3D Registration for Verification of Humanoid Justin's Upper Body Kinematics

Nadia Figueroa, Haider Ali and Florian Schmidt
Institute of Robotics and Mechatronics.
DLR (German Aerospace Center)
Oberpfaffenhofen, Germany
nadia.figueroa@dlr.de, haider.ali@dlr.de,florian.schmidt@dlr.de

***Abstract*—Humanoid robots such as DLR's *Justin* are built with light-weight structures and flexible mechanical components. These generate positioning errors at the TCP (Tool-Center-Point) end-pose of the hand. The identification of these errors is essential for object manipulation and path planning. We proposed a verification routine to identify the bounds of the TCP end-pose errors by using the on-board stereo vision system. It involves estimating the pose of 3D point clouds of *Justin's* hand by using state-of-the-art 3D registration techniques. Partial models of the hand were generated by registering subsets of overlapping 3D point clouds. We proposed a method for the selection of overlapping point clouds of *self-occluding* objects (*Justin's* hand). It is based on a statistical analysis of the depth values. We applied an extended metaview registration method to the resulting subset of point clouds. The partial models were evaluated with detailed based surface consistency measures. The TCP end-pose errors estimated by using our method are consistent with *ground-truth* errors.**

I. Introduction

DLR's complex humanoid robot *Justin* is composed of two Light Weight Robot (LWR) arms [1]. The light-weight structures and mechanical flexibilities in the joints and links of these arms enable human-like mobility and compliant interaction. These mechanical flexibilities as well as bending effects in the light-weight structures produce rotational and translational errors at the TCP end-pose of the kinematic chains. Thus, the positioning accuracy of *Justin's* upper body kinematics can be verified by the identification of these TCP end-pose errors.

We propose a procedure for the identification of the bounds of *Justin's* TCP end-pose errors based on 3D registration techniques, using the on-board stereo vision system. These errors are identified by estimating the TCP end-pose applying a pair-wise registration between a 3D point cloud of the hand at a random pose to a pre-generated model with a fixed TCP origin.

A pair-wise registration aims at finding a rigid transformation between a pair of data views. In a typical scenario, it is implemented as a two step procedure, a coarse registration and a fine registration [10]. The coarse registration generates an initial guess of the motion between the two point clouds. This initial guess is obtained by matching correspondences. These correspondences are computed with different types of features (Point Signatures based on surface/color, Spin Images, Surface Models and Principal Curvatures) [10].

In this work four types of correspondence algorithms are evaluated: (i) David Lowe's Scale-Invariant Feature Transform (SIFT) [13], based on the color information, (ii) Fast Point Feature Histogram (FPFH) Descriptors[11], based on surface relations between a points neighborhood, (iii) the Signature of Histograms of Orientations (SHOT)[17] , also based on surface relations and (iv) a descriptor combining color and surface description introduced by Tomardi *et al*(CSHOT)[18]. These descriptors are used to find correspondences between point clouds and compute a coarse registration. A fine registration is used to converge to a more accurate solution. The Iterative Closest Point (ICP) algorithm [14][15] is used for fine-tuning rough alignments [11][10]. These methods have been implemented using the Open Source Point Cloud Library (PCL)[2].

The estimation of *Justin's* hand TCP end-pose relies on a model of the hand. 3D CAD models were generated when designing the hand. However, these do not accurately reflect the data observed by the actual sensing device. We generate a 3D model of the hand by registering multiple point clouds of different views. An offline multiple view registration method has been introduced by Pulli [7]. The method uses pair-wise registrations to obtain a global registration. Chen and Medioni [16] proposed a *metaview* approach to register and merge views incrementally. A simple pair-wise incremental registration would suffice to obtain a full model if the views contain no alignment errors. This becomes a challenging task while dealing with noisy datasets. The existing approaches use an additional offline optimization step to compensate the alignment errors for the set of rigid transformations [9].

Blind areas and/or *occlusions* are present in sequential views, when dealing with a *self-occluding* object, such as *Justin's* hand. Using the full dataset will generate an erroneous model. The solution to this problem is to generate partial models of the object. Huber and Hebert proposed a graph-based method to tackle this problem [6]. It relies on constructing a graph from all possible combinations of pair-wise registrations and discards faulty matches to create partial models. We propose a method that discards faulty matches *before* computing pair-wise registrations. It is based on an approach used to select window candidate pixels in depth images of building facades, introduced by Ali *et al* [19]. They use a statistical analysis on the distribution of the local depth variations, by applying an adaptive threshold value. We adapt this approach to select the overlapping views of a *self-occluding* object. An adaptive threshold value is applied to the min/max depth values to find the potential subset of overlapping views. This subset of views is registered with an extended metaview approach.

The generated models require a fixed origin w.r.t. the camera for pose estimation. We obtain several estimates from our extended metaview registration approach. These estimates are averaged to compute an absolute origin. The averaging of translational components is computed as a simple mean. However, the averaging of rotational components is not as straight forward. Sharf et al [20] review the existing formulations of rotation averaging and classify them based on their metric, either Riemannian or Euclidean solutions. We use a Euclidean solution, since it is considerably faster than the Riemannian. An implicit loop calibration configuration is defined [8] to measure the 3D coordinates of the origin of the hand w.r.t. the stereo vision system.

