Pacific Graphics (2014)
J. Keyser, Y. J. Kim, and P. Wonka (Editors)

Automatic 3D Posing from 2D Hand-Drawn Sketches

Alexandros Gouvatsos[1,2], Zhidong Xiao[1], Neil Marsden[2], Jian J. Zhang[1]

[1]Centre for Digital Entertainment, National Centre for Computer Animation, Bournemouth University, UK
[2]Hibbert Ralph Animation, UK

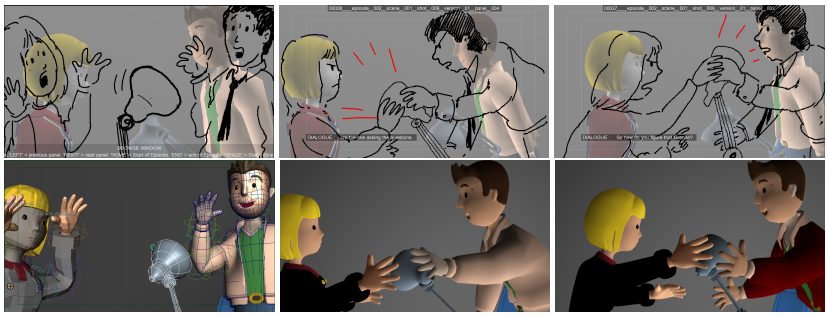

Figure 1: *From initial storyboard to pre-visualisation. The lamp in these three shots is automatically posed using the storyboard drawing. On the top row, drawings are overlaid on top of the 3D scene with the assets unposed. On the bottom row, the lamp has been posed automatically using our method.*

Abstract

Inferring the 3D pose of a character from a drawing is a non-trivial and under-constrained problem. Solving it may help automate various parts of an animation production pipeline such as pre-visualisation. In this paper, a novel way of inferring the 3D pose from a monocular 2D sketch is proposed. The proposed method does not make any external assumptions about the model, allowing it to be used on different types of characters. The 3D pose inference is formulated as an optimisation problem and a parallel variation of the Particle Swarm Optimisation algorithm called PARAC-LOAPSO is utilised for searching the minimum. Testing in isolation as well as part of a larger scene, the presented method is evaluated by posing a lamp and a horse character. The results show that this method is robust and is able to be extended to various types of models.

Categories and Subject Descriptors (according to ACM CCS): I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—Heuristic methods I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—Shape I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.4.9 [Image Processing and Computer Vision]: Applications—

1. Introduction

The field of animation has advanced rapidly, offering a plethora of ways to bring characters to life from sketching and joint manipulation to puppet posing and motion capture.

Since animation has its roots in hand drawings, being able to show one's ideas on paper is a skill most animators possess. As such, many have argued sketching is an intuitive interface for artists of traditional and modern backgrounds, be it for posing [MQW06], animating [DAC*03], modelling [YW10], stylising motion [LGXS03] or simulating secondary movement [JSMH12] and crowds [MQW07]. By providing an intuitive link between the initial sketching process and the end result, the cognitive load and the necessary training for the artist may be significantly reduced.

Initial sketching is part of most animation pipelines already, usually in the form of storyboarding during pre-

DOI: 10.2312/pgs.20141264

production. These drawings are then used by animators or pose artists to set up the initial pose layout before going into the animation stage. Rather than introducing a new workflow or step, the aim of the proposed method is to automate one of the existing steps. Specifically, the focus is on making the initial posing stage of a production less labour intensive.

Posing 3D models from 2D sketches is a non-trivial open problem, closely related to the computer vision problem of inferring a 3D pose from a 2D photograph but with several differences. Photographs are usually accurate in terms of scale and include colour information. Sketches may have squashing and stretching, usually lack colour and contain noise in the form of auxiliary strokes e.g. for shading.

This paper focuses on using sketches to pose 3D characters automatically, for layout or pre-visualisation.

2. Related work

The mainstream method of posing 3D characters is by manipulating controls of a skeletal structure, usually joints [BW76]. Even with the invention of Inverse Kinematics [ZB94], this remains a time consuming process which requires the user to be trained in specific suites of software.

A drawing represents an internal model the artist has about a scene and it contains no depth information. As such, the problem of inferring the depth from a drawing is under-constrained, creating problems such as the forward-backward ambiguity problem (Figure 2).

