Hierarchical Spacetime Control of Linked Figures

Zicheng Liu and Steven J. Gortler and Michael F. Cohen

Department of Computer Science
Princeton University

These course notes are excerpted from *Hierarchical Spacetime Control*, by Zicheng Liu, Steven J. Gortler, and Michael F. Cohen, *SIGGRAPH, 1994*.

Abstract

Specifying the motion of an animated linked figure such that it achieves given tasks (e.g., throwing a ball into a basket) and performs the tasks in a realistic fashion (e.g., gracefully, and following physical laws such as gravity) has been an elusive goal for computer animators. The *spacetime constraints* paradigm has been shown to be a valuable approach to this problem, but it suffers from computational complexity growth as creatures and tasks approach those one would like to animate. The complexity is shown to be, in part, due to the choice of finite basis with which to represent the trajectories of the generalized degrees of freedom.

The functions through time of the generalized degrees of freedom are reformulated in a hierarchical wavelet representation. This provides a means to automatically add detailed motion only where it is required, thus minimizing the number of discrete variables. In addition the wavelet basis is shown to lead to better conditioned systems of equations and thus faster convergence.

1 Introduction

The spacetime constraint method, proposed in 1988 by Witkin and Kass [36], and extended by Cohen [7], has been shown to be a useful technique for creating physically based and goal directed motion of linked figures. The basic idea of this approach can be illustrated with a three-link arm and a ball (see Figure 4). The problem statement begins with specifying *constraints*, examples being specifying the position of the arm at a given time, requiring the ball to be in the hand (end effector) at time t_0, and that the arm is to throw the ball at time t_1 to land in a basket at time t_2. In addition, the animator must specify an *objective* function, such as to perform the tasks specified by the constraints with minimum energy or some other style consideration. The solution to such a series of specifications is a set of functions through time (or *trajectories*) of each degree of freedom (DOF), which in this case are the joint angles of the arm. Thus the unknowns span both space (the joint angles) and time, and have led to the term *spacetime constraints*.

Related approaches to the spacetime constraint paradigm are reported in [34, 25]. In each of these papers, feedback control strategies are the fundamental unknown functions rather than DOF trajectories. The goal is set, for example, for the creature to move in some direction as far as possible in 10 seconds, and a *score* for a particular motion is defined as the distance traveled. An initial control strategy is selected, a dynamic simulation is run and the results are scored. Iterations change the control strategy, as opposed the motion curves, producing a simulation that, hopefully, has a higher score. The results of these studies are encouraging, however, they are distinctly different from that in the previous spacetime constraint work (and the work described in this paper) in which the aim is to provide the animator with the overall control of the motion.

The spacetime constraint formulation leads to a non-linear constrained variational problem, that in general, has no closed form solution. In practice, the solution is carried out by reducing the space of possible trajectories to those representable by a linear combination of basis functions such as cubic B-splines. Finding the finite number of coefficients for the B-splines involves solving the related constrained optimization problem, (i.e., finding the coefficients to create motion curves for the DOF that minimize the objective while satisfying the constraints). Unfortunately, general solutions to such a non-linear optimization problem are also unknown.

Based on this observation, Cohen developed an interactive spacetime control system using hybrid symbolic and numeric processing techniques [7]. In this system, the user can interact with the iterative numerical optimization and

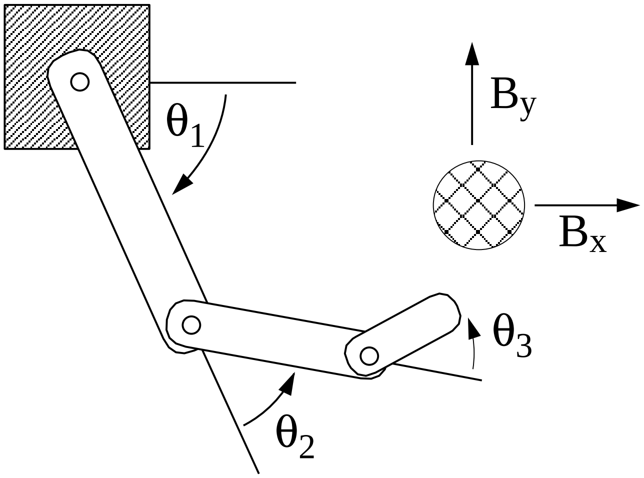

Figure 4: A planar three-link arm.

