



Multi-view point cloud fusion for LiDAR based cooperative environment detection

B. Jaehn, P. Lindner, and G. Wanielik

Professorship of Communications Engineering, Chemnitz University of Technology, Chemnitz, Germany

Correspondence to: B. Jaehn (bjae@hrz.tu-chemnitz.de)

Received: 15 December 2014 – Revised: 2 February 2015 – Accepted: 4 March 2015 – Published: 3 November 2015

Abstract. A key component for automated driving is 360° environment detection. The recognition capabilities of modern sensors are always limited to their direct field of view. In urban areas a lot of objects occlude important areas of interest. The information captured by another sensor from another perspective could solve such occluded situations. Furthermore, the capabilities to detect and classify various objects in the surrounding can be improved by taking multiple views into account.

In order to combine the data of two sensors into one coordinate system, a rigid transformation matrix has to be derived. The accuracy of modern e.g. satellite based relative pose estimation systems is not sufficient to guarantee a suitable alignment. Therefore, a registration based approach is used in this work which aligns the captured environment data of two sensors from different positions. Thus their relative pose estimation obtained by traditional methods is improved and the data can be fused.

To support this we present an approach which utilizes the uncertainty information of modern tracking systems to determine the possible field of view of the other sensor. Furthermore, it is estimated which parts of the captured data is directly visible to both, taking occlusion and shadowing effects into account. Afterwards a registration method, based on the iterative closest point (ICP) algorithm, is applied to that data in order to get an accurate alignment.

The contribution of the presented approach to the achievable accuracy is shown with the help of ground truth data from a LiDAR simulation within a 3-D crossroad model. Results show that a two dimensional position and heading estimation is sufficient to initialize a successful 3-D registration process. Furthermore it is shown which initial spatial alignment is necessary to obtain suitable registration results.

1 Introduction

Humans were able to use indirect signals over mirrors or through the windows of other vehicles to observe relevant areas e.g. at difficult crossroad situations. To achieve a similar understanding of the surrounding with depth sensors, one important aspect for future autonomous systems will be the cooperative exchange of environment information denoted as car to car (C2C) communication. Together with the use of real time cloud or local infrastructure based services this will improve the recognition and reaction capabilities towards objects located outside the direct field-of-view (FoV) of the own sensors.

E.g. the green car of Fig. 1 could provide useful information about the blue truck for the red car. The foundation of such a dense data fusion is the availability of an highly accurate relative pose estimation which can be transferred into a homogeneous transformation matrix. This is hard to derive by traditional satellite based relative localization methods but their estimate could be improved using the gathered environment data of both cars. A method for achieving this is the content of this paper which is mainly based on the master thesis by Jähn (2014). For the rest of this paper we will refer to the red car as target and to the green car as source and we will use the coordinate definitions of the right picture from Fig. 1.

To support the understanding of the reader the present paper is organized as follows: Sect. 2 provides relevant background information's which are necessary to ensure comprehensibility. Afterwards the general outline and the novel ideas of the presented method are shown in Sect. 3. An evaluation of the presented approach is then given in Sect. 4. The conclusion in Sect. 5 contains a summary of all findings, it's limitations as well as improvements and future work.

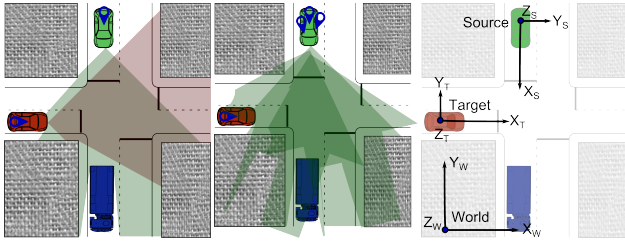


Figure 1. The figure shows two cars which could exchange their environment information if an accurate enough pose estimation would be available (left picture). If the pose estimation is inaccurate this would yield an alignment error (center picture). The right picture shows the coordinate system definitions which were used in the paper.

2 Background

2.1 Depth sensors

Typical depth sensors, like LiDARs and stereo cameras, measure the surface of the surrounding environment. Using the projective geometry their range images can be converted into a point cloud which represents the surface shape of the surrounding. During this conversion the pixel wise neighborhood information is maintained which is useful for further processing steps like normal feature estimation. In the case that two of these sensors measure the same surface area from different perspectives, this data can be used to align them against each other if their relative pose estimation is inaccurate. The process of doing this is called registration or shape matching which is part of Sect. 2.3.