Figure 2: *An example of the forward-backward ambiguity issue in inferred 3D pose of a lamp when not using the internal lines for additional information.*

Automatic 3D posing: In general, inferring a 3D pose from a drawing can be broken down into two sub-problems. Firstly, match the 3D model projection to the drawing. Secondly, infer the missing depth. For 3D productions, the camera represents the position from which the characters in the drawing are viewed, according to the artist's internal model.

A comprehensive overview of progress in this computer vision problem [Gav99] highlights two main types of approach, the *explicit shape model* approach and the *database* approach. In this case the focus is on data from drawings rather than photographs or video frames.

Database approach: Previous computer vision research has used database approaches not only for tracking and posing [JSH09] but also for positioning [XCF*13] in a scene or simply classifying [EHA12]. In animation, Jain et al. [JSH09] follows a database approach in order to infer the missing depth and pose humanoid characters. However, it is able to handle only orthogonal camera projections, while the method proposed in this paper can handle both orthogonal and perspective projection. Building a database of poses may be expensive, especially since motion capture cannot be easily used for non-humanoids that are so common in 3D animation productions. As such, an approach that does not require a database was preferred.

Explicit shape model approach: This approach is very effective when there are multiple camera views because the problem is no longer under-constrained (no occlusions). It consists of comparing the 3D render pose to the 2D image in order to estimate how close it is to a correct pose. One of its limitations in computer vision is that it requires a 3D model that is a good match to the 2D image. In 3D productions this limitation does not apply, since a 3D model exists anyway.

Favreau et al. successfully pose animals using Radial Basis Functions of 3D pose examples [FRDC06]. Ivekovic et al. [ITP08] infer 3D poses of humans in video sequences using multi-view images. Ivekovic et al. still make human-specific assumptions limiting its use to a small subset of characters while both Favreau et al. and Ivekovic et al., use video data and exploit temporal relations between frames.

Posing interfaces: Interfaces to replace the mainstream skeleton manipulation process have been the focus of previous research e.g. tangible devices [JPG*14]. Examples of sketch-based interfaces make use of differential blending [OBP*13] or the line of action to pose 3D characters [GCR13]. These sketch-based interfaces may offer improvements, but are still operated by artists. The proposed method is aiming to reduce the process to an automatic method a user with no artistic background is able to perform.

This paper proposes a method for posing 3D models from 2D drawings, aiming at pose layout and pre-visualisation. It makes no assumptions about the 3D model in order to deal with many types of characters. Unlike Jain et al. [JSH09], it does not rely on a database. Instead, a particle swarm optimisation (PSO) variant called Parallel Convergence Local Optima Avoidance PSO (PARAC-LOAPSO) is used. This allows for posing of both cooperative (e.g. humans) and non-cooperative (e.g. animals) characters. It does not require the model to have a skeleton; any type of controller is suitable. Unlike Ivekovic et al. [ITP08] the input data are monocular, stand-alone drawings with no temporal relations between them. Minimal user input is required, in the form of pinpointing the joints on top of the drawing. To avoid forward-backward ambiguity, the proposed method uses a combination of various descriptors [FAK12]. There are currently no other methods that perform automatic posing for the layout phase of animation, that work irrespective of the character model and that can be applied without the need for a pre-existing database.

3. Methodology

3.1. Overview

To propose a general posing method which can automate the initial layout pass or pre-visualisation of a production, the solution takes into account several requirements:

- Make no assumptions about the type of character, in order to be versatile (e.g. humans, horses, lamps, octopuses).
- Provide results which are good enough for the initial posing and not the final animation.
- Ensure necessary user input is quick and requires little learning from the user side.
- Do not restrict the traditional pipeline structure.
- Be feasible for both small and big budget productions.

Since the proposed method is focusing on posing and not generating a 3D model from a 2D sketch, it is assumed that there is already a 3D model that corresponds to the character in the drawing. Therefore, the *explicit shape model* approach is utilised for solving the problem.

The 2D data is an artistic drawing e.g. a storyboard panel. The 2D sketch itself does not need to be of high quality, but it does need to be accurate in terms of the shape and scale of the object that will be posed.

To create a search space containing the wanted 3D pose, an interface is used which allows users to overlay drawings on top of a 3D space (Figure 1), moving the camera and placing unposed 3D models. This reduces the search space to consist of only local poses of a character, not translation in 3D space (Section 3.2). To compare the 2D to the 3D data in the search space, the system extracts information from the sketch. That is when the only user input required - the pinpointing of the joints - takes place (Section 3.3). To navigate the search space and find the wanted pose, the system iteratively poses and renders the 3D model of the character to match the drawing (Section 3.2).