can *guide* the optimization process to converge to an acceptable solution. One can also focus attention on subsets or *windows* in spacetime. This system produces physically based and goal directed motions, but it still suffers from a number of computational difficulties, most notably as the complexity of the creature or animation increases.

An important difficulty in the spacetime system is that the user is required to choose the discretization of the B-spline curves. If not enough control points are selected, there may be no feasible solution (i.e., one that meets all constraints), or the restriction to the curve is so severe, that the resulting motion curves have a much higher objective cost than necessary. If too many control points are selected, then the computational complexity is increased unnecessarily due to the larger number of unknowns as well as the resulting ill-conditioning of the linear subproblems that arise in the solution [32]. This complexity issue is addressed by reformulating the DOF functions in a *hierarchical* basis, in particular, in a B-spline wavelet (B-wavelet) basis. Wavelets provide a natural and elegant means to include the proper amount of local detail in regions of spacetime that require the extra subdivision without overburdening the computation as a whole.

2 System overview

The interactive spacetime control system is shown in Figure 5. Input to the system includes user defined constraints and objectives and a creature description from which the symbolic equations of motion are generated automatically. The equations of motion define the torque at each joint as a function of the position and velocity of all joints as well as physical properties such as mass and length of the links. These expressions for torque are central to the definition of a minimum energy objective. The expressions are next symbolically differentiated and compiled to create concise evaluation trees.

The main focus of the current discussion is on the next section, the numerical process that solves for the coefficients of the chosen B-spline or hierarchical wavelet basis. Finally, the intermediate and final animations are displayed graphically. The animator can simply watch the progress of the optimization procedure or can interact directly with the optimization by creating starting motion curves for the DOF and/or by modifying intermediate solutions.

3 Wavelets

An elegant and concise hierarchical basis, and one that leads naturally to an adaptive basis, is offered by a *wavelet* construction. This section concentrates on the advantages of wavelets and wavelet formulations in the spacetime animation problem.

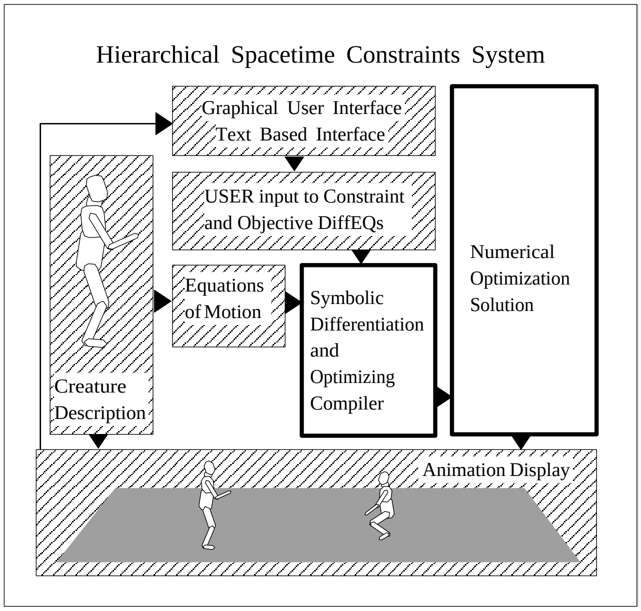

Figure 5: The Hierarchical Spacetime Constraints System. This paper focuses on the Symbolic Differentiation and Optimizing Equation Compiler, and the Numerical Optimization System

The wavelet construction results in a non-redundant basis that provides the means to begin with a low resolution basis and then *adaptively refine* the representation layer by layer when necessary without changing the representation above. If refinements are required in only part of the interval, then only those coefficients whose bases have support in this region need to be added.

