THE JOURNAL OF VISUALIZATION AND COMPUTER ANIMATION, VOL. 4: 153–163 (1993)

# Sequential-goal Constraints for Computer Animation

WEN-BANG LIU
*Department of Computer and Information Science, National Chiao-Tung University, Hsinchu, Taiwan*

MING-TAT KO
*Institute of Information Science, Academia Sinica, 128 Yen-Chiu-Yuan Road, Sec. II, Taipei, 11529, Taiwan*

AND

RUEI-CHUAN CHANG
*Department of Computer and Information Science, National Chiao-Tung University, Hsinchu, Taiwan*

## SUMMARY

**The dynamic constraints technique has been proposed for building geometrical models composed of rigid bodies, which are made to act naturally, according to Newtonian laws, by specifying constraints on their states. In computer animation, the dynamic constraints technique alleviates the work-load of animators who formerly had to plan animated sequences in detail by intuition alone. Nevertheless, for some real-world applications, it is desirable to have a mechanism that makes physically-based elements move according to a given scenario by providing some control states. The control states can be represented by transient constraints that are to be met and then released immediately. In this paper, a technique called the sequential-goal constraints technique is proposed to provide such a mechanism. With the sequential-goal constraints technique, it is easy to specify transient constraints according to a given scenario and derive proper forces and torques to drive an element to meet each transient constraint exactly at a specified time so that the whole motion of the element is continuous and integral.**

KEY WORDS Animation Dynamic constraints Inverse dynamics Physically-based model

## 1. INTRODUCTION

In traditional animation techniques,[1] objects are modelled purely by using their geometrical characteristics, such as position, orientation and shape. The idea behind animating an object is simply putting the object in the desired position at the appropriate time. That is, the motion of objects is developed through the animator's intuition by using detailed descriptions of the position, orientation and shape of the objects. Though tedious, this approach is easy to use, and the concept behind it is simple. When dealing with complex compound objects, however, animation using traditional techniques becomes virtually impossible. Moreover, the animation effects produced by such traditional techniques are usually not realistic.[2] Most animators

1049–8907/93/030153–11$10.50

*Received 2 July 1991*
*Revised 24 October 1992*

would prefer to focus their attention on high-level designation of object actions and to be free of the need to provide a detailed description of object behaviour. A more advanced technique providing more realistic animated pictures can help them to satisfy these desires.

The *dynamic constraints technique*,[3–9] proposed by Barr *et al.* in 1988, is a technique that partially satisfies the desire of animators for objects composed of rigid bodies. In the dynamic constraints technique, physically-based elements (objects whose motion is driven by forces and torques according to Newtonian laws) are used to develop realistic animated pictures. Users may assemble designated compound elements by specifying certain geometrical constraints between elements, such as *point-to-nail*, *point-to-point*, and so on. Using inverse dynamic techniques, the mechanism automatically derives proper constraint forces and torques over time to drive elements to meet the specified constraints; it then maintains those constraints forever. As long as the constraints are maintained, the compound element responds to external forces in the environment naturally and realistically.

When using dynamic constraints, the emphasis is on building compound elements that behave naturally, and the behaviour of elements in the assembly phase (that is, the time from the initial states of the elements to the point when they begin to meet the constraint) is of less concern. As it happens, in dynamic constraints, the behaviour in the assembly phase is straightforward and the only factor that can be adjusted is the assembly rate.

However, some real-world applications require more sophisticated control over elements in the assembly phase. For example, in a self-assembly mechanism system using dynamic constraints, the behaviour of components should be controlled in order to avoid improper collision. Moreover, one may want a physically-based element to perform any given action scenario. For example, just for dramatic effect, one might want to have an element turn a somersault somewhere in its motion. In a case where a scene with interactions among several elements is to be generated, say a bird catching a fish at certain place, the elements should be controlled so that they move to the same place at the same time. In other words, planning of the behaviour of elements driven by constraint forces is needed. In addition to the path of movement of elements, sometimes the timing of how they pass certain critical positions must be specified and fulfilled.

