Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 72 (2018) 398–403

www.elsevier.com/locate/procedia

51st CIRP Conference on Manufacturing Systems

A Motion Reuse Framework for Accelerated Simulation of Manual Assembly Processes

Felix Gaisbauer[a,b], Jannes Lehwald[a], Philipp Agethen[a,b], Michael Otto[a,b], Enrico Rukzio[b]

[a] Daimler AG, Wilhelm-Runge-Straße 11, 89081 Ulm, Germany
[b] Ulm University, James-Frank-Ring, 89081 Ulm, Germany

* Corresponding author. Tel.: +49-505-24-50; fax: +49-711-3052156294. E-mail address: felix.gaisbauer@daimler.com

Abstract

The simulation of digital humans and installation paths within virtual environments is a key-technology for production planning departments to plan and assess manual assembly processes. Depending on the scenario, recent path planning algorithms need several minutes up to hours to compute a collision free installation path. However, the expensively computed data-sets are currently not further utilized in subsequent simulations, which induces unnecessary computational effort. Within this paper a novel concept is introduced, which enables to reuse paths stemming from previous simulation runs, thus significantly reducing computation times while ensuring identical path quality. A comprehensive evaluation underlines the technical performance of the proposed approach. With this method we show that simulations in virtual environments could be accelerated by a factor of 50 using an exemplary K-Nearest Neighbor interpolation approach.

Keywords: Motion Reuse; Motion Blending; Path Planning; Human Simulation; Assembly Path Planning; DHM

1. Introduction

With growing product portfolios and ongoing mass customization within the automotive industry, it is increasingly hard to manually assess the production processes of each variant. To nevertheless cope with these challenges, digital human simulations are widely used. In production industry, virtual scenes are used for ergonomic assessments, process quality verification and product optimizations [1]. Therefore motions have to be simulated multiple times with slight adaptations of the scene, where the results may be similar but not equal. Moreover, since the motion computation oftentimes takes place in highly constrained environments (e.g. engine compartment) the computational costs are especially expensive and may result in several minutes of calculation time. Even though these calculations are highly time-consuming, existing approaches often ignore pre-knowledge of previous simulation runs during these computations, thus rejecting potentially useful information. Avoiding the re-computation of those expensive motions appears as a promising measure to further improve the computation time and reactiveness.

Within this paper a novel framework concept is proposed in which motion trajectories of objects and digital human simulations are continuously stored and reused (see Figure 1). The novel concept can be utilized for learning agents and environments which can highly accelerate the computation processes,

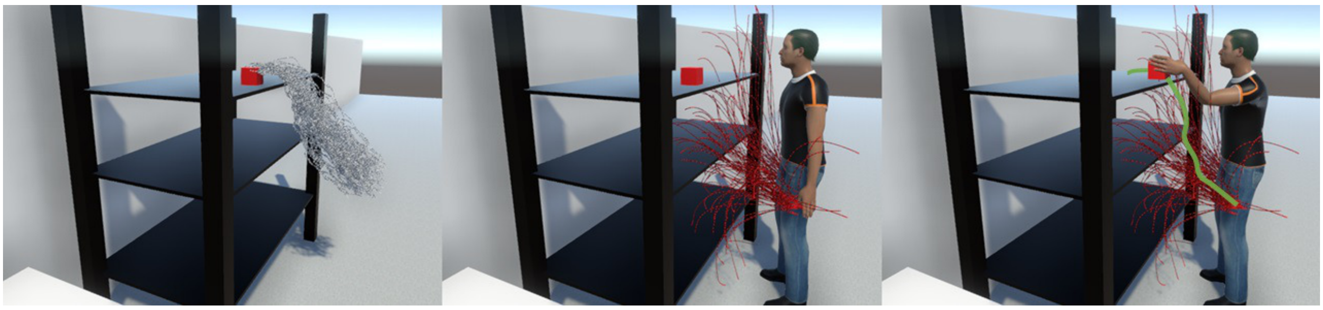

Fig. 1. Motion databases attached to a scene object (left) and a digital human (middle). These motion databases are used to accelerate and generate human motions within virtual environments (right).

