspondences do not fulfill the required rigidity constraint of the scene and so are detected as outliers. Unlike random outliers, relative motion outliers are grouped where each group represents a moving rigid object and can be described by a certain relative motion to the camera.

The epipolar geometry can be described by the F-matrix $\mathbf{F}$ which encodes the intrinsic (e.g. focal length) and extrinsic (translation and rotation) camera parameters:

$$\mathbf{F} = \mathbf{K}^{'-1}\mathbf{TRK}^{-1} \tag{17}$$

with $\mathbf{K}$ and $\mathbf{K}^{'}$ describing camera intrinsics [4]:

$$\mathbf{K} = \begin{pmatrix} \frac{f}{p_x} & s & c_x \\ 0 & \frac{f}{p_y} & c_y \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{K}' = \begin{pmatrix} \frac{f'}{p_x} & s & c_x \\ 0 & \frac{f'}{p_y} & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{18}$$

where f is the focal length of the camera generating frame 1, f' is the focal length of the camera generating frame 2, s is the skew of a pixel, c_x, c_y are coordinates of the principal point and p_x, p_y describe the pixel size. Except f', all intrinsic parameters are assumed to be known.

$\mathbf{R}$ is the rotation matrix:

$$\mathbf{R} = \begin{pmatrix} s_\varphi s_\vartheta s_\rho + c_\varphi c_\rho & s_\varphi s_\vartheta c_\rho - c_\varphi s_\rho & s_\varphi c_\vartheta \\ c_\vartheta s_\rho & c_\vartheta c_\rho & -s_\vartheta \\ c_\varphi s_\vartheta s_\rho - s_\varphi c_\rho & c_\varphi s_\vartheta c_\rho + s_\varphi s_\rho & c_\varphi c_\vartheta \end{pmatrix} \tag{19}$$

with φ, ϑ and ρ rotation angles and

$$\begin{aligned} s_\varphi = \sin(\varphi),\ s_\vartheta = \sin(\vartheta)\ s_\rho = \sin(\rho) \\ c_\varphi = \cos(\varphi),\ c_\vartheta = \cos(\vartheta)\ c_\rho = \cos(\rho).\end{aligned} \tag{20}$$

The components of camera translation are $T_1,\ T_2,\ T_3$ and $\mathbf{T}$ is defined as:

$$\mathbf{T} = \begin{pmatrix} 0 & T_3 & -T_2 \\ -T_3 & 0 & T_1 \\ T_2 & -T_1 & 0 \end{pmatrix}. \tag{21}$$

With a correct F-matrix the inlier point lies on the appropriate epipolar line as shown in fig. 4.

$\mathbf{F}$ fulfills the epipolar condition [4]:

$$\bar{\mathbf{x}}_{\mathrm{hom}}^{'\top}\mathbf{F}\,\bar{\mathbf{x}}_{\mathrm{hom}} = 0 \tag{22}$$

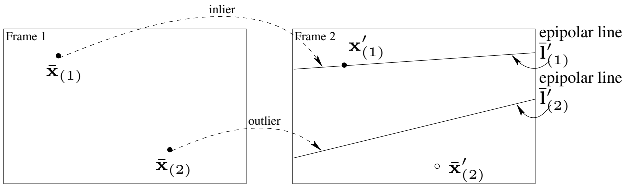

Figure 4: *Epipolar lines and corresponding feature points (inlier and outlier).*

where

$$\bar{\mathbf{x}}'_{\text{hom}} = \begin{pmatrix} \bar{x}' \\ \bar{y}' \\ 1 \end{pmatrix} \text{ and } \bar{\mathbf{x}}_{\text{hom}} = \begin{pmatrix} \bar{x} \\ \bar{y} \\ 1 \end{pmatrix} \tag{23}$$

are noise free homogeneous feature points. The *epipolar line* $\mathbf{l}$ is defined by:

$$\mathbf{l}' = \mathbf{F}\mathbf{x}_{\text{hom}}. \tag{24}$$

The epipolar condition (22) can also be described by:

$$\text{distance}(\bar{\mathbf{l}}', \bar{\mathbf{x}}'_{\text{hom}}) = 0. \tag{25}$$

In the case where the points are noisy the distance between the epipolar line and the appropriate point is not zero so it can be used as an indicator for the correspondence being an inlier or not.