Since the wavelet coefficients encode differences, in smooth portions of the trajectory the coefficients encoding finer scale detail will be zero. Thus, only those basis functions with resulting coefficients greater than some ϵ will have a significant influence on the curve and the rest can be ignored. In other words, given an *oracle* function [15, 14], that can predict which coefficients will be above a threshold, only the corresponding subset of wavelets needs to be included.

Solutions to the non-linear spacetime problem, involve a series of quadratic subproblems for which the computational complexity depends on the number of unknown coefficients. The smaller number of significant unknown coefficients in the wavelet basis provide faster iterations. In addition, the wavelet basis provides a better conditioned system of equations than the uniform B-spline basis, and thus requires less iterations. The intuition for this lies in the fact that there is no single basis in the original B-spline basis that provides a global estimate of the final trajectory (i.e., the locality of the B-spline basis is, in this case, a detriment). Thus, if the constraints and objective do not cause interactions across points in time, then information about changes in one coefficient travels very slowly (in $O(n)$ iterations) to other parts of the trajectory. In contrast, the hierarchical wavelet basis provides a shorter ($O(\log(n))$) "communication" distance between any two basis functions. This is the basic insight leading to *multigrid* methods [32], and the related hierarchical methods discussed here.

The wavelet representation also allows the user to easily lock in the coarser level solution and only work on details simply by removing the coarser level basis functions from the optimization. This provides the means to create small systems that solve very rapidly to develop the finest details in the trajectories.

3.1 B-wavelets

In the literature, there are many wavelet constructions, each with its own particular functions ϕ an ψ, with varying degrees of orthogonality, compactness, and smoothness. The particular wavelet construction used in this work are derived in [5], and were chosen because of the semi-orthogonality of the basis, the associated ϕ is a cubic B-spline (i.e., C^2), and the wavelet function ψ is symmetric with compact support.

3.2 Wavelets on the Interval

In a classical wavelet construction, the domain goes from $-\infty \dots \infty$. In an animation context, only functions over some fixed finite interval of time need to be expressed, and it is important to only deal with a finite number of basis functions. Therefore, the function space V_L used here is defined to be the space of all C^2 functions defined over the interval $[0 \dots 2^L]$ that are piecewise cubic between adjacent integers (simple knots at the inner integers and quadruple knots at the boundaries). A basis for V_L is made up of *inner* basis functions, which are just those translational B-spline basis functions $\phi_{L,j}$ whose support lies completely within the interval, as well as three special *boundary* B-spline basis functions at each end of the interval. For the boundary basis functions, one may either choose to include the translational basis functions $\phi_{L,j}$ themselves whose support intersects the boundaries by just truncating those basis functions at the boundary, or else one may use the special boundary basis functions that arise from placing quadruple knots at the boundaries [2]. This complete set of basis functions will be denoted $\phi_{L,j}$ with j in $\{-3 \dots 2^L - 1\}$, where it is understood that the first and last three basis functions are the special boundary B-spline basis functions.

A two-part basis for V_L can be constructed with the wider B-spline functions $\phi_{L-1,j}$ with j in $\{-3 \dots 2^{L-1} - 1\}$ where again the first and last three basis functions are scaled versions of the special boundary B-splines functions. The two-part basis is completed with the wavelet functions $\psi_{L-1,j}$ with j in $\{-3 \dots 2^{L-1} - 4\}$. Here too, the *inner* wavelet basis functions are just those translational functions $\psi_{L-1,j}$ that do not intersect the boundaries, while the first three and the last three interval wavelet basis functions must be specially designed to fit in the interval and still be orthogonal to the $\phi_{L-1,j}$. A full description of this construction is given in [4, 28].

