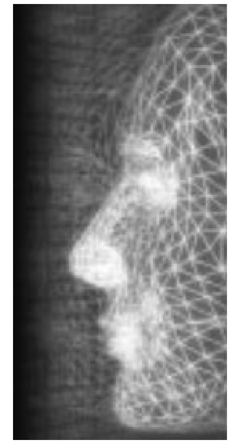
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2008; **19**: 445–454
Published online 7 August 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/cav.252

Grasp synthesis from low-dimensional probabilistic grasp models



By Heni Ben Amor, Guido Heumer, Bernhard Jung and Arnd Vitzthum*

We propose a novel data-driven animation method for the synthesis of natural looking human grasping. Motion data captured from human grasp actions is used to train a probabilistic model of the human grasp space. This model greatly reduces the high number of degrees of freedom of the human hand to a few dimensions in a continuous grasp space. The low dimensionality of the grasp space in turn allows for efficient optimization when synthesizing grasps for arbitrary objects. The method requires only a short training phase with no need for preprocessing of graphical objects for which grasps are to be synthesized.
Copyright © 2008 John Wiley & Sons, Ltd.

Received: 24 June 2008; Accepted: 25 June 2008

KEY WORDS: grasp synthesis; principal component analysis; Gaussian mixture models

Introduction

Realistic and natural synthesis of human-like grasping is still a challenging task when animating virtual characters in movies, games, and virtual environments. What makes the animation of grasping particularly difficult is that a high number of interdependent degrees of freedom in the hand must be controlled. Furthermore, at the same time, the contacts of the hand with the object surface must be optimized to ensure the stability of the grasp. Pure motion capture does not solve the problem because it provides no support for the animator in the task of closely fitting the hand shape to new objects, that is, the retargeting problem.[1] To cope with the retargeting problem, model-driven approaches are typically applied for automatic grasp synthesis which operate by optimizing the contacts between the hand and the object; however, such model-driven approaches often result in unnatural hand shapes as they do not account for constraints of human finger motion. Therefore, combined data-driven and contact-optimizing methods are called for in order to generate both natural looking and physically plausible grasping animations.

In this paper, we propose a data-driven method for the efficient synthesis of natural looking hand shapes for grasping animations. A first step is to collect a wide range of possible hand shapes using motion capture techniques. Then, principal component analysis (PCA) is applied to create a low-dimensional grasp space. It was already reported by Santello and colleagues that the first two and three principal components (PCs) account for more than 80% and resp. 87% of the variance in hand posture.[2] The result of applying PCA to the recorded hand shapes is conceptualized as continuous space which is by many orders of magnitude smaller than the space spanned by the original degrees of freedom. Furthermore, a Gaussian mixture model (GMM) is constructed to constrain the PCA-based grasp space to plausible hand configurations. The resulting probabilistic low-dimensional grasp space is small enough to be searched through at interactive rates using optimization strategies.

While a main motivation of the presented approach is its efficiency, it was also designed with the intention of providing the animator with a high degree of control over the type of the synthesized grasp. Generally, there are often many ways in which an object can be grasped. For example, a cup would be grasped in a very different way when drinking from it as opposed to placing it in the dishwasher. The different ways of grasping objects is a thoroughly researched topic and different grasp taxonomies are presented, for example, in References [3,4]. Accordingly, it is possible for the proposed method to synthesize grasps that optimize the grasp stability but may not meet the animator's original intent. Rather than

*Correspondence to: H. Ben Amor, VR and Multimedia Group, Institute of Informatics, TU Bergakademie Freiberg, Bernhard-von-Cotta Str. 2, 09599 Freiberg, Germany.
E-mail: amor@tu-freiberg.de

Copyright © 2008 John Wiley & Sons, Ltd.

attempting to build a complex inference mechanism for reasoning about a grasp's intent, our solution to the problem of synthesizing specific ways for grapsing an object is to let the animator specify a preferred region of where to grasp the object as well as a concrete grasp type. From an algorithmic point of view, the synthesis of a specific grasp type can be achieved in our method by simply training a specific model from motion-captured hand shapes conforming to this grasp type.

Related Work

Generally, the approaches to realistic virtual grasping can be divided along two criteria axes. While *geometry-based* approaches, such as Reference [5] or [6], solely rely on graphical geometry for their calculations, other approaches use an underlying physical simulation of the human hand and can be referred to as *physics-based* or *simulation-based*, for example, References [7–9]. Another criterion is whether data of grasp examples is used for the grasp synthesis process or whether the grasps are the product of a set of controller functions or other algorithmic models. The former type of approach is often called *data-driven* or *example-based*, for example, Reference [10] whereas the latter one is called *model-driven* or *behavioral*, for example, References [5,11,12].

