INTERACT – Interactive Manual Assembly Operations for the Human-Centered Workplaces of the Future

Grant Agreement Number	**:**	**611007**
	:	**INTERACT**
Project Start Date	**:**	**1st October, 2013**
Consortium	**:**	**DAIMLER AG (DAIMLER)- Project Coordinator**
		ELECTROLUX ITALIA S.P.A. (ELECTROLUX)
		INTRASOFT INTERNATIONAL SA (INTRASOFT)
		IMK AUTOMOTIVE GMBH (IMK)
		EMPHASIS TELEMATICS AE (EMPHASIS)
		HADATAP SP ZOO (HADATAP)
		UNIVERSITY OF PATRAS (LMS)
		UNIVERSITAET ULM (IMI)
		DEUTSCHES FORSCHUNGSZENTRUM FUER KUENSTLICHE INTELLIGENZ GMBH (DFKI)

Title	**:**	**Motion Modeling — Methodology Applied and Lessons Learned**
Reference	**:**	**D2.2.3**
Availability	**:**	**Public (PU)**
Date	**:**	**2016-08-30**
Author/s	**:**	**Daimler, DFKI, IMK**
Circulation	**:**	**EU**

Summary: To achieve the goal of INTERACT to investigate and develop tools used to support design, verification, validation, modification and continuous improvement of human-centered, flexible assembly workplaces, technologies to synthesize motions which appear realistic to a human user are of key importance. D2.2.3 summarizes the final methodology applied in the best fit simulation of INTERAT and the lessons learned in the course of the project.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

Table of Contents

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

1 INTRODUCTION

INTERACT aims to utilize workers' knowledge on executing manual assembly tasks and include it in the digital tools used to support design, verification, validation, modification and continuous improvement of human-centered, flexible assembly workplaces. To achieve this goal INTERACT proposed a so-called best fit simulation which is able to simulate the execution of worker instructions for manual assembly tasks which are given in as controlled language input.

The data-driven motion synthesis module represents the core component of this best fit simulation. For this purpose a statistical motion synthesis method based on existing work by Min et al. [1] was implemented and extended. We construct a statistical motion model from motion capture data of manual assembly tasks. At runtime new motions can then be generated by searching in the motion model based on user defined constraints. The method was integrated with a text based user interface that creates constraints based on a scene knowledge base.

The rest of the document describes in detail the methodology for the construction of the statistical motion model and the constrained motion synthesis. Furthermore we describe important lessons learned from applying the statistical motion synthesis method in practice.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

2 FINAL METHODOLOGY APPLIED

Figure **1Fehler! Verweisquelle konnte nicht gefunden werden.** shows an overview of the offline motion modeling and of the online motion synthesis pipelines. Using MoCap data as input a motion model is constructed offline. The motion generator module uses this motion model at runtime to generate a new motion fitting the input constraints. Section 2.1 describes the motion modeling pipeline in detail and Section 2.2 describes the motion generator loop that converts constraints into a new motion and uses an independent service to generate constraints in order to avoid collision.

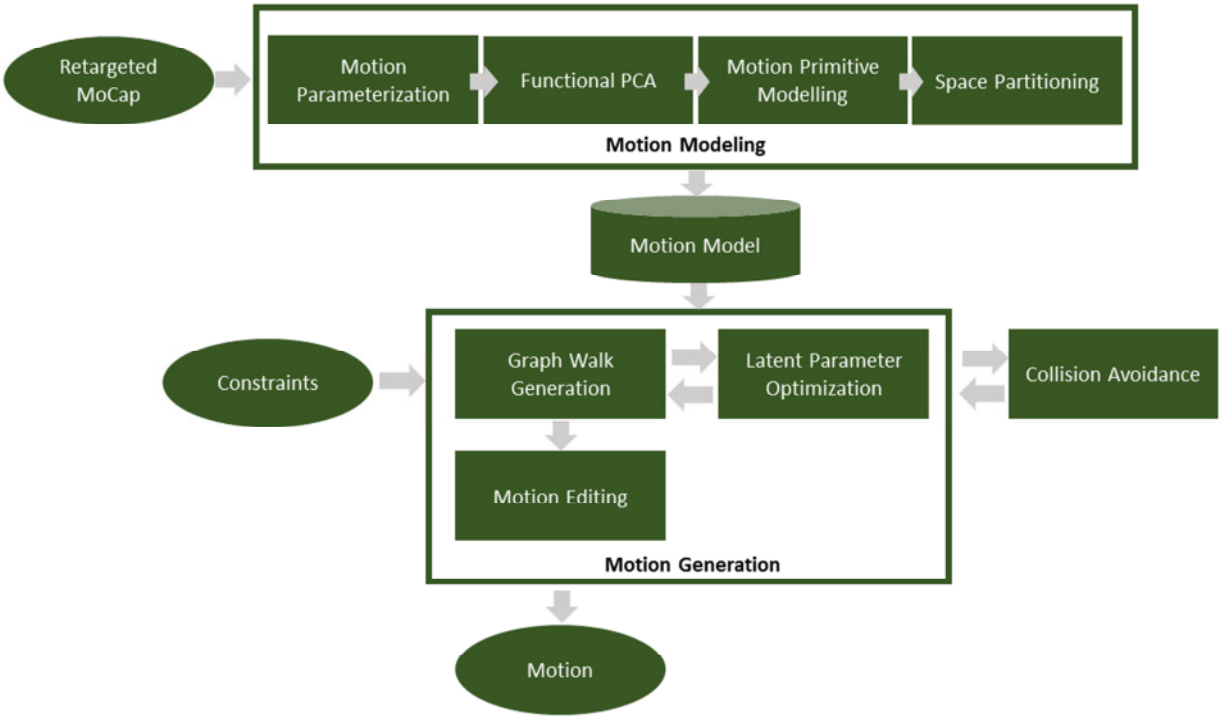

Figure 1: Motion modeling pipeline overview**.**

2.1 Motion Model

The goal of motion model is to construct a statistical human motion database from motion capture data to simulate animation in assembly workshop scenarios. In Best-fit simulation software prototype, motion model module provides input for motion generator to simulate a large amount of variant motions for different constraints.