3.3 Completing the Wavelet Basis

The reasoning that was used to construct the two-part basis can now be applied recursively $L - 3$ times to construct a multilevel *wavelet basis*. Noting that roughly half of the basis functions in the two-part basis are themselves B-spline basis functions (only twice as wide), to continue the wavelet construction, keep the basis functions $\psi_{L-1,j}$ and re cursivelyapply the reasoning above to replace the $\phi_{i,j}$ with $\{\phi_{i-1,j}, \psi_{i-2,j}\}$.

Each time this reasoning is applied, the number of B-spline functions in the hierarchical basis is cut in half (roughly), and the new basis functions become twice as wide. After $L - 3$ applications, the wavelet basis

$$\{\phi_{3,k}, \psi_{i,j}\} \tag{6}$$

is obtained, with i in $\{3 \dots L-1\}$, k in $\{-3 \dots 7\}$ and j in $\{-3 \dots 2^i - 4\}$, where the inner basis functions are defined by

$$\begin{aligned} \phi_{i,j}(t) &= \phi(2^{(i-L)}t - j) \\ \psi_{i,j}(t) &= \psi(2^{(i-L)}t - j) \end{aligned} \tag{7}$$

This basis is made up of eleven wide B-splines, and translations (index j) and scales (index i) of the wavelet shape (as well as scales of the boundary wavelet basis functions).

The wavelet basis is an alternate basis for V_L, but unlike the B-spline basis, it is an $L - 3$ level hierarchical basis. At level 3 there are eleven broad B-splines, and eight broad wavelets. These basis functions give the coarse description of the function. At each subsequent level going from level 3 to $L - 1$, the basis includes twice as many wavelets, and these wavelets are twice as narrow as the ones on the previous level. Each level successively adds more degrees of detail to the function.

Since each wavelet coefficients represents the amount of local detail of a particular scale, *if the function is sufficiently smooth in some region, then very few non-zero wavelet coefficients will be required in that region*[4].

[4] In this case, non-zero can be defined to be having an absolute value greater than some epsilon without incurring significant error in the representation.

3.4 Scaling

One final issue is the scaling ratio between the basis functions. Traditionally [5] the wavelet functions are defined with the following scaling:

$$\begin{aligned} \phi_{i,j}(t) &= 2^{(i-L)/2}\ \phi(2^{(i-L)}t - j) \\ \psi_{i,j}(t) &= 2^{(i-L)/2}\ \psi(2^{(i-L)}t - j) \end{aligned} \tag{8}$$

This means that at each level up, the basis functions become twice as wide, and are scaled $\frac{1}{\sqrt{2}}$ times as tall. While in many contexts this normalizing may be desirable, for optimization purposes it is counter productive. For the optimization procedure to be well conditioned [8] it is advantageous to emphasize the coarser levels and hence use the scaling defined by

$$\begin{aligned} \phi_{i,j}(t) &= 2^{L-i}\ \phi(2^{(i-L)}t - j) \\ \psi_{i,j}(t) &= 2^{L-i}\ \psi(2^{(i-L)}t - j) \end{aligned} \tag{9}$$

where the wider functions are also taller.

4 Implementation

The input to the wavelet spacetime problem includes the creature description, the objective function (i.e., symbolic expressions of joint torques generated from the creature description), and user defined constraints specifying desired actions (throw, catch, etc.), and inequality constraints such as joint limits on the elbow.

Each trajectory of a DOF, $\theta(t)$, is represented in the uniform cubic B-spline basis. The unknowns are then the B-spline coefficients, $\mathbf{b}$, or the equivalent wavelet coefficients, $\mathbf{c}$, scaling the individual basis functions. This finite set of coefficients provide the information to evaluate the $\theta(t)$, $\dot{\theta}(t)$, and $\ddot{\theta}(t)$ at any time t, that comprise the leaves of the DAGs. This finite representation transforms the variational problem into a constrained non-linear optimization problem. An unconstrained problem can then be derived by penalizing violations to the constraints.