A classical geometry-based, model-driven approach is described in Reference [5]. Objects are represented as one of three primitive shapes which are then mapped to an appropriate grasp in a grasp taxonomy. Fingers are closed until contact between the geometrical hand representation and the object occurs. Drawbacks of this approach are not very lifelike movements, no real interactions with virtual objects and the restriction to primitive shapes.

In Reference [8], a physics-based approach is outlined, where controller parameters are derived from a database of example grasps. The proposed controller combines passive and active control by calculating torques for each finger joint. This enables the hand to be influenced by moving objects in addition to the hand manipulating scene objects in a physically plausible way. Although being general, relatively simple and consistent for movement with or without contact, the approach still has some limitations. No automatic hand orientation is done and the type of grasps supported is limited to cylindrical enveloping hand shapes.

The problem of orienting the hand toward the object is addressed in Reference [10]. Backed by a grasp database, several possible ways to grasp an object are identified and evaluated w.r.t. a given grasp quality metric. With a shape matching algorithm, a hand orientation and pose is identified which fulfills user-specified task requirements. These requirements refer to forces applied to the object and are evaluated on the basis of an anatomical hand model. While producing a flexible set of enveloping power grasps, some limitations remain. The approach is not real-time capable (runtimes of several minutes), only semi-automatic (animator has to pick a candidate) and limited to hand models very similar to the captured hand.

Instead of guessing or user-specifying hand compliance parameters, in Reference [9] contact force parameters are measured during user interactions with (real) objects via specialized hardware. This enhanced version of motion capture is referred to as interaction capture.

In terms of the categorization above, our approach falls into the category of geometry based and data driven. The aim is to find an adequate positioning of hand and fingers on an arbitrary virtual object while maintaining the style and naturalness of human-demonstrated examples, as such being related to Reference [10]. However, our approach is automatic, fast enough to be used in interactive scenarios, and can be used for arbitrary types of grasps.

Data Acquisition

As capturing device for hand shape data, an optical "fingertracking" system by A.R.T.[13] was used, as seen in Figure 1 (left) . The system tracks movements of all five fingers of the hand and comes with software that computes, among other data, position and orientation of the hand, fingertip positions, and rotations of finger joints. Before application, the fingertracking system is calibrated in order to estimate the size and parameters of the user hand. Calibration is performed using the vendor provided software tool and typically takes around 2 minutes per user.

In the capturing session, the demonstrator performs grasps with various physical or imaginary objects. Hand shape data is collected continuously, during closing and opening phases of the hand. In our experiments, the typical duration of the data acquisition phase was about 1–2 minutes. Hand poses in the database are stored as rotations of finger joints (three ball joints per finger, i.e., 45 degrees of freedom in total). The first post-processing step is to transform the joint rotation values data into an *exponential map* representation.[14] The result is a set of 45-dimensional vectors, each of which represents a single hand posture. The reason for using the exponential map representation is that it transforms the rotation into a

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

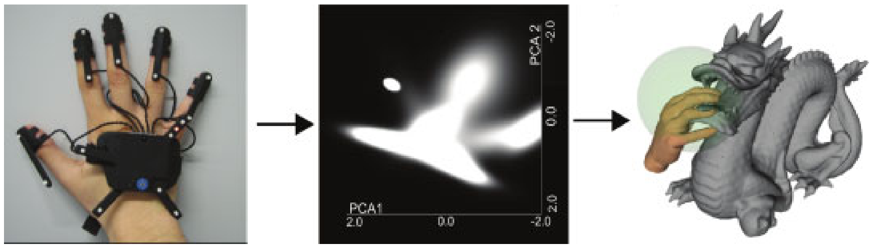


Figure 1. The proposed grasp synthesis method: recorded motion capture data (left) is transformed into a probabilistic low-dimensional grasp space (middle). Using optimization techniques, the space can then be searched for appropriate grasps of new objects.

linear space. PCA, the dimensionality reduction method used in our approach, is a linear transformation technique and therefore benefits from this type of representation.

Probabilistic Low-dimensional Grasp Space

The process of constructing a low-dimensional grasp space from recorded finger movements can be decomposed into two main steps: (1) performing PCA and (2) computing a GMM. This section also shows how a low-dimensional grasp space can be used to synthesize new grasps.