2.2 Normal feature estimation

To gain information about the underlying surface, which the point cloud data represents, it is very common in 3-D data processing to determine normal features¹. We apply here the standard weighted normal averaging scheme which uses triangle combinations incorporating the neighbourhood points (Klasing et al., 2009). In this work we use a weighting factor w_j which is chosen as the reciprocal product of the distance between the neighbours $\mathbf{q}_{i,j}$ of \mathbf{p}_i according to Eq. (1). We refer to this method as distance weighted cross product (DWCP).

$$w_j = \frac{1}{|\mathbf{q}_{i,j} - \mathbf{p}_i| \cdot |\mathbf{q}_{i,j+1} - \mathbf{p}_i|} \quad (1)$$

2.3 Registration of point clouds

Registration, or shape matching, is the process to align two or more surfaces against each other in such a way that a

¹Vectors assigned to each point in space representing a local plane patch of the approximated surface

mathematical metric is minimized. Although other registration methods exist (compare Pottmann et al., 2002) the most popular one is the so called ICP algorithm introduced by Besl and McKay (1992).

The general idea is to find at first point correspondences between two slightly misaligned data sets. This is done usually by searching the currently closest point in the other data set. Secondly a locally optimal rigid transformation matrix is calculated which minimizes a certain distance metric in order to align them against each other. Using the result from the previous iteration this procedure is repeated. New correspondences were determined and the alignment is optimized again until some stopping criteria is fulfilled.

The original formulation of the algorithm by Besl and McKay (1992) delivers bad results if the incorporated data sets do not overlap completely or if there are holes due to occlusion effects. As a consequence a still growing number of different variations has been proposed. The main goals are to speed up the convergence rate and improve the robustness against outliers. Therefore, several heuristics and modifications are applied and additional features, like normals, are incorporated. A comparison study of several methods can be found at Pomerleau et al. (2013).

In this work we utilize a point to plane (Chen and Medioni, 1991) based ICP method which utilizes additional rejection methods (Rusinkiewicz and Levoy, 2001). First all correspondences where the angle between the corresponding normals exceeds 30° are ignored and secondly we remove the worst 10% due to the Euclidean distance (Pulli, 1999).

3 Method

The primary goal of this work is to include the captured point cloud data of a sensor platform, like the green car (compare Fig. 1), into the coordinate system of another, e.g. the red car. Each of them has its own coordinate system indicated with the superscript prefix S and T respectively. Thus, an arbitrary point ${}^S\mathbf{p} = [x_S, y_S, z_S]^T$ in the source coordinate system can be transformed in the target coordinate system, using \mathbf{T}^{TS} . It follows that the corresponding point ${}^T\mathbf{p} = [x_T, y_T, z_T]^T$ from ${}^S\mathbf{p}$ can be calculated with the following equation.

$$\begin{bmatrix} {}^T\mathbf{p} \\ 1 \end{bmatrix} = \mathbf{T}^{TS} \cdot \begin{bmatrix} {}^S\mathbf{p} \\ 1 \end{bmatrix} \quad (2)$$

Unfortunately, the exact transformation matrix \mathbf{T}^{TS} is in general not known and has to be estimated by $\hat{\mathbf{T}}^{TS}$. In order to achieve an initial guess of this matrix, it is further assumed that the target platform estimates the relative position and orientation of the source with a suitable, e.g. satellite based, relative localization system which delivers a state

²E.g. points which do not have a corresponding partner representing the same physical object surface within the other data set.



Figure 2. Overview of the processing steps within the presented approach.

$\mathbf{x} = [x, y, z, \phi, \theta, \psi]^T$ and its uncertainty, as covariance matrix Σ_{Cov} . The accuracy of this transformation can be improved, using a registration algorithm with environment information captured by both sensor platforms simultaneously.