A quasi-Newton method, BFGS [10], is used to solve the resulting non-linear problem. Iterations begin with a user provided initial guess of wavelet coefficients (that can be derived from B-spline coefficients) and a guess of the inverse of the Hessian (usually an identity matrix leading to the first iteration being a simple gradient descent).

Each subsequent iteration involves finding the gradient of the modified constraint/objective function and performing a matrix-vector multiply. The newly obtained solution is then transformed into B-spline coefficients and sent to the graphical user interface for display.

If the initial function space is restricted to a coarse representation consisting of the broad B-splines and a single level of wavelets, after each iteration a simple *oracle* function adds wavelets at finer levels only when the wavelet coefficient above exceeds some tolerance. This procedure quickly approximates the optimal trajectory and smoothly converges to a final answer with sufficient detail in those regions that require it.

An important feature of the system discussed in [7] is also available in the current implementation. The user can directly modify the current solution with a simple key frame system to help *guide* the numerical process. This is critical to allow the user, for example, to move the solution from an underhand to an overhand throw, both of which represent local minima in the same optimization problem. The next iteration then begins with these new trajectories as the current guess.

5 Results

A set of experiments was run on the problem of a three-link arm and a ball (see Figure 4). The goal of the arm is to begin and end in a rest position hanging straight down, and to throw the ball into a basket. The objective function is to minimize energy, where energy is defined as the integral of the sum of the squares of the joint torques. Gravity is active.

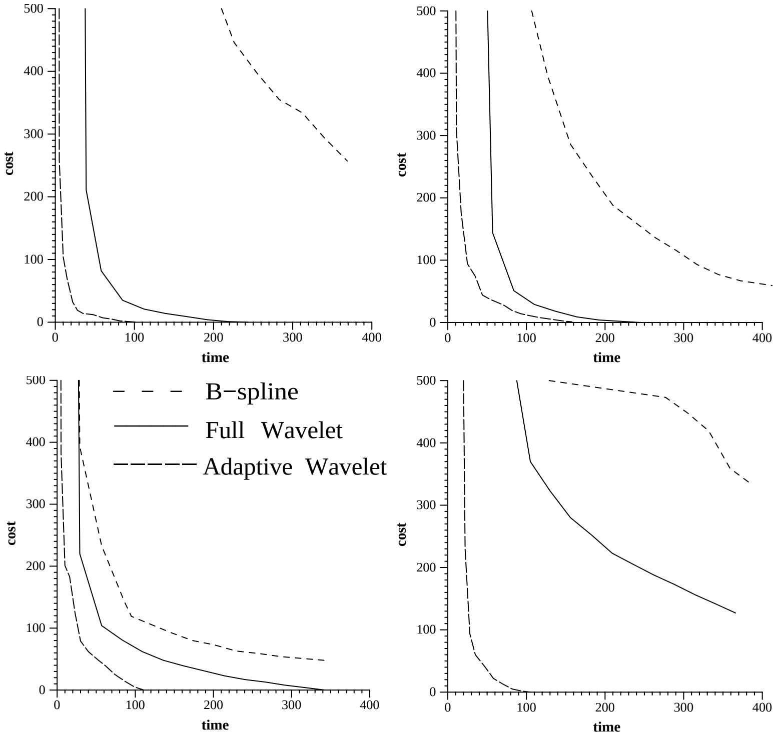

Figure 6: Convergence of Arm and Ball example for 4 different starting trajectories. The first and fourth examples resulted in underhand throws, and the rest overhand. Time is in seconds, and the cost is a weighted sum of constraint violations and energy above the local minimum.

The four graphs in Figure 6 show the convergence of five different test runs of the arm and ball example. Each plot differs only in the starting trajectories of the arm DOF. Each run converged to either an underhand or overhand throw into the basket. The full B-spline basis contained 67 basis functions for each of the three DOF, thus there were 201 unknown coefficients to solve for. Iterations took approximately 7 seconds each on an SGI workstation with an R4000 processor. Convergence was achieved on each, but only after many iterations due to the ill-conditioning of the B-spline formulation.