Behaviour planning can be implemented, in a general form, by specifying a sequence of transient constraints $c_i$, $i = 1,\ldots,n$, in temporal order, with some of them specified as time-critical (to be met precisely at a specified time). In the course between consecutive constraints, say $c_i$ and $c_{i+1}$, the element is released from constraint $c_i$ and is on its way to meeting constraint $c_{i+1}$. As the constraints are sucessively met and released, the elements should move in accordance with the designated behaviour.

Consider constraints between consecutive time-critical constraints, say $c_i$ and $c_j$ with specified time $t_i$ and $t_j$, respectively. Since the times at which the constraints to be met are not critical, the routeing time, $t_j - t_i$, is usually distributed to courses between $c_k$ and $c_{k+1}$, $i \le k \le j - 1$, by the animator's intuition or by the distance between consecutive constraints. Routeing times for courses can be allocated in order to make the elements move in a designated rhythm.

Finally, one may assume that a sequence of constraint–time pairs $(c_i, t_i)$, $t_i < t_{i+1}$, $i = 1,\ldots,n - 1$, is given. These pairs are called *goals*. The problem, called the *sequential-goal problem*, is to provide a mechanism for deriving proper driving forces and torques for the constrained element so that the element will move to meet constraint $c_i$ exactly (exactness) at a specified time $t_i$ (punctuality), $i = 1,\ldots n$, and so that the entire behaviour is continuous and integral (integrity). For goals at which interaction between elements occurs, exactness and punctuality are rigorously required. Although the entire behaviour is achieved by a sequence of transient constraints, it has to be integral from an outsider's point of view. In general, the behaviour along the course between pairs of consecutive constraints should be continuous and natural. The main effort that must be expended lies in linking the behaviour in the transition from course to course in a continuous and natural way. Continuity and integrity of motion are basic requirements in producing natural animated sequences.

In this paper, a mechanism that we call *sequential-goal constraints* is proposed to solve the sequential-goal problem. Since there is a sequence of transient constraints to be met, it seems that the problem can be resolved simply by applying dynamic constraints repeatedly. However, as we will see later, this simple strategy cannot provide element behaviour of the required integrity, exactness and punctuality.

The rest of this paper is organized as follows. In Section 2, we review the dynamic constraints technique and discuss its feasibility for solving the sequential-goal problem. In Section 3, a new behaviour differential equation is introduced to provide a mechanism for solving the sequential-goal problem. Section 4 presents an animated sequence developed using the sequential-goal constraints technique. Concluding remarks are presented in the last section.

## 2. DYNAMIC CONSTRAINTS AND THE SEQUENTIAL-GOAL PROBLEM

In the dynamic constraints approach proposed by Barr *et al.*,[7] a constraint is a specified configuration of states of elements in the model. The states of elements are composed of their geometrical attributes: position and orientation. A point-to-nail constraint, for example, specifies that a certain point on an element must coincide with a given nail (a position in the world space). The position and orientation of the $i$th element in the model are denoted by $X_i(t)$ and $R_i(t)$, respectively. The state of the model is thus represented by

$$Y_s(t) = \{X_1(t), R_1(t), \ldots, X_n(t), R_n(t)\}$$

To measure the deviation of the model at time $t$ from a given constraint, a *constraint deviation function* $\mathbf{D}(Y_s(t),t)$, a vector with any number of components, is defined such that $\mathbf{D}(Y_s(t),t) = 0$ if and only if the constraint is met. For example, for a point-to-nail constraint, $\mathbf{D}(t) = \mathbf{X}(t) - \mathbf{X}_0$, where $\mathbf{X}(t)$ and $\mathbf{X}_0$ are the co-ordinates of the constrained point and the nail, respectively. The deviation function $\mathbf{D}(Y_s(t),t)$, with non-zero initial value typically, is required to decrease to zero over a period of time in order to meet the constraint and then stay at zero to maintain conformance to the constraint.

Second-order differential equations are used to describe the behaviour of deviation functions. The terms in the first-order differentiation of $\mathbf{D}(Y_s(t),t)$ with respect to time are interpreted to be the linear momenta and angular momenta of constrained elements, and those in the second-order differentiation of $\mathbf{D}(Y_s(t),t)$ are interpreted to be forces and torques. With this interpretation, a system of linear equations on forces and torques for constraints can be derived. The constraint forces (needed to propel the element so that it meets all the constraints and satisfies the behaviour differential equations) can then be derived by solving this system of equations.