To achieve this the environment data, which was captured by both sensor platforms simultaneously, is determined. Therefore the state and the corresponding uncertainty information, namely the standard deviation values $\text{Diag}(\Sigma_{Cov}) = [\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2]$, were used to estimate the theoretical field of view (FoV) of the source sensor within the targets coordinate system in order to determine the data within the overlapping FoV. Further on occlusion effects were resolved to achieve the relevant points for the registration process within the target point cloud (denoted in the following as \mathbf{T}) and the source point cloud (\mathbf{S}).

Once an accurate transformation is determined it can be applied to the whole source point cloud. Both data sets are now fused within the target coordinate systems and available for further processing. An overview of the previously explained steps is given in Fig. 2 and some steps are explained in detail in the following.

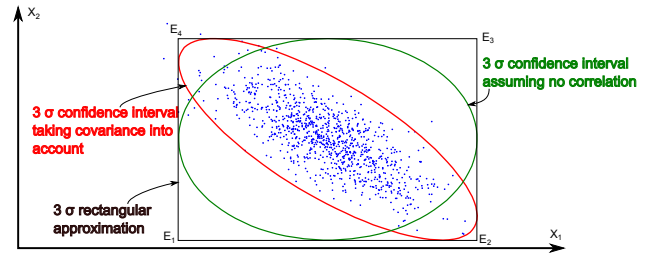


Figure 3. Within this chart three different 3σ ($\gamma = 3$) confidence intervals of the same two dimensional correlated normal distributed sample set are shown. The red one uses the full covariance matrix whereas the green one just neglects the covariance values. The rectangular confidence interval includes both ellipses and thus acts as an upper bound.

3.1 Overlapping FoV estimation

The FoV is the geometrical area, which is theoretically visible for a sensor, without taking occlusion effects into account. For the most typical range sensors, like automotive LiDAR, this can be approximated as a pyramid. All sensor data have to lie within this field, thus it can be used as a rough geometrical boundary. Such a pyramid can be defined for the source and target sensor. In order to determine the overlapping part it is necessary to transform them into the same coordinate system.

3.1.1 Uncertainty field of view

Problematic in this context is the accuracy of $\hat{\mathbf{T}}^{TS}$. The exact origin and orientation of the source FoV is thus not known. To overcome this problem the track uncertainty is used to increase the source FoV such that it includes the error space, which is considered as normal distributed, up to a certain confidence interval. How this can be achieved efficiently is dependent on its original shape. In this work a pyramid model is used and therefore the basic idea to derive a proper uncertainty field approximation is described in the following. The full derivation can be found at Jähn (2014).

As mentioned already the uncertainty space is assumed to be normal distributed with an expectation value of $\mathbf{0}$. Consequently a surrounding confidence interval $\gamma \cdot \sigma$ would result in a six-dimensional elliptical error space. If this is used to transform the pyramid model into the uncertainty space this would result in an extremely complex mathematical shape. Therefore just the diagonal values of Σ_{Cov} are used to approximate this with a rectangular upper boundary space. With this simplification it doesn't matter if one would take cross correlation values into account or not. To support the understanding of the reader a two-dimensional example is shown in Fig. 3.

Using this rectification, the confidence interval can be described just with the maximum divergence from the state space in positive and negative direction for each dimension.

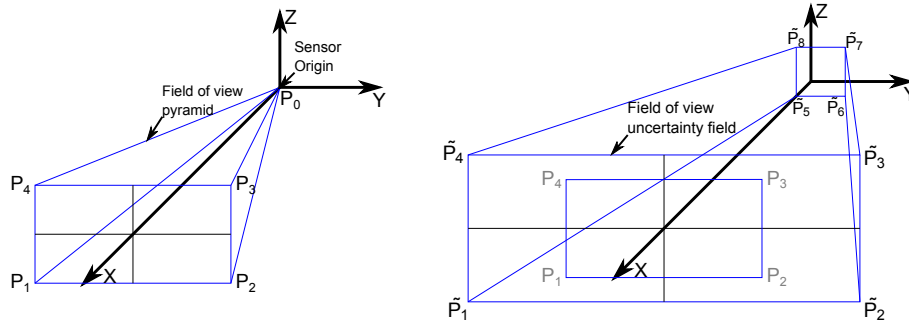


Figure 4. The two pictures show the coordinate system definitions and the nomenclature of the original field of view (left) and the increased uncertainty field of view (right).