The full wavelet basis also contained 67 basis function per DOF (11 B-splines at the top level and 56 wavelets below), thus iterations also took approximately the same 7 seconds. Figure 6 clearly shows the improved convergence rates of the wavelet formulations over the B-spline basis, due to better conditioned linear systems. The adaptive wavelet method with the oracle was the fastest since the number of unknowns was small in early iterations, leading to a very fast approximation of the final trajectories, in addition to the better conditioning provided by the hierarchical basis. The final few iterations involved more wavelets inserted by the oracle to complete the process. Note that in each case, a good approximation to the complete animation was achieved in less than a minute of computation.

6 Conclusion

The spacetime constraint system first suggested by Witkin and Kass [36] for animating linked figures has been shown to be an effective means of generating goal based motion. Cohen enhanced this work by demonstrating how to focus the optimization step on *windows* of spacetime and methodologies to keep the user in the optimization loop. These notes discuss extentions to this paradigm by removing two major difficulties.

A major improvement lies in the representation of the trajectories of the DOF in a wavelet basis. This resulted in faster optimization iterations due to less unknown coefficients needed in smooth regions of the trajectory. In addition, even with the same number of coefficients, the systems become better conditioned and thus less iterations are required to settle to a local minimum. Results are shown for a planar three-link arm.

Acknowledgements

The authors owe a debt to Charles Rose who implemented the user interface for this work. This research was supported in part by the National Science Foundation, Grant CCR-9296146.

References

[1] BARTELS, R., AND BEATTY, J. A technique for the direct manipulation of spline curves. In *Graphics Interface 1989* (1989), pp. 33–39.

[2] BARTELS, R., BEATTY, J., AND BARSKY, B. *An Introduction to Splines for Use in Computer Graphics and Modeling.* Morgan Kaufmann, 1987.

[3] CELNIKER, G., AND GOSSARD, D. Deformable curve and surface finite-elements for free-from shape design. *Computer Graphics 25*, 4 (July 1991), 257–266.

[4] CHUI, C., AND QUAK, E. Wavelets on a bounded interval. *Numerical Methods of Approximation Theory 9* (1992), 53–75.

[5] CHUI, C. K. *An Introduction to Wavelets*, vol. 1 of *Wavelet Analysis and its Applications*. Academic Press Inc., 1992.

[6] COHEN, A., DAUBECHIES, I., AND FEAUVEAU, J. C. Biorthogonal bases of compactly supported wavelets. *Communication on Pure and Applied Mathematics 45* (1992), 485–560.

[7] COHEN, M. F. Interactive spacetime control for animation. *Computer Graphics 26*, 2 (July 1992), 293–302.

[8] DAHMEN, W., AND KUNOTH, A. Multilevel preconditioning. *Numerische Mathematik 63* (1992), 315–344.

[9] FINKELSTEIN, A., AND SALESIN, D. Multiresolution curves. In *Computer Graphics, Annual Conference Series, 1994* (1994), Siggraph, pp. 261–268.

[10] FLETCHER, R. *Practical Methods of Optimization*, vol. 1. John Wiley and Sons, 1980.

[11] FORSEY, D., AND BARTELS, R. Hierarchical b-spline refinement. *Computer Graphics 22*, 4 (August 1988), 205–212.

[12] FORSEY, D., AND WENG, L. Multi-resolution surface approximation for animation. In *Graphics Interface* (1993).

[13] FOWLER, B. Geometric manipulation of tensor product surfaces. In *Proceedings, Symposium on Interactive 3D Graphics* (1992), pp. 101–108.

[14] GORTLER, S., AND COHEN, M. F. Hierarchical and variational geometric modeling with wavelets. *1995 Symposium on Interactive 3D Graphics", year =*.