2.1.1 Motion Parameterization

In order to provide a compact and scalable solution to model a large amount of motion capture data with rich variation in INTERACT, a highly structured motion data representation is applied (c. [1]). The assumption is that although human motion appears to have infinite variations, the fundamental high-level structures (motion primitives) are always finite. For example, normal walking can be regarded as a sequence of alternating left and right stances, and picking can be decomposed as reaching and retrieving. A directed

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

graph is employed to model motion primitives and the possible transitions between each other. Figure **2** shows an example of high-level structures of normal walking. Long recorded motions are decoupled into small clips, and structurally similar motion clips are categorized into each note in the graph. A motion primitive is a statistical model, which describes the distribution of motion clips in one note.

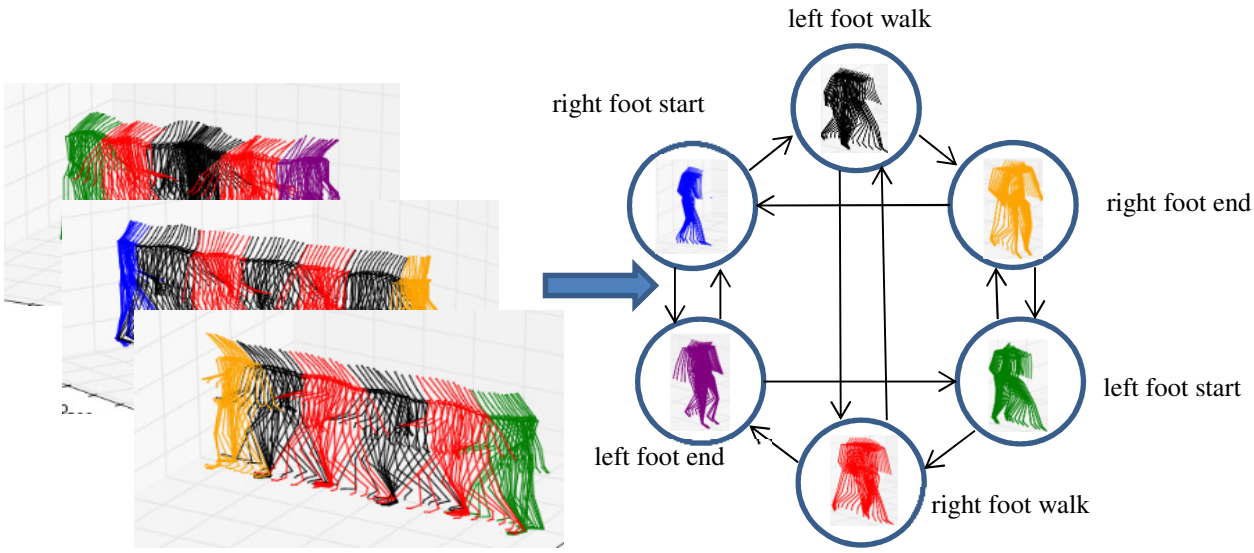

Figure 2: High-level graph representation for elementary action walking.

2.1.1.1 Motion Segmentation

The previously captured motion data is decomposed into semantically and structurally meaningful clips to embed high-level semantic information like left step, right step and so on, into motion modeling. The segmentation is done by defining and extracting key frames in motion data. Key frames are instances with contact state transitions occurring (e.g. feet contact on ground) or with significant visual content changing (e.g. pick reaching and retrieving). In INTERACT, a rule-based approach is applied to automatically extract pre-defined key frames in motion data. However, for complex actions, such as inserting, screwing, manual work is still required to guarantee the good quality of segmentation. The key frames are taken as border frames to cut motion data into small clips. The motion clips which share the same starting and ending key frames are categorized into the same motion primitive.

2.1.1.2 Low-level Semantic Annotation

After motion segmentation, the high-level semantic information is embedded into motion primitives. For instance, samples in walk_leftStance motion primitive should be all one step walking with left leg moving. Besides the high-level semantic information, we are also interested in some low-level semantic information, for example, foot-ground contact for locomotion or hand-object contact for manipulation action. Although the high-level semantic information is the same for one motion primitive, however, the low-level semantic information varies for different motion clips. Similar as key frame extraction, we apply a rule-based approach to automatically annotate frames in motion capture data.

2.1.1.3 Motion Alignment

Motion clips which are within the same motion primitive, could have different root position, orientation and number of frames. For statistical learning, motion clips in one motion primitive should not only look similar,

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

but have similar data representation as well. Therefore, motion clips within the same motion primitive are normalized to have the same starting root position and orientation, and frames are aligned to a reference motion clip by using Dynamic Time Warping (DTW). The reference motion clip defines a canonical timeline for motion clips in one motion primitive. We choose the one with minimum average frame distance to other motion clips as reference. Figure 3 illustrates left toe height of walking leftStance before alignment and after alignment. After motion alignment, the original motion data is decomposed as warped motion clips and their corresponding time warping indices.

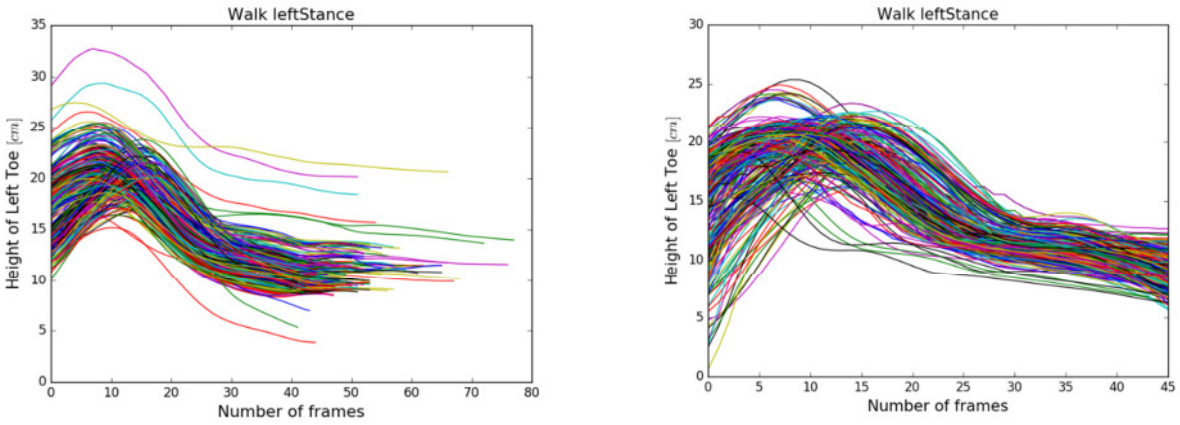

Figure 3: (Left) motion data before alignment; (Right) motion data after alignment.