In dynamic constraints, deviation functions must decrease exponentially to zero and remain at zero from then on. Let $D(t)$ be a component of a deviation function. The behaviour differential equation for $D(t)$ used by Barr *et al.*[7] is

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}D(t) + \frac{2}{\tau}\frac{\mathrm{d}}{\mathrm{d}t}D(t) + \frac{1}{\tau^2}D(t) = 0 \qquad \text{for } t > t_0 \tag{1}$$

with initial conditions

$$D(t_0) = D_0\,, \qquad \frac{\mathrm{d}}{\mathrm{d}t}D(t_0) = D_0'$$

where $D_0$ is the initial state of $D(t)$ and the parameter $\tau$ is used to control the rate at which the constraint is met.

The deviation function according to equation (1) is

$$D(t) = \mathrm{e}^{-t/\tau}(C_1 + C_2 t) \tag{2}$$

where $C_1$ and $C_2$ are dependent on both $D_0$ and $D_0'$. The smaller $\tau$ is, the more rapid the deviation function decays to zero and the faster the element meets the constraint.

By equation (2), analytically, $D(t)$ will approach zero asymptotically. Nevertheless, it does not really reach zero. In the dynamic constraints technique, a common method is to set an error-tolerance $\epsilon > 0$, and a constraint is said to be met if $|D(t)| < \epsilon$. The drawback of dynamic constraints for the sequential-goal problem appears in the inevitable case where both $\tau$ and $\epsilon$ are small. To satisfy exactness, especially for time-critical goals, $\epsilon$ should be sufficiently small; otherwise, an element may be considered to have already met a goal while still far away from it and then sent on its way to the next goal. With a small $\epsilon$, if the routeing time happens to be short, the parameter $\tau$ should also be set small to ensure that the goal is met at the specified time. In this case, $D(t)$ first approaches $\epsilon$ rapidly in a short period of time, and then takes the remaining relatively long amount of time to reach $\epsilon$ (see Figure 1). From a spectator's point of view, the constrained point of the element

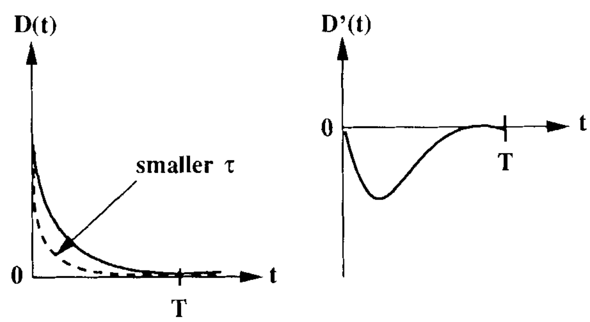


*Figure 1. Deviation function of dynamic constraints*

first moves rapidly towards its goal, then remains near the goal for a rather long time, and then moves rapidly again towards the next goal. While remaining near the constrained point, due to the acting forces and torques, the element displays some superfluous motion, usually rotation or oscillation. The behaviour of the element appears discontinuous during the transition from course to course. Thus, the entire behaviour is not integral.

## 3. SEQUENTIAL-GOAL CONSTRAINTS

In this section, we will propose a mechanism—sequential-goal constraints—for resolving the sequential-goal problem. This method avoids the drawbacks involved in applying dynamic constraints directly.

In sequential-goal constraints, first of all, users need to specify the routeing time for each course between consecutive constraints. Once the routeing times are obtained, a new behaviour differential equation for deviation functions is introduced to control the motion along the courses. Routeing time allocation can be used to control the average speed of each course. As pointed out in the preceding section, in order to yield exactness and punctuality in the element's movement, the deviation functions satisfying the new equation should reach zero rather than be close to zero at the specified time. In addition, the speed of the deviation function should be stable, relative to the average speed, over the entire course so as achieve integrity of the whole motion. In particular, the speed at the end of a course should be kept sufficiently large relative to the average speed along the course.