[15] GORTLER, S., SCHRÖDER, P., COHEN, M., AND HANRAHAN, P. Wavelet radiosity. In *Computer Graphics, Annual Conference Series, 1993* (1993), Siggraph, pp. 221–230.

[16] GORTLER, S. J. *Wavelet Methods for Computer Graphics*. PhD thesis, Princeton University, January 1995.

[17] HALSTEAD, M., KASS, M., AND DEROSE, T. Efficient, fair interpolation using catmull-clark surfaces. In *Computer Graphics, Annual Conference Series, 1993* (1993), Siggraph, pp. 35–43.

[18] JAFFARD, S., AND LAURENÇOT, P. Orthonormal wavelets, analysis of operators, and applications to numerical analysis. In *Wavelets: A Tutorial in Theory and Applications*, C. K. Chui, Ed. Academic Press, 1992, pp. 543–602.

[19] LIU, Z., GORTLER, S. J., AND COHEN, M. F. Hierarchical spacetime control. In *Computer Graphics, Annual Conference Series, 1994* (August 1994), pp. 35–42.

[20] LOUNSBERY, M., DEROSE, T., AND WARREN, J. Multiresolution analysis for surfaces of arbitrary topological type. Tech. Rep. TR 93-10-05b, Department of Computer Science and Engineering, Princeton University, October 1993.

[21] LYCHE, T., AND MORKEN, K. Spline wavelets of minimal support. In *Numerical Methods in Approximation Theory*, D.Braess and L. L. Schumaker, Eds., vol. 9. Birkhauser Verlag, Basel, 1992, pp. 177–194.

[22] MALLAT, S. G. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE PAMI 11* (July 1989), 674–693.

[23] MEINGUET, J. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics (ZAMP) 30* (1979), 292–304.

[24] MORETON, H., AND SEQUIN, C. Functional optimization for fair surface design. *Computer Graphics 26*, 4 (July 1992), 167–176.

[25] NGO, J. T., AND MARKS, J. Spacetime constraints revisited. In *Computer Graphics, Annual Conference Series, 1993* (August 1993), Siggraph, pp. 343–350.

[26] PENTLAND, A. Fast solutions to physical equilibrium and interpolation problems. *The Visual Computer 8*, 5 (1992), 303–314.

[27] QIAN, S., AND WEISS, J. Wavelets and the numerical solution of partial differential equations. *Journal of Computational Physics 106*, 1 (May 1993), 155–175.

[28] QUAK, E., AND WEYRICH, N. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval. Tech. Rep. 294, Center for Approximation Theory, Texas A&M, 1993.

[29] RANDO, T., AND ROULIER, J. Designing faired parametric surfaces. *Computer Aided Design 23*, 7 (September 1991), 492–497.

[30] SZELISKI, R. Fast surface interpolation using hierarchical basis functions. *IEEE PAMI 12*, 6 (June 1990), 513–439.

[31] TERZOPOULOS, D. Image analysis using multigrid relaxation methods. *IEEE PAMI 8*, 2 (March 1986), 129–139.

[32] TERZOPOULOS, D. Image analysis using multigrid relaxation methods. *IEEE PAMI 8*, 2 (March 1986), 129–139.

[33] TERZOPOULOS, D. Regularization of inverse visual problems involving discontinuities. *IEEE PAMI 8*, 4 (July 1986), 413–424.

[34] VAN DE PANNE, M., AND FIUME, E. Sensor-actuator networks. In *Computer Graphics, Annual Conference Series, 1993* (August 1993), Siggraph, pp. 335–342.

[35] WELCH, W., AND WITKIN, A. Variational surface modeling. *Computer Graphics 26*, 2 (July 1992), 157–166.

[36] WITKIN, A., AND KASS, M. Spacetime constraints. *Computer Graphics 22*, 4 (August 1988), 159–168.

[37] YSERENTANT, H. On the multi-level splitting of finite element spaces. *Numerische Mathematik 49* (1986), 379–412.