3.1. RANSAC

The RANSAC based approach utilizes the 7-point algorithm [4]. The 7-point algorithm may have 1 or 3 solution candidates per iteration where all candidates are tested on the correspondence set. Therefore, the required number of iterations can be multiplied by an average factor of 2. RANSAC is an iterative algorithm and the number of iterations r required to find a correct sample of 7 correspondences depends on the number of inliers, the number of correspondences and the desired success probability P_s:

$$r = 2\frac{\ln(1 - P_s)}{\ln(1 - (1 - P_7))} \tag{26}$$

where

$$P_7 = \frac{k}{N} \cdot \frac{k-1}{N-1} \cdots \frac{k-6}{N-6}. \tag{27}$$

3.2. Error model for epipolar line - point distance

We assume that the points in frame 1 are noise free and the points in frame 2 have Gaussian PDF. This assumption is sustainable for the KLT tracker [9]. We identify the diameter of the frame by $2R$. Assuming that the PDF of the position of an outlier point $\mathbf{x}_o$ is constant:

$$q_{\mathbf{x}_o}(\mathbf{x}_o) = \frac{1}{A} \tag{28}$$

we face the same situation as in section (2.3). The generalized PDF $p_F(d)$ for the distance is:

$$p_F(d) = \alpha \frac{2e^{-\frac{d^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} + (1-\alpha)\overline{q}_d(d) \tag{29}$$

where $d = \mathrm{d}(\mathbf{l}, \mathbf{x})$ is now defined as the distance between the epipolar line and the appropriate point in the current frame.

3.3. Cost function

In order to determine the cost function for the robust estimation we take a closer look at the PDF $\overline{q}_d(d)$ defined in (13) and calculate the expected value $E_{\overline{q}_d(d)}[d]$:

$$E_{\overline{q}_d(d)}[d] = \int_0^{2R} \overline{q}_d(t)\; t\; dt = \frac{5}{8}R \tag{30}$$

For simplicity and performance reasons, we approximate $\overline{q}_d(d)$ by a Gaussian PDF $g_o(d)$ with the same expected value. The expected value for a generic Gaussian PDF defined whithin $[0, 2R]$ is:

$$\begin{aligned} E_{\text{Gaussian}}[d] &= \frac{1}{\text{erf}(\frac{R\sqrt{2}}{\sigma_o})} \int_0^{2R} \frac{2e^{-\frac{t^2}{2\sigma_o^2}}}{\sqrt{2\pi}\sigma_o}\; t\; dt \\ &= \frac{1}{\text{erf}(\frac{R\sqrt{2}}{\sigma_o})} \sqrt{\frac{2}{\pi}}\sigma_o \left[1 - \exp(-2\frac{R^2}{\sigma_o^2})\right] \end{aligned} \tag{31}$$

The equation

$$E_{\overline{q}_d(d)}[d] \overset{!}{=} E_{\text{Gaussian}}[d] \tag{32}$$

is used to determine the parameter σ_o of the approximated Gaussian PDF. The result is obtained numerically:

$$\sigma_o = \gamma\; R,\; \gamma = 0.811511153. \tag{33}$$

With this result, the approximated Gaussian PDF is fully determined:

$$g_o(d) = \frac{1}{\text{erf}(\frac{\sqrt{2}}{\gamma})} \frac{2e^{-\frac{d^2}{2\sigma_o^2}}}{\sqrt{2\pi}\sigma_o}. \tag{34}$$

Fig. 5 shows the plots of $g_o(d)$ and $\overline{q}_d(d)$ for $R = 1$. After all, our PDF $p_F(d)$ becomes:

$$p_F(d) = \alpha\; \frac{2e^{-\frac{d^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} +\; (1-\alpha)\; B\frac{2e^{-\frac{d^2}{2\sigma_o^2}}}{\sqrt{2\pi}\sigma_o} \tag{35}$$

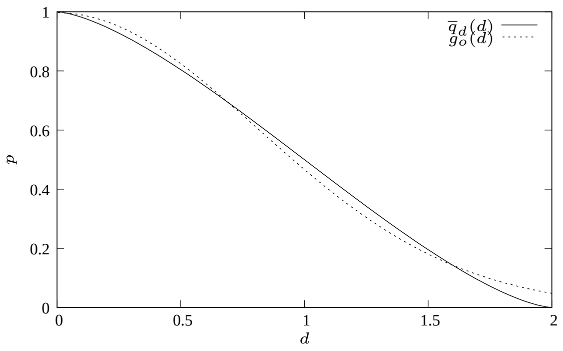

Figure 5: *Mean PDF $\overline{q}_d(d)$ and the approximated PDF $g_o(d)$ for $R = 1$.*

