Drafting 2D Characters with Primitive Shape Scaffolds

Golam Ashraf
School of Computing
National University of Singapore
gashraf@nus.edu.sg

Kaiser Md. Nahiduzzaman
School of Computing
National University of Singapore
kaisernahid@gmail.com

Nguyen Kim Hai Le
School of Computing
National University of Singapore
dcslnkh@nus.edu.sg

Li Mo
School of Computing
National University of Singapore
limo1985@gmail.com

Abstract **- Primitive shapes, namely circle, triangle and square, form the basis of design and cognition of a large number of objects we find around us. Inspired by this, we have recently proposed a primitive shape field representation that simplifies detailed convex shapes. In this paper, we apply this representation in image annotation, character drafting, and repurposing of 2D artwork. We implement a character design system that allows artists to sketch rough drafts using body-part scaffolding and then retarget a pre-annotated library image onto the draft mannequin. We also compare this algorithm with Free Form Deformation lattice deformers. This allows them to quickly create an estimate of the desired character without spending effort in inking and painting. The key technical algorithms presented here have a wide range of applications, such as structured shape design (humanoid cartoons, vehicles, consumer products, etc.), content retrieval, morphing, cartoonification/stylization, and procedural detailing (textures/shapes). The key technical contributions of this paper are vector fitting of strokes and a novel primitive cage field image-warping algorithm to warp articulated characters.**

Keywords - character design, shape scaffold, deformation, image warping.

I. INTRODUCTION

One of the first things children learn is to manipulate and express with primitive shapes. Even as adults, we naturally tend to decompose complex compositions into primitives. Basic shapes like triangles, circles and squares are so well understood, that even a textual/verbal description of structures in terms of these shapes elicits a natural visualization in our brain. Basic shapes play an important role in design drafts [2, 4, 7, 10, 11, 12, 19, 21]. For example, artists use shape scaffolding to pre-visualize the final form, using basic shapes to represent each component or part. Apart from establishing the volume and mass distribution of the figure, these shapes may also help portray a certain personality, as is widely seen in stylized cartoon drawings. For example, in Pixar's recent animated feature titled "UP", the main protagonist had distinctively square features to highlight his "cooped-in" life. The square features were amplified by contrasting with a large round nose, as well as distinctly rounded supporting characters. Depending on the art style, primitive shapes may become less apparent with the addition of details; e.g. clothes, accessories, and hair for humanoid figures.

The main contribution of this paper is the formulation of a continuous primitive-shape field to address art creation, manipulation and understanding. It is inspired by three strong potentials: a) Complex shapes are often cognitively processed as groups of primitives; b) Complex shape design often begins as a scaffolding of primitive shapes; c) Though details may obscure component shape cues in different degrees, and it may be hard to synthesize these details from bare scaffold versions, the underlying scaffold could provide useful anchors for manipulation. We feel that primitive shape fields add strong intuition to the manipulation and understanding of organic shapes. This paper provides implementation details on vector shape fitting of input-strokes (for intuitive art interface), and shape-field based image warping (for retargeting existing character art to the input drawing). The proposed algorithms produce compelling results with minimal setup and an uncomplicated interface.

Without losing generality, we limit the scope of the problem of designing complex shapes with a known structure to humanoid character drawings. The paper addresses practical challenges of implementing a 2D character visualizing system for pre-production artists. Artists usually create several draft drawings for each character, first with rough shape scaffolds, and then with details for a few promising ones. Our proposal can substantially speed up this process by allowing them to draw a rough shape scaffold, and quickly map appropriate detail templates from the image library. It also allows for filter and stroke based shape refinement on the drawn scaffold. This gives artists a lot of freedom, and takes away the tedium of manually detailing every prospective sample, most of which will be probably rejected by the art director anyways.

The rest of the paper is organized as followed. Section 2 has the related work to our paper. In Section 3, we will explain our approach and system in details. Section 4 will have some comparisons of our results with the Free Form Deformation (FFD) algorithm to prove our better performance in warping method regarding to discontinuity artifact caused by FFD.