Let $c_1,\ldots,c_n$ be a sequence of constraints in temporal order, with time-critical constraints, $c_{i_h}$, $1 \leq h \leq k$, which have to be met at the specified time $t_{i_h}$. We denote the deviation function of course $i$, between consecutive constraints $c_{i-1}$ and $c_i$, by $\mathbf{D}_i$. In general, users allocate a routeing time to each course on the basis of their intuition and experience. Usually, all the distances between consecutive constraints are known in advance. For example, all the constraints between consecutive time-critical constraints might be point-to-nail constraints with respect to some coordinate system. In this case, one may assign each course a weight according to the designated rhythm of motion. Let $w_i$ be the weight and $t_i$ the routeing time to be allocated for course $i$. The average speed of course $i$ is then $|\mathbf{D}_i(0)|/t_i$. The routeing times are allocated such that the ratio of average speeds between courses $i$ and $j$ is $w_i/w_j$, and the total time for the courses between the time-critical constraints $c_{i_h}$ and $c_{i_{h+1}}$ is $t_{i_{h+1}} - t_{i_h}$, for $1 \leq h \leq k - 1$.

Next, we present the behaviour control for a particular course. Let the routeing time of the course be $T$ and one component of the deviation function be $D(t)$. Without loss of generality, we assume that the initial time is 0 and $D(0) > 0$. In order to obtain more flexibility in behaviour control, we introduce one extra parameter in the new behaviour differential equation for deviation functions; this parameter is not used in the dynamic constraints technique. The new behaviour differential equation with two parameters $\alpha$ and $\beta$ is

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} D(t) - 2\alpha \frac{\mathrm{d}}{\mathrm{d}t} D(t) + (\alpha^2 + \beta^2) D(t) = 0 \tag{3}$$

with initial conditions $D(0) = D_0$ and $\mathrm{d}D(0)/\mathrm{d}t = D_0'$. The parameter $\alpha$ plays a role similar to that of the assembly rate $\tau$ in the dynamic constraints approach. The deviation function, $D(t)$, turns out to be

$$D(t) = e^{\alpha t}[C_1\cos(\beta t) + C_2\sin(\beta t)]$$

and its derivative is

$$\frac{d}{dt}D(t) = e^{\alpha t}[(\alpha C_1 + \beta C_2)\cos(\beta t) + (\alpha C_2 - \beta C_1)\sin(\beta t)]$$

where $C_1 = D_0$ and $C_2 = (D_0' - \alpha D_0)/\beta$. See Figure 2 for an illustration.

To achieve exactness and punctuality in the element's motion, the parameters $\alpha$ and $\beta$ must satisfy the following equation:

$$D(T) = 0 \tag{4}$$

To ensure the integrity of the motion, the parameters are chosen to satisfy

$$\frac{d}{dt}D(T) = -k\frac{D_0}{T} \tag{5}$$

where $k > 0$ is a constant factor for the course. In general, the factor $k$ is set such

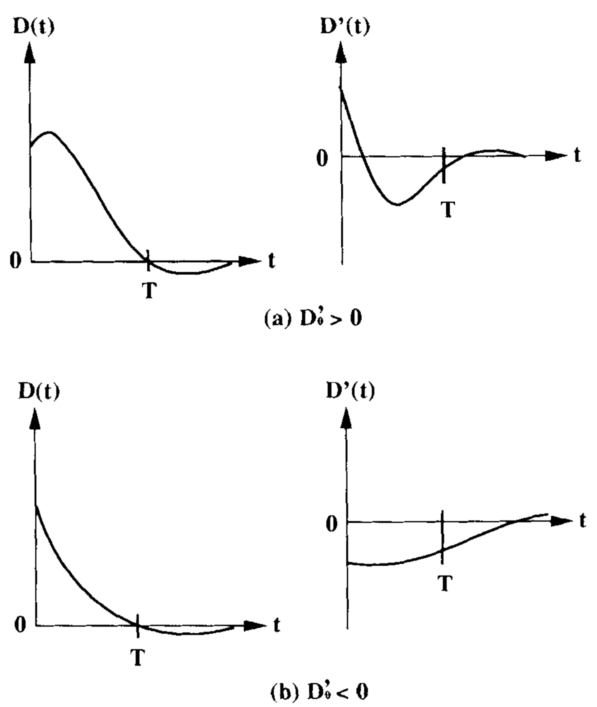


*Figure 2. Deviation function of sequential-goal constraints*

that the ending velocity $dD(t)/dt$ is close to the average velocity of the succeeding course, in order to make the behaviour in the transition between courses continuous and to avoid superfluous rotation or oscillation.