where

$$B = \frac{1}{\mathrm{erf}(\frac{\sqrt{2}}{\gamma})}. \qquad (36)$$

Having N point correspondences, which are assumed to be statistically independent, the compound PDF which we need for the construction of our cost function is:

$$L = \Pi_{i=1}^{N} p_F(d_i) \qquad (37)$$

where

$$d_i = \mathrm{distance}(\hat{\mathbf{l}}_i, \mathbf{x}_i) \qquad (38)$$

is the distance of the i-th point to its corresponding estimated epipolar line for a particular estimated F-matrix. The estimation of the F-matrix requires a maximization of L:

$$L = \Pi_{i=1}^{N} p_F(d_i) \rightarrow \max \qquad (39)$$

In order to ease the numerical computation we utilize the logarithmic likelihood:

$$\begin{aligned} & \log(L) \\ &= \log\left[\Pi_{i=1}^{N} p_F(d_i)\right] \\ &= \sum_{i=1}^{N} \log\left[p_F(d_i)\right] \\ &= \sum_{i=1}^{N} \log\left[\frac{2}{\sqrt{2\pi}}\left[\alpha\frac{e^{-\frac{d_i^2}{2\sigma^2}}}{\sigma} + (1-\alpha)B\frac{e^{-\frac{d_i^2}{2\sigma_o^2}}}{\sigma_o}\right]\right] \\ &= \sum_{i=1}^{N} \log\left[\frac{2}{\sqrt{2\pi}}\right] + \log\left[\alpha\frac{e^{-\frac{d_i^2}{2\sigma^2}}}{\sigma} + (1-\alpha)B\frac{e^{-\frac{d_i^2}{2\sigma_o^2}}}{\sigma_o}\right] \\ &= \sum_{i=1}^{N} \log\left[\frac{2}{\sqrt{2\pi}}\right] + S_F(d_i) \end{aligned} \qquad (40)$$

where

$$S_F(d_i) = \log\left[\alpha\frac{e^{-\frac{d_i^2}{2\sigma^2}}}{\sigma} + (1-\alpha)B\frac{e^{-\frac{d_i^2}{2\sigma_o^2}}}{\sigma_o}\right]. \qquad (41)$$

The term $\log\left[\frac{2}{\sqrt{2\pi}}\right]$ is constant and can be discarded. We are interested in efficient computation so we try to find an approximation for $S_F(d_i)$ which can be calculated more efficiently. The Taylor series developed at $d = 0$ of the term $S_F(d)$ is:

$$\begin{aligned} S_F(d) \quad = \quad & \log\left[\frac{\alpha\sigma_o + B\sigma - B\sigma\alpha}{\sigma\sigma_o}\right] \\ & - \frac{B\sigma^3 - B\sigma^3\alpha + \alpha\sigma_o^3}{2\sigma^2\sigma_o^2(\alpha\sigma_o + B\sigma - B\sigma\alpha)} d^2 \\ & + \mathcal{O}(d^4). \end{aligned} \qquad (42)$$

For small d, we can approximate $S_F(d)$ by $S_A(d)$:

$$S_A(d) = \exp\left[\frac{B\sigma^3 - B\sigma^3\alpha + \alpha\sigma_o^3}{2\sigma^2\sigma_o^2(\alpha\sigma_o + B\sigma - B\sigma\alpha)} d^2\right] \qquad (43)$$

which has the Taylor series expansion (at $d = 0$) of:

$$\begin{aligned} S_A(d) \quad = \quad & 1 \\ & - \frac{B\sigma^3 - B\sigma^3\alpha + \alpha\sigma_o^3}{2\sigma^2\sigma_o^2(\alpha\sigma_o + B\sigma - B\sigma\alpha)} d^2 \\ & + \mathcal{O}(d^4). \end{aligned} \qquad (44)$$

One can see that the second order terms match. Fig. 6 shows the plots of both $S_F(d)$ and $S'_A(d)$ where

$$S'_A(d) = S_A(d) - 1 + \log\left[\frac{\alpha\sigma_o + B\sigma - B\sigma\alpha}{\sigma\sigma_o}\right] \qquad (45)$$

equals $S_A(d)$ up to a constant with the property that

$$S'_A(0) = S_F(0). \qquad (46)$$

The cost function utilized by this approach is now fully determined:

$$\sum_{i=1}^{N} \exp\left[-\frac{1}{2\kappa} d^2\right] \rightarrow \max \qquad (47)$$

where

$$\kappa = \frac{\sigma^2\sigma_o^2(\alpha\sigma_o + B\sigma - B\sigma\alpha)}{B\sigma^3 - B\sigma^3\alpha + \alpha\sigma_o^3}. \qquad (48)$$