II. RELATED WORK

Shape description and shaper representation is a well studied field because of its tremendous importance in pattern recognition and computer vision [5, 17, 23, 25]. These methods can be classified according to several criteria [14]. The first classification is based on the use of shape boundary points as opposed to the interior of the shape. These two approaches are known as external and internal, respectively. Another classification can be made according to whether the result is numeric or non-numeric. The scalar transform techniques map the image into an attribute vector description, while the space-domain techniques transform the input image into an alternative spatial domain representation. The third classification can be made on the basis of whether a transformation is information preserving or information losing. There is also an approach called mathematical morphology which is a geometrical based approach for image analysis [14]. It provides a potential tool for extracting geometrical structures and representing shapes in many applications. Inspired by all these developments and from the fact that primitive shapes like circle, triangle and squares play a central role in human perception we developed the shape descriptor with a scaled/rotated/blended combination of these three primitive shapes [15]. Our descriptor can approximate any convex shape with a mixture of these three primitives. Every arbitrary shape is represented as a vector of height, width, rotation, centroid-position and three weight values for circle, triangle, and rectangle.

Sketching: Schmidt et al. [25] explain the importance of the scaffolding technique in their review of sketching and inking techniques used by artists. In this method, artists construct characters from basic blocks representing different body parts. Our paper addresses this need for rapid abstraction of these basic blocks from rough strokes. Thorne et al. [35] proposed the concept of sketching for character animation, but do not include shape modeling. Orzan et al. [22] propose "Diffusion Curve" primitives for the creation of soft color-gradients from input strokes, along with an image analysis method to automatically extract Diffusion Curves from photographs. Schmidt et al. [26] propose “ShapeShop”, a 3D sketch authoring system generating implicit surfaces, with non-linear editing via a construction history tree. Although these curve-based methods are intuitive, they require a fair amount of detailing. Thus they are inappropriate for rapid drafting. Our primitive blocks are a lossy abstraction of detailed convex shapes, and thus are easier to represent, construct and perceive.

Deformation: Laplacian deformation allows user-specified tweaks to one or a few points on the deformable surface, to be smoothly propagated to the vicinity. The tweaks are treated as hard constraints and the aim is to find an optimal deformation to satisfy them [3, 14, 31, 32]. Igarashi et al. [14] first proposed an interactive system that lets user deform a two-dimensional shape using a variant of constrained Laplacian deformation. In this system the shape is represented by a triangle mesh and the user moves several vertices of the mesh as constrained handles. The system then computes the positions of the remaining free vertices by minimizing the distortion of each triangle. A two-step closed algorithm is used instead of physically based simulation in order to achieve real-time interaction.

By combining locally optimal block matching with as-rigid-as-possible shape regularization, Sykora et al. [33] proposed a geometrically motivated iterative scheme to register images undergoing large FFD and appearance variations. They also demonstrated the usability of their scheme in tasks required for the cartoon animation production pipeline including unsupervised tweening, example-based shape deformation, auto-painting, editing and motion retargeting. The embedding lattice in this system consists of several connected squares. In this case local rigid transformations are computed individually for each square and then the global smoothing step is used to ensure consistency. This extension is performed to enable more flexible deformation and to preserve local rigidity of the original shape. The algorithm produces similar results to [24] but allows smooth control over shape rigidity.

In FFD methods [27], the displacement of a cage control-point influences the entire space inside the lattice. However, specifying mesh deformations this way is both cumbersome and counterintuitive. Griessmair and Purgathofer [9] extended this technique to employ a trivariate B-spline basis. Though these methods are simple, efficient and popular in use, they suffer from the drawback of a restrictive original volume shape. Parallelepiped volumes rarely bear any visual correlation to the objects they deform and typically have a globally uniform lattice point structure that is larger than is required for the deformations to which they are applied. EFFD [5] is an improvement as it allows user-specified base-shapes, but manual lattice creation and deformation are still cumbersome [6].

MacCracken and Joy [18] use a volume equivalent of the Catmull-Clark subdivision scheme for surfaces to iteratively define a volume of space based on a control point structure of arbitrary topology. This is a significant step in increasing the admissible set of control lattice shapes. The technique is powerful and its only real shortcoming is the potential continuity problems of the mapping function (a combination of subdivision and interpolation) of points within the volume. The approach also suffers from the same discontinuity problems as Catmull-Clark surfaces at extraordinary vertices [30].