Consequently there are $2^6 = 64$ different maximum error variations. If all of these error variations are applied to the FoV pyramid (compare Fig. 4, left panel) the resulting shape can be approximated by a frustum pyramid as it is shown in Fig. 4 (right panel).

This approximation of the uncertainty FoV can be used to clip the points of the target sensor which could have no corresponding partner within the source data set.

3.1.2 Point uncertainty ranges

In this section it is estimated which parts of both point clouds could be visible to both sensors taking occlusion effects into account. E.g. if an object is located within both fields of view but it is occluded through another object from just one perspective. To figure this out the uncertainty of the initial transformation matrix $\hat{\mathbf{T}}^{TS}$ have to be taken into account again. Therefore the same assumptions, namely the rectification of the uncertainty space, from the previous section are applied. This time the uncertainty space is determined for each point of the source point cloud separately and the resulting space is approximated by a sphere around the point ${}^S p_i \in \mathbf{S}$ with a radius r_i according to Eq. (3).

$$r_i = \max \{ r_{\min}, \max \{ r_{i,j} \forall j \} + \| \mathbf{t}_{\text{err}} \|_{\infty} \} \quad (3)$$

$$j \in \{ 1, 2, \dots, |\Sigma_{\text{Rot}}| \} \quad \mathbf{t}_{\text{err}} \in \Sigma_{\text{Transl}} \quad (4)$$

$$r_{i,j} = | \mathbf{p}_{S,i} - \text{Rot} \{ \mathbf{r}_j \} \cdot \mathbf{p}_{S,i} | \quad \mathbf{r}_j \in \Sigma_{\text{Rot}} \quad (5)$$

$$\Sigma_{\text{Transl}} = \{ \pm \sigma_x \} \times \{ \pm \sigma_y \} \times \{ \pm \sigma_z \} \quad (6)$$

$$\Sigma_{\text{Rot}} = \{ \pm \sigma_\phi \} \times \{ \pm \sigma_\theta \} \times \{ \pm \sigma_\psi \} \quad (7)$$

Transferred into the target coordinate system using $\hat{\mathbf{T}}^{TS}$ these spheres represents the area where $\begin{bmatrix} \mathbf{p}_{T,i} \\ 1 \end{bmatrix} = \mathbf{T}^{TS} \cdot \begin{bmatrix} \mathbf{p}_{S,i} \\ 1 \end{bmatrix}$ could be if \mathbf{T}^{TS} would be known exactly. Further it represents the area where corresponding points of \mathbf{T} could be which were captured from the same object. Thus each source point, which has no target point within its range defined by r_i , is clipped. The other way around each target point is clipped

if it is not within at least one of the source uncertainty spheres (compare Fig. 5).

4 Evaluation

The objective of the following evaluation is to show the applicability of the presented approach for cooperative environment recognition applications. Compared to the master thesis (Jähn, 2014), where this paper is based on, the data which was used for the evaluation was re-evaluated to fit the limited scope of this paper. For a more exhaustive evaluation please refer to Jähn (2014).

4.1 Evaluation strategy

To show the effects of different steps of the presented approach a detailed cross road 3-D model was built with SketchUp^{®3}. This was used to simulate range data by two virtual LiDAR depth sensors, which were moved through the static model such that they overlap partially and enough non-parallel surface details for registration were included. The recorded data includes ground truth range data as well as position and orientation of both sensors. This was then superimposed by range and pose (position and orientation) noise. These point clouds were afterwards aligned and the output was compared with the applied pose error vector. This procedure was then repeated 21 times with different data set combinations each with 100 different range and tracking noise samples. Thus the quality of the registration process can be statistically appraised.

4.1.1 Range data generation

The range data was captured using a simple ray tracing approach. Each laser beam of the virtual LiDAR has been modelled as a single ray. In contrast to the simulation applied by Gabriel (2010) the complete sensor optic simulation and the 3 dimensional beam structure was not taken into account.