The parameter κ is quite invariant regarding the inlier rate and σ_o. Furthermore, it is

$$\sigma \ll \sigma_o \Rightarrow \kappa \sim \sigma^2. \qquad (49)$$

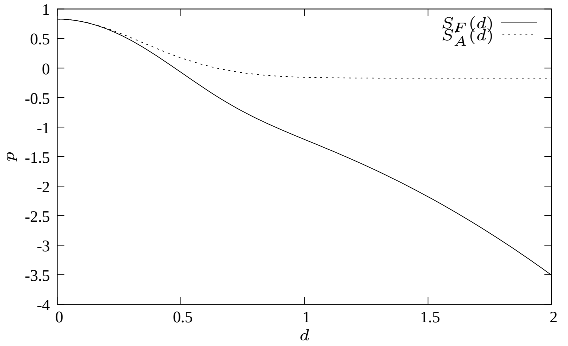

Figure 6: *Plot of the functions $S_F(d)$ and $S'_A(d)$ which match in their second order Taylor terms. The parameters in this plot are: $\sigma = 0.3$, $\sigma_o = \gamma$, $R = 1$ and $\alpha = 0.5$.*

It proves that the estimation succeeds for a

$$\kappa_{\mathrm{cf}} > \kappa \tag{50}$$

up to a certain upper bound for κ_{cf}. This bound depends on the inlier rate and on the threshold τ for the distance by which a correspondence is decided to be an inlier or an outlier. For example it is not necessary to set $\tau = 0$ for noise free points ($\sigma = 0$). The reason for utilizing a greater κ is that the smoothness of the search space hypersurface increases and also the region of the global optimum becomes easier to find.

3.4. Parameterization of the F-matrix

The magnitude of the camera translation can not be determined by the estimation so $\mathbf{T}$ can be parameterized by 2 values. With 3 parameters for the rotation, 2 parameters for the translation and one parameter for the focal length, $\mathbf{F}$ is entirely determined by 6 parameters.

The rotation is parameterized by 3 angles:

$$\varphi, \vartheta, \rho \in [-0.2\frac{2R}{f} \text{ rad}, 0.2\frac{2R}{f} \text{ rad}] \tag{51}$$

The translation is parameterized by 2 parameters:

$$\begin{aligned} \zeta, \eta &\in [0, \pi] \\ T_1 &= \sin(\zeta)\ \cos(\eta) \\ T_2 &= \sin(\zeta)\ \sin(\eta) \\ T_3 &= \cos(\zeta) \end{aligned} \tag{52}$$

where ζ and η are angles describing the orientation of a vector in 3D-space (spherical coordinates). For the generation of frame 1 it is assumed that the focal length f is known so $\mathbf{K}$ is completely known. The focal length f' of the camera which generates frame 2 is unknown and must be estimated. With an estimated $\hat{f}'$, $\mathbf{K}'$ is completely determined so the F matrix is constructed by equation (17). The search space is 6 dimensional and the task of the optimization is to find the estimate $\hat{\mathbf{F}}(\hat{\varphi}, \hat{\vartheta}, \hat{\rho}, \hat{\zeta}, \hat{\eta}, \hat{f}')$ (of the true $\bar{\mathbf{F}}(\bar{\varphi}, \bar{\vartheta}, \bar{\rho}, \bar{\zeta}, \bar{\eta}, \bar{f}'))$ yielding a globally optimal cost. This problem can be solved by global optimization (GO). The GO technique proposed is a new approach called *Non Linear Repulsive Particle Swarm Optimization* (NLRPSO) which is a modified *Particle Swarm Optimization* (PSO) [5, 8] and belongs to the class of stochastic global optimizers.

The search method of the new algorithm is fundamentally different compared to the RANSAC algorithm. The RANSAC algorithm constructs solution candidates by randomly chosen samples whereas the GO-based algorithm constructs solution candidates by interactions within a population of different solution candidates and thus makes use of more information (e.g. by sharing 'experiences' within the population). This is one of the reasons for this new approach outperforming RANSAC at high outlier rates. However, the NLRPSO does not fine tune the estimated parameters since it is not necessary to apply a global optimizer when the region of the global optimum is known. Instead, it allows for finding that region by utilizing a greater τ. The fine tuning is done by a subsequent gradient search method which does not take considerable effort.