Exposing mathematical parameters for indirect manipulation via a GUI interface has two major disadvantages. Firstly, there is no intuitive connection between these parameters and the user-desired manipulation. Secondly, deformations defined using the handles of a specific representation cannot be trivially applied to other shape representations or even different instances of the same shape representation [1]. Integrated bone and cage deformation systems avoid potential artifacts that may arise in case of independent localized cages [34]. An interactive system that lets users move and deform a two-dimensional shape without manually establishing a skeleton or FFD domain beforehand was presented by [14].

Image Warping: Previous works use FFDs [8] or point features [16] to warp images. They typically use PCA to reduce the feature space, but the final deformation proceeds in the detailed triangulated mesh space. We propose a continuous pixel-based warp algorithm that does not need triangulation and hence avoids sampling and fold-over problems. We present some details for completeness here, and discuss additional details on resolving ambiguous pixels shared by multiple body parts. This algorithm provides a stable, fully automated and economical alternative to linear blend skinning based methods. Furthermore, we also compare its performance with FFD lattice deformation.

III. System and Implementation

We describe a system that allows body-part annotation of pre-existing orthographic character images, and then correctly retargets them to any valid draft scaffold sketched by the artist.

We briefly describe our prior work in shape representation outline in [15], and then describe the novel image warping algorithm, as we will develop on these to implement character image retargeting to draft scaffold drawings. The key technical contributions of this paper are vector fitting of outline strokes, and shape-field based image warping to resolve the artifact on human body warp.

A. *Algorithms*

Since we currently implement only front view drawings and images, we propose the following blocks to address these challenges: a) Vector shape representation that generates artifact-free continuous transitions between circle, triangle and square; b) Fitting a set of mouse/stylus generated strokes to the most representative vector shape for that body part; c) Vector field image warp that retargets library images to draft scaffolds.

As shown in Fig. 1, we store each of the three normalized primitive shapes as a set of eight quadratic Bezier curves. The solid points represent segment boundaries and the ragged blotches represent mid-segment control points. Note how a null segment (1-2) had to be created for the apex of the triangle. The reason why our piecewise curve segments work so well is that we were able to carefully identify the corresponding segments for the diverse topologies of circle, triangle and square. As a result, even under simple linear interpolation, we do not notice any tears or inconsistent shapes.

1) *Vector Shape Representation*

The normalized shapes can be affine transformed to any location, scale and rotation. Finally, the shape weights are applied to blend the corresponding Bezier control points, to yield an in-between shape. Note that start-end-mid control points of only corresponding segments are interpolated, as shown in Eqns. 1 and 2.

$$p'_j = \sum_{i=1}^{3} (w_i \cdot p_{i,j}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$m'_j = \sum_{i=1}^{3} (w_i \cdot m_{i,j}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where, $\sum_{i=0}^{2} w_i = 1$

And, $j \in \{1,2,3,4,5,6,7,8\}$

In the above equations, p'_j and m'_j represent the *j-th* blended segment boundary and midpoints respectively, while $p_{i,j}$, and $m_{i,j}$ represent the corresponding control points in the *i-th* primitive shape (circle, triangle, square). w_i is the weight contribution from the *i-th* primitive shape. Results of some blend operations are shown in Fig. 1. The cross hair under the shapes indicates the shape weights.

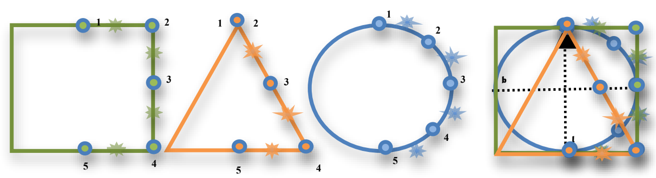

Figure 1. Consistent interpolation of circle, triangle, and square [15]

Results of some blend operations are shown in Fig. 2. The cross hairs under the shapes indicate the shape weights. With this background information about our primitive representation, we are now ready to describe vector fitting of stroked body-part line drawings. We assume that the input shapes are roughly symmetric about their medial axis, and generally convex.

2) *Vector Fitting*

A closed input stroke can be treated as a set of connected points, where the first and last points are fairly close to each other. We first resample the stroke at fixed angular intervals about the centroid of the input points. This helps avoid any bias due to variances in stylus pressure and stroke timing. A standard projection variance maximization algorithm, commonly employed to compute Oriented Bounding Boxes, is used to find the medial axis. In this algorithm, a ray is cast through the centroid, then all the boundary points are projected onto the ray, and the variance of the projected point distances from the centroid is noted. The ray that produces maximum variance is estimated to be the medial axis. Once the medial axis is noted, the axial-length and lateral-breadth of the shape can be easily calculated.