This work presents two key contributions:- (i) a model generation method for self-occluding objects that avoids the pre-computation of point cloud overlap and (ii) a functional verification procedure for *Justin's* upper body kinematic chains using 3D registration.

This paper is organized in five sections. The data acquisition and pre-processing of 3D point clouds is discussed in the Sec.II. Sec.III provides a comparison of local registration methods. The proposed model generation method is described in Sec.IV. It also provides a detailed evaluation of our method based on a synthetic model. Sec.V describes the pose estimation of *Justin's* hand using 3D registration. Finally, Sec.VI is the detailed description and evaluation of the verification routine used to estimate the bounds of the TCP end-pose errors.

II. Data acquisition and processing

Justin is equipped with a head mounted pair of calibrated cameras. This stereo system has a horizontal field of view (FOV) of 32° and vertical FOV of 20°. The left and right images (780x580) (Fig.1) are obtained via the SensorNet Library [4], which provides a small and fast mechanism for distributing real time streaming data. We process these stereo images with a Semi-Global Matching (SGM) algorithm [5].

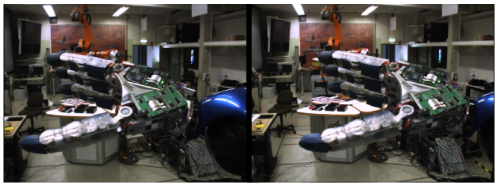

Figure 1: Left and Right Stereo Images of *Justin's* right hand

The SGM algorithm calculates a disparity map from the dense stereo images using a Census pixel-matching method based on a Non-parametric cost [12]. The output is a 2.5D image, containing color (RGB) and depth information per pixel. In Table I the depth resolutions (Δ_z) corresponding to different depth values (Z) are shown.

Table I: Stereo Depth Resolution

Depth (Z) [m]	0.1	0.3	0.5	0.6	1
Resolution (Δ_z) [cm]	0.013	0.12	0.33	0.49	1.35

Table II: Point Cloud Down-sampling

Leaf Size (mm)	0	1	1.5	2	2.5
Points	93,914	51,140	28,150	17,780	12,250

The calibration parameters from the Stereo Vision System are used to project each pixel of this 2.5D image into 3D world coordinates of a 3D point cloud. The region of interest in the point clouds are *Justin's* hands. *Justin's* arms have a workspace reach of approx. 1m. A pass-through filter [3] is applied at 1m. to cut-off any 3D points related to the background. A Statistical Outlier Removal filter [3] is used to remove small isolated point blobs generated by the Stereo Processing Algorithm. Since the acquired point clouds are dense, they are down-sampled with a Voxel Grid filter [3] to reduce computation time. In Table II, we show the impact of using different Voxel Leaf Sizes on our dataset.

III. Pair-wise registration of overlapping point clouds

In this section we evaluate several local feature-based correspondence methods, based on texture, surface and combined texture-surface descriptors.

A. Texture-based Initial Alignment

We use SIFT (Scale-Invariant Image Transforms) keypoints [13] to find interest points from two point clouds and obtain the rigid motion between them. The SIFT keypoints are highly descriptive points in an image that are obtained by comparing a pixel to its neighbors. The original SIFT algorithm was developed for the 2D image space. The algorithm we use is an adaptation to 3D point clouds which is available in PCL [2].The rigid transformation between the SIFT keypoints of two point clouds is computed by applying the ICP algorithm with Singular Value Decomposition.

B. Surface-based Initial Alignment

In 3D point clouds the description of the surrounding surface of a point is obtained by computing relations between it's neighboring points. The descriptors evaluated in this work use the estimated surface normals $n_i = (n_x, n_y, n_z)$ to compute these relations. For the following feature descriptors a sample-based method for initial alignment is used to compute the rigid transformation between correspondences. It is the Sample Consensus Initial Alignment (SAC-IA) method proposed by Rusu *et al*[3][11]. It allows two datasets to fall into the same convergence basin of a local non-linear optimizer, without trying all correspondence combinations.

1) Fast Point Feature Histograms (FPFH): FPFH Descriptors [11] are pose-invariant local features that represent the underlying surface of a point within a user-defined search radius. The relation between each point in this underlying

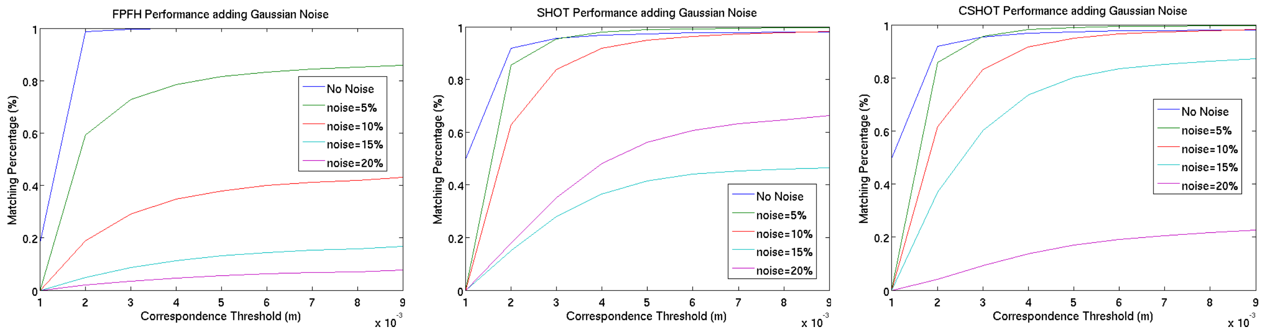

Figure 2: Descriptor Evaluation with increasing random Gaussian noise: (left) FPFH (center) SHOT (right) CSHOT

surface is a triplet of angles $<\alpha,\phi,\theta>$ between the normals and the Euclidean distance d between the points. The angles are computed by defining a fixed Darboux coordinate frame at the query point. For our data set, the suitable value for search radius is 2cm to have the best correspondence matching with down-sampled data at 1-2mm leaf size.