2.1.1.4 Motion Parameterization

The success of statistical learning approaches generally depends on the data representation. For motion capture data, an informative parameterization needs to address two properties: smoothness and similarity. Motion capture data is smooth, time-series data from observation, so the motion parameterization should be corresponding to this observation. Motion clips in the same motion primitive are similar after motion alignment, so data representation should be similar as well. In INTERACT, we represent motion capture data as root position and orientation, and orientation of joints in the skeleton. The orientation is parameterized as quaternion. The singularity issue caused by antipodal points in unit quaternion space is solved by smoothing quaternion values for each dimension (c. [6]).

2.1.2 Functional Principal Component Analysis for Motion Data

In general, motion capture data is high-dimensional because of high degree of freedom of human body. However, the data is redundant and the dimensionality can be reduced from two perspectives: the movement of each part of the body is highly coordinated and the adjacent frames are very similar due to high frame rate. Functional Principal Component Analysis (FPCA) [5] is applied to reduce the dimensionality of motion capture data from spatial and temporal domain.

2.1.2.1 Functional Data Analysis

The key idea of functional data analysis of motion data is to represent each motion clip as a vector of continuous functions rather than a sequence of frames. Each dimension of discrete motion data is interpolated by a linear combination of cubic B-spline functions. Each continuous function is parameterized as a vector of weights of B-spline functions.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

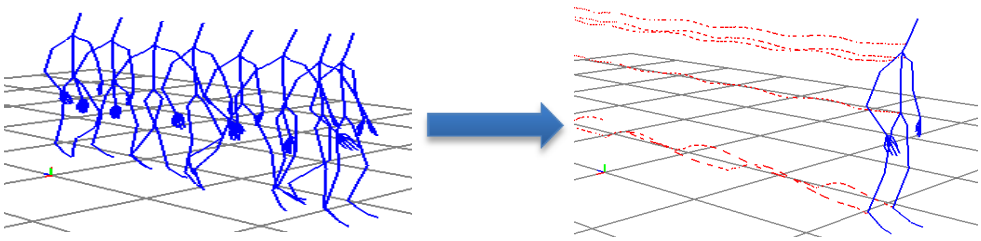

Figure 4: Functional data representation for motion clip. A sequence of frames is represented as a set of continuous functions.

The time warping indices and semantic annotations of motion capture data are convert from discrete data to functional data as well. An additional step for time warping indices is that in order to keep the monotonic increasing property, the Z transform (c. [1]) is employed on the functional data of time warping indices.

2.1.2.2 Functional Principal Component Analysis

Functional data analysis provides a compact representation to reduce the redundancy in temporal domain. We further reduce the dimensionality of data by applying Principal Component Analysis (PCA) on functional data, which consists of the weights of B-spline functions. The number of principal components is chosen by keeping 95% explained variance. For temporal and semantic functional data, PCA is applied to further reduce dimensionality as well.

2.1.3 Motion Primitive Modeling

In order to model the distribution of motion primitive in low dimensional space, we concatenate the low dimensional projections of spatial, temporal and semantic information of each motion clip as a vector, and model the distribution of motion clips using Gaussian Mixture Model (GMM). The parameters of GMM are estimated by applying Expectation-Maximization (EM) algorithm. The number of Gaussians is selected by empirically evaluating a set of values to maximize the Akaike Information Criterion (AIC)[1] for training data. AIC calculates the sum of negative log likelihood of input samples, with a penalty of the complexity of the model. A low AIC score indicates the model is generative and fits training data well. A high AIC score indicates that either the model cannot fit the data well, or the model is over fitted due to insufficient number of training samples.

2.1.4 Space partitioning

We build a space partitioning data structure for each motion primitive by recursively applying the K-means algorithm on random samples from the statistical model. The optimal number of random samples is set to 1000 and the number of subdivisions per level to 4 based on experimental results. The motivation is to accelerate the motion synthesis by quickly discarding a part of the latent space based on the observation that samples close in latent space resulting from FPCA are close in constrained space for interesting joints, e.g. the right hand in the pickRight motion primitive.

[1] https://en.wikipedia.org/wiki/Akaike_information_criterion

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

```
DIVIDE (data, k):
    children = [ ]
    clusters = k-means-partition(data, k)
    For c in clusters do
        children.push(DIVIDE(c, k))
    return Node(mean(data), children)
```

Algorithm 1: Recursive cluster tree construction

2.2 Motion Generation

Based on the motion primitive graph generate a new motion fitting input constraints in three steps: First, a graph walk through the graph of motion primitives is generated sequentially and the latent parameters of each motion primitive are optimized to fit the constraints. Then the motion primitives in the resulting graph walk are back projected into a motion splines and concatenated into a single spline. Finally the motion spline is discretized into a frame based representation that is edited using inverse kinematics to reach constraints outside of the range of the training data. In order to handle collisions with the environment we make use of a separate service that was developed that provides constraints that avoid collisions with the environment.

2.2.1 Graph Walk Generation

The motion synthesis supports two types of constraints: trajectory and key frame constraints. Trajectory constraints define the position of a joint during an entire elementary action. Key frame constraints allow to constrain the position or orientation of a joint on one frame of the motion. Furthermore, they can be used to constrain the time on which the constraint must be reached. For the definition of key frame constraints we semantically annotate the canonical timeline of motion primitives with semantic labels such as start_contact and find on runtime the frame that meets the semantic labels associated with the key frame constraint.

The motion synthesis algorithm takes constraints on elementary action level and breaks them down to motion primitive level. The breakdown uses edges in a manually defined motion primitive graph and the annotation of motion primitive as start, transition or end primitives to create the graph walk. An overview of the sequential motion synthesis algorithm is shown in Algorithm 2.

```
Initialize empty graph walk
For (action, constraints) in elementary action list do
    While state is not end of action do
        Transition to new state
        Generate state constraints from action constraints
        Optimize latent parameters of step
        Check for collisions and re-estimate latent parameters if necessary
        Add state with optimized parameters to graph walk
    Optimize all steps of current elementary action
Return graph walk
```

Algorithm 2: Sequential motion synthesis algorithm.