³A 3-D modeling software by Trimble Navigation

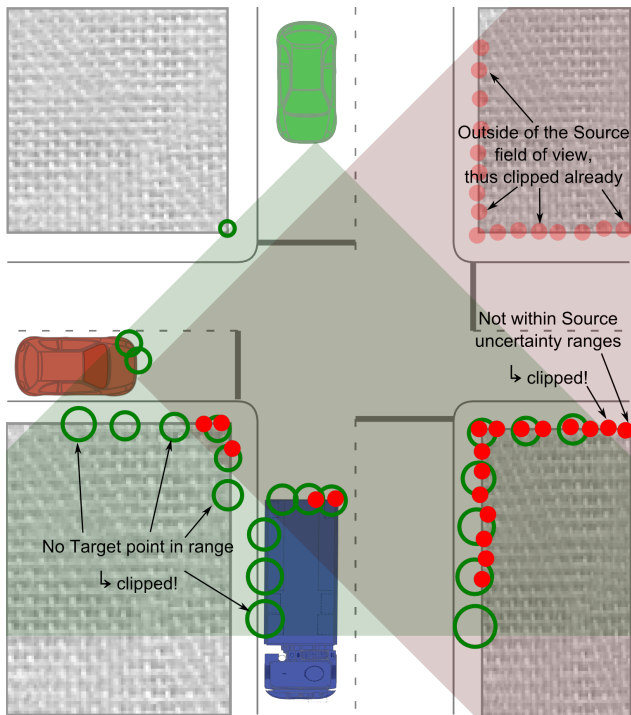


Figure 5. To each source point an uncertainty range is assigned. Within each sphere at least one target point has to be present and each target point have to lie within at least on sphere.

This approach allows a much faster simulation but it does not reflect the special properties of LiDAR sensors, e.g. just the first echo is used. Anyhow this simple sensor model can be applied also to other range sensors, like stereo and time of flight cameras, hence it is used here. For the evaluation in this work we used a LiDAR model with 40 layers each with 451 beams covering 72° horizontally and 10° vertically.

The range image taken directly from the simulation is noiseless and therefore not very realistic. That’s why it is superimposed by Gaussian noise. According to the manual (Ibeo Automotive Systems GmbH, 2008) of the Ibeo LUX laser scanner family, a typical standard deviation value σ_r is 10 cm, which is used here.

To simulate tracking noise n random error sample from a normal distributed error space is superimposed to \mathbf{x} according to Eq. (8). The uncertainty standard deviation values $\sigma_x, \sigma_y, \sigma_z, \sigma_\phi, \sigma_\theta, \sigma_\psi$ were taken from Table 1 to simulate different tracking conditions and quality. As described in Sect. 3.1 this will influence how the overlapping area of the sensor data is estimated.

$$\mathbf{x} = \mathbf{x} + \mathbf{x}_{\text{Err}} \quad \mathbf{x}_{\text{Err}} \in \mathcal{N}\left(0, [\sigma_x, \sigma_y, \sigma_z, \sigma_\phi, \sigma_\theta, \sigma_\psi]^T\right) \quad (8)$$

Table 1. Tracking noise setups.

	$\sigma_\phi, \sigma_\theta, \sigma_\psi$	$\sigma_x, \sigma_y, \sigma_z$
TN1	1°	30 cm
TN2	2°	60 cm
TN3	3°	120 cm
TN4	4°	240 cm

Table 2. Percentiles from $\mathcal{N}(0, \sigma)$ intervals for direct comparison with non-normal distributed error spaces.

$\mathcal{N}(0, \sigma)$	Percentile	Name
$ x < 0.5\sigma$	$\hat{=}$ 38.29 %	p_{38}
$ x < 1.0\sigma$	$\hat{=}$ 68.27 %	p_{68}
$ x < 1.5\sigma$	$\hat{=}$ 86.64 %	p_{86}
$ x < 2.0\sigma$	$\hat{=}$ 95.45 %	p_{95}
$ x < 2.5\sigma$	$\hat{=}$ 98.76 %	p_{98}

4.1.2 Analysis of the results

Within each test cycle the resulting alignment vector \mathbf{x}_{Alg} is subtracted from the previously applied tracking noise vector \mathbf{x}_{Err} which results in the remaining error vector \mathbf{x}_{Rem} (compare Eq. 9). Over all, in total 2100, test cycles the expectation value of this is assumed to be $\mathbf{0}$. To allow an easy comparison between the input error space, which is normal distributed, and the output error space, which is in general not normal distributed, the equivalent percentiles are used according to Table 2. These percentiles are calculated for each component of \mathbf{x}_{Rem} . If the resulting value of the percentile is smaller than the equivalent interval from the input error space normal distribution the alignment was improved during the registration.