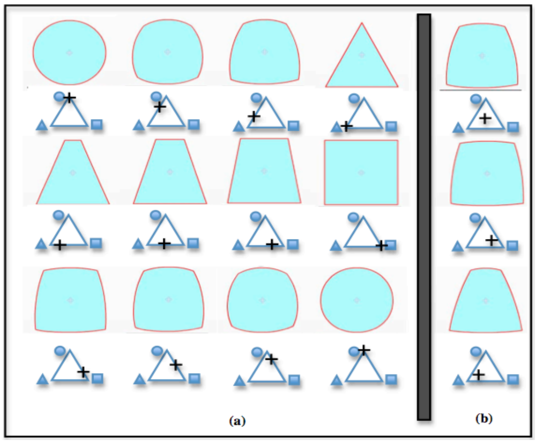

Figure 2. Blended shapes after consistent interpolation (shape weights indicated by cursor positions) [15]

We then perform a normalization affine transform to align the input shape to the Y-axis and scale it into a unit square. This simplifies shape error checking while ensuring rotation/translation/scale invariance during the fitting process. Lastly, we compute the best primitive shape combination, by minimizing boundary distance errors between our template shape combinations and the input points. In practice, this is a simple 2-level for-loop, incrementing shape weights by a fixed small value, and measuring the accumulated shape error. The shape error is calculated by accumulating slice-width errors over 40 lateral segments (along the medial axis). We have achieved decent fitting results for most cases. However, there are some cases where shapes computed with boundary distance errors do not match with human perception. We are currently working to improve the qualitative results through a perception regression model.

3) Space Parameterization

As shown in Fig. 3, we use a data structure {s,t} for parameterizing the cage and performing image warping, where t is a floating point number whose integral part holds the bezier segment number of the curve and s is the measurement of of distance along the line joining the center of a cage and the point on the bezier-segment-curve. Each pixel in Cartesian coordinates {x,y} can be easily converted into polar shape coordinates {s,t} and vice versa.

4) Image Warping

The challenge for articulated characters is that the limbs may be rotated significantly, and deformation along joints thus is a prime concern. We first explain the basic algorithm and then discuss how influences from multiple cages are resolved at overlapping and boundary regions.

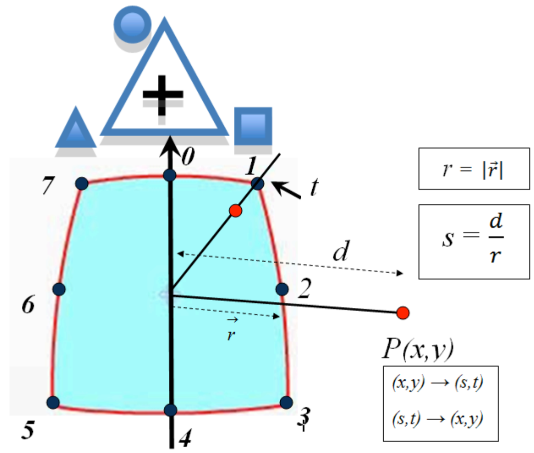

Figure 3. Polar Coordinate Parameterization of a Cage

The image based warp idea proceeds in a scan line manner, closely resembling texture fetches in the graphics pipeline. For each pixel *p* in the final image, an *{s,t}* coordinate is first calculated with respect to the corresponding vector cage. The source pixel p_0 can be extracted by converting the same polar coordinates in the corresponding source cage to Cartesian coordinates. The color for the morphed pixel can then be fetched from this source pixel.