10.1016/j.procir.2018.03.282

while the simulation results are improved and enriched during each simulation run. It can be used with any kind of motion planning algorithms and acts independently of the specific interpolation methods.

The remainder of the paper is structured as follows: First, the state of art in the context of digital human simulation and motion modeling is reviewed. Second, the novel concept is described enabling to reuse already computed motions. Furthermore, an exemplary implementation for human pick and place motions is presented, outlining the practical applicability of the concept. Consecutively, the technical performance of the proposed method is verified in an evaluation considering a pick up motion in a collision afflicted environment. Finally, a conclusion and an outlook on further optimization potentials are given.

2. Related Work

Digital human simulations in virtual environments are increasingly important for various industries. There are numerous specialized solutions for ergonomic aspects, buildability [2–4] as well as static and dynamic posture analysis [5]. Others, such as Smartbody [6] focus on interaction or gaming. The available tools differ in terms of utilized motion synthesis methods, ranging from fully analytical approaches, like Santos [7] to data driven probabilistic approaches [8] and methods based on artificial intelligence [9]. Even though there is a wide variety of different systems available, all of these approaches commonly rely on static, precomputed motion capture data or compute the paths on the fly without using the previously obtained knowledge. Trying to simulate a full assembly line with multiple agents, the motions have to be recomputed for each agent, leading to expensive computations.

Realistic and collision free human motions can be either obtained by using pre-recorded motion capture data or can be artificially computed using motion planning algorithms like [10]. Moreover, if the recorded motion capture data is collision afflicted, or does not fulfill required constraints, the data is oftentimes further adjusted using path planning algorithms and trajectory refinement approaches as described in [11,12]. Even though the motions are adjusted within these approaches, the data is not further stored for further calculations. Consequently, subsequent simulation runs do not benefit from the previously generated motions. Thus, similar and identical motions will be always re-synthesized. Beside the motion generation based on path planning, motion blending is a widely spread technique which is used to generate fitting motions based on a reference motion database. According to Feng et al. [13] the blending techniques can be clustered into Barycentric interpolation, Radial basis functions (RBF), K-Nearest Neighbor (KNN) interpolation and Inverse Blending. Additionally, motions can be also obtained using geostatistical interpolation approaches [14]. Within this paper, the KNN motion blending approach has been used to enrich and interpolate the datasets.

Lim et al. [15] proposed a concept to store and reuse motion capture data for gathering realistic motions. Whereas this approach only considers the offline storage of fine grained motion capture data for so called motion primitives, the proposed approach within this work, makes no assumptions about the motion data and can be seen as a more generic method focusing on reutilization and online learning. Geng and Yu presented a generic overview of motion reuse techniques [16]. While the work by Geng and Yu gives an overall overview of motion reuse operations, within this paper a specific approach is presented. Moreover, Kallman and Thalmann [17] introduced the concept of smart objects, in which objects are annotated with additional meta informations. However, the concept does not cover the storage of motion paths. Thus the proposed work within this paper can be seen as an extension of this concept to cover and accelerate motion planning in virtual environments. To the best of our knowledge, the proposed motion reuse concept by continuously storing and integrating motions, as proposed within in this paper, is not in scope of research so far.

3. Approach

The main objective of the proposed approach is to accelerate and support the simulation process and motion synthesis by reusing existing motion data stored within so-called motion databases which are directly attached to the subjects (e.g. digital human or scene object). Figure 2 depicts an overview of the process flow introduced by the novel approach. Within the framework, the available motion data is reused as often as possible by accessing the databases and storing every artificially computed trajectories at a suitable motion storage. The proposed framework can be used independently of the respective subject, thus allowing to store motion trajectories related to humans or scene-objects (e.g. assembly part) separately. Based on this concept, the attached motions paths of the respective objects can be utilized in subsequent simulation runs, as well as within varying scenarios and environments. The following section will give an in-depth overview of the different components:

Motion Database. One of the key concepts of the proposed framework is the modeling and storing of arbitrary motions within so-called motion databases. These motion databases can contain motions of different kinds, either being recorded with motion capture techniques, or artificially generated by using motion planning algorithms. Independent of the way the motion is gathered, the individual databases should contain homogeneous motion (e.g. grasp motion within one database and walk motions within another database). Furthermore, the databases can be annotated according to the current environment or due to various properties (e.g. size of human, velocities).