$$\mathbf{x}_{\text{Err}} - \mathbf{x}_{\text{Alg}} = \mathbf{x}_{\text{Rem}} \quad (9)$$

4.2 Contribution of clipping & confidence interval

To figure out the contribution of the FoV and uncertainty range clipping from Sects. 3.1 and 3.1.2 the influence of the applied confidence interval will be investigated now. Therefore the confidence interval parameter γ is modified and an additional test was done where both clipping methods were deactivated.

According to Table 3 the smallest remaining error is for $\gamma = 1.75$. With $\gamma = 1.75$ and without clipping at all the error increases. This effect decreases with higher tracking noise due to the huge uncertainty ranges where occlusion and shadowing effects cannot be detected. Obviously the rejection step works very efficiently such that bad correspondences are detected reliably. Consequently this pre-processing does affect the alignment accuracy just slightly but noticeable. Thus we conclude that the FoV and uncertainty range clipping stabilize the registration process. Furthermore it has an influence on the number of points, which have to be aligned and

Table 3. The table shows the results of the remaining error percentiles for the point to plane metric with different clipping confidence intervals γ . The light green colour indicates nearly perfect alignment results whereas dark red stands for unsuitable results above 1° or 20 cm. The achieved accuracy and the necessary number of iterations is quite similar in all cases. Especially for higher tracking noise levels where the uncertainty ranges were huge.

	Point to Plane [DWCP] @ $\gamma = 1,75$							Point to Plane [DWCP] @ $\gamma = 2,5$							Point to Plane [DWCP] without Clipping							
	φ [°]	θ [°]	ψ [°]	x [cm]	y [cm]	z [cm]	Iter	φ [°]	θ [°]	ψ [°]	x [cm]	y [cm]	z [cm]	Iter	φ [°]	θ [°]	ψ [°]	x [cm]	y [cm]	z [cm]	Iter	
TN1	P ₃₈	0,03	0,02	0,04	2,34	1,41	0,82	4	0,02	0,03	0,05	2,87	1,79	0,81	5	0,05	0,04	0,09	4,47	3,03	1,14	5
	P ₆₈	0,06	0,08	0,10	5,35	3,49	1,51	5	0,05	0,10	0,12	7,07	4,47	1,50	6	0,09	0,13	0,23	10,66	7,57	2,15	6
	P ₈₆	0,10	0,12	0,21	11,04	6,53	2,57	6	0,10	0,15	0,28	14,52	9,15	2,44	7	0,19	0,20	0,41	21,27	13,48	3,53	7
	P ₉₅	0,13	0,15	0,32	26,89	10,15	3,41	7	0,15	0,17	0,52	25,44	16,16	3,42	9	0,47	0,27	0,70	37,67	20,86	6,88	9
TN2	P ₃₈	0,16	0,16	0,42	41,16	13,65	4,97	8	0,19	0,19	0,68	41,35	22,33	4,75	11	0,57	0,44	1,46	54,86	36,95	9,60	12
	P ₆₈	0,04	0,03	0,06	3,52	2,32	1,07	5	0,04	0,04	0,07	4,34	3,00	1,08	6	0,05	0,04	0,08	4,32	2,99	1,20	6
	P ₈₆	0,08	0,11	0,16	8,62	5,84	2,01	6	0,07	0,12	0,18	10,17	7,20	2,04	8	0,09	0,13	0,22	10,71	8,12	2,24	7
	P ₉₅	0,11	0,17	0,34	16,38	11,47	3,29	7	0,12	0,17	0,42	20,66	14,65	3,57	10	0,19	0,20	0,43	22,80	14,24	3,68	9
TN3	P ₃₈	0,21	0,21	0,59	34,52	18,33	5,27	9	0,22	0,22	0,91	42,03	28,96	5,74	13	0,46	0,27	0,87	40,93	23,80	7,41	11
	P ₆₈	0,25	0,22	0,87	47,67	26,11	8,57	11	0,28	0,24	2,75	80,49	112,58	8,37	17	0,56	0,43	2,64	83,89	109,92	9,42	15
	P ₈₆	0,05	0,04	0,10	4,86	3,62	1,37	6	0,05	0,04	0,09	4,37	3,16	1,27	6	0,05	0,05	0,10	4,90	3,38	1,31	7
	P ₉₅	0,10	0,14	0,24	12,28	9,37	2,61	6	0,09	0,13	0,23	11,79	8,04	2,43	8	0,10	0,14	0,26	13,38	9,49	2,48	9
TN4	P ₃₈	0,18	0,21	0,55	27,43	18,69	4,12	10	0,19	0,20	0,48	24,98	16,49	4,26	10	0,24	0,21	0,65	33,29	20,15	4,93	11
	P ₆₈	0,30	0,26	2,44	79,39	104,60	8,31	13	0,33	0,25	2,86	94,78	115,22	8,69	15	0,47	0,33	2,90	98,05	118,19	8,76	16
	P ₈₆	0,35	0,32	3,96	170,26	155,42	11,95	18	0,41	0,37	6,51	232,36	212,45	15,91	22	0,58	0,50	6,22	210,83	205,89	17,27	32
	P ₉₅	0,06	0,06	0,14	7,09	5,22	1,74	8	0,06	0,08	0,16	8,16	6,06	1,77	10	0,06	0,07	0,13	6,83	4,75	1,80	8
TN4	P ₃₈	0,15	0,17	0,43	21,82	15,41	3,79	11	0,18	0,18	1,12	52,51	40,05	5,59	17	0,15	0,17	0,51	32,00	18,80	4,51	12
	P ₆₈	0,41	0,29	3,65	171,35	151,58	11,82	17	0,43	0,38	6,80	258,97	282,38	21,37	23	0,46	0,35	5,68	229,45	266,33	18,78	21
	P ₈₆	0,60	0,61	9,18	419,63	393,30	36,35	32	0,87	0,80	11,36	486,18	463,53	49,81	49	0,80	0,71	10,58	484,31	445,62	42,42	42
	P ₉₅	1,37	1,32	17,78	883	612,24	72,87	49	1,65	1,50	17,02	932,33	776,66	79,49	49	1,42	1,25	17,08	816,36	789,18	81,89	49