2) Signature of Histograms of Orientations (SHOT): SHOT Descriptors [17] rely on the definition of a local reference frame based on Eigen Value Decomposition of the scatter matrix of the underlying surface. A 3D grid is superimposed on the local reference frame of a query point. Local histograms containing geometric information of the 3D volumes generated by the grid are computed and grouped together to form the signature descriptor. The geometric relation used to compute the local histograms is $\cos(\theta_i)$, where θ_i is the angle between the surface normal vectors of a query point and a neighboring point. For our data set, the suitable value for search radius is 3cm to have the best correspondence matching with down-sampled data at 1-2mm leaf size.

C. Combined (Surface-Texture) Initial Alignment

CSHOT (Color SHOT) Descriptors [18] add texture representation of the underlying surface of a point to the original SHOT Descriptors. The textured-based description of the underlying surface uses the same formulation as SHOT. The texture relation between each point is the *L1* norm l(.) applied to the color triplets in *CIELab* space. For our data set, the suitable value for search radius is 3cm to have the best correspondence matching with down-sampled data at 1-2mm leaf size.

D. Local Feature Descriptor Comparison

We evaluate the performance of each local descriptor type with the Feature Evaluation Framework available in PCL [2]. We extended this framework to add artificial noise. The procedure involves the following steps:

1) Generate a random point cloud of the hand.
2) Duplicate this point cloud and apply a synthetic rigid transformation.
3) Apply a random gaussian noise of zero-mean to the new modified cloud.
4) Features are extracted from the original and modified point clouds.
5) Search for correspondences in feature space using a nearest neighbor approach based on a kd-tree.

The matching percentage is determined as the number of correct matches from the total number of points of the point cloud. It is plotted vs. the correspondence threshold for each feature type, which is the Euclidean distance between correspondences. FPFH, SHOT and CSHOT descriptors were evaluated by applying a Gaussian noise with increasing standard deviation of 5%-20% of the search radius (Fig.2). The precision of these methods is not dramatically affected by the Gaussian noise. However, SHOT appears to be more robust to noise than FPFH with this synthetic data.

We have also created an adaptation for evaluating the performance of point-to-point correspondence matching. This was applied to test the performance of the SIFT Keypoints. A random Gaussian noise with an increasing standard deviation of 1-4mm was applied (Fig.3). The SIFT Keypoints show a stable behavior, reaching 60% matching rate under high noise. However, the precision (which is represented as the correspondence threshold) is highly affected.

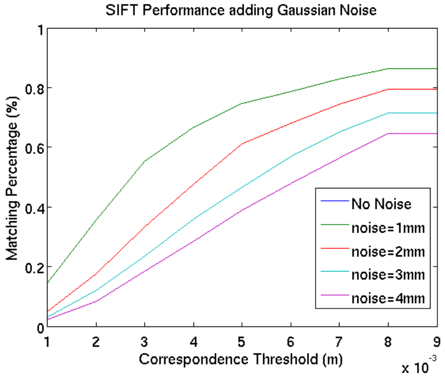

Figure 3: SIFT Evaluation with increasing random Gaussian noise

E. Fine-Tuning Method

The coarse rigid transformations obtained by any of the previous methods need to be fine-tuned. We use the standard ICP, which minimizes the point-to-point error between 3D correspondences.

IV. Model Generation

Two datasets for the model creation were generated. The first is a recording of an upright frontal configuration of *Justin's* right arm, the hand was rotated around the z-axis of the Tool-Center-Point (TCP) in 10° increments from $-30°$ to $150°$. The second is a recording of a sideways frontal configuration of the right arm, the hand was rotated around the z-axis of the TCP 10° increments from $-360°$ to $0°$. We implemented a pair-wise registration over all the views from both datasets with our local descriptor based methods. This resulted in a faulty registration (Fig. 4).

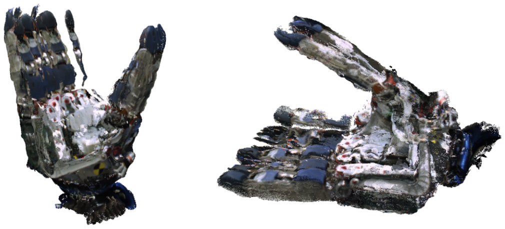

Figure 4: Faulty pair-wise aligned datasets: upright configuration (right) and sideways configuration (left)

We tried to compensate these registration errors by applying a global registration step, carried out with the external interactive tool *Scanalyze* [7]. We can see in Fig.5, that not even a supervised global registration tool can find a set of rigid transformations that aligns all of the views from both of our datasets. Views need to be rejected to create a functional model.