Since a morphed pixel can be expressed as a member of multiple cages (due to the overlap among body parts, i.e. neck, torso and head), pixel *p* in the final image might end up being sourced from a few different pixels in the source image (due to different *{s,t}* coordinates in different cages). To resolve this issue, we perform distance-weighted influence blending of the different source pixel positions contributed by different cages. We avoid pixel color blending to prevent texture artifacts. The blending weights are derived as inversely proportional to pixel *p's* distance to the nearest boundary point on the associated cage, as shown in Eqn. 3, where d_i is the distance between *p* and the center of $cage_i$, and r_i is the corresponding center-boundary distance.

$$b_i = \frac{1}{d_i - r_i} \text{........(3)}$$

By doing this, we can effectively reduce undesirable stretching artifacts at boundaries between body parts and at joints undergoing large rotations. This simple weighting scheme does away with the need for manually specified deformation weights (e.g. in Linear Blend Skinning), and saves the artist a lot of extra manual work.

B. *System Implementation*

The system should be able to let artist annotate body parts of character images, and then to transform them into any hand-drawn scaffolds. Our system consists of three

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-110-6

main modules: a) Body part annotation; b) Scaffold sketching c) Image Retargeting. The interface has been designed for browsers to enable remote markup and drawings, with a server-side image-library.

1) Annotation

Fig. 3 shows the stroke annotation tool that can be used to trace out body parts. Each part is vector-fitted using least square error minimization after rotation and scale normalization, as described in Sec. A.2. The annotation process takes only a few minutes per image. We currently mark-up the following parts: head, neck, torso, upper arms, lower arms, hands, abdomen/hip, upper legs, lower legs and feet. We also construct a body silhouette to mark the whole body shape (blue outline in Fig. 4). The character image and its set of body cages are then stored into our image library, ready to be retargeted onto new drawings.

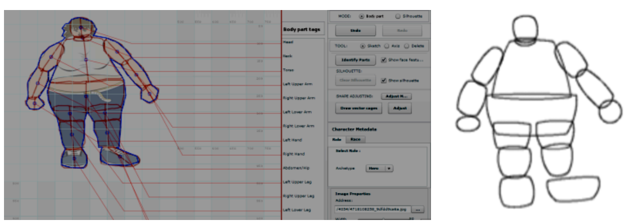

Figure 4. Manual Shape Annotation and Vector Fitting

2) Scaffold Sketching

Our system allows artist to design whole body scaffold that contains the same number of body parts. Based on the position of each part in relation with the others, it can support automatic deduction of body parts. However, the accuracy depends on the eccentricity of the proportions. The artist can easily correct wrongly annotated parts with a few mouse clicks in the drawing interface. A completed scaffold drawing is also automatically vector-fitted using the algorithm mentioned in Sec. A.2. Newly drawn scaffolds can be saved into a library for review and modification purposes. The whole process of scaffold design and image mark-up typically takes only a few minutes to complete.

3) Image Retargeting

Once the scaffold design is complete, the artist can then choose the character in the image library to retarget onto the scaffold. They can directly tweak source annotations to create deformations (as shown in Fig. 5), or they can retarget a selected (pre-annotated) image onto a newly drawn scaffold (as shown in Fig. 6). Fig. 7 illustrates a variety of scaffold retargeting results for two source character images. Results are obtained in real time, given the economical performance of the warping algorithm. We have achieved substantial acceleration for GPU implementation only for images above HD resolution.

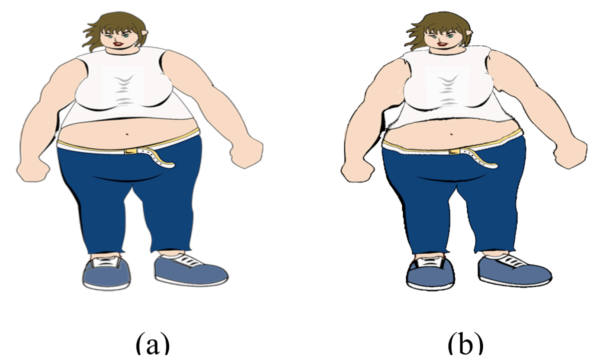

Figure 5. Interactive Deformation Mode: Original vector cage for torso in (a) made slimmer in (b) by tweaking the torso cage directly in vector space. (a) Shape(tri:36, sqr:12, cir:52); (b) Shape(tri:16, sqr:62, cir:22).