For motions that following trajectory constraints on the hip joint, such as walk and carry, we have generate constraints for each step until the end of the trajectory has been reached. The constraints of individual steps consist of the position and orientation of the projected hip joint at the end of each step and are generated based on a heuristic using the median step length of the motion primitive. Due to structural differences in the motion, we have to separate motion primitives such as sidestep and a standard step. In order to choose the

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

appropriate motion primitive at runtime, the path following algorithm can evaluate multiple options per step to select the best option given the current step constraints.

In order to prevent collisions with the environment, each step is evaluated by a remote Collision Avoidance service. If the CA-service detects a collision, new constraints are generated and the latent parameters are modified in order to generate a motion that avoids the collision.

After the generation of the graph walk with optimal low dimensional parameters, the low dimensional parameters of the graph walk are back projected into motion splines, concatenated and discretized into frames for visualization. During this discretization we generate an annotation of the key frames that is used in the Simulator for the scene manipulation.

2.2.2 Latent Parameter Optimization

Inside of the breakdown loop of Algorithm 2 the optimal parameters for each step are found using latent parameter optimization. First an initial guess is found by a search in the cluster tree of the motion primitive. During the search the mean of each node in the tree is evaluated using the following objective function that sums up the distance to the constraints.

$$\underset{s}{argmin} \sum_{k=0}^{N} \|fk(M(s)) - c_k\|^2 \tag{1}$$

Here, s is the low dimensional parameter vector, N is the number of constraints, fk is the forward kinematics function and M is the back projection from low dimensional parameters into a motion spline. For the search in the cluster tree we keep multiple candidates at each level of the tree to avoid getting stuck in local minima.

The resulting parameter vector is then further optimized using the Levenberg-Marquardt algorithm. In order to prevent the algorithm to produce unnatural motions, equation (1) is extended with a naturalness term as shown in equation (2).

$$\underset{s}{argmin} \sum_{k=0}^{N} \|fk(M(s)) - c_k\|^2 - ln\, pr(s) \tag{2}$$

The likelihood term is the negative log likelihood of the motion primitive model $pr(s)$ constructed as a GMM.

By minimizing the negative log likelihood of the distribution function, we can maximize the likelihood of the motion in respect to the original samples. In order to handle different value ranges for the kinematic error and the likelihood term we added weight factors to equation (2), that can be adapted based on the types of constraints.

By optimizing the parameters, the natural motion that most likely fits the constraints can be estimated. Additionally, if a constraint is unreachable by the range defined in the original samples, the naturalness term prevents the optimization from producing unnatural parameters.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

For elementary actions that follow a trajectory-constraint such as walk and carry further optimization over multiple steps is required to ensure that the end of the trajectory has been reached correctly. For this purpose we concatenate the latent space parameters of T steps into one long vector.

$$\underset{0\ldots s_T}{argmin} \sum_{J=0}^{T} \frac{\sum_{k=0}^{N} \left\| fk\left(M_j(s_j)\right) - c_k \right\|^2}{2\sigma^2} - ln\, pr_j(s_j)$$

The result of the step-wise optimization is used as an initial guess and the resulting parameters of the multi-step optimization replaced in the graph walk. In our evaluation T is set to 3 steps based on experimental results.

2.2.3 Collision Avoidance

In order to avoid collisions, it is important to determine the points where the avatar collides with the scene geometries. Usually, collision detection is computationally expensive. In the present work the open source physics engine Bullet Physics Library[2] is used for detecting the collisions. The collision detection module developed on top of the Bullet engine detects the collision in the input motion provided by the motion generator. It is decided by the consortium that the entire motion which is given as input to the collision avoidance module need not be modified. Only the frames in the input motion which exhibit collisions need to be modified. Continuous frames in the input motion which exhibits collision are defined as "Collision Island" in this work. A synthesized motion may comprise of several such collision islands. The collision avoiding constraints are generated only for these collision islands.

Currently, the collision avoidance module provide positions of the end-effector (hand joint) for reach and place elementary actions for their single handed and two handed variant as constraints. A standard A Star algorithm is used for computing the constraints. The collision avoiding constraints are ensured to be within the reach space of the avatar. This is done by using the captured reach space data as the base. Position of generated constraints is checked against the captured reach space data. The constraints which are positioned out of the reach space data are disqualified to be considered as collision avoidance constraints.

2.2.4 Motion Editing

The range the training data of the motion models is limited, therefore user defined constraints can sometimes not be reached by the latent parameter optimization. In order to increase the reachability of constraints, we use inverse kinematics to edit pose parameters of individual frames of the discrete motion representation.

The constraints used during the motion synthesis are reused for motion editing by mapping the semantic labels to frames of the discrete motion representation based on the same method that is also used for the generation of the frame annotation. We then directly optimize on the quaternion parameters of free joints in the kinematic chain influencing the constrained joints to reach those constraints using the L-BFGS-B algorithm. The free joints are defined manually for each end effector in a simple skeleton model. By using the pose parameters of the motion synthesis as initial guess for the optimization the result stays closes to the training data.

[2] http://bulletphysics.org/

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

The modified frame parameters are then blended with the original frame parameters using Spherical Linear Interpolation (Slerp). We first synthetically create a smooth transition motion to and from the modified frame. This synthetic transition motion is then blended with the original motion using a sliding weight function.

Due to the lack of hand motion models the hand motion is generated during motion editing. For this purpose predefined hand poses for open and closed hands are used to procedurally create the hand animations using Slerp.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

3 LESSONS LEARNED

3.1 Data Requirement Estimation for Statistical Modeling

In general, the real distribution of motion capture data can be better modelled with sufficiently large amount of samples. However, motion capture and data preprocessing are time-consuming and costly. The connection between variation of synthesized data and required number of samples is not clearly stated in the MG++ approach [1], which is the basis of motion synthesis for INTERACT. So it is important to learn the required number and distribution of training samples for constructing a good statistical model. However, it is nontrivial to find out the necessary number of samples for good models, since the samples in different motion primitives have different number of dimensions, and the range of the variation in the motion is also different.

Our experimental results lead to two conclusions. In principle, if the motion samples display large variation—for instance, several clusters in low dimensional space, in order to capture the distribution of the data—a complex model which contains several Gaussians is required to achieve a high likelihood of training samples. In this case, more samples are required to avoid overfitting the parameters of the model and to fill the gaps between clusters (c. [2]). The other case is when the motion is relatively complex, which requires more dimensions in low-dimensional space, in order to capture the same variation compared to simple motions. More training samples are preferred in this case to avoid overfitting the GMM model due to the curse of dimensionality.