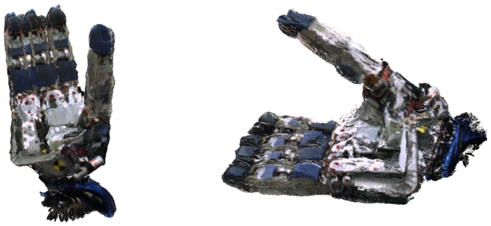

Figure 5: Faulty global registrations (notice the finger areas): upright configuration (right) and sideways configuration (left)

A. Extended Metaview Registration Method

We propose a method to select subsets of views, without pre-computing the overlapping regions. Following the approach introduced by Ali *et al* [19] we analyze the distribution of maximum depth values in a sequential order for all the views in a dataset (Fig.6).

Occlusions are identified by a significant variation of depth values in sequential views. We identify a clear occlusion between views 12 and 16 for the upright configuration dataset. The sideways configuration dataset has occlusions within a range of 0 to 22. The point clouds of non-occluded views represent the inner model of the hand. We analyze the minimum depth values to obtain the views for the outer model of the hands (Fig.7). The upright configuration dataset has only a 180° view of the hand, therefore it has no stable region. In the sideways configuration occlusions exist in depth values between view 25 and 35. These non-occluded views represent the outer model of the hand.

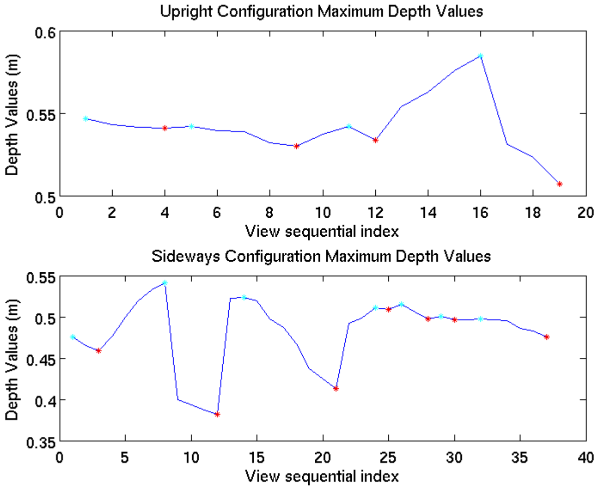

Figure 6: Max depth values of upright (top) and side (bottom) configurations

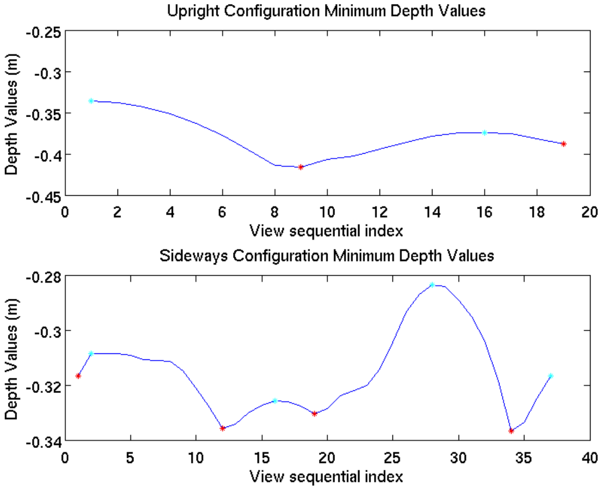

Figure 7: Min depth values of upright (top) and side (bottom) configurations

Based on the previous analysis, we identified that for *self-occluding objects* global maxima, minima and peak behavior on the max/min depth values are signs of occlusion. We developed a global thresholding process, that rejects the views that lie in these unstable areas. The *global thresholding process* involves the following steps:

1) Extract global minima and maxima from the min/max depth values, views neighboring the global max/min are discarded.
2) To compute an upper cut-off threshold, the mean or interquartile (IQR) of the mean are used to discard the views with higher depth values.
3) We extract the local maxima and the views between first and last are the final subset.

In this process we have discarded all the views that can possibly lead to a faulty registration. The view with the local minimum depth value is the *best view*. The *best view* is used as reference for the extended metaview registration method. To choose which view to register next, we compute a distance metric between the query view and the closest neighboring views. This *next view metric* (nvm) is computed by calculating the difference between the depth values and the sequential view index. The view that has the lowest *nvm* value to the query view is the *next-best-view*. This view is registered and merged to the reference view. This procedure is repeated until all of the views are registered.

$$nvm(x,y) = \sqrt{(x_d - y_d)^2 + (x_i - y_i)^2} \tag{1}$$

In Eq.(1) x is the query view and y is the next view candidate. Subscript d represents the depth value and i the view index value. We applied this algorithm to the overlapping views of the inner model of the upright configuration and the outer model of the sideways datasets. (Fig. 8). The upright configuration dataset recording with limited 180° rotation has no large variation of the depth values. We found the mean as an optimal upper cut-off threshold for this dataset. The sideways configuration has higher variations of the depth values because of the 360° rotation, therefore more occlusions are present. To deal with these occlusions the upper IQR of the mean was found to be as an optimal threshold.