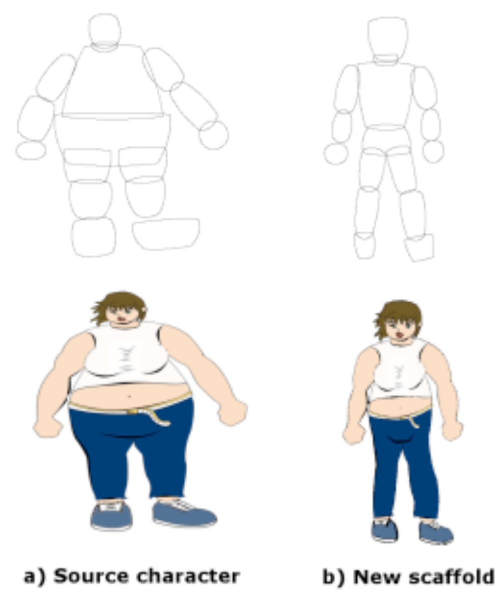

Figure 6. Warped output with a pair of source character image and new body scaffold. a) Source character image and its vector shape representation of body cages. b) Vector shape representation of artist's new scaffold and result produced by the system

As evident from the results, our mesh-less algorithm performs decently for retargeting between different shaped/proportioned characters. It can handle minor change in postures as well. We are currently working on improvements to the warping algorithm that allow drastic changes in postures, and also drawings from different views that will enable texture retargeting from 3D models.

IV. Comparison With Free Form Deformation

We now cite several desirable properties from deformation survey papers [1,6], and then do a head-to-head comparison with a well known deformation method.

Users expect deformation operations to preserve the shape of the object locally. They also wish that the deformation could be produced with intuitive operations, and with a minimal number of control handles. The resulting deformation should be smooth (differentiable) and predictable (can arrive at the same result for the given configuration, irrespective of the previous state). The algorithm should be efficient allowing real-time interactivity for at least a low-resolution model. Lastly, it is desirable to have shape operators to aide deformation tools for personality-driven, stylized character design.

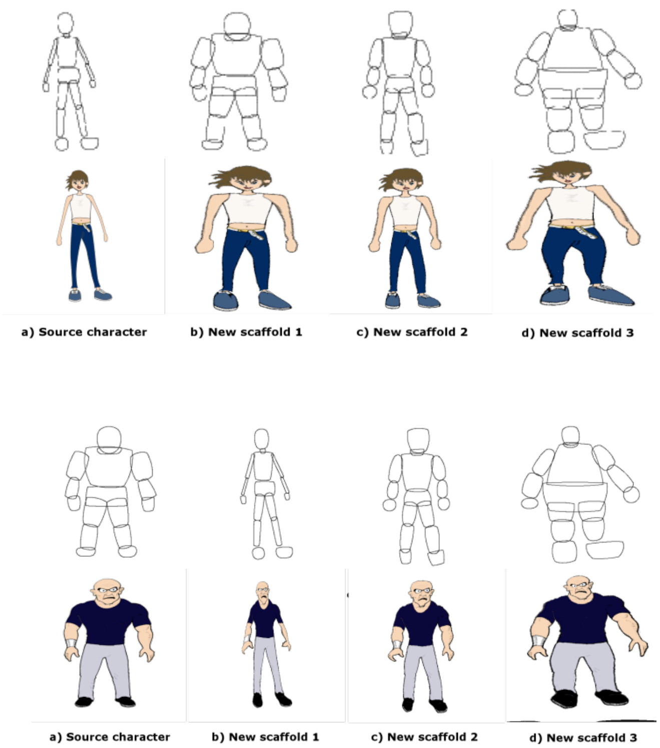

Figure 7. More examples of retargeting characters to draft scaffolds. a) Source image and its vector shape representation. b) c) and d) Result image with new scaffold 1, 2 and 3 from the artist.

In order to compare our results with FFD lattice deformers, we have implemented a shape-driven setup of FFD cage control points using Maya^{TM}. Since FFDs are setup as regular XY grids for 2D deformation, we can regularly sample our vector cage in *s-t* coordinate space and place the control points in corresponding Cartesian coordinate locations. For example, all boundary FFD points must lie on locations where $s=1$ for a given cage. One problem with regular FFDs is that the base deformer points must follow a regular grid pattern, causing accuracy problems when morphing from non-rectangular body shapes. An alternative would be to use more flexible representations like EFFD [1], but these come at the cost of multiple iterative computations.