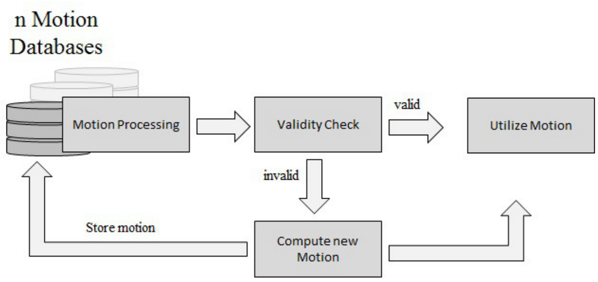

Fig. 2. Illustration of the proposed motion reuse framework. The motion data is stored within so-called motion databases. These datasets can be further processed using interpolation approaches or methods of artificial intelligence. After having checked the data for validity, the motions are either directly utilized or new motions have to be computed. In case a re-computation is necessary, the generated motions are then further stored in the suitable database, being utilizable within subsequent simulations.

An essential aspect of the proposed concept is to directly assign the motion databases to the respective scene objects or to a digital human model (DHM). For instance, by attaching all setup paths of a specific assembly part to the object itself, the path information might be reused in different scenarios (e.g. varying orientation/scene) or subsequent simulations. Consequently the scene objects and DHM act as smart objects containing various motions without relying on the specific environment and external information.

Motion Processing. Since the motion databases serve as a storage for arbitrary motions, the datasets have to be extracted and further processed in order to be re-utilized. The motions can be either directly fetched by performing a database search, be further processed using interpolation approaches like [18,19] or methods of artificial intelligence. Whereas a simple search only returns the identical trajectories, the latter can generate novel motions out of the existing database, which can cover certain areas which have not been simulated in before. For instance, utilizing a KNN blending approach, as described in [13], novel motions laying in between existing datasets can be efficiently generated. Therefore these methods serve as a promising approach to further increase the probability of being able to utilize already present motions and avoid recomputations.

Motion Validity Check. After having obtained a motion from the existing database, the motion might be checked respective to specified constraints. These constraints can be related to collision affliction, path shape, spatio-temporal parameters or other properties. If the obtained motion fulfills the specified constraints, the motion can be directly utilized and further applied. This step does not have to be necessarily in a separate stage. For instance, the constraints could be directly considered by using inverse blending approaches like [20] within the motion processing stage. These blending approaches set the interpolation weights in order to minimize the error considering the respective constraints.

Furthermore, instead of utilizing a binary acceptance/rejection scheme, in case of unsatisfied constraints, motion trajectories can also be partially reused. Since oftentimes Rapidly Exploring Random Tree (RRT) approaches are utilized for path planning, which rely on sampling configurations within the search space, collision free states gathered from existing motions could be directly inserted into the search space. In this case the search can be even accelerated if no path which satisfies all constraints is available in before.

Re-computation of novel Motions. In case the gathered motion does not fulfill the desired constraints (e.g. collision affliction), the database and motion processing is not able to produce the desired paths given the existing datasets. Thus the motion database must be further enriched with newly computed motions to produce motion trajectories being able of satisfying the constraints. In this case, the motions are recomputed using an arbitrary motion planning algorithm (e.g. RRT or Probabilistic Roadmap). Since this part is the most time consuming aspect of the simulation, it is tried to avoid the re-computation as often as possible. As mentioned within the previous subsection, in case the motions do not satisfy the constraints entirely, they can be nevertheless utilized to initialize and accelerate the motion planning process. After having computed a new trajectory, satisfying the desired constraints, the motion is further stored in the suitable database, thus enriching the existing datasets. This process is performed during the simulation execution and can be principally integrated in arbitrary simulation environments to continuously enrich the motion databases.

4. Accelerated Human Assembly Simulation: An Implementation

Whereas in the previous section, the generic principle of the framework is introduced, within this section a specific implementation for accelerating human assembly simulations is presented. In general, human assembly simulation predominantly consists of tasks in which a digital human interacts with virtual objects. Mostly, an assembly trajectory is computed for a given object first and the digital human tries to follow the gathered trajectory in a second step, considering the intrinsic constraints [21]. Since the proposed motion reuse concept is applicable for different kind of objects, the implementation distinguishes between motions of digital humans and assembly paths of the objects. Therefore the motions of the digital human and assembly parts are represented using different motion databases.