3.5. Non Linear Repulsive Particle Swarm Optimization

The estimation of the F-matrix utilizing the cost function defined in (47) equals a search for the global optimum in a 6 dimensional search space. It is obvious that there are many local optima, depending on the outlier rate. The approach proposed in [12] introduced the RPSO algorithm which produced good results compared to RANSAC at high outlier rates.

RPSO utilizes a population Ψ of solution candidates, called *particles*. A Particle has a position $\mathbf{p}$ and a velocity $\mathbf{v}$ in the search space. In this case the position $\mathbf{p}^{(r)}$ of a particle at iteration step r is determined by 6 parameters:

$$\mathbf{p}^{(r)} := (\varphi_p^{(r)}, \vartheta_p^{(r)}, \rho_p^{(r)}, \zeta_p^{(r)}, \eta_p^{(r)}, f_p'^{(r)})^\top$$

The velocity $\mathbf{v}^{(r)}$ of a particle determines the next position $\mathbf{p}^{(r+1)}$ after an iteration step:

$$\begin{aligned} \mathbf{v}^{(r)} &:= (\Delta\varphi^{(r)}, \Delta\vartheta^{(r)}, \Delta\rho^{(r)}, \Delta\zeta^{(r)}, \Delta\eta^{(r)}, \Delta f'^{(r)})^\top \\ \mathbf{p}^{(r+1)} &:= \mathbf{p}^{(r)} + \mathbf{v}^{(r)} \end{aligned} \tag{53}$$

Each particle knows its best position $\hat{\mathbf{p}}$ it has achieved so far measured by the cost function. The equation for the velocity $\mathbf{v}^{(r+1)}$ of a particle in RPSO for the next iteration is:

$$\begin{aligned}\mathbf{v}^{(r+1)} = \quad & \omega\, \mathbf{v}^{(r)} \\ + \quad & a_{\mathrm{RPSO}}\, \chi_1^{(r)}\, (-\mathbf{p}^{(r)} + \hat{\mathbf{p}}^{(r)}) \\ + \quad & b_{\mathrm{RPSO}}\, \chi_2^{(r)}\, \omega(-\mathbf{p}^{(r)} + \hat{\mathbf{y}}^{(r)}) \\ + \quad & c_{\mathrm{RPSO}}\, \chi_3^{(r)}\, \omega\, \mathbf{z}^{(r)} \qquad (54)\end{aligned}$$

with

- $\chi_1^{(r)}$, $\chi_2^{(r)}$, $\chi_3^{(r)}$: random numbers $\in [0,\ 1]$
- ω: inertia weight $\in [0.01,\ 0.7]$
- $\hat{\mathbf{p}}^{(r)}$: best position of a particle
- $\hat{\mathbf{y}}^{(r)}$: best position of a randomly chosen other particle from Ψ
- $\mathbf{z}^{(r)}$: a random velocity vector
- $a_{\mathrm{RPSO}}, b_{\mathrm{RPSO}}, c_{\mathrm{RPSO}}$: constants.

The term with the a_{RPSO}-scalar on the right side of (54) leads to a motion of the particle towards its best position and so represents the local optimization method of the particle. The term with the b_{RPSO}-scalar leads to a repulsion between the particle and the best position of a randomly chosen other particle in order to explore new areas in the search space and to prevent the population to get stuck in a local optimum. This represents the global search method. The term with the c_{RPSO}-scalar generates noise in the velocity of a particle to enhance the exploration of new areas in the search space. The inertia weight ω decreases in steps of 0.05 from 0.7 to 0.01 when no progress at all is encountered for Δr_ω iteration steps.
In the proposed Non Linear Repulsive Particle Swarm Optimization (NLRPSO) algorithm, the components of the velocity $\mathbf{v}$ and the components of the position $\mathbf{p}$ are constrained to be within $[0, 1]$. The rotation parameters are mapped by e.g.

$$\varphi_p \in [0,1] \rightarrow \varphi = -0.2 + 0.4\varphi_p \qquad (55)$$

and the focal length is mapped by:

$$f'_p \in [0,1] \rightarrow f' = (0.9 + 0.2 f'_p) f. \qquad (56)$$

In case of a particle position component bound infringement the new value is calculated by a ’reflection’: e.g.

$$p_1 = 1.2 \notin [0,1] \Rightarrow p_1 \rightarrow 1 - (1.2 - 1) = 0.8 \qquad (57)$$