thus the computational complexity is decreased drastically. This holds true especially if the overlap is small compared to the rest.

A smaller confidence interval reduces the range where potential correspondences could be found. This has positive effects if the error lies within this interval. In this case it does not happen that some of the potential good corresponding partners were clipped. However, if the current error is located outside of the confidence interval, points were clipped which could have a good corresponding partner in the other set. Thus the number of details, which can be used for the alignment, is reduced together with the overall possibility that ideal pairings can be found in general. This limits the alignment abilities drastically if the current initial error is high.

Thus we conclude that the uncertainty FoV and range clipping is especially useful for low tracking noise. There the number of points, which have to be aligned is reduced significantly and potential outliers were detected even before they could take any bad influence on the registration process. The confidence interval should not be chosen too small to avoid the clipping of potential useful point pairings. For higher tracking noise the uncertainty range clipping is less useful because occlusion effects can be detected just roughly. Therefore, outliers within the overlapping area were not thrown out. The FoV clipping on the other hand is always useful because the number of points, which have to be aligned, is decreased at low computational costs and potential wrong pairings were avoided.

5 Conclusions

The results show that an alignment of two data sets can be achieved with the help of a pose tracking system although they were captured from completely different perspectives. During the previous evaluation the contribution of certain system design aspects have been examined and the findings are summarized in the following sections.

The primary goal of the FoV clipping was to determine the overlapping area of the sensors FoV due to the pose estimation uncertainty. This reduces the number of points, which have to be aligned, significantly especially if the overlapping part is just a small subset of the complete data. Thus it further reduces the amount of possible outliers, which were not detected by the applied heuristics. That's why we conclude that the utilization of this method is always recommendable.