Human Motion Representation. For representing the human motions, a motion database has been utilized which contains several reach motions of the left and right hands, recorded using a motion capture system. For interpolating novel motions from the motion capture data, the KNN interpolation approach [13] is used. Since the accuracy of the KNN interpolation strongly relies on the density of the reference datasets the method proposed by Kovar and Gleicher [22] is further applied.

Within the utilized blending space, six degree of freedom motions of the human hands are stored, whereas the motions are arranged in the local coordinate system of the agent. As DHM, the in-build Mecanim animation system of the Unity 3D engine has been used. By utilizing inverse kinematics functionalities, the animation system can generate realistic full-body motions, while only relying on the relevant end-effector trajectories and transformations. Figure 1 (middle) visualizes the pre-recorded motion capture data for a reach motion of the virtual human. For determining a fitting motion, the end-position and orientation of the target hand poses are used to derive a suitable motion from the blending space. The gathered motion is further evaluated concerning collision affliction and Euclidean distance error to the target position. If the Euclidean distance is larger than a specified threshold, or the trajectory is collision afflicted, a new path has to be computed. This path is further annotated and integrated into the motion database, being utilizable within subsequent simulations.

Assembly Part Trajectory Representation. For representing the motion database of the assembly part the identical KNN blending approach is used. The motion database is directly attached to the respective scene-object, leading to a distributed system of smart objects which can be reused in subsequent scenarios. For the computation of a novel motion, initially the existing motion database of the specific object is used to derive a fitting motion. If the Euclidean Distance of the end-position of the gathered motion and the target destination is larger than a threshold value, or the trajectory is collision afflicted, a novel trajectory is computed. The trajectories for describing the motions of the object are computed based on an RRT implementation. Figure 1 (left) visualizes motion trajectories which are stored in the motion database of an exemplary scene object.

Realizing Pick & Place Motions. Since the previously explained motion databases only contain partial trajectories, they must be linked together to accomplish specific assembly tasks. The novel approach has been implemented for pick & place scenarios, in which pick-up and put down motions with arbitrary objects are performed. To adapt the motions to the proposed framework, they are separated into two different phases: namely reach and return. In case of a pick-up task, the reach motion is determined by the DHM, whereas the return motion is computed, utilizing the attached motion database of the object to be picked up. For modeling the put-down motion it is done vice versa.

For modeling a pick-up task, first the target hand position for grasping the object must be determined. Subsequently, this position is used to derive a reach motion from the agent's motion database. Having determined a suitable reach path, a return motion must be computed afterwards. Therefore, initially the target transformation of the scene object is determined. This target transformation corresponds to the carry position and rotation in which the scene object is located after the return motion is finished. Subsequently, a new path from the current position of the object to the carry destination is obtained by using the motion database of the scene object. A put-down task is performed in an analogous way, whereas the reach destination corresponds to the place position and the return destination describes the hand position after the task is finished. In this case, the scene object is responsible for determining the reach motion while the DHM computes the return motion. Since the motion computations of the DHM and scene object can be performed independently, the performance can be further increased by a parallel computation of multiple paths simultaneously. Having computed the corresponding reach and return paths, the trajectories are traversed according to a fixed update timespan for each frame. Within each update step the inverse kinematics of the human avatar is recomputed according to the current pose of the obtained path.

5. Evaluation

To validate the proposed concept, the generation of collision-free motions within a collision-afflicted environment is analyzed. The performance is examined respective to the computation times and path lengths for varying region sizes, representing the tolerance of the start position. To analyze the long term behavior (e.g. 24 hour assembly lines), each experiment is simulated for 5000 runs. Hereby, the novel approach is compared to a default motion planning implementation in which the motions are recomputed within every simulation run using a RRT algorithm.