Principal Component Analysis

Due to the anatomy of the human hand, movement of fingers and finger segments is highly correlated. Often when trying to move a finger in a particular way we find our other fingers automatically moving in a similar fashion. This suggests that hand shapes lie on a low-dimensional subspace of the 45-dimensional space used for data acquisition. Reducing the dimensionality of the recorded grasps with PCA allows us to capture the correlations between the degrees of freedom of the human hand. PCA reduces the dimensionality of a dataset while retaining as much of the variance, and thus information, of the dataset as possible. To perform PCA, the mean grasp x_m is subtracted from all recorded data points and the covariance matrix M of the resulting points is computed. A singular value decomposition (SVD) on M yields matrices U, V, and W, such that

$$M = UWV^{\mathrm{T}} \tag{1}$$

The columns of matrix V contain orthonormal vectors called the *eigenvectors* or PCs of matrix M. The matrix W is a diagonal matrix containing the singular values. Each PC has a corresponding singular value which indicates how much information of the dataset is covered by this PC. Therefore, for projecting the dataset onto a d-dimensional space, we take the d PCs with the highest singular values in order to retain as much information as possible. These PCs are then used as the axes of our lower dimensional PCA space. Given a new data point, we can compute its coordinates in PCA space by subtracting the mean grasp and calculating the dot product with each of the PCs. Another interesting feature of PCA is the ability to synthesize new grasps by re-projecting a point back into the 45-dimensional space of finger joint rotations. This allows us to generate new grasps with a small number of control parameters. Given a d-dimensional point $a = \{a_1, \ldots, a_d\}$, the corresponding re-projected point a' can be computed by

$$a' = x_m + \sum_{i=1}^{d} a_i e_i \tag{2}$$

where e_i is the ith PC. For purpose of illustration, a 2D PCA is used throughout this paper, thus $d = 2$. However, the approach is independent of the number of dimensions d. In our application domain, grasp spaces with two to three dimensions proved to be sufficient.

Taking any arbitrary point $a = (a_1, a_2)^{\mathrm{T}}$, we can re-project it in order to synthesize a corresponding grasp. In Figure 2, we see the result of synthesizing grasps along the first two PCs. In our experiments, we found that the first PC controlled the opening and closing of the hand shape, while the remaining PCs controlled the relative movement of fingers to each other. While Figure 2 shows the power of PCA to interpolate and extrapolate grasps, it also reveals a pitfall: the anatomical limitations and constraints of the human hand are not accounted for by the

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

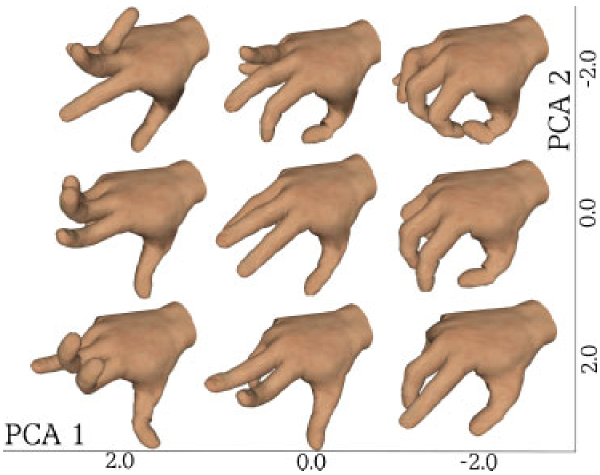


Figure 2. Hand postures synthesized by re-projecting points from the 2D PCA space. Some postures bend the finger segments in unnatural and exaggerated ways as the PCA space does not account for anatomical constraints of the human hand.

PCA. This means, that it is possible to synthesize grasps which bend the finger segments in exaggerated and, thus, unnatural ways. This effect can, for instance, be seen at position $(2, -2)^{\mathrm{T}}$ of the PCA space.

Gaussian Mixture Model

To ensure that only natural grasps are synthesized, it is necessary to derive a model of the anatomical constraints of the human hand from the fingertracking data. Such a model can then be used to discriminate between anatomically feasible and unfeasible grasps. In our approach, we accomplish this by learning a GMM of the training grasps in PCA space. Taking the set $\{x_1, \ldots, x_N\}$ of projected grasps (where N is the number of grasps), a GMM estimates the probability density function of the dataset by a weighted sum of K Gaussian distributions. The probability density function can therefore be written as

$$p(x) = \sum_{k=1}^{K} \pi_k p(x|k) \tag{3}$$

with π_k being the weight of the kth Gaussian and $p(x|k)$ being the conditional density function. The conditional density function is a d-dimensional Gaussian distribution:

$$p(x|k) = \frac{1}{\sqrt{2\pi}^{d}\sqrt{det(C_k)}} e^{-(1/2)\left((x-\mu_k)^{\mathrm{T}}\, C_k^{-1}(x-\mu_k)\right)} \tag{4}$$

with mean μ_k and covariance matrix C_k. The above $p(x|k)$ can also be written as $\mathcal{N}(x|\mu_k, C_k)$. To estimate the parameters $\{\mu_k, C_k, \pi_k\}$ for each of the Gaussian kernels, the expectation-maximization (EM)[15] algorithm is used. However, performing the EM algorithm in high dimensional spaces can be very time consuming. It is therefore convenient that our data is already projected to the low-dimensional PCA space as this ensures fast convergence of EM.

Making a correct choice for the number K of Gaussians is a difficult and critical task. High values can lead to overfitting while low values can result in inaccurate models. Following a similar approach as in Reference [16], we use the *Bayesian Information Criterion* (BIC) for determining an optimal number of K. The BIC value measures how well the model fits the training data while also incorporating a penalty term for the model complexity. To find a good tradeoff, we compute a set of GMMs with increasing values for K and take the one with minimal BIC.

Once the GMM is trained, we can use Equation (3) to compute the probability of a given PCA point x with respect to the modeled probability density function. Using a generative interpretation of the GMM, we can regard it as the statistical model from which the training grasps were sampled. Computing the probability of p w.r.t. the GMM, therefore, estimates the likelihood of x being generated by the same statistical model as the recorded training grasps. Assuming that we use enough grasps for training, we can therefore use the GMM results as an indicator for the anatomical feasibility of the grasp corresponding to a given point x. All points in PCA space with $p(x) > \tau$ are similar enough to the training grasps to be considered a feasible grasp. Figure 1 (middle) shows the GMM embedded in the PCA space of Figure 2.

Grasp Optimization

The goal of the grasp optimization process is to find a natural looking hand shape leading to a stable grasp on a user provided 3D object. The hand shape should have the same style and appearance as the demonstrated grasps during data acquisition. As a prerequisite for this, we fitted our virtual hand model with proximity sensors, as seen in Figure 3. The sensor model consists of one spherical sensor attached to each of the finger segments and three additional sensors in the palm. Using these sensors allows us to detect collisions or compute distances without having to resort to the complete geometry of the hand. The number of employed sensors can further

Copyright © 2008 John Wiley & Sons, Ltd.

Comp. Anim. Virtual Worlds 2008; **19**: 445–454
DOI: 10.1002/cav

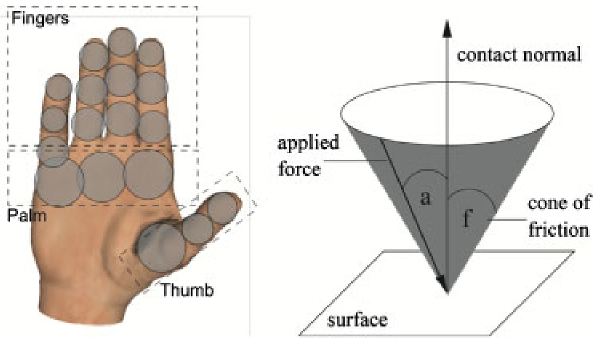


Figure 3. Left: placement of spherical sensors in the hand model with opposition groups thumb, fingers, and palm. Right: cone of friction at a contact point.

be adapted according to the desired precision or calculation time. We divided the sensors into three different opposition groups *thumb, fingers, and palm*. Sensors from different opposition groups are considered to be antagonistic sensors. This means that they can exert forces in opposing directions so as to create stable grasps. A similar sensor model for the human hand has been introduced in Reference [11].

In order to find an optimal grasp, we need to search for a set of parameters leading to a hand shape which optimizes a given *grasp metric*. Various metrics and quality measures for grasps have been proposed in the robotics literature,[17] many of which are based on physical properties of the object and the performed grasp. In this paper, we use a simple metric which consists of three components: the distance of the sensors to the object, an estimate of the stability of the grasp, and a penalty value.