B. 6DOF origin estimation

Each rigid transformation generated from the extended metaview method, yields an estimate of the absolute origin of the model. To compute the absolute origin for the model, we average these estimates. The translational component t of the origin is computed as the mean of t of all estimates. The rotation average is the least-squares solution to a metric-based optimization problem. We use a Euclidean metric surveyed by Sharf *et al* [20]: $d_F = ||R_1 - R_2||_F$. This metric is bi-invariant based on the Frobenius norm, which describes the difference between two rotation matrices. The average of N rotation matrices is the solution of the minimization problem based on this norm.

$$\bar{R_F} = min\sum_{i}^{N} ||R_i - R||_F^2 \tag{2}$$

This minimization problem has an exact solution. It is the orthogonal projection of the arithmetic mean R_{arith} in the Rotation Group *SO(3)*.

$$R_{arith} = \frac{1}{N}\sum_{i}^{N} R_i \tag{3}$$

As demonstrated in [20] this orthogonal projection can be calculated as the UV matrices of the singular value decomposition (SVD) of R_{arith}.

$$\bar{R_{SVD}} = UV \tag{4}$$

In Fig.9 the estimated origins $(TCP_1,..,TCP_n)$ and the averaged origin (TCP_O) of the inner model are shown.

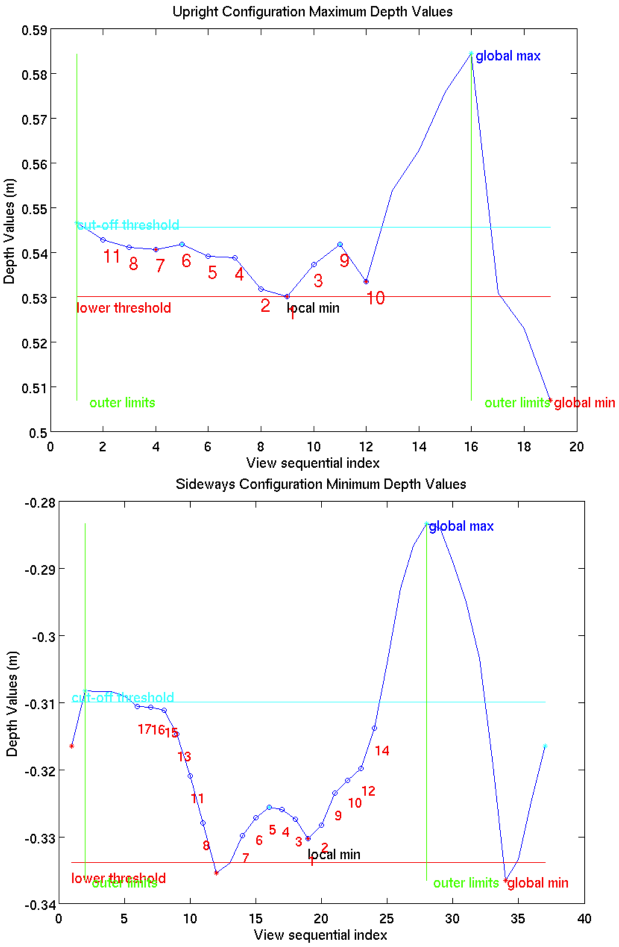

Figure 8: Global Thresholding Process applied to Best views for: (top) upright inner model (bottom) sideways outer model

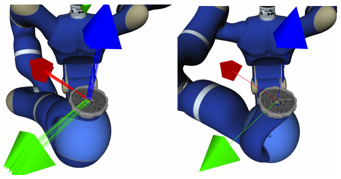

Figure 9: (left) Origin Estimates $(TCP_1,..,TCP_n)$ (right) Averaged Origin (TCP_O)

C. Model Evaluation

We created synthetic models (Fig.10) to evaluate the registered models obtained with each pair-wise registration method. These synthetic models are generated from the known relative rigid transformations between the forward kinematics of each registered view.

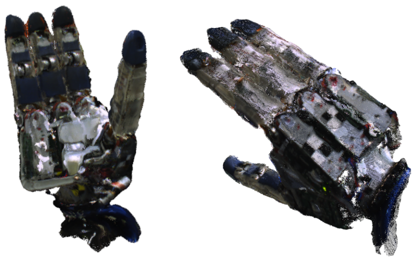

Figure 10: Synthetic model: (left) Inner and (right) Outer

Table III: Inner model rigid transformations

Dev. Metrics	SIFT	FPFH	SHOT	CSHOT
Max (($^\circ$)	2.190	2.264	2.215	2.254
Mean (($^\circ$)	0.879	0.887	0.88	0.885

Table IV: Outer model rigid transformations

Dev. Metrics	SIFT	FPFH	SHOT	CSHOT
Max ($^\circ$)	8.314	8.304	8.317	8.308
Mean ($^\circ$)	3.062	3.057	3.06	3.058

In Table III and IV we show deviations between the rigid transformations of registered and synthetic models in an Angle-Axis representation. The mean deviations within the four methods are almost negligible (around 0.01°). The inner models show a small max. deviation of approx. 2.2°, however the outer model has large max. deviation of 8.3°. With metrics, we can anticipate that the registered outer models will have poor evaluation metrics compared to the inner models. We further evaluate the partial models by applying surface consistency measures. These measures represent how the overlapping data of two surfaces can represent the same physical object [6]. These consist of projecting rays from the center of the camera to each point of the 3D model [6]. It is implemented by generating z-buffers or range images of a 3D model. We projected our synthetic and registered models into z-buffers (Fig.11) with an angular resolution of 0.15°/pixel.