Experimental Design. During the experiment, a virtual human avatar is placed in front of a target object which is occluded by an obstacle. The center of mass position of the human is randomly adjusted within a spherical region S (see Figure 3). Moreover, the radius of the this spherical region S is adjusted in order to analyze the performance of the novel approach. For each sampled agent position within S, a collision-free path from the specified target object to the hand of the virtual human has to be computed. While the position of the scene-object is fixed, the target position is varying since the agent's position is randomly adjusted within S.

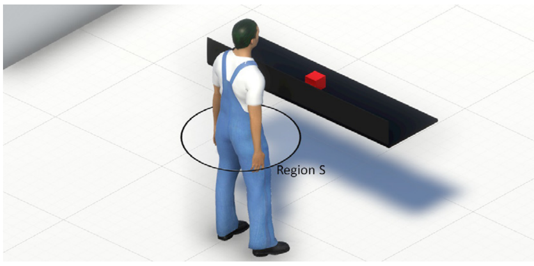

Fig. 3. Overview of the experimental design within the evaluation. A collision free path has to be computed from the red object to the left hand of the agent. Moreover, the position of the agent is adjusted within a spherical region S.

The experiment has been carried out with the implementation explained in section 4. The motion database has been modeled with a KNN motion blending approach. In order to improve the sampling-coverage of the blending space, five artificially generated motions are added for each inserted motion path. To align the motion trajectories an Euclidean nearest neighbor approach has been utilized. The translation accuracy of the utilized RRT algorithm has been set to 0.01 m. For improving the greediness of the algorithm, the goal bias heuristic [23] has been used (value 0.1). The bounds of the search region of the RRT algorithm have been set to 2.00 x 2.00 x 2.00 m, while the algorithm is stopped after the first valid solution has been found.

The performance of the novel approach has been compared to the above mentioned RRT algorithm in terms of path length and computation time and has been carried out for different sizes of the spherical region S (0.05 m, 0.10 m, 0.25 m, 0.50 m, 1.00 m) in which the starting position of the DHM is determined. The position is obtained by means of uniform sampling within this spherical region. For each size of S, a separate simulation and motion database has been used, while 5000 simulation runs were computed. The motions have been computed for the left hand of the agent only. For the path computation the collision geometry of the target object (size 0.10 x 0.10 x 0.10 m) is explicitly considered. The target object is placed on a plate with dimensions of 1.50 x 0.02 x 0.40 m, whereas it is occluded by a vertically aligned obstacle with a respective size of 1.50 x 0.20 x 0.02 m. Within the novel approach, the motion trajectories are gathered from the motion database using the KNN approach. If the obtained motion offset is below 0.01 m Euclidean distance it is directly used. Otherwise a new trajectory is computed using the RRT with the above-mentioned parametrization. The default reference implementation recomputes the same path within each simulation run, utilizing the identical RRT implementation and parameters.

Results. After having computed in total 250 000 motion trajectories (5000 per scenario), it can be seen, that the novel approach predominantly results in lower computation times compared to the default implementation, while fulfilling the accuracy requirements of 0.01 m. Figure 4 visualizes the computation times for the specific region sizes of S depending on the number of calculated trajectories. The computation time is set relative to the mean calculation time of the reference implementation. Note that the reference RRT-implementation always results in a straight line slightly oscillating around 1.0.

In general, it can be seen that the novel approach reduces the computation times for all considered region sizes ranging from 0.05 m to 1.00 m. The calculation times are decreasing with a growing amount of calculated trajectories, however the gradient of the individual curves varies strongly. For the region size 0.05 m, the computation time can be instantly reduced and results in a stable plateau only requiring 2% of the time of the naive reference implementation. Analogously for the region size of 0.10 m, the curve is also instantly decreasing, however it requires more calculated trajectories to achieve asymptotic behavior, in which 2.5% of the computation time of the reference implementation is required. Considering a region size of 0.25m, the curve has a similar shape compared to the previous scenarios. However, the gradient is flatter and a stable plateau is reached later (10% of computation time). For the region sizes of 0.50 m and 1.00 m the curve shape transforms to a more linear shaped curve. In both cases, a stable plateau is not reached and the computation times are around 25% (0.50 m) and 80% (1.00 m) after 5000 computations. Note that the trend-lines in Figure 4 have been generated using a moving average computation to illustrate the overall trend. In particular, asymptotic behavior is reached earlier for the smaller region sizes, whereas in larger region sizes more oscillation can be registered.