The translation parameters are mapped by e.g.

$$\zeta_p \in [0,1] \rightarrow \zeta = 0 + \pi\zeta_p. \qquad (58)$$

As an exception, the translation components of the particle position are not constrained.
In case of a velocity component bound infringement the new value is calculated by a ’cut off’: e.g.

$$v_1 = 1.2 \notin [0,1] \Rightarrow v_1 \rightarrow 1. \qquad (59)$$

One disadvantage of the RPSO algorithm is the fact that the local optimization ability of a particle does not include any interaction with other particles but only relies on information about itself. This increases the robustness of the convergence but also slows down its speed. NLRPSO is an approach where both the local optimization method and the global search method of a particle are realized by swarm interactions. This enables a better utilization of available information and so a faster convergence. The main difference of NLRPSO compared to RPSO is the particle propagation operator (PPO). The PPO assigns a new velocity (and so a new position) to a particle by processing available information such as the positions and velocities of all particles. Before we represent the proposed PPO for NLRPSO, we define the following attractive part:

$$V_{\mathrm{attr}}^{(r)} = a \sum_i^m \chi_i^{(r)}\, (-\mathbf{p}^{(r)} + \hat{\mathbf{p}}_i^{(r)}) \qquad (60)$$

and the repulsive part:

$$V_{\mathrm{rep}}^{(r)} = b\, \chi_R^{(r)} \sum_i^S \frac{-\mathbf{p}^{(r)} + \hat{\mathbf{p}}_i^{(r)}}{|-\mathbf{p}^{(r)} + \hat{\mathbf{p}}_i^{(r)}|} \exp\left[-\frac{d(\hat{\mathbf{p}}_i^{(r)}, \hat{\mathbf{p}}^{(r)})^2}{2c\overline{D}^2} \right]. \qquad (61)$$

The proposed PPO is:

$$\mathbf{v}^{(r+1)} = V_{\mathrm{attr}}^{(r)} + V_{\mathrm{rep}}^{(r)} \qquad (62)$$

with

- $\chi_i^{(r)}$, $\chi_R^{(r)}$: random numbers $\in [0,\ 1]$
- m: number of randomly chosen attractive particles
- S: swarm size
- $\overline{D}$: mean distance of current particle to all other particles
- $d(\mathbf{p}_i^{(r)}, \mathbf{p}^{(r)})$: distance between current particle and particle number i
- a, b, c: scalar constants.

In contrast to RPSO, the velocity does not depend on its previous vector. This means that there is no inertia demanded in the movement of the particles so the movement pattern of the swarm is more like a gas. Another difference is that there is no local optimization part as in the case of RPSO. Instead, there is an attraction towards the best positions of other particles by the term (60). This term produces a vector which is an element of a space spanned by

$$(-\mathbf{p}^{(r)} + \hat{\mathbf{p}}_i^{(r)}),\ i = 1, ..., m \tag{63}$$

and leads to an attraction of the particle towards m randomly chosen particles $\hat{\mathbf{p}}_i^{(r)}$. This has the effect that the attractive vector is less constrained then in the case of RPSO. The parameter a determines the size of the spanned space. A small value for a results in faster convergence but at the same time the ability to explore new areas suffers and so the robustness.

The repulsion strategy is also different. The term (61) involves all particles, in contrast to RPSO where there is only a repulsion between 2 particles at a time. If the best position of the current particle gets closer to the best position of another particle the repulsion gets stronger nonlinearly. This property leads to a prevention of premature convergence and so helps avoiding local optima. On the other hand, it also enhances exploring new locations in the search space and speeds up the convergence. The parameter b adjusts the repulsion strength. For a too big value of b, the repulsion gets too strong and this may lead to a state where each particle is surrounded by a barrier which it cannot overcome. NLRPSO is quite sensible on both a and b. These parameters have to be balanced to achieve high performance or even a global convergence.

The parameter c scales the mean distance range where the repulsion between particles is effective and so is coupled with the parameter b in its effect.

Though the cumulation of particles in a suboptimal region is made less probable by the PPO of NLRPSO, in more complex search spaces it can occur that the best position of a particle, once arrived in a suboptimal region, does not change and so gets stuck in that region. If this happens to more than one particle in the same region then we speak of a *collision*. In NLRPSO, a collision detection system recognizes collisions. If the distance of the *best* position of the current particle to the best position of any other particle is less than a specified threshold τ_{col} the current particle's best position is displaced by the vector

$$\frac{1}{b} c_{\mathrm{col}} V_{\mathrm{rep}} \tag{64}$$

where c_{col} is a constant.