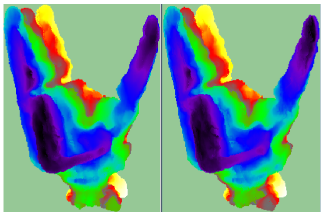

Figure 11: z-buffers: (left) synthetic and (right) registered models

Two surface consistency measures are computed:

- Out of bounds percentage:

$$out = \sum f(yi, j)/N_y \tag{5}$$

  f is a binary function that is 1/0 if the pixel is inside/outside the bounds of the synthetic model. Ny is the total number of pixels of the registered model.

- Mean Square Error from z-buffers:

$$MSE = \sum (x_{i,j} - y_{i,j})^2/N_y \tag{6}$$

  x and y are the depth values of single pixels from the synthetic and registered model's z-buffer respectively.

We also use an *Origin Error* to evaluate the models. It is the Euclidean distance between the origin of the synthetic model and the estimated origin of the registered model. We have tested our methods in an Intel(R) Pentium(R) D CPU 2.80GHz with 2GB RAM. The error metrics shown in Table V and VI are applied to partial models generated from 11 (inner) and 10 (outer) point clouds. Each individual point cloud contains approx. 50k points.

Table V: Inner model rigid transformations

Error Metric	SIFT	FPFH	SHOT	CSHOT
3D RMS (mm)	0.00106	0.00106	0.00106	0.00106
z-buffer MSE (mm)	0.05373	0.04969	**0.03907**	0.04932
Out of Bounds (%)	1.129	1.129	1.129	1.129
Origin Error (mm)	**0.861**	0.864	0.862	0.865
Comp. Time (s)	1918	76422	191173	180875

Table VI: Outer model rigid transformations

Error Metric	SIFT	FPFH	SHOT	CSHOT
3D RMS (mm)	0.00276	0.00275	0.00275	0.00279
z-buffer MSE (mm)	0.6056	0.5276	0.6423	**0.5046**
Out of Bounds (%)	11.511	8.748	10.130	**8.288**
Origin Error (mm)	3.366	**3.323**	3.435	3.357
Comp. Time (s)	1154	41419	188177	127767

Overall, the four evaluated methods show similar behavior. The SHOT-based methods show the best z-buffer MSE and Out of Bounds %. However, the SIFT- and FPFH-based methods show the best Origin Error. The SIFT-based method is the most efficient w.r.t. computation time. This is due to the n-dimensional search space during correspondence matching (SIFT-3d, FPFH-33d, SHOT- 358d, CSHOT-1344d). The datasets behave differently. The outer models shows poor error metrics compared to the inner models, therefore only the inner model is used for pose estimation. These poor metrics are due to the outer surface of the fingers. Their surface consists of transparent material and shows specular highlights, which show difficulties for any stereo processing.

V. Pose Estimation

The basic principle of estimating *Justin's* TCP end-pose by using 3D registration is to register a point cloud of the hand in a random pose TCP to the model of the hand with known pose TCP_O (Fig.12).

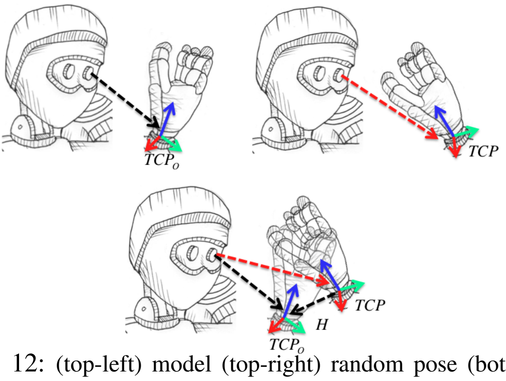

Figure 12: (top-left) model (top-right) random pose (bottom) H is the rigid motion of $TCP \rightarrow TCP_O$

The pose of TCP is computed in the sensor coordinate system as follows:

$$TCP = TCP_O(H)^{-1} = T_s^{tcp} \tag{7}$$

The implicit loop closure (Fig.13) of *Justin's* upper body kinematics enables us to relate our estimated pose TCP_{reg} to the measured pose from forward kinematics TCP_{fk}.

$$TCP_{reg} = T_w^h T_h^s T_s^{tcp} \tag{8}$$

$$TCP_{fk} = T_w^h T_h^a T_a^{tcp} \tag{9}$$

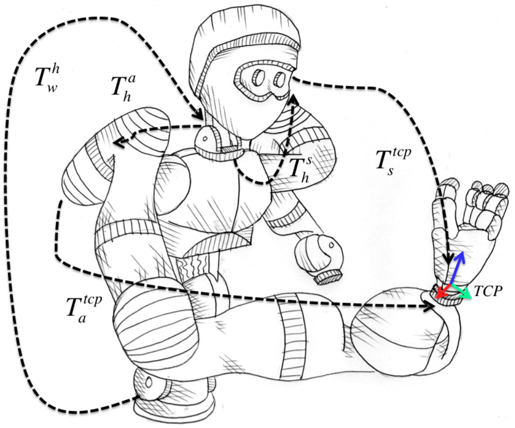

Figure 13: Implicit Loop Closure using the Stereo Vision System

We use *Justin's* head as the reference coordinate system for the loop closure. *Justin's* base is the world coordinate system. T_w^h is the transformation of the world coordinate system to the head joint, computed by simple forward kinematics. T_h^s is the transformation of the head joint to the sensor origin, generated by the stereo calibration process. T_h^a is the transformation of the head to the arm base, computed by a simple forward kinematic model. T_a^{tcp} is the transformation from arm base to TCP, computed by a forward kinematic model considering the measured torques and gear stiffness. T_s^{tcp} depends on H (Eq.7), which is obtained by a pair-wise registration. H is the fine-tuned rigid motion from an initial guess H_i (Sec.III). An initial guess H_{fk} was also computed by using forward kinematics. To improve the quality of the initial guess H_i we use H_{fk} as a *pre-guess* to the registration procedure. We also try using only H_{fk} as a direct initial guess. Therefore, we compare five methods for estimating H.