In INTERACT, AIC score (see Section 2.1.3) is employed as a measure of model quality to address the aforementioned observations. Table 1 in the appendix shows the evaluation of parts of the motion primitive models in INTERACT. The statistical evaluation results are corresponding to the user study of motion primitive models in D2.3.1. The models with a larger number of input data such as walk_leftStance, walk_rightStance, achieve very good model quality. For walk_sidestepLeft, although the average log likelihood is not low, the high AIC score indicates that the model is over fitted, since there are 5 Gaussians for 40 samples.

3.2 Functional Data Analysis for Motion Data

Many statistical motion modeling approaches take motion capture data as a sequence of frames, and parameterize it as a long vector. The smoothness and sequence order of frames cannot be represented well by this representation. An appealing property of statistical motion synthesis approach presented by Min et al. [1] is that they claim that they apply functional PCA to model motion, which can intrinsically address the smoothness of motion data. However, details about the functional representation and the use of FPCA are missing in the paper and the authors refer to their previous work [4], which clearly states PCA is used for dimensionality reduction of frame data. Furthermore the dissertation of the author [8], was not yet made available to the public. We therefore had to derive the details of the approach experimentally.

Following the information given in the paper, we first evaluated the performance of applying PCA on frame-wised motion capture data. We constructed a set of motion primitive models for walking, picking, carrying and placing using a frame-wise motion representation for M12 prototype. However, in order to address the requirement of smoothness of the synthesized motion, different motion parameterizations (c. [6]) were evaluated and functional data analysis was explored. Based on previous research (e.g. [5], [7]), intensive

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

experimental efforts were made to evaluate different basis functions for spline interpolation and different choices of parameters to achieve a good representation for motion capture data. In the M18 prototype, the previous motion primitive models were replaced with cubic B-spline-based models. Furthermore, more elementary actions which are required in INTERACT were modeled successfully with functional data analysis in the following prototypes.

Our results indicate that the cubic B-spline basis can offer a compact representation of our motion data. Figure **5** shows the reconstruction error between B-spline representation and original discrete data. The reconstruction error is measured by evaluating functional data at canonical frames compared to original data. Generally, we are interested in using less basis functions while keeping the functional data as close to the original data as possible. It seems that some complex motion, for instance picking and placing, require more basis functions to achieve the same reconstruction error, however, the number of canonical frames is different for different motion primitives. Usually the length of an aligned motion clip in a complex motion primitive is longer than in a simple motion. So the reduction rate of the functional representation is similar for most motions, which indicates that the cubic B-spline basis works well for our motion data.

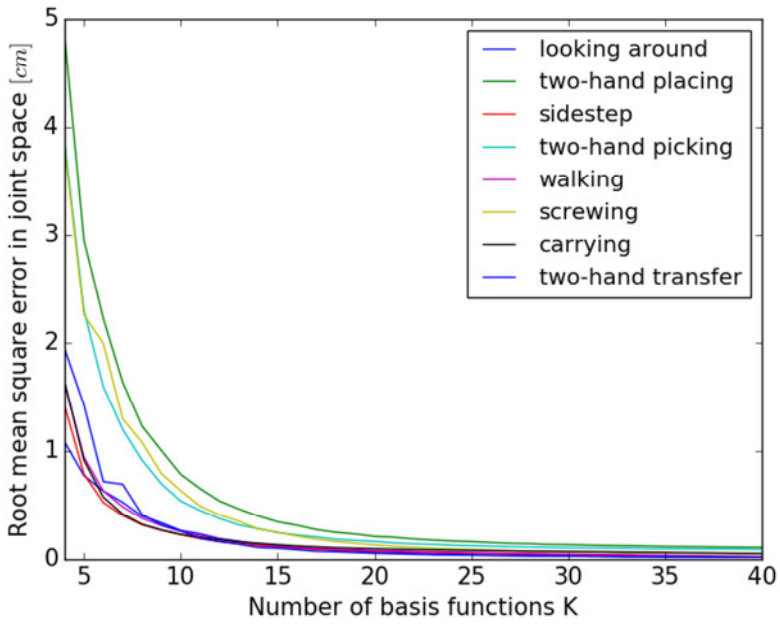

Figure 5: Evaluation of functional data representation for different motion. Root mean squared error (RMSE) between functional data and discrete data are measured in joint space. RMSE for each motion type is plot with increasing of number of basis functions K.

3.3 Transition Model

Transition model between motion primitives is one of the main appealing properties of MG++ approach. In INTERACT, since M12, great efforts were deployed to investigate the implementation of transition model, and evaluate the performance of transition model, due to lacking implementation details in the original paper [1]. A test version of transition model based on Gaussian processes was implemented by the cooperation of Daimler and DFKI. In addition, a great number of experiments were done by DFKI and LMS to find the

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

optimal parameters for the model. However, the results were not natural enough for practical usage and the variation of the generated model was not big enough for INTERACT.

From our experiments, we found that for a good transition model, every motion clip should have a range of different styles as possible next transition. However, this is not supported by the design of motion capture in the INTERACT project, which has limited variation. Therefore, the transition model trained by our data also only displays a limited range of variation, so it fails to fulfill the requirement that the generated motion should support a rich set of constraints while keeping as natural as possible.

3.4 Acceleration using Space Partitioning Data Structures

The original paper by Min et al. [1] proposed random sampling to find a good guess that they use to initialize the optimization in latent space. Depending on the variation of a motion primitive and the number of evaluated random samples, this can end up in a guess that is far away from the global minimum. Instead of using random sampling a more stable result, that requires less sample evaluations, can be achieved by taking knowledge about clusters in the latent space into account.