Figure 5 (top) visualizes the computation times for all examined sizes S. The mean and median computation times of the novel approach are always below the reference implementation. Furthermore, the delta between the novel and default approach is decreasing with growing region size. For $S = 0.05\ m$ the mean calculation time can be reduced from 0.091 $s \pm 0.028\ s$ to 0.003 $s \pm 0.007\ s$ which corresponds to an improvement of 97% regarding the mean value. At a size of 0.10 m, the times can be improved from 0.099 $s \pm 0.027\ s$ to 0.005 $s \pm 0.015\ s$ (mean value improvement $\approx$ 95%). For $S = 0.25\ m$ an improvement from 0.091 $s \pm 0.035$ to 0.017 $s \pm 0.035$ can be registered ($\approx$ 81%), while for $S = 0.50\ m$ the mean times can be more than bisected from 0.097 to 0.043 s ($\approx$ 56%). For a large region size of 1.00 m the mean calculation time can be reduced from 0.123 $s \pm 0.097$ s to 0.096 ± 0.096 ($\approx$ 22%). While for the region sizes 0.05 m, 0.10 m and 0.25m, all computations times are below the reference implementation, for 0.50 and 1.00 m, occasionally there are outliers requiring more calculation time.

After having analyzed the performance of the novel approach in terms of calculation time, the quality of the gathered paths has to be analyzed considering the obtained path lengths. Figure 5 (bottom) shows the results respective to the

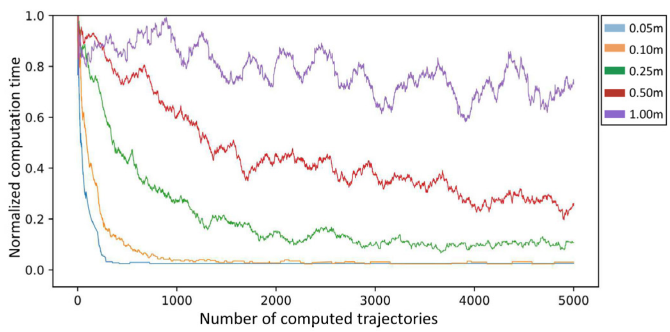

Fig. 4. Plots displaying the required computation time considering the amount of calculated motions for the five examined region sizes 0.05 m, 0.10 m, 0.25 m, 0.50 m and 1.00 m. The computation time is set relative to the mean calculation time of the respective RRT implementation for the specific region size S.

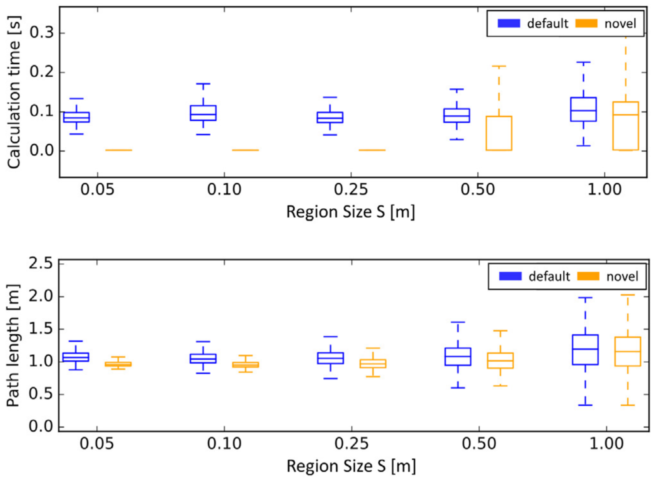

Fig. 5. Top: Boxplots of the measured computation times for gathering a motion depending on the region size S. Whereas the blue boxes represent the default implementation in which a trajectory is recomputed in every simulation run, the orange boxes represent the novel approach. The individual boxes contain the median, 5-, 25-, 75- and 95-percentile. Bottom: Visualization of the obtained path lengths for the varying region sizes of S.

path length of the obtained paths. In general, the novel approach produces similar but slightly shorter trajectories compared to the reference implementation. Whereas the difference within the smaller region sizes is higher (reduction of mean length for $S = 0.05\ m \approx 10\%$; $S = 0.10\ m \approx 9\%$; $S = 0.25\ m \approx 8\%$), it decreases for the larger region sizes ($S = 0.50\ m \approx 5\%$; $S = 1.00\ m \approx 2\%$).