A. Method Limitations

Our pose estimation method has limitations concerning the arm kinematics (physical) and stereo vision system. The physical limitations concern Justins reach d=0.6m and the closest configuration to the head without colliding a=0.2m. The complete hand has to be in the FOV of the Stereo Vision System for pose estimation. The minimum depth that complies with this limitation is b=0.3m. The depth resolution increments as the hand is further away. We identified our maximum depth value as c=0.55m. Considering these limitations, we created a *Verification Volume* (Fig.14).

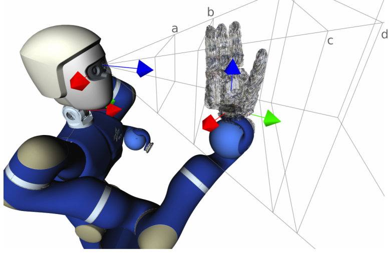

Figure 14: Verification Volume *(a/d:min/max physical limits, b/c:min/max limits of the 3D registration method)*

Table VII: Estimated TCP end-pose errors

Errors	ART	fk-ICP	SIFT	FPFH	SHOT	CSHOT
max $\|\|e_t\|\|$ (cm)	0.86	0.85	0.85	0.85	0.85	0.84
$\overline{\|\|e_t\|\|}$ (cm)	0.62	0.65	0.65	0.65	0.69	0.68
std $\|\|e_t\|\|$ (cm)	0.1	0.13	0.13	0.12	0.1	0.11
max e_θ (°)	1.48	2.3	2.07	2.3	2.06	2.06
$\overline{e_\theta}$ (°)	0.94	0.85	0.83	0.84	0.88	0.8
std e_θ (°)	0.29	0.56	0.54	0.56	0.56	0.52
$\overline{fit}$ (μm)	-	2.63	2.63	2.63	2.63	2.63
rejected	-	9	9	9	9	9

Table VIII: Overall Performance of Pose Estimation by using 3D Registration

Method	fk-ICP	SIFT	FPFH	SHOT	CSHOT
$\overline{Comp.Time}$ (s)	64	170	329	731	512
Succes Rate (%)	62.42	62.42	61.74	58.39	57.72

B. Error Identification

Ideally $TCP_{reg} := TCP_{fk}$ (Eq.8, Eq.9), however, this is not the case. We represent this error with a tuple $e =< e_t, e_\theta >$ extracted from the $\Delta T = (TCP_{reg})^{-1} TCP_{fk}$. Where $e_t = \Delta T(t)$ and $e_\theta = angleAxis(\Delta T(R))$.

C. Experimental Results

To evaluate our identified errors we compare them to an error *ground-truth*. We generated this ground truth by estimating the TCP end-pose errors with the ART (Advanced Realtime Tracking) IR tracking system, as described in previous work [21]. N-random poses within the *Verification Volume* are generated. Point clouds of these poses are created and registered to the model. An error tuple ($e =< e_t, e_\theta >$) and a fitness score ($\overline{fit}$) are computed for each registration. To ensure the quality of our estimated errors, we apply a RANSAC outlier rejection algorithm on the $\overline{fit}$ of the complete set of poses. In Table VII we show the estimated errors of one test with 28 random poses. Overall the $||e_t||$ estimated by all methods are consistent to the *ground-truth* (ART). However, the estimated e_θ varies and exhibits a small deviation from the *ground-truth* (0.5-0.8°). The SHOT-based methods show the least deviation. To further evaluate our methods we calculated a success rate which is formulated as follows:

$$SuccessRate(\%) = \frac{N_t - R_t}{N_t} \times 100 \qquad (10)$$

Where N_t is the total number of poses and R_t is the total number of rejected poses from faulty registrations. The success rates and average comp. times of all methods are shown in Table VIII. These metrics were computed from 5 tests with different environmental conditions. 150 poses were estimated with downsampled point clouds with a 2mm voxel leaf size (model-65k points, random-20k points). The success rates from all methods are within the same range (around 60%). If fast computation time is crucial then the fk-ICP method is the best suited, with the draw-back that the e_θ may be inconsistent with the ground truth. If the accuracy of e_θ is of higher importance than comp. time, the SHOT-based methods should be used.