The process is not distruptive or encumbering on pipelines, as it is not different to a traditional pipeline with a 3D layout pass. Additionally, drawings exist early on in preproduction in the form of storyboards. The pinpointing of the joints can be done in a few seconds during the clean-up phase and does not require technical training.

3.2. PARAC-LOAPSO

The problem of inferring the 3D pose from 2D data is treated as a minimisation problem, by navigating a formulated search space using a variation of PSO called PARAC-LOAPSO. It performs global search and a reason for choosing it is that it is parallel by nature, which makes it straight-forward to implement on a GPU [MIC10]. As metaheuristic, it makes no assumptions about the problem.

The problem is formulated by defining each particle as a possible pose or *solution*. Specifically, each solution is a vector $X = (j_1, ..., j_n)$, where n is the number of joints to manipulate on the model. Each element j represents a joint and is a three dimensional vector $j = (x, y, z)$ where x, y and z are its Euler angle rotations in local space. Therefore, the search space contains all the possible poses for that model.

While issues of uncertain convergence, speed and getting stuck at local minima have been explored [BK07], it is important to re-examine them for this particular problem.

Improved search: To deal with slower convergence, a constriction factor K (equation 1) [BK07] is preferred over the inertia weight used by Ivekovic et al. [ITP08] or Salehizadeh et al. [SYM09]. To speed up convergence and overall runtime, the algorithm is implemented to run in parallel on the GPU. To deal with getting stuck at local minima, the method proposed by Salehizadeh et al. is used to add variation in the population [SYM09]. Additionally, the optimisation process is faster and more accurate by navigating a smaller search space. That is achieved by using joints' rotation limits from the given 3D model as constraints. Instead of using biological data about human joint limits, only the joint limit data from the 3D model are acquired. Any hierarchical assumptions would decrease the method's versatility.

$$K = \frac{2}{|2 - a - \sqrt{a^2 - 4a}|} \tag{1}$$

$$a = \phi_1 + \phi_2 \tag{2}$$

The cognitive (influence by own findings) and social (influence by others' findings) weights are defined as ϕ_1 and ϕ_2.

Initialisation: The overall performance of the algorithm is affected by its initialisation. Since no temporal data is assumed, as would be in the case of a video, a previous pose cannot be used as an initial point. Therefore, each element j_n of position X_i of each particle i is initialised from a uniform random distribution: $j_n \sim U(X_{min}, X_{max})$. The upper and lower boundaries $X_{min}, X_{max} \in [-360, 360]$ differ for each element of X_i, since the joint rotation limits are used to set them for each particle element.

Fitness evaluation: After the initialisation step a fitness function f allows for the evaluation of each particle. Based on this, the personal best position B_i^t of each particle i and the global best position B_g^t from all particles are stored.

$$f = w_1 D_j + w_2 D_s + w_3 D_e \tag{3}$$

where D_j is a distance metric for the joint positions, D_s one for the shape silhouette and D_e for the edges (both internal and external), while w_1, w_2 and w_3 are the respective weights for each component (e.g. $\frac{1}{3}$ for equal weighting of all parts).

Update: At the beginning of epoch t, the population of N particles is split into two subgroups of size γN and $(1-\gamma)N$, where $\gamma = r, r \sim U(0,1)$ if $t < T_c t_{max}$, or 1 otherwise. T_c is denoted as the threshold of convergence, which controls how many epochs are dedicated to exploration and how many to exploitation. For example, for $T_c = \frac{3}{4}$ only the last quarter

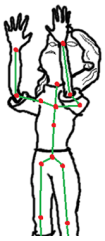

Figure 3: *Joints overlaid by a user on top of a drawing of a human.*

Figure 4: *Edge map of a drawing of a human, extracted using the Canny algorithm.*

Figure 5: *Silhouette of a drawing of a human.*

of the epochs will be dedicated to exploitation. t_{max} is the maximum number of epochs for a run.