Our warp algorithm gave better results than the FFD deformers, especially at cage boundaries. For example, in Fig. 8, the torso and upper arms are drawn unnaturally towards each other at the joints in the FFD case, for a triangular shape morph. Our warp algorithm better preserves the original area, as every pixel transformation is a weighted influence between all shape fields (as opposed to isolated FFD deformation with distance fall-off influences). This property is also evident in the Fig. 5b and Fig 7 (second row, last column), where abrupt discontinuities in the underlying cages are smoothly reflected in the warped image.

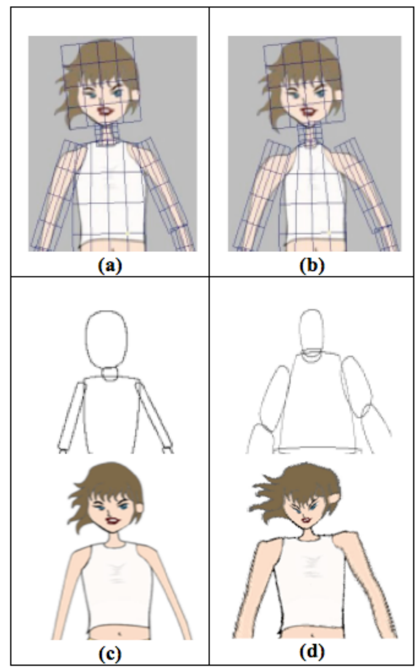

Figure 8. Comparison of FFD with our Primitive Cage Field deformation. a) Source image before applying FFD. b) Result image after apply FFD with artifact among torso and upper arms. c) Source image and its vector shape representation. d) Result image and its vector shape representation of artist's new scaffold with smooth connection among torso and upper arms.

V. Conclusion and Future Work

We have successfully demonstrated a system that allows artists to retarget a pre-annotated character image onto the draft scaffold. This allows them to quickly create an estimate of the desired character without spending effort in mapping detail templates. We have demonstrated decent quality results for a variety of scaffold proportions and shapes for three different humanoid characters. We have also shown how our warping algorithm yields more stable results than an equivalent FFD setup. Our method does not require any expensive mesh calculations, and is free from fold-over or hole artifacts, typical of mesh deformation methods.

We are currently working on a number of improvements: support for multi-stroked outlines, auto-segmentation of

Copyright (c) IARIA, 2010
ISBN: 978-1-61208-110-6

body part cages from a full-body outline drawing, and better warp robustness to large posture and view changes. We hope that these contributions will open up paths for more intuitive tools and easier character design.

ACKNOWLEDGMENT

This research is funded by MDA GAMBIT fund (WBS: R-252-000-357-490), sponsored by Media Development Authority of Singapore.

REFERENCES

[1] A. Angelidis , K. Singh, “Space deformations and their application to shape modeling,” ACM SIGGRAPH 2006 Courses, Boston, 2006.

[2] N. Beiman, “Prepare to Board! : Creating Story and Characters for Animated feature,” Focal Press, 2007.

[3] M. Botsch, M. Pauly, M. Wicke, and M. H. Gross, Adaptive space deformations based on rigid cells. Computer Graphics Forum 26, 3, 339–347, 2007.

[4] S. Camara, “All About Techniques in Drawing for Animation Production,” 1st ed. 2006, Barron’s Education Series, Inc.

[5] S. Coquillart, “Extended free-form deformation: A sculpturing tool for 3D geometric modeling,” Comput. Graph. 24, 4, 187–196.

[6] J. Gain and D. Bechmann, “A survey of spatial deformation from a user-centered perspective,” ACM Transactions on Graphics (TOG), v.27 n.4, p.1-21, October 2008.

[7] L. Garrett, “Visual design : A Problem-Solving Approach,” Huntington, N.Y., R. E. Krieger Pub. Co., 1975.

[8] B. Gooch, E. Reinhard and A. Gooch, “Human facial illustrations: Creation and psychophysical evaluation,” ACM Trans. Graph.23, 1 (2004), 27–44.

[9] J. Griessmair and W. Purgathofer, Deformation of solids with trivariate B-splines, Eurographics 89, 137–148.

[10] J. Hamm, “Cartooning the Head & Figure”.

[11] C. Hart, “Cartoon cool : How to Draw New Retro-Style Characters,” Watson-Guptill, 2005.

[12] C. Hart, “Simplified Anatomy for the Comic Book Artist : How to Draw the New Streamlined Look of Action-adventure Comics,” Watson-Guptill, 2007.