The information utilization ability of NLRPSO is considered higher than that of RPSO due to the differences in the attractive and repulsive parts, respectively.

4. EXPERIMENTAL RESULTS

This approach is tested on synthetic data and on a real image sequence. Some of the parameters of the algorithm are fixed while others are adaptively changed in dependence of the outlier rate β. The required number of iterations for the three algorithms is compared where an iteration is defined by processing all correspondences. A swarm-iteration is defined by processing all particles so a swarm with S particles has to process S iterations per swarm-iteration.

The NLRPSO algorithm is used to find the region of the global optimum. After a minimum number of swarm-iterations $r_{\min}$, the solution of the NLRPSO algorithm is used as a start position for a levenberg-marquardt gradient search algorithm for fine-tuning. If the fine-tuning does not succeed, the NLRPSO algorithm is continued until the gradient search algorithm succeeds.

4.1. Synthetic Data Tests

A virtual camera with rotation angles restricted to ± 0.2 rad and a maximum focal length variation of 10% between two consecutive frames is used.

At first a 3D point cloud is randomly constructed. To simulate relative-motion-outliers, moving objects with varying number of supporting points are randomly generated. With a focal length of 1 the virtual camera determines the first projection of the 3D points into the camera plane to generate frame 1. After translating, rotating and changing the focal length of the virtual camera frame 2 is generated. The corresponding feature points are determined. To simulate additional random-outliers, mismatches are generated randomly. A Gaussian noise is added to the feature point coordinates to simulate real world conditions. With $N = 400$ and varying outlier rates the required number of iterations for a 99% successful search is experimentally determined.

The search area is defined by $R = 0.5u$ and the coordinates of the feature points lie within $[-0.5, 0.5]u$ where u is any length unit. The true variance $\bar{\sigma}^2$ of the Gaussian noise is $\bar{\sigma}^2 = 10^{-7}u^2$. Assuming that a real camera has 720x576 pixel resolution this is equivalent to a Gaussian error with a standard deviation of $\sigma \sim 0.18$ pel.

The fixed parameters are shown in tab. 1.

c	κ_{cf}	m
2	$40 \cdot 10^{-6}$	6

Table 1: *Fixed parameters.*

This setup results in

$$\kappa \sim 10^{-7}. \tag{65}$$

As stated in sec. 3.3, empirical studies show that the factor κ_c specified in the cost function may be greater than κ

without loosing significant accuracy of the estimated parameters. Tab. 2 shows the parameters of the NLRPSO algorithm depending on the outlier rate. Each time the NLRPSO algorithm finds a new globally best position, the appropriate inlier rate is calculated and the other parameters are adapted accordingly.
These parameters are determined empirically, since there is no analytical derivation available yet.

β	τ_{col}	a	b	c_{col}	$\tau[u]$	S	$r_{\min}$
0.5	0.0	0.63	-0.039	0.0	$7 \cdot 10^{-3}$	15	120
0.6	0.0	0.64	-0.039	0.0	$6 \cdot 10^{-3}$	20	100
0.7	0.02	0.64	-0.04	-0.02	$4 \cdot 10^{-3}$	30	110
0.8	0.05	0.65	-0.04	-0.075	$3 \cdot 10^{-3}$	30	200
0.9	0.1	0.66	-0.043	-0.1	$3 \cdot 10^{-3}$	30	–

Table 2: *Adaptive parameters of the NLRPSO algorithm.*

4.2. Real Data Tests

The proposed approach was also tested on an outdoor sequence. In this sequence, a sportsman is moving in the foreground while the camera keeps focus on him. Some of the parts of the background are moving as well so many outliers are produced. Fig. 7 shows the detected inlier rates.

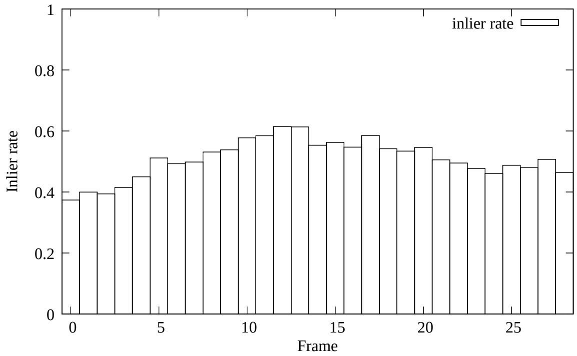

Figure 7: *Inlier rates in tested outdoor sequence.*

The sequence was tracked and a virtual blue cube was placed into the scene as shown in Fig. 8.