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{4}$$

For every particle i, its position X_i is updated (equation 4). If the particle belongs to the first population subgroup, its velocity component V_i^{t+1} is updated using equation 5. Otherwise, to attempt and avoid local minima by adding variation, it is updated using equation 6.

$$V_i^{t+1} = K[V_i^t + c_1(B_i^t - X_i^{t+1}) + c_2(B_g^t - X_i^{t+1})] \tag{5}$$

$$V_i^{t+1} = K\{V_i^t - L^t[r_1(B_i^t - X_i^{t+1}) + r_2(B_g^t - X_i^{t+1})]\} \tag{6}$$

$$L^t = 2 - \frac{2t}{t_{max}} \tag{7}$$

$$r_1, r_2 \sim U(0,1) \tag{8}$$

$$c_1 = \phi_1 r_1 \tag{9}$$

$$c_2 = \phi_2 r_2 \tag{10}$$

Algorithm: The above components are combined in the following steps to form the overall algorithm.

1. Initialise the swarm population, with global best B_g^t.
2. For each particle i, with personal best B_i^t (in parallel):
   a. Update particle position (equation 4).
   b. Update descriptors of 3D pose (Section 3.3).
   c. Evaluate fitness of particle (equation 3).
   d. If current fitness $< B_i^t$, set B_i^t to current fitness.
   e. If current fitness $< B_g^t$, set B_g^t to current fitness.

3.3. Comparing drawings to renders

Three ways to compare the drawing with the 3D render were chosen given their previous reported advantages and disadvantages. Joints are used by Jain et al. in order to find close poses from a database. In order to make use of the internal edges, edge maps are simple, fast and as effective [TR07] as more expensive methods like Shape Context Histograms [AT06]. Finally, silhouettes have been used to find poses generatively [ITP08] or to train models [AT04].

Joints: The user overlays the positions of the joints on top of the drawing (Figure 3). The position of the overlaid joints on the drawing is compared to that of the 3D positions of the joints, transformed into screen space coordinates.

Edge map: To use internal lines in the drawing, the Canny algorithm [Can86] is used to extract a binary map of the external and internal edges (Figure 4). Then the binary map of the drawing is compared with that of the rendered image.

Silhouette: Finally, for the overall shape, the silhouette of the drawing is used as a binary map (Figure 5). The *silhouette distance* between the rendered image and the drawing is calculated using a previously successful method [FAK12].

4. Results

The system was tested on two problems of different dimensionality and difficulty: a lamp (Figure 6) and a horse (Figure 7). The reason for choosing them was to evaluate whether the proposed method is suitable irrespective of the problem. Both models would be difficult to be posed via methods such as motion capture, especially the horse. Most motion capture solutions focus on humanoids and are not affordable for smaller studios. Apart from posing models in isolation, the proposed method is evaluated by posing a lamp within a scene using an existing storyboard drawing (Figure 1).

For these tests, the algorithm was ran for $t_{max} = 1000$ epochs. The values ϕ_1 and ϕ_2 as defined in Section 3, were set to a value of 2.05, which is the default value in the canonical PSO [BK07] and ensures convergence of the swarm. The minimum and maximum velocity of each particle were set to -360.0 and 360.0 respectively, since this is the space of possible rotations. This means that in one epoch, a joint can rotate at most by one complete rotation either clockwise or counter-clockwise. The minimum and maximum position of each particle was set dynamically based on the angle rotation limits of each joint of the model, reducing the search space of possible poses. The exploration phase (Section 3.2), was set to $T_c = \frac{3}{4}$, so the first $t_{max}T_c = 750$ epochs were dedicated to exploration and the last quarter of epochs was focused on exploitation.

Figure 6: *Side by side comparison of the drawings (top) and the estimated 3D pose (bottom) for the lamp model.*

1000 epochs took approximately 81 seconds on average on a system with 8GB of RAM, an Intel HD Graphics 3000 GPU and an Intel Core i5-2520M 2.50GHz CPU.

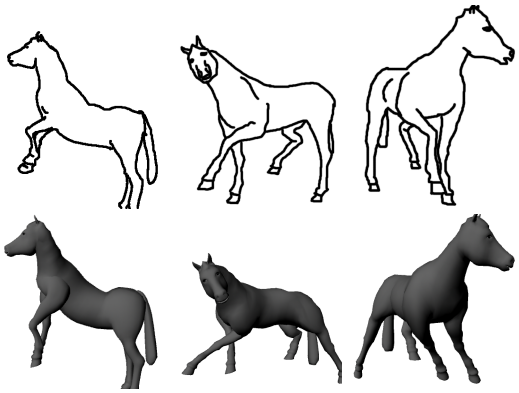

Figure 7: *Side by side comparison of the drawings (top) and the estimated 3D pose (bottom) for the horse model.*

5. Discussion

It is important to note that this method does not aim to produce final animated output. It aims at automating the initial pose layout phase or pre-visualisation, which is currently performed manually by skilled operators.