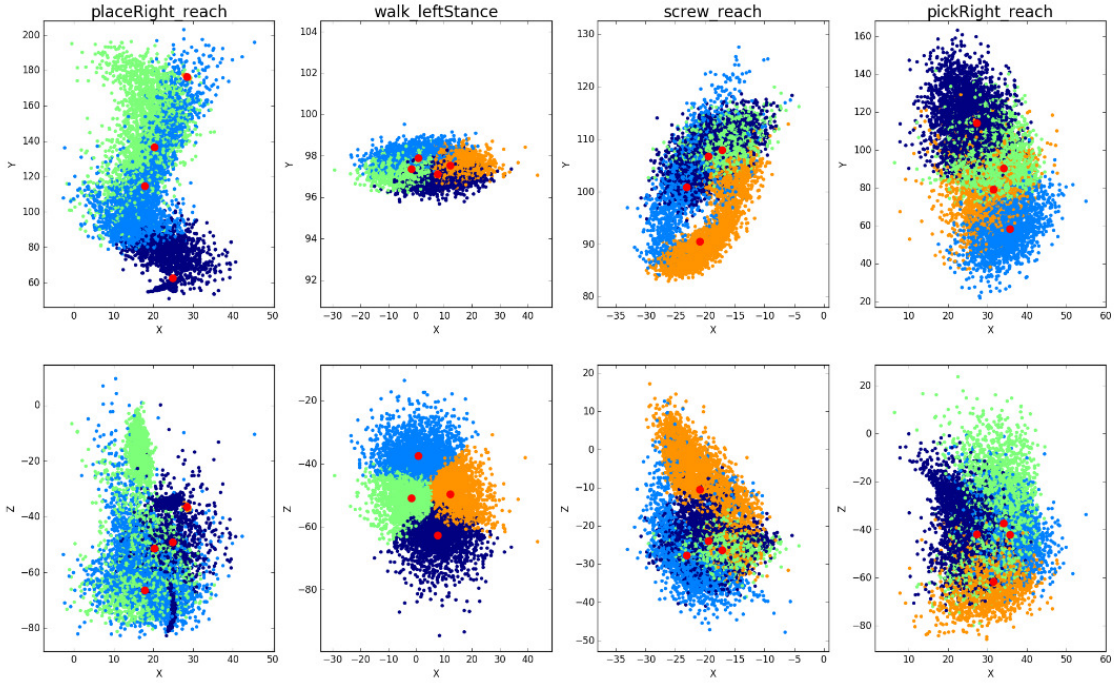

Figure 6: Clusters that were found in latent space visualized using X, Y and Z coordinates of joints of 10.000 random samples. Green, light blue, orange and dark blue represent labels and red dots represent the projected centers of the four latent space clusters. For each motion primitive, a different joint is used for the visualization. For pickRight and placeRight the right hand for walk the hip and for screw the left hand is visualized. The clustering is not perfect but segments the space into overlapping regions.

We therefore use a directed search inside of a space partitioning data structure constructed on the latent space to find an initial guess for the optimization. Experiments using the k-means algorithm for the clustering of

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

the latent space, showed that the search result can also come closer to the optimum than random sampling, which results in a reduced time needed for the optimization. However, the experiments were only successful for spatial constraints and only for constraints on joints that have high variation. For those joints clusters that were found in latent space resulting from the linear dimension reduction method can be mapped to clusters of the global positions and orientations of the joints. Figure 6 shows examples of clusters that were found in latent space visualized on joint positions sampled from the motion models.

3.5 Reachability of Constraints

In general the motion synthesis approach can reproduce variation in the motion capture data well. However, motion capturing and motion processing into motion primitive models is a time consuming task. Therefore, it is difficult to cover every possible scenario with training data and the available variation is limited.

The motion synthesis integrated into the best-fit pipeline does not work well in practical user experiments because the available variation in the motion model is too limited and the constraint generation does not take the range of the training data into account so constraints outside of the reachability of the training data are generated. This problem, however, was only considered very late in the project after tests using the M24 prototype. Daimler worked on the evaluation of the variation of the motion data. However, it was not possible to take it into account for the constraint generation due to limited resources very late in the project.

One approach to increase the reliability of reaching constraints, that was tested in the INTERACT project, is the combination of the data-driven motion synthesis with procedural motion editing based on inverse kinematics. Inverse kinematics is a standard technique used to calculate the joint parameters directly based on constraints defined on the joint positions and/or orientations. Tests using a basic unconstrained implementation show that altering the motion using procedural methods does not necessarily destroy the naturalness of the motion, as long as the difference of the motion sampled from the motion model and the synthetic pose parameters resulting from inverse kinematics is small and blending is used to create a smooth transition. However, the motion loses the stylistic information, if constraints that are far away from the training data need to be reached. Motion editing was added very late to the pipeline after M24, therefore joint boundaries are missing in the existing implementation, which would improve the result.

In general, a large variation in the training data is necessary to generate motions that reach constraints and have the quality that is required for the intended application area of ergonomic analysis. Therefore, an efficient, ideally automated, processing pipeline for training data is needed, to create models that cover the required variation for manual assembly tasks.

3.6 Collision Avoidance Integration

The original proposal for the integration of the separate Collision Avoidance module was to run the motion synthesis algorithm twice, once based on user constraints and then again with additional constraints that avoid collisions with the environment. A first version of the collision avoidance service became available after M18. Tests showed that this approach was problematic due to the not yet optimized implementation of the motion synthesis algorithm, which resulted in a long processing time. Furthermore, the constraints could only be provided on elementary action level. As a result the motion synthesis algorithm internally had to run multiple times in order to find the step in the graph walk that should be constrained based on the shortest distance and then modify this step. Additionally, due to the limited variation in the motion models, the

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

collision avoidance constraints could result in a changed path during trajectory following motions such as walk and carry. This can invalidate the following collision avoidance constraints for the same elementary action by making them unreachable. For example, by applying a collision avoidance constraint on the hand during a walk action in addition to the constraints on the hip joint can end up in changing the path, if the combination of constraints on the hand and hip joints lies outside of the range of the training data. If there was another collision in the following motion primitives, the corresponding collision avoidance constraints might not be reached anymore due to the changed path. Additionally, the modified motion can have new collisions.

In order to correctly handle collisions during motion synthesis, using the available variation of the training data, the motion synthesis needs to directly react to a collision so the motion has to be generated only once. After M24, the collision avoidance module was therefore re-integrated using a direct connection inside of the motion synthesis loop. However, the change in the architecture was implemented very late in the project and could not be tested and improved sufficiently which results in a long run time. Due to the long time required to run the implementation, further optimizations need to be done before it can be applied in user tests.