VI. Application: Bound-Identification of Justin's TCP errors

The aim of this work was to create a verification routine to identify the bounds (e_b) of the TCP end-pose errors. It was designed to be conducted before any robot interaction. Therefore, e_b can be used to pre-adjust the obstacle clearance in path planning techniques. As an offline step for this routine, a model of the hand has to be generated (Sec. IV). New models are necessary only if the calibration of the stereo system has been modified. The routine (Fig.15) consists of the following steps:

1) N-random TCP poses are created within the *Verification Volume*.
2) For every pose $n \in N$ an individual *Error Identification Pipeline* is executed, where the output is an error tuple $e_k =< e_t, e_\theta >$ and a fitness score f_k (Sec. V-B).
3) Once the set of error tuples $E = (e_1, \cdots, e_N)$ and fitness scores $F = (f_1, \cdots, f_N)$ are estimated, a RANSAC outlier rejection algorithm is applied to the fit. scores $F^* = RANSAC(F)$. A subset of error tuples E^* corresponding to F^* is created.
4) The maximum bounds e_b of the TCP end-pose errors are calculated as:

$$e_b =< \max(||e_t|| \in E^*), \max(e_\theta \in E^*) > \qquad (11)$$

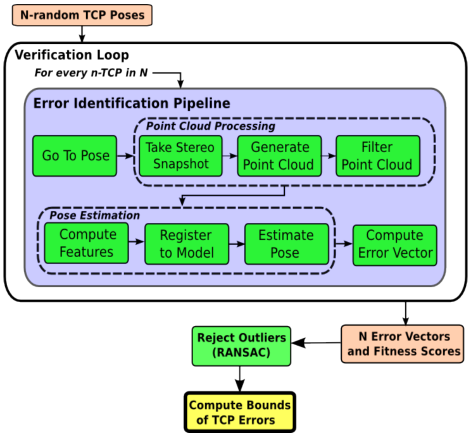

Figure 15: Verification Routine

The total computation time of the *Verification Loop* depends on three factors:- (i) the number of random poses (N), (ii) the cores available to compute individual *Error Identification Pipelines* and (iii) the chosen registration method. For 30 random poses using fk-ICP with one available core the total comp. time is around 45 min. However, if 10 cores are available it is approx. 4.5 min.

VII. Conclusions and Future Work

We presented a functional verification routine using 3D registration to compute the bounds of *Justin's* TCP end-pose errors. This involved extensively evaluating 3D registrations methods and proposing a method for generating partial models of *self-occluding* objects. We plan on using the identified bounds as an error compensation reference for path-planning algorithms. Furthermore, a detailed analysis of the identified errors could be performed to generate a hypothesis of which joint/joints are consistently affecting the TCP end-pose.

References

[1] A. Albu-Schäffer, S. Haddadin, Ch. Ott, A. Stemmer, T. Wimbck, and G. Hirzinger. The DLR lightweight robot: Design and Control concepts for Robots in Human Environments. *Industrial Robot: An International Journal*, 34(5):376-385, 2007.

[2] R.B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). *ICRA*, 2011, pages 1-4, May 2011.

[3] Radu Bogdan Rusu, *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD Thesis, Computer Science Department, TUM, Germany, October 2009.

[4] Tim Bodenmueller, *Streaming Surface Reconstruction from Real Time 3D Measurements*. PhD Thesis, TUM, Germany, 2009.

[5] Heiko Hirschmueller, Stereo Processing by Semi-Global Matching and Mutual Information, *TPAMI*, vol. 30, 2008, pp 328-341.

[6] D.F. Huber and M. Herbert, Fully Automatic Registration of Multiple 3D Data Sets, The Robotics Institute, CMU, Pittsburgh, Pennsylvania, 2001.

[7] Kari Pulli, Multiview Registration for Large Data Sets, *3DIM (1999)*, pp 160-168.

[8] W. Khalil and E. Dombre, Geometric calibration of robots, *Modeling, Identification and Control of Robots*, Kogan Page Science, 2002

[9] A. Makadia, E. Patterson and K. Daniilidis, Fully automatic registration of 3d point clouds, *CVPR*, 2006, pp 1297-1304

[10] J. Salvi, C. Matabosch, D. Fofi and Josep Forest, A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, vol. 25, No.5, May 2007. pp 578-596

[11] R.B. Rusu, N. Blodow and M. Beetz, Fast Point Feature Histograms (FPFH) for 3D Registration, *ICRA*, May 2009.

[12] R. Zabih and J. Woodfill, A non-parametric approach to visual correspondence. *ECCV*, 1994, pp 82-89.

[13] David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, vol. 60, No. 2, January, 2004.

[14] P.J. Besl and N.D McKay, A Method for Registration of 3-D Shapes, *TPAMI*, vol. 14, 1992.

[15] Zhengyou Zhang. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *IJCV*, 13(2):119-152, 1994.

[16] Y.Chen and G.Medioni, Object modeling by registration of multiple range images, *ICRA*, Sacramento, USA, 1991.

[17] F. Tombari, S. Salti and L. Di Stefano, Unique Signatures of Histograms for Local Surface Description, *ECCV*, 2010

[18] F. Tombari, S. Salti and L. Di Stefano, A combined texture-shape descriptor for enhanced 3D feature matching, *ICIP*, 2011

[19] H. Ali, R. Sablatnig, and G. Paar. Window detection from terrestrial laser scanner data - a statistical approach. In *VISAPP (1)*, pp 393-397, 2009.

[20] I. Sharf, A. Wolf and M.B. Rubin, Arithmetic and geometric solutions for average rigid-body rotation, *Mechanism and Machine Theory*, vol. 45, pp 1239-1251, 2010.

[21] Nadia Barbara Figueroa Fernandez, *3D Registration for Verification of Humanoid Justin's Upper Body Kinematics*. MS Thesis, ETIT-IRF, TU Dortmund, Germany, March 2012.