Grasp Metric

As a first component in our metric, we use the distance of each finger sensor to the object. This distance needs to be minimized to ensure that the fingers touch the object. The second component, the stability measure, follows the definition of a stable grasp found in Reference [18]. In particular, we adopted the concept of *cone of friction*. In Figure 3, we see the graphical representation of the cone of friction at a contact point on the surface of the grasped object. According to MacKenzie and Iberall,[18] the cone of friction is a geometric interpretation of the maximally allowed angle ϕ between the surface normal and the applied force vector. If the applied force at the contact point makes an angle α with $|\alpha| < \phi$, then no slip will be produced at the fingertip. The angle ϕ is determined by the coefficient of friction of the grasped object. To determine if a grasp induced by two antagonist finger sensors is stable, we first compute the nearest positions on the surface of the object. These are taken as estimates of the contact points of the sensors with the object. Then we compute the connecting line. If this line lies within both cones of friction at the intersection points with the object surface, we regard the grasp as stable. In other words, for achieving a stable grasp with two fingers we need to minimize the angles α_1 and α_2 between the connecting line and the contact normals, until both are smaller than ϕ. Finally, as a last component to our grasp metric, we also incorporated a penalty term v (violation) for every sensor that penetrates the object:

$$v_i = \begin{cases} V : & \text{sensor } i \text{ penetrates object} \\ 0 : & \text{sensor } i \text{ does not penetrate object} \end{cases} \tag{5}$$

This ensures that no finger unrealistically enters the object during the grasp. In our case, a value of $V = 100$ produced satisfactory results. For every pair of antagonistic finger sensors, we compute the above components and combine them into one metric according to the formula

$$M = \sum_{i=0}^{\mathrm{SP}} (d_{i,1} + \alpha_{i,1} + v_{i,1}) + (d_{i,2} + \alpha_{i,2} + v_{i,2}) \tag{6}$$

where SP is the number of antagonistic sensor pairs, $d_{i,1}$ and $d_{i,2}$ are the distances of each respective sensor to the object surface, $\alpha_{i,1}$ and $\alpha_{i,2}$ are the angles between the connecting line and the contact normals and values $v_{i,j}$ are the penalty terms. Lower values for M indicate better grasps.

Optimization Process

We use the low-dimensional grasp space to synthesize grasps in the optimization process. This reduces the hand shape parameters to be optimized to d values defining the coordinates of the grasp in PCA space. These values will subsequently be called the *intrinsic parameters*. Additionally, we also need to specify a set of *extrinsic parameters*. Such parameters can for instance be orientation, position, or distance of the hand w.r.t. the grasped object. These extrinsic parameters can either be set by the user or be included into the optimization process if no specific values are available.

Given intrinsic and extrinsic grasp parameters and a grasp metric, we can employ optimization algorithms to find a suitable grasp for the provided 3D object. This is done by searching the space spanned by intrinsic and

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

extrinsic parameters for a set of parameters which minimizes the value of M. Additionally, during optimization, for each evaluated grasp the GMM likelihood is computed using Equation (3). If the likelihood is lower than τ, then the corresponding grasp is discarded from the optimization by receiving a high penalty.

In order to determine an optimization algorithm which is best suited for our application domain, we evaluated several optimization algorithms as reported below in the results section. In particular, we are interested in fast optimizers which scale well with increasing complexity of the objects.

Computational Speedups and Improvements

A simple and efficient way to speed up the above grasp synthesis algorithm is to use a low-polygon version of the object model for the optimization process. For graphical display, however, the original model can still be used. For example, when optimizing grasps for the Stanford bunny, we use a high resolution version with 26 332 polygons for display and a low-polygon version with 453 polygons for optimization. In general, however, it is also important that the shapes of the original model and the reduced model do not deviate too much. Another way to speedup computation while at the same time increasing the control over the optimization process, is to specify an *area of interest*. This allows the user to define a specific part of the object within which a grasp should be generated. In Figure 4, we see an example for a spherical area of interest. In this example, we want the optimizer to find a grasp which grasps the left ear of the bunny model. Therefore, we placed the area of interest in such a way that it envelopes all triangles of the left bunny ear. All polygons outside of the area of interest are discarded from the optimization process. In our specific case this further reduces the number of evaluated polygons from 453 to about 120 polygons. Generally, using an area of interest leads to noticeably lower computational demands.

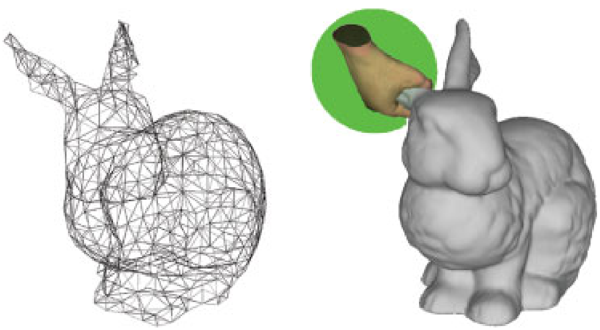


Figure 4. Left: a low-polygon version of the Stanford bunny is used for fast optimization. Right: using an area of interest (green circle) allows to specify which part of the object should be grasped.