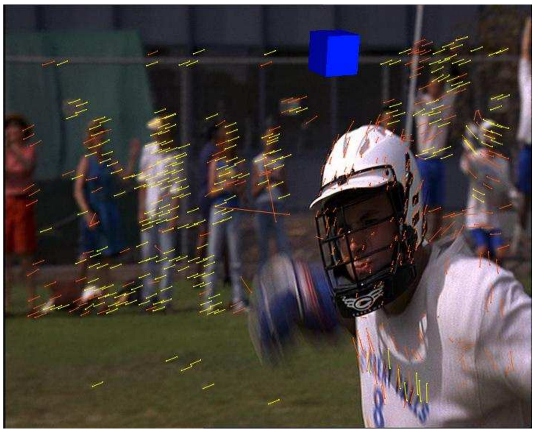

Figure 8: *Snapshot from the outdoor image sequence.*

4.3. Results

Fig. 9 shows the results of the synthetic data tests where the required number of iterations is plotted.
For outlier rates of 53% and higher, the proposed approach requires less iterations compared to the RANSAC based approach.

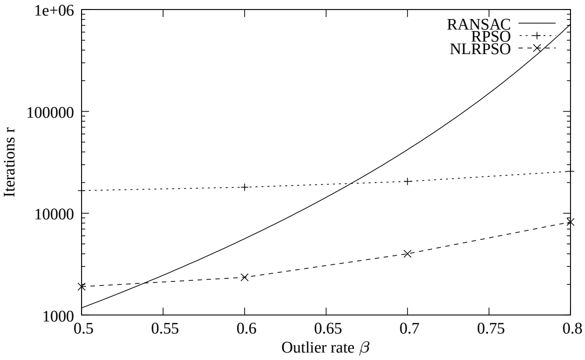

Figure 9: *Mean iterations required for 99% successfull search: RANSAC vs. RPSO vs. NLRPSO*

5. CONCLUSIONS

The introduced NLRPSO algorithm for robust estimation shows promising results. In robust estimation of epipolar geometry and focal length, it outperforms the well known RANSAC based approach at high outlier rates. Compared to the RPSO algorithm, NLRPSO shows considerable improvements in global search ability.

Though the proposed technique requires more iterations than the RANSAC algorithm at outlier rates below 53%, the upper bound for the required computational effort of the estimation of camera rotation, translation and focal length is significantly decreased. The real time estimation at high outlier rates can be facilitated. Another advantage of the proposed estimator is that it can be configured to estimate any subset of parameters, e.g. only rotation and one direction translation, so it is possible to benefit from any constraints in the camera parameters.

6. REFERENCES

[1] O. Chum and J. Matas. Randomized ransac with $t_{d,d}$ test. In *Proc. British Machine Vision Conference*, volume 2, pages 448–457, Cardiff, Sept. 2002.

[2] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Deutsche Arbeitsgemeinschaft fuer Mustererkennung Symposium*, pages 236–243, Sept. 2003.

[3] R. M. A. Fischler and C. Bolles. Random sample consensus: A paradigm for model fitting with application 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*. Cambridge University Press, 2000.

[5] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[6] D. Myatt, P. Torr, S. Nasuto, J. Bishop, and R. Craddock. Napsac: High noise, high dimensional robust estimation - it's in the bag. In *Proc. British Machine Vision Conference*, pages 458–467, 2002.

[7] D. Nister. Preemptive ransac for live structure from motion estimation. In *IEEE International Conference on Computer Vision*, pages 199–206, Oct. 2003.

[8] J. Shi and R. C. Eberhart. Parameter selection in particle swarm optimization. In *Proc. Evolutionary Programming Conference*, pages 591–600, 1998.

[9] C. Tomasi and T. Kanade. Detection and tracking of point features, Apr. 1991. Carnegie Mellon University Technical Report CMU-CS-91-132.

[10] B. Tordoff and D. Murray. Guided sampling and consensus for motion estimation. In *European Conference on Computer Vision*, pages 82–96, May 2002.

[11] P. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[12] O. Urfalıoḡlu. Robust estimation of camera rotation, translation and focal length at high outlier rates. In *Proc. International Canadian Conference on Computer and Robot Vision*, pages 464–471, May 2004.