[13] L. N. K. Hai, W. Y. Peng and G. Ashraf, “Shape Stylized Face Caricatures,” unpublished.

[14] T. Igarashi, T. Moscovich, and J. F. Hughes, “As-rigid-aspossible shape manipulation,” ACM Trans. Graphics 24(3), 1134–1141 (2005).

[15] M. T. Islam, K. M. Nahiduzzaman, Y. P. Why and G. Ashraf, “Learning from Humanoid Cartoon Designs,” *Advances in Data Mining, Applications and Theoretical Aspects (LNCS Springer),* 6171:606-616, Berlin, 2010..

[16] J. Liu, Y. Chen and W. Gao, “Mapping Learning in Eigenspace for Harmonious Caricature Generation,” ACM Multimedia 2006: 683-686

[17] S. Loncaric, “A survey of shape analysis techniques,” Pattern Recognition 31 (1998), pp. 983–1001.

[18] R. MacCracken and K. Joy, “Free-form deformations with lattices of arbitrary topology,” In: SIGGRAPH 96 Conference Proceedings, pp. 181–188 (1996).

[19] M. D. Mattesi, “Force : Dynamic Life Drawing for Animators”.

[20] L. Moccozet and N. M. Thalmann, Dirichlet free-form deformations and their application to hand simulation. Computer Animation, 93–102, 1997.

[21] J. A. Mugnaini, “Drawing: A Search for Form”.

[22] A. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, and D. Salesin, “Diffusion Curves: A Vector Representation for Smooth-Shaded Images,” ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008), Volume 27 – 2008.

[23] T. Pavlidis, “A review of algorithms for shape analysis,” Comput. Graphics Image Process. 7 (1978) (2), pp. 243–258.

[24] A. R. Rivers, and D. L. James, FastLSM: Fast lattice shape matching for robust real-time deformation. ACM Transactions on Graphics 26, 3, 82, 2007.

[25] R. Schmidt, T. Isenberg, P. Jepp, K. Singh, and B. Wyvill, “Sketching, Scaffolding, and Inking: A Visual History for Interactive,” 3D Modeling, 2007.

[26] R. Schmidt, B. Wyvill, M.C. Sousa, and J.A. Jorge, “ShapeShop: Sketch-Based Solid Modeling with BlobTrees,” 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling, pp. 53-62, 2005.

[27] T. W. Sederberg and S. R. Parry, “Free-form deformation of solid geometric models,” Comput. Graph. 20, 4, 151–160.

[28] J. Serra, Image Analysis and Mathematical Morphology, Academic, New York (1982).

[29] A. Sheffer and V. Kraevoy, “Pyramid coordinates for morphing and deformation,” In: Proceedings of 3DPVT (2004).

[30] K. Singh, E. Kokkevis, Skinning Characters using Surface Oriented Free-Form Deformations. Graphics Interface 2000: 35-42. 2000.

[31] O. Sorkine, and M. ALEXA, As-rigid-as-possible surface modeling. In Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 109–116. 2007.

[32] R. W. Sumner, J. Schmid, and M. Pauly, Embedded deformation for shape manipulation. ACM Transactions on Graphics 26, 3, 80. 2007.

[33] D. Sýkora, J. Dingliana and S Collins, As-rigid-as-possible image registration for hand-drawn cartoon animations. NPAR 2009: 25-33. 2009.

[34] J. Tao, Z. Qian-Yi, P. Michiel, D. Cohen-Or, U. Neumann, “Reusable Skinning Templates Using Cage-based Deformations,” Dec 2008, Proceedings of ACM SIGGRAPH Asia 2008.

[35] M. Thorne and D. Burke, “Motion Doodles: An Interface for Sketching Character,” Motion University of British Columbia, 2004.

[36] R. Wang, “Image Understanding,” China. 1995.

[37] Y. Wang, K. Xu, Y. Xiong and Z.-Q. Cheng, 2D shape deformation based on rigid square matching. Computer Animation and Virtual Worlds 19, 3–4, 411–420, 2008.

[38] Y. Weng, W. Xu, Y. Wu, K. Zhou, and B. Guo, “2D shape deformation using nonlinear least squares optimization,” The Visual Computer 22(9), 653–660, 2006.