Evaluation and Results

First, we evaluated our approach in combination with different optimization algorithms in order to determine the best suited optimizer for our domain. Three optimizers namely genetic algorithms (GA), dynamic hill-climbing (DHC), and simulated annealing (SA) have been tested.

During the experiment, grasp synthesis was performed on the bunny 3D model from the Stanford 3D scanning repository.[19] For this, a general grasp space was trained from the motion capture data of grasping a large variety of objects. The intrinsic parameters were the x and y coordinates in the low-dimensional grasp space. As extrinsic parameters, we used the rotation angles of the hand which results in a total number of five parameters. The position of the hand was user specified.

For each optimizer, the grasp synthesis was repeated 100 times and the grasp quality was measured using metric M as specified in Equation (6). As already stated, lower values for M indicate better grasps. In order to have a fair comparison, the number of function evaluations for each optimizer was limited to 600. In Figure 5, the average resulting grasp metric values of the different algorithms are

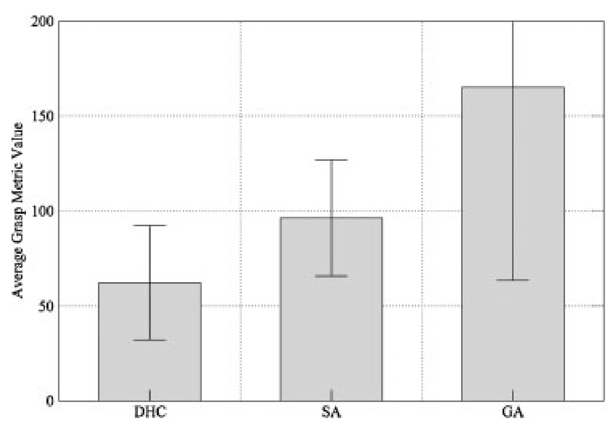


Figure 5. Average grasp metric of a dynamic hill climbing, simulated annealing, and genetic algorithm after performing 100 optimization (minimization) trials. The error bars show the standard deviation.

Copyright © 2008 John Wiley & Sons, Ltd.

Number of polygons	Time (seconds)
<1000	1
3851	2
16 301	8
146 148	17

Table 1. Runtimes of optimization with dynamic hill climbing

shown. As can be seen, DHC systematically performed better than the other algorithms. This is in particular due to the fact, that it is far less sensitive to changes in parameters than GA. We also experimented with some gradient-based optimization algorithms such as gradient descent or RPROP. However, we found that they are not suited in this domain. This is due to the fact that the grasp metric is a non-differentiable function and as such not suited for gradient-based techniques.

Next, we evaluated the scalability of grasp synthesis with DHC as optimizer to objects with higher complexity, that is, higher number of polygons. For this, we prepared different versions of the Stanford bunny with increasing number of polygons and tested the runtime of the algorithm. The results in Table 1 show that for reasonable object sizes of up to 150 000 polygons, optimization is finished in less then 20 seconds. Of course, the optimization speed can be further increased using the techniques explained in the preceding section.

Figures 1 and 6 show the results of synthesizing grasps for the Stanford dragon and bunny. As can be seen, even on fairly complex objects the algorithm is able to synthesize a variety of different natural looking grasps. The user can choose parts of the object such as the tail, nose, back, or mouth of the dragon and synthesize a valid grasp for it.

Taxonomy-Based Grasp Synthesis

Some application domains of the proposed grasp synthesis algorithm make use of grasp taxonomies for specifying different grasp types. Grasp taxonomies are categorizations of grasps based on form or function. Such taxonomies can provide the user with a means to exactly specify the way in which an object should be grasped. A similar control of grasp type can be integrated into our optimization algorithm by training a set of type specific grasp spaces. In the following example, we trained the grasp types of the Schlesinger taxonomy[3] using PCA and GMMs.

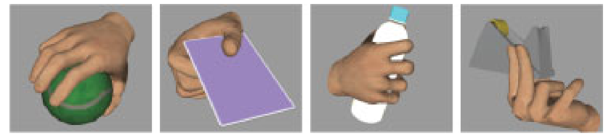


Figure 7. Grasps from the Schlesinger grasp taxonomy synthesized using the introduced optimization approach. From left to right: spherical, lateral, cylindrical, and palmar grasp.