Discussion. As indicated by Figure 5, the novel approach predominantly requires less computation times, while generating similar and collision-free paths. Whereas for smaller region sizes the mean computation time can be accelerated by a factor of 50, the improvement decreases for larger region sizes. One reason therefore is, that within smaller sizes of S, less samples are required to cover the full space. Consequently, the probability is higher that a fitting motion can be obtained, compared to larger region sizes in which drastically more samples are required. Moreover, the outliers occurring at region sizes of 0.50 m and 1.00 m can be interpreted as a direct result of this decreased probability to interpolate a suitable path from the available data. Since the blending spaces in these cases contain a lot of samples, the time consumption for performing the KNN search and inserting new samples is additionally increasing. Thus, these outliers were generated in a later stage where already various samples were contained in the blending-space and no valid path could be extracted.

Evaluating the curvature of the examined region sizes (Figure 4), different curve-shapes can be determined. Again, the reason for this behavior is the amount of required samples to cover the respective spaces and the additional computational effort in large spaces. Thus within a small space the approach can directly accelerate the computation time, while in a large region significantly more computations are required.

Additionally, the novel approach can reduce the mean trajectory lengths. Since the RRT algorithm might produce unrealistic and jittery paths, the linear interpolation within the blending space can be denoted as a possible reason. However, the overall mean path length is increasing from region size 0.05 m to

1.00 m. As reason for this behavior it can be denoted, that with increasing region size, more paths have to be computed to cover the full space. Consequently less interpolation is applied and the original paths are inserted, whereas the ratio of generated paths to interpolated paths is increasing.

Limitations and possible Extensions. In summary, it can be seen that the novel approach can improve the overall performance regarding computation time and path length. The improvement differs depending on the region size. Whereas the performance gain for smaller regions is high (up to factor 50), it shrinks for larger sizes (factor 1.25). Consequently, the presented implementation can be best utilized with small region sizes from 0.05 m to 0.25 m. Transferring these findings to the use-case domains, the proposed approach is best suited for use-cases in which the geometrical variance of the processes is low (e.g. highly repetitive tasks within assembly). Note, that within the evaluation the starting positions have been varied within each simulation run. If performing the identical motions multiple times, the proposed approach can instantly reuse the already present motions.

In the experiment, the computation was stopped after the first valid solution was found. In principle, the approach can be also utilized to compute paths fulfilling specific constraints and criteria such as optimality. By applying motion interpolation using a motion database, the generated results might loose these properties. Nevertheless, these motions could be used to initialize the path planning algorithm, which again accelerates the overall planning. In this case, the overall computation time increases for both, the naive reference implementation and the novel approach. In general, it can be denoted that with growing amount of constraints and criteria, the need for re-computation increases, since the motion interpolation increasingly fails to produce results satisfying all constraints. However, it is expected, that the novel approach can still provide vast computational benefits over a naive implementation, since the already present informations can be considered by the path planner.

For the evaluation, a KNN interpolation approach has been used and the trajectories have been rejected if not satisfying the constraints. By applying inverse blending techniques, in which constraints like collision affliction can be considered during the motion interpolation stage, it is expected that the need for recomputation decreases. Moreover, the blending data could be additionally enriched by using various sub-trajectories of the existing motions, ultimately improving performance.

6. Conclusion and Outlook

In this paper a motion reuse framework has been proposed which allows to speed up human motion generation in collision afflicted environments. The evaluation indicates that the approach can improve the overall simulation performance in terms of computational time and path length. Since the performance improvements decrease with growing translational variance, the proposed concept can be best utilized for highly repetitive tasks (e.g. tasks within automotive assembly). Within this paper only the computational performance was compared, however within future work the accuracy and parametrization of the RRT and the reuse of motion data will be further examined. Additionally, the performance regarding the computation of optimal and highly constrained motion paths will be analyzed in subsequent studies.