The choice of the method of comparing the drawing to the render may have a notable effect on the end result. For example, by not using the internal lines of a drawing, the forward-backward ambiguity problem can become an issue as seen in Figure 2.

The method proposed in this paper easily fits into traditional animation pipelines. Unlike Jain et al. [JSH09], it does not require pre-existing data and can use perspective and orthographic camera models. The pinpointing of the joints user input can take place as part of the clean up phase. The system is flexible and applicable to a broad range of models, from objects like lamps, to quadrupeds like horses.

However, it is this flexibility which leads to the main disadvantage of this method. Since the approach is generative, it requires a rendering step in every iteration, which may mean it takes a longer time to return a result. It is important to note that the time to return a result is computational only, meaning that this method is still able to contribute to reducing costs. Furthermore, implementing the method on a GPU reduces the effects of this disadvantage significantly.

Since the PARAC-LOAPSO algorithm is stochastic, result accuracy may vary between runs. However, a system with a powerful graphics card can normalise the results through the use of a large population of particles, which helps both in searching as well as in having a more varied initialisation.

6. Conclusion

This paper proposes a method to automatically pose 3D models using information from monocular drawings. This approach is generative and deals with inferring a 3D pose as an optimisation problem. The results show that it is general enough to deal with many types of characters, and a lamp and a horse model are successfully posed in different scenarios. Moreover, it does not require changes in the pipeline to accommodate for it. The focus of the proposed method is to pose models for the pose layout phase or pre-visualisation, not final animated output. It may serve as a direct link between storyboarding and pose layout phases of a pipeline.

It is worth examining more general or accurate descriptors and methods of comparing drawings to renders, such as a context-based approach on joints [JSH09]. Completely removing the need for user input may be possible by using medial axis methods [ABCJ08] to extract the curve skeleton automatically from the drawing and then fit the character skeleton to the curve skeleton.

A hybrid between the current optimisation approach and a data-driven approach such as Jain et al. [JSH09] is another area where future work may expand to, to get results faster while remaining more general than a pure database method.

References

[ABCJ08] ABEYSINGHE S., BAKER M., CHIU W., JU T.: Segmentation-free skeletonization of grayscale volumes for shape understanding. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on* (June 2008), pp. 63–71. doi:10.1109/SMI.2008.4547951. 5

[AT04] AGARWAL A., TRIGGS B.: Learning to track 3d human motion from silhouettes. In *In International Conference on Machine Learning* (2004), pp. 9–16. 4

[AT06] AGARWAL A., TRIGGS B.: Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 1 (Jan 2006), 44–58. doi:10.1109/TPAMI.2006.21. 4

[BK07] BRATTON D., KENNEDY J.: Defining a standard for particle swarm optimization. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE* (April 2007), pp. 120–127. doi:10.1109/SIS.2007.368035. 3, 4

[BW76] BURTNYK N., WEIN M.: Interactive skeleton techniques for enhancing motion dynamics in key frame animation. *Commun. ACM 19*, 10 (Oct. 1976), 564–569. URL: http://doi.acm.org/10.1145/360349.360357, doi:10.1145/360349.360357. 2

[Can86] CANNY J.: A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-8*, 6 (Nov 1986), 679–698. doi:10.1109/TPAMI.1986.4767851. 4

[DAC*03] DAVIS J., AGRAWALA M., CHUANG E., POPOVIĆ Z., SALESIN D.: A sketching interface for articulated figure animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 320–328. URL: http://dl.acm.org/citation.cfm?id=846276.846322. 1

[EHA12] EITZ M., HAYS J., ALEXA M.: How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012), 44:1–44:10. 2

[FAK12] FLEISCHMANN P., AUSTVOLL I., KWOLEK B.: Particle swarm optimization with soft search space partitioning for video-based markerless pose tracking. In *Advanced Concepts for Intelligent Vision Systems*, Blanc-Talon J., Philips W., Popescu D., Scheunders P., Zemcik P., (Eds.), vol. 7517 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 479–490. URL: http://dx.doi.org/10.1007/978-3-642-33140-4_42, doi:10.1007/978-3-642-33140-4_42. 2, 4