Let $\gamma = D_0'/D_0$. When combined with $D(0) = C_1$ and $dD(0)/dt = \alpha C_1 + \beta C_2$, equation (4) is equivalent to

$$\alpha - \gamma - \beta \cot(\beta T) = 0 \tag{6}$$

Substituting $\gamma + \beta \cot(\beta t)$ for $\alpha$ in equation (5), we obtain

$$\beta T e^{[T\gamma + \beta T \cot(\beta T)]} - k \sin(\beta T) = 0 \tag{7}$$

The values of parameters $\alpha$ and $\beta$ can be obtained from equations (6) and (7) by numerical methods.

There are infinitely many solutions. Only solutions where $0 < \beta < \pi/T$ are used, since all the other solutions will result in redundant oscillations of $D(t)$ along the course.

Exploring equations (6) and (7), we find that the relation among $\alpha$, $\beta$ and $\gamma$ is as follows:

$$\alpha T = \log\left(\frac{k \sin(\beta T)}{\beta T}\right) \tag{8}$$

$$\gamma T = \log\left(\frac{k \sin(\beta T)}{\beta T}\right) - \beta T \cot(\beta T) \tag{9}$$

Since $\beta$ is in the range $(0, \pi/T)$, $\gamma T$ is an increasing function of $\beta T$ and $\alpha T$ is a decreasing function of $\beta T$. By evaluating equations (8) and (9) at $\beta T = 0$ and $\beta T = \pi$, we find that $\alpha T$ ranges from $\log k$ down to $-\infty$ and $\gamma T$ ranges from $\log k - 1$ up to $\infty$.

To achieve integrity of motion, the value of $k$ needs to be set properly in each component of the deviation function. By the above discussion, $k < e^{\gamma T+1}$. The ending velocity $-kD_0/T$ should also be as close to the average velocity of the succeeding course as possible.

Next, we consider the components of the deviation function together. In practice, a constraint is composed of several basic constraints, such as a point-to-nail constraint, an orientation constraint, and so on. Each group of components of the deviation function for a particular basic constraint can be dealt with separately. Without loss of generality, in the following discussion, we assume that the dimension of the deviation function is three.

In a fixed co-ordinate system, although we may assign the ratio of average speeds of the deviation functions for consecutive courses by intuition, componentwise ratios can differ tremendously from the given ratio due to large differences in the magnitude of components. Such differences in ratios may result in a small ending velocity for some courses or a drastic variation in velocity along the whole motion, both of which should be avoided.

To avoid drastic variations in componentwise velocities, a special local normalized co-ordinate system is used for each course. We denote the normalized co-ordinate system for course $h$ by $S_h$. Let $\mathbf{D}_h(t) = (D_{h,1}(t), D_{h,2}(t), D_{h,3}(t))$ and $\boldsymbol{\Delta}_{h+1}(t)$ denote the deviation functions of courses $h$ and $h + 1$, respectively, in co-ordinate system $S_h$, for all $1 \leq h \leq n$. We make a practically reasonable assumption that the angle between $\boldsymbol{\Delta}_{h+1}(0)$ and $\mathbf{D}_h(0)$ is within $\pi/6$. The co-ordinate system $S_h$ is defined by an orthonormal transformation from the world co-ordinate such that

$$\mathbf{D}_h(0) = \frac{|\mathbf{D}_h(0)|}{\sqrt{3}}(1, 1, 1)$$

and $\boldsymbol{\Delta}_{h+1}(0)$ is on the plane spanned by the vectors (1, 1, 1) and (1, 0, 0). We will set the values of the $k_{h,j}$s with respect to the components of the deviation function of course $h$ in the normalized co-ordinate system $S_h$.