For each of the grasp types, we created a separate grasp model which is based on data of only this particular grasp type. Therefore, if during synthesis the user explicitly specifies a grasp type to be used, we perform optimization on the corresponding grasp model. Grasp types other than the one specified are not contained in the particular space and will hence not be produced by the optimization process. Figure 7 shows the resulting grasps synthesized for the types *spherical*, *cylindrical*, *palmar*, and *lateral* of the Schlesinger taxonomy. Each of the grasps in Figure 7 took about 1 second for optimization.

Discussion

A strength of the presented method for grasp synthesis is that it supports the training of general grasp models that can be applied to a wide range of differently shaped objects. Grasps generated from the general model usually appear visually plausible as they both correspond to natural hand poses and enforce stable contact between the hand and the grasped object. When a general model is used for grasp synthesis, the method may, however, produce hand postures which, although stable and representing one way to grasp an object, can be different

Figure 6. The results of applying grasp optimization on the Stanford dragon model. The position of the hand and the area of interest (green) were specified by the user. Both the hand shape, and orientation were computed by the optimization process.

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/cav

Figure 8. Grasp synthesis for a virtual human.

from the grasp type expected by the animator. For a finer control over the type of generated grasp, the animator is provided with three principal means of parameter adjustment: (1) a region of interest can be defined as specification of where to grasp the object; (2) specific grasp models can be trained and applied to control the grasp type; and (3) different grasp stability criteria could be employed in the optimization method to fine-tune hand-object contacts.

The training of specific grasp models was demonstrated with the example of the Schlesinger taxonomy. Clearly, the method can also be applied to other grasp taxonomies. In extreme cases, the animator might even choose to train highly specific grasp models for individual objects. When physical objects are grasped, further degrees of freedom of hand shape might be induced from the contact between the hand and the object. If a high degree of realism is demanded in the animation, such effects could be reproduced by training models from grasp demonstrations on physical objects only.

As grasp stability measurement, the method has been tested with a simple criterion from the literature based on the cone of friction. We intentionally opted for a purely geometric criterion as this frees the animator from the necessity of modeling physics-related object properties. The presented method itself does, however, not depend on a particular optimization criterion and is compatible with more complex grasp quality measures as researched, for example, in the GraspIt project.[20] To reproduce effects of physics-related properties such as object mass when synthesizing grasps using geometric grasp criteria only, it is again recommended to train specific grasp models which reflect the object's physical properties. An interesting idea for further work would be to learn the optimization function directly from interactions in virtual reality, by exploiting contact point information. For this, we also plan to experiment with more elaborate hand models based, for example, on additional sensors on the palm and finger sides. In this way, it might be possible to generate highly specific grasps which resemble the demonstrated grasp possibly less in shape yet more in "intent."

Conclusion

We introduced a new method for the synthesis of human-like grasping for virtual characters. In the proposed approach, motion capture data of grasps performed by a human are used to train a probabilistic model of the human's grasp space, that is, the space of possible hand postures assumed during grasping actions. Grasp synthesis is then realized by searching the grasp space for a hand shape that optimizes the given grasp quality metrics. Algorithmic techniques such as PCA and GMMs are used to create a probabilistic low-dimensional grasp space which can be searched through efficiently while still being large enough to contain a large range of plausible grasps. In our experiments, DHC was able to search the grasp space particularly efficiently and turned out as the optimization method of choice. The combination of data-driven motion capture with model-driven optimization techniques results in the generation of both naturally looking and physically plausible grasps.

ACKNOWLEDGEMENTS

The research described in this contribution is partially supported by the DFG (Deutsche Forschungsgemeinschaft) in the Virtual Workers project.

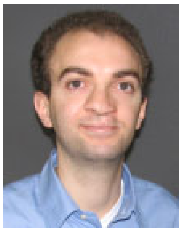
References

1. Gleicher M. Retargetting motion to new characters. In *SIGGRAPH'98 Conference Proceedings*, Computer Graphics Annual Conference Series, ACM, 1998; 33–42.
2. Santello M, Flanders M, Soetching JF. Postural hand synergies for tool use. *The Journal of Neuroscience* 1998; **8**(23): 10105–10115.

Copyright © 2008 John Wiley & Sons, Ltd.