The idea of the uncertainty range clipping was to find the points, which could have a corresponding partner in the other data set, taking occlusion effects into account. This works fine if the tracking accuracy is already good because in this case the uncertainty ranges are small and a lot of potential outliers are ignored. Additionally the number of points, which have to be aligned is reduced further, which improves the overall performance. However, if the tracking noise is high, the contribution of the uncertainty range clipping is reduced drastically. That's why the uncertainty ranges are too big to detect occlusion effects, caused by small and medium sized objects like cars and trees, effectively. Nevertheless, occlusions caused by huge objects like buildings, were detected properly and thus points, which cannot have a corresponding

point in the other data set are ignored. However, it is heavily situation dependent, whether the benefits justify the computational costs of this method.

According to Table 3 a rotation and translation accuracy below 0.5° and below 15 cm is attainable in most cases. Under good circumstances with a lot of details this falls even under 0.1° and 10 cm. This should be enough for a lot of practical applications. Using the improved transformation result it is now possible to fuse the complete data of both sensors.

5.1 Limitations of the presented approach

The presented approach is suitable for scenarios where two depth sensors observe partly the same surface of an arbitrary scene from completely different viewpoints. Additionally their relative pose have to be roughly known, due to some also known confidence interval. However some additional conditions must be met. First of all, enough details should be included in the overlapping part, such that an unambiguous registration result is possible. Therefore, it must contain at least 3 non parallel surface patches. This implies further that the point density in this region is sufficient to estimate proper normal features and point correspondences. Additionally the extent of this region has to be significantly higher than the measurement noise of the sensor data and the tracking error.

5.2 Further improvements and future work

So far there is no method applied which checks if the above mentioned requirements were fulfilled. For practical applications this have to be checked for each sensor data pair separately before the presented approach can be applied. If that is successfully done the plausibility of the registration result should be checked. Currently there is no method applied which is able to determine the quality of the registration result. A completely wrong alignment can be detected easily by evaluating, if the alignment result lies within the confidence interval of the applied tracking mechanism. For a further assessment of the achieved alignment accuracy it is necessary to develop a suitable metric. This could be done based on the remaining average distance between the correspondences found in the last ICP iteration. Further on the similarity of the normal features between the point pairs could be a helpful measurement.

Edited by: M. Chandra

Reviewed by: two anonymous referees

References

- Besl, P. and McKay, N. D.: A method for registration of 3-D shapes, *Pattern Analysis and Machine Intelligence*, IEEE Transactions, 14, 239–256, doi:10.1109/34.121791, 1992.
- Chen, Y. and Medioni, G.: Object modeling by registration of multiple range images, in: *Robotics and Automation, 1991, Proceedings, 1991 IEEE International Conference*, Vol. 3, 2724–2729, doi:10.1109/ROBOT.1991.132043, 1991.
- Gabriel, M.: LiDAR-Signaturberechnung von räumlich ausgedehnten Zielen im Fahrzeugumfeld, Bachelor thesis, Chemnitz University of Technology, 2010.
- Ibeo Automotive Systems GmbH: Operating Manual ibeo LUX[®] Laser scanner, ibeo Automobile Sensor GmbH, Merkurring 20, 22143 Hamburg, 2.5 Edn., 2008.
- Jähn, B.: Fusion of multi-view point cloud data for cooperative object detection, Master thesis, Chemnitz University of Technology, 2014.
- Klasing, K., Althoff, D., Wollherr, D., and Buss, M.: Comparison of surface normal estimation methods for range sensing applications, in: *Robotics and Automation, 2009, ICRA '09, IEEE International Conference*, 3206–3211, doi:10.1109/ROBOT.2009.5152493, 2009.
- Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S.: Comparing ICP variants on real-world data sets, *Auton. Robots*, 34, 133–148, doi:10.1007/s10514-013-9327-2, 2013.
- Pottmann, H., Leopoldseder, S., and Hofer, M.: Registration without ICP, *Comput. Vis. Image Und.*, 95, 54–71, 2002.
- Pulli, K.: Multiview registration for large data sets, in: *3-D Digital Imaging and Modeling, 1999, Proceedings, Second International Conference*, 160–168, doi:10.1109/IM.1999.805346, 1999.
- Rusinkiewicz, S. and Levoy, M.: Efficient variants of the ICP algorithm, in: *3-D Digital Imaging and Modeling, 2001, Proceedings, Third International Conference*, 145–152, doi:10.1109/IM.2001.924423, 2001.