[FRDC06] FAVREAU L., REVERET L., DEPRAZ C., CANI M.-P.: Animal gaits from video: Comparative studies. *Graph. Models 68*, 2 (Mar. 2006), 212–234. URL: http://dx.doi.org/10.1016/j.gmod.2005.04.002, doi:10.1016/j.gmod.2005.04.002. 2

[Gav99] GAVRILA D.: The visual analysis of human movement: A survey. *Computer Vision and Image Understanding 73* (1999), 82–98. doi:10.1006/cviu.1998.0716. 2

[GCR13] GUAY M., CANI M.-P., RONFARD R.: The line of action: An intuitive interface for expressive character posing. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 205:1–205:8. URL: http://doi.acm.org/10.1145/2508363.2508397, doi:10.1145/2508363.2508397. 2

[ITP08] IVEKOVIC S., TRUCCO E., PETILLOT Y. R.: Human body pose estimation with particle swarm optimisation. *Evolutionary Computation 16* (2008), 509–528. doi:10.1162/evco.2008.16.4.509. 2, 3, 4

[JPG*14] JACOBSON A., PANOZZO D., GLAUSER O., PRADALIER C., HILLEGES O., SORKINE-HORNING O.: Tangible and modular input device for character articulation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 33*, 4 (2014). 2

[JSH09] JAIN E., SHEIKH Y., HODGINS J. K.: Leveraging the talent of hand animators to create three-dimensional animation. In *Symposium on Computer Animation* (2009), pp. 93–102. doi:10.1145/1599470.1599483. 2, 5

[JSMH12] JAIN E., SHEIKH Y., MAHLER M., HODGINS J.: Three-dimensional proxies for hand-drawn characters. *ACM Trans. Graph. 31*, 1 (Feb. 2012), 8:1–8:16. 1

[LGXS03] LI Y., GLEICHER M., XU Y.-Q., SHUM H.-Y.: Stylizing motion with drawings. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 309–319. URL: http://dl.acm.org/citation.cfm?id=846276.846320. 1

[MIC10] MUSSI L., IVEKOVIC S., CAGNONI S.: Markerless articulated human body tracking from multi-view video with gpu-pso. In *Proceedings of the 9th International Conference on Evolvable Systems: From Biology to Hardware* (Berlin, Heidelberg, 2010), ICES'10, Springer-Verlag, pp. 97–108. URL: http://dl.acm.org/citation.cfm?id=1885332.1885344. 3

[MQW06] MAO C., QIN S. F., WRIGHT D. K.: Sketching-out virtual humans: From 2d storyboarding to immediate 3d character animation. In *Proceedings of ACM SIGCHI International Conference On Advances In Computer Entertainment Technology* (June 2006), ACM. 1

[MQW07] MAO C., QIN S. F., WRIGHT D.: Sketch-based virtual human modelling and animation. In *Proceedings of the 8th International Symposium on Smart Graphics* (Berlin, Heidelberg, 2007), SG '07, Springer-Verlag, pp. 220–223. URL: http://dx.doi.org/10.1007/978-3-540-73214-3_24, doi:10.1007/978-3-540-73214-3_24. 1

[OBP*13] ÖZTIRELI A. C., BARAN I., POPA T., DALSTEIN B., SUMNER R. W., GROSS M.: Differential blending for expressive sketch-based posing. In *Proceedings of the 2013 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2013), SCA '13, ACM. 2

[SYM09] SALEHIZADEH S. M. A., YADMELLAT P., MENHAJ M.-B.: Local optima avoidable particle swarm optimization. In *Swarm Intelligence Symposium, 2009. SIS '09. IEEE* (March 2009), pp. 16–21. doi:10.1109/SIS.2009.4937839. 3

[TR07] TRESADERN P., REID I.: An evaluation of shape descriptors for image retrieval in human pose estimation. In *Proc. 18th British Machine Vision Conf., Warwick* (Sept. 2007), vol. 2, pp. 800–809. 4

[XCF*13] XU K., CHEN K., FU H., SUN W.-L., HU S.-M.: Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics 32*, 4 (2013), 123:1–123:12. 2

[YW10] YANG R., WÃIJNSCHE B. C.: Lifesketch - a framework for sketch-based modelling and animation of 3d objects. In *in âĂŸProceedings of the Australasian User Interface Conference (AUIC 2010* (2010), p. 2010. 1

[ZB94] ZHAO J., BADLER N.: Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics 13* (1994), 313–336. 2