3. Schlesinger G. Der Mechanische Aufbau der Künstlichen Glieder. In *Ersatzglieder und Arbeitshilfen für Kriegsbeschädigte und Unfallverletzte*, Borchardt M, Hartmann K, Leymann H, Radike R, Schlesinger G, Schwiening H (eds). Springer-Verlag: Berlin, Germany, 1919; 321–661.
4. Cutkosky MR. On grasp choice, grasp models and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation* 1989; **5**(3): 269–279.
5. Sanso RM, Thalmann D. A hand control and automatic grasping system for synthetic actors. *Computer Graphics Forum* 1994; **13**(3): 167–177.
6. Weber M, Heumer G, Ben Amor H, Jung B. An animation system for imitation of object grasping in virtual reality. In *Proceedings of 16th International Conference on Artificial Reality and Telexistence, ICAT*. Springer, Berlin/Heidelberg, 2006; 65–76.
7. Borst CW, Indugula AP. Realistic virtual grasping. In *VR'05: Proceedings of the 2005 IEEE Conference on Virtual Reality*, 2005; 91–98.
8. Pollard NS, Zordan VB. Physically based grasping control from example. In *SCA'05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 2005; 311–318.
9. Kry PG, Pai DK. Interaction capture and synthesis. *ACM Transactions on Graphics* 2006; **25**(3): 872–880.
10. Li Y, Fu J, Pollard N. Data driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics* 2007; **13**(4): 732–747.
11. Rezzonico S, Huang Z, Boulic R, Thalmann NM, Thalmann D. Consistent grasping interactions with virtual actors based on the multi-sensor hand model. In *Virtual Environments'95*, Springer, Wien, 1995; 107–118.
12. Rijpkema H, Girard M. Computer animation of knowledge-based human grasping. *SIGGRAPH Computer Graphics* 1991; **25**(4): 339–348.
13. Advanced Realtime Tracking GmbH. Homepage. http://ar-tracking.eu, last visited on 18 January 2008.
14. Grassia FS. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools* 1998; **3**(3): 29–48.
15. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 1977; **39**(1): 1–38.
16. Pelleg D, Moore A. *X*-means: extending *K*-means with efficient estimation of the number of clusters. In *Proceedings of 17th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 2000; 727–734.
17. Moon A, Farsi M. Grasp quality measures in the control of dextrous robot hands. *Physical Modelling as a Basis for Control (Digest No: 1996/042), IEEE Colloquium on Physical Modelling as a Basis for Control*, 1996; 6/1–6/4.
18. MacKenzie CL, Iberall T. *The Grasping Hand*. Elsevier-North Holland: Amsterdam, 1994.
19. Stanford. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/last visited on 22 January 2008.
20. Miller AT, Christensen HI. Constraint stabilization for time-stepping approaches for rigid multibody dynamics with joints, contact and friction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003; 2262–2268.

Authors' biographies:



Heni Ben Amor studied computational visualistics at the University of Koblenz-Landau, Germany. After a research stay at the Intelligent Robotics Lab in Osaka, Japan, he joined the Technical University Bergakademie Freiberg as a research assistant and PhD Student in 2006. His research interests include machine learning, android robots, and virtual reality.

Guido Heumer holds a Diploma (equivalent to MSc) degree in computer science which he earned in 2003 from the University of Bielefeld, Germany. Currently, the 33 year old is aspiring to his PhD in the field of virtual reality at the Technical University Bergakademie Freiberg. His main research work is focused on the Analysis of user interactions with Virtual Environments in the DFG-funded Virtual Workers project. He followed his PhD supervisor Professor Dr. Bernhard Jung and the project on the move from the International School of New Media (ISNM) to Freiberg in October 2005.

Bernhard Jung studied computer science and linguistics at the University of Stuttgart, Germany, and the University of Missouri, Saint Louis. He received his Doctorate degree from the University of Bielefeld with a Thesis on dynamic knowledge representation as well as a Habilitation degree for a Thesis on intelligent virtual environments. From 2003 to 2005, he was

Copyright © 2008 John Wiley & Sons, Ltd.

Full Professor for Media Informatics at University of Lübeck's International School of New Media. Since October 2005, Bernhard Jung chairs the Virtual Reality and Multimedia group of the Technical University Bergakademie Freiberg, Germany. Bernhard Jung's research interests are in the fields of Virtual Reality, Pervasive and Mobile Computing, Artificial Intelligence, and Human-Computer Interaction in general.

Arnd Vitzthum has studied Computer Science at the Dresden University of Technology. From March 2003 to January 2008, he was a PhD student and Research Assistant at the University of Munich (LMU) and member of the Media Informatics Group there. Currently, he is a member of the Virtual Reality and Multimedia group at the TU Bergakademie Freiberg. His primary research interests are directed toward the development of virtual reality and augmented reality applications.

Copyright © 2008 John Wiley & Sons, Ltd.

DOI: 10.1002/cav