For future projects, the optimization of the runtime of the code should be given extra focus, when dealing with potentially computationally intensive tasks. Furthermore, the separation of work on closely coupled services such as the handling of collisions and the motion synthesis is problematic, because it requires close collaboration between partners in order to be successful. Based on experience with the speed-up of the core of the motion synthesis algorithmic structure, it could be already proved that significant further speed-up of the motion synthesis would be possible. However, this most likely would involve many or even multi-core hardware which would result in a complete redesign of the motion synthesis component. The creation of such an implementation would require deep expertise in many and/or multi core programming and can only be the last step of optimization in the design of a practical system because the effort for the implementation is very high and basically all algorithmic decisions need to be validated before.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

4 CONCLUSIONS AND FUTURE WORK

4.1 Summary and Conclusions

In INTERACT, the core of the statistical motion synthesis method presented by Min et al. [1] was re-engineered. It provides a generative, compact, semantic-aware representation for the large amount of motion capture data. Because the PhD thesis [8] with details of the method was not made available to the public by the author, the implementation required more effort than initially planned. The functional representation and the FPCA implementation had to be derived experimentally and might differ from the paper. Furthermore, the transition model as described in the paper could not be successfully integrated into the motion synthesis pipeline despite of a lot of effort due to issues reaching constraints while keep the naturalness of the motion.

A statistical human motion modeling pipeline was designed and implemented. Using this pipeline, a statistical motion database which contains 13 elementary actions was created: walking, two-hand carrying, single-hand carrying, two-hand picking, single-hand picking, two-hand placing, single-hand placing, retracting, sidestep, screwing, inserting, transferring and looking around, was constructed. The database is compact compared to the original motion capture data, requiring only 42MB for memory. However, the original motion capture data takes 4.2 GB for storage. Any number of motion variations with different styles can be efficiently generated by sampling the motion primitive model, and the quality of the generated motion clips is good in general according to an evaluation by users.

Given a list of constraints for different elementary actions, such as walk and pick, a constrained motion can be synthesized. The controlled motion synthesis algorithm breaks down the constraints to motion primitive level in order to generate a sequence of motion primitives. The latent model parameters of each motion primitive are first optimized individually to fit the constraints. For path following motions, multiple steps are additionally optimized together. The optimization of individual steps is accelerated using space partitioning data structures in latent space of the statistical motion models. Furthermore, motion editing is applied in order to increase the variation of the generated motion outside of the range of the training data.

Individual motion primitives can inherently generate realistic motions inside of the range of the training data. However, according to a user study that was conducted as part of the INTERACT project, the resulting motion of the synthesis algorithm was judged as not yet acceptable for the intended application of ergonomic analysis. There are issues regarding the motion quality when constraints outside of the training data need to be reached.

Although problems to work on remain, like for example that paths cannot be followed close enough or given constraints break the naturalness of motions, the motion synthesis approach produces high quality motions for a significant number of tasks.

4.2 Future Work

Currently, our motion primitive modeling pipeline is not fully automatic. Our rule-based frame recognition approach uses geometric information of motion data, however, some key frames are semantically similar but quite different in geometry, especially for complex actions like screw, transfer. So for extracting key frames from complex actions, manual work is still required. This manual work can be very time-consuming due to the large number of samples in the motion data required for modeling. In order to make our motion synthesis

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

prototype more practical for industry usage, deep learning based human motion recognition approaches could be a promising option to investigate (e.g. [11], [12], [13]). The goal would be to extract structurally and semantically similar key frames from recorded motion sequences, so the motion primitive construction can be fully automatic.

The statistical motion database constructed in INTERACT provides a compact representation for the variation in motion capture data. Any number of motion variations with different styles can be efficiently generated by simply sampling the motion primitive models we construct. However, the variation in motion data differs for different motion parameterization. We found that the variations of poses in Euclidean space closely matched visually observed variations in the motion data. This observation should be included in motion modeling. For future improvements of motion primitives, we propose to reduce the dimensionality of motion data by minimizing the reconstruction error in the Euclidean space of joint positions, instead of the feature space. This can be achieved by applying a scaled version of Functional PCA.

An important feature that is missing in the motion synthesis implementation is the Gaussian Process (GP) transition model described in the paper by Min et al. [1]. As a result the concatenated motion primitives are not very natural. To fix this problem further work investigating the transition model is necessary. One idea, described in [4], would be to investigate a frame wise (or in case of FPCA coefficient wise) optimization approach instead of optimizing the latent parameters once for the entire motion primitive. This way the GP could be used to predict smooth transitions and still have the entire variation of the motion primitive model to reach constraints without requiring too much training data. Another missing feature is the synthesis of parallel actions by different body parts. For this purpose, the extension of the motion primitive models into a hierarchical motion primitives models would need to be investigated. An interesting hierarchical dimension reduction approach for this purpose was proposed by Lawrence et al. [10]. Furthermore, in this context hand motion models need to be added. As basis for future work, the method presented by Zhao et al. [14] could be used. This is an extension of the motion model approach by Min et al. [1] with a physical model of the hand, which is used to handle the contact with objects correctly.

In order to better reach constraints, it might be useful to investigate a separate foot step planning algorithm, before the actual motion synthesis process. A similar data-driven two-step approach was developed by Agrawal et al. [9], which could be combined with motion primitives.

In order to handle contact with the environment correctly, a physical model of a skeleton in a simulation using forces could be used. This way collisions would be inherently detected and the motion would be inherently physically correct regarding balance and weight of manipulated objects. Furthermore, joint trajectories that avoid collisions outside of the variation of the training data could be generated using a particle filter based approach as proposed by [3]. In context of the physics integration another option for future work would be to evaluate a frame-based motion model, which can simplify the integration with a physics simulation because it reduces the differences in the update rate of the data driven motion synthesis and the physics engine. Furthermore this has the potential to increase the variation that can be generated from the available training data due to the larger possible combinations resulting from a concatenation of frames instead of motion primitives.

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

REFERENCES

[1] Min, J. & Chai, J. Motion Graphs++: A Compact Generative Model for Semantic Motion Analysis and Synthesis. ACM Trans. Graph., ACM, 2012, 31, 153:1-153:12

[2] Manns M, Mengel S, Mauer M. Experimental Effort of Data Driven Human Motion Simulation in Automotive Assembly. Procedia CIRP. 2016 Dec 31;44:114-9.