References

[1] Otto, M., Prieur, M., Agethen, P., Rukzio, E.. Dual reality for production verification workshops: a comprehensive set of virtual methods. Procedia CIRP 2016;44:38–43.

[2] Delmia website. 2017. URL: www.3ds.com; access date: 2018-02-11.

[3] Hoegberg, D., Hanson, L., Bohlin, R., Carlson, J.S.. Creating and shaping the DHM tool IMMA for ergonomic product and production design. International Journal of the Digital Human 2016;1(2):132.

[4] Fritzsche, L., Jendrusch, R., Leidholdt, W., Bauer, S., Jckel, T., Pirger, A.. Introducing ema (Editor for Manual Work Activities)-A New Tool for Enhancing Accuracy and Efficiency of Human Simulations in Digital Production Planning. In: HCI (17). Springer; 2011, p. 272–281.

[5] HS Group - Produkte Mobility - RAMSIS Automotive website. 2017. URL: www.human-solutions.com; access date: 2017-10-18.

[6] Thiebaux, M., Marsella, S., Marshall, A.N., Kallmann, M.. Smartbody: Behavior realization for embodied conversational agents. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems; 2008, p. 151–158.

[7] Santos digital human website. 2017. URL: www.santoshumaninc.com; access date: 2017-12-20.

[8] Min, J., Chai, J.. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. ACM Transactions on Graphics (TOG) 2012;31(6):153.

[9] Holden, D., Saito, J., Komura, T.. A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) 2016;35(4):138.

[10] Kagami, S., Kuffner, J., Nishiwaki, K., Okada, K., Inaba, M., Inoue, H.. Humanoid arm motion planning using stereo vision and RRT search. In: Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on; vol. 3. IEEE; 2003, p. 2167–2172.

[11] Pan, J., Zhang, L., Lin, M.C., Manocha, D.. A hybrid approach for simulating human motion in constrained environments. Computer Animation and Virtual Worlds 2010;21(3-4):137–149.

[12] Yamane, K., Kuffner, J.J., Hodgins, J.K.. Synthesizing animations of human manipulation tasks. In: ACM Transactions on Graphics (TOG); vol. 23. ACM; 2004, p. 532–539.

[13] Feng, A., Huang, Y., Kallmann, M., Shapiro, A.. An analysis of motion blending techniques. In: International Conference on Motion in Games. Springer; 2012, p. 232–243.

[14] Mukai, T., Kuriyama, S.. Geostatistical motion interpolation. In: ACM Transactions on Graphics (TOG); vol. 24. ACM; 2005, p. 1062–1070.

[15] Lim, B., Ra, S., Park, F.C.. Movement primitives, principal component analysis, and the efficient generation of natural motions. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE; 2005, p. 4630–4635.

[16] Geng, W., Yu, G.. Reuse of motion capture data in animation: a review. Computational Science and Its ApplicationsICCSA 2003 2003;:982–982.

[17] Kallmann, M., Thalmann, D.. Modeling behaviors of interactive objects for real-time virtual environments. Journal of Visual Languages & Computing 2002;13(2):177–195.

[18] Rose, C., Cohen, M.F., Bodenheimer, B.. Verbs and adverbs: Multidimensional motion interpolation. IEEE Computer Graphics and Applications 1998;18(5):32–40.

[19] Ge, Q.J., Kang, D.. Motion interpolation with G2 composite Bezier motions. Journal of Mechanical Design 1995;117(4):520–525.

[20] Huang, Y., Kallmann, M.. Motion Parameterization with Inverse Blending. In: MIG. Springer; 2010, p. 242–253.

[21] Hanson, L., Hgberg, D., Carlson, J.S., Bohlin, R., Brolin, E., Delfs, N., et al. IMMAIntelligently moving manikins in automotive applications. In: Third International Summit on Human Simulation (ISHS2014). 2014,.

[22] Kovar, L., Gleicher, M.. Automated extraction and parameterization of motions in large data sets. In: ACM Transactions on Graphics (TOG); vol. 23. ACM; 2004, p. 559–568.

[23] Akgun, B., Stilman, M.. Sampling heuristics for optimal motion planning in high dimensions. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE; 2011, p. 2640–2645.