[3] Hämäläinen, P.; Eriksson, S.; Tanskanen, E.; Kyrki, V. & Lehtinen, J. Online Motion Synthesis Using Sequential Monte Carlo. ACM Trans. Graph., ACM, 2014, 33, 51:1-51:12

[4] Min J, Chen YL, Chai J. Interactive generation of human animation with deformable motion models. ACM Transactions on Graphics (TOG). 2009 Dec 15;29(1):9.

[5] Ramsay J, Silverman B. Functional Data Analysis. Springer; 2005.

[6] Du, H.; Manns, M.; Herrmann, E. & Fischer, K. Joint Angle Data Representation for Data Driven Human Motion Synthesis. Procedia CIRP, Elsevier, 2016, 41, 746-751

[7] Ormoneit D, Black MJ, Hastie T, Kjellström H. Representing cyclic human motion using functional analysis. Image and Vision Computing. 2005 Dec 12;23(14):1264-76.

[8] Min, Jianyuan. Intuitive Generation of Realistic Motions for Articulated Human Characters. Doctoral dissertation, Texas A&M University. 2013.

[9] Agrawal, S. & van de Panne, M. Task-based Locomotion ACM Transactions on Graphics (Proc. SIGGRAPH 2016), 2016, 35.

[10] Lawrence, N. D. & Moore, A. J. Hierarchical Gaussian process latent variable models. Proceedings of the 24th international conference on Machine learning, 2007, 481-488

[11] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011, November). Sequential deep learning for human action recognition. InInternational Workshop on Human Behavior Understanding (pp. 29-39). Springer Berlin Heidelberg.

[12] Du, Y., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1110-1118).

[13] Wang, C., Wang, Y., & Yuille, A. L. (2016). Mining 3d key-pose-motifs for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2639-2647).

[14] Zhao, W.; Zhang, J.; Min, J. & Chai, J. Robust realtime physics-based motion control for human grasping. ACM Transactions on Graphics (TOG), ACM, 2013, 32, 207

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

5 APPENDIX

elementary action	Motion primitive	Number of input samples	Length of low dimensional vector	AIC score (the lower the better)	Optimal number of Gaussians	Average log likelihood of training data (the higher the better)
walk	beginLeftStance	123	33	-7488.9862	3	44.9470
	beginRightStance	123	33	-7490.2250	3	44.9521
	leftStance	1366	42	-74088.5716	8	59.2667
	rightStance	1366	44	-78710.4901	8	63.1164
	endLeftStance	182	32	-12227.2735	6	51.9462
	endRightStance	131	31	-7998.1986	5	50.9114
	sidestepLeft	40	15	24.9662	5	16.5485
	sidestepRight	40	16	-58.9246	3	12.1865
	turnRightLeftStance	15	12	112.4796	2	8.3173
	turnLeftRightStance	15	12	112.4796	2	8.3173
Two-hand carry	beginLeftStance	12	13	28.4221	1	7.4824
	beginRightStance	12	13	28.4221	1	7.4824
	leftStance	280	28	-12309.8280	6	31.2684
	rightStance	280	27	-11971.1889	6	29.7114
	endLeftStance	84	14	-176.1567	3	5.3223
	endRightStance	84	14	-176.1567	3	5.3223
	sidestepLeft	43	13	856.3731	2	-5.0973
	sidestepRight	43	13	933.8753	5	1.0359
	turnRightLeftStance	19	13	156.8395	3	12.3989
	turnLeftRightStance	19	13	156.8395	3	12.3989

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

Left-hand carry	beginLeftStance	60	11	3.4619	2	2.5544
	leftStance	166	28	-8306.6932	2	30.2551
	rightStance	193	30	-9212.7247	2	29.0018
	endLeftStance	13	10	-22.1927	1	5.8535
	endRightStance	43	10	-433.8836	2	8.0916
	sidestepLeft	75	14	956.3415	5	1.6730
	sidestepRight	75	11	344.4085	2	-0.2293
	turnRightLeftStance	16	13	177.6495	2	7.5109
	turnLeftRightStance	63	23	241.6399	5	21.7318
Right-hand carry	beginRightStance	64	24	-2426.3558	1	24.0184
	leftStance	279	22	-2657.1714	2	10.3225
	rightStance	280	25	-12818.5436	4	27.9009
	endLeftStance	84	21	-2318.2464	3	22.8228
	endRightStance	55	18	-951.9676	3	18.9997
	sidestepLeft	75	23	-1639.1073	3	22.9140
	sidestepRight	75	23	-1880.6664	3	24.5244
	turnRightLeftStance	63	23	239.1885	5	21.8508
	turnLeftRIghtStance	16	13	177.6495	2	7.5109
Two-hand pick	reach	374	45	-22461.7969	7	50.5741
	retrieve	356	48	-22077.7663	7	56.2962
Left-hand pick	reach	29	15	63.9624	3	12.9316
	retrieve	31	16	182.0964	2	6.9016
Right-hand pick	reach	29	15	64.0411	3	12.9316
	retrieve	31	17	-329.1722	2	16.3092
Two-hand place	reach	182	37	-6204.8289	6	41.0388
	retrieve	180	37	-5516.8709	5	36.1137

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.

Left-hand place	reach	37	17	-348.1842	3	17.9834
	retrieve	38	18	-405.5442	2	15.3097
Right-hand place	reach	37	17	-306.8400	3	17.9843
	retrieve	38	18	-405.5442	2	15.3097
LookAt	LookAt	88	15	129.0858	10	10.8541
Transfer	transfer	82	22	-821.5554	5	21.2225
Two-hand insert	reach	54	17	-221.9901	5	17.8702
	retrieve	55	17	-61.6436	6	16.0876
Right-hand insert	reach	12	12	123.5129	2	9.9369
	retrieve	11	8	104.6566	3	7.4246
Left-hand insert	reach	12	12	123.5129	2	9.9369
	retrieve	11	8	104.6566	3	7.4246
Left-hand screw	reach	231	25	-98.5422	9	13.6356
	retrieve	229	29	-948.2564	8	18.3594
	transfer	79	20	393.2646	6	15.1911
Right-hand screw	reach	231	25	-185.0918	10	15.2924
	retrieve	229	29	-1070.6439	10	22.7935
	transfer	79	20	356.7561	6	15.2736

Table 1: Evaluation of Motion Primitive Models

The INTERACT project (611007) is co-funded by the European Commission under the 7th Framework Programme.