In the following, let us consider course $i$. In $S_i$, denote the ratio of the average velocities along courses $i$ and $i + 1$ of the $j$th component,

$$\frac{\Delta_{i+1,j}(0)}{t_{i+1}} \Bigg/ \frac{\mathbf{D}_{i,j}(0)}{t_i}$$

by $\rho_{i,j}$. Let $M$ be the co-ordinate transformation from $S_i$ to $S_{i+1}$. We then have $M(\boldsymbol{\Delta}_{i+1}(t)) = \mathbf{D}_{i+1}(t)$.

By our assumption that the angle between $\boldsymbol{\Delta}_{i+1}(0)$ and $\mathbf{D}_i(0)$ is within $\pi/6$, we obtain

$$1/\sqrt{3} \leq \Delta_{i+1,1}(0)/|\boldsymbol{\Delta}_{i+1}(0)| \leq (3 + \sqrt{6})/6$$

and

$$(6 - \sqrt{6})/12 \leq \Delta_{i+1,j}(0)/|\boldsymbol{\Delta}_{i+1}(0)| \leq 1/\sqrt{3}$$

for $j = 2, 3$.

Then, from the above, we have

$$\frac{\omega_{i+1}}{w_i} \leq \rho_{i,1} \leq \frac{\sqrt{3} + \sqrt{2}}{2} \frac{w_{i+1}}{w_i} \tag{10}$$

and

$$\frac{2\sqrt{3} - \sqrt{2}}{4} \frac{w_{i+1}}{w_i} \leq \rho_{i,j} \leq \frac{w_{i+1}}{w_i} \tag{11}$$

for $j = 2, 3$.

From equations (10) and (11), there is no drastic variation in average velocity componentwise.

In our scheme, for the $j$th component of the deviation function in $S_i$, when $\rho_{i,j} < 1$, $k_{i,j}$ is set such that

$$k_{i,j} < 1 \quad \text{and} \quad k_{i,j} D_{i,j}(0)/t_i = c\Delta_{i+1,j}(0)/t_{i+1} \quad \text{for some } c > 0{\cdot}8 \tag{12}$$

and when $\rho_{i,j} \geq 1$,

$$k_{i,j} \geq 1 \quad \text{and} \quad k_{i,j} D_{i,j}(0)/t_i = c\Delta_{i+1,j}(0)/t_{i+1} \quad \text{for some } c < 1 \tag{13}$$

In order to guarantee that the $k_{i+1,j}$s for the succeeding course can be set properly, the $k_{i,j}$s are set such that the ending velocity of course $i$ in co-ordinate system $S_{i+1}$ satisfies

$$-\frac{\mathrm{d}}{\mathrm{d}t} D_{i+1,j}(0) < \frac{D_{i+1,j}(0)}{t_{i+1}} \tag{14}$$

It is not difficult to find $k_{i,j}$s to satisfy equations (12), (13) and (14).

From equations (12) and (13), when $\rho_{i,j} < 1$, i.e. $\Delta_{i+1,j}(0)/t_{i+1} < D_{i,j}(0)/t_i$,

$$0{\cdot}8\frac{\Delta_{i+1,j}(0)}{t_{i+1}} < \frac{\mathrm{d}}{\mathrm{d}t}\Delta_{i+1,j}(0) = \frac{\mathrm{d}}{\mathrm{d}t}D_{i,j}(t_i) < \frac{D_{i,j}(0)}{t_i}$$

and when $\rho_{i,j} \geq 1$, i.e. $\Delta_{i+1,j}(0)/\mathrm{t}_{i+1} \geq D_{i,j}(0)/t_i$,

$$\frac{D_{i,j}(0)}{t_i} \leq \frac{\mathrm{d}}{\mathrm{d}t}\Delta_{i+1,j}(0) = \frac{\mathrm{d}}{\mathrm{d}t}D_{i,j}(t_i) \leq \frac{\Delta_{i+1,j}(0)}{t_{i+1}}$$

It is easy to see that the magnitude of the ending velocity of course $i$ is between that of the average velocities of course $i$ and course $i + 1$. It follows that the velocities in the transitions will not be too small. Superfluous rotation or oscillation are thus avoided.

Let

$$\gamma_j = \frac{\mathrm{d}}{\mathrm{d}t}D_{i+1,j}(0)/D_{i+1,j}(0)$$

From equation (14), in co-ordinate system $S_{i+1}$,

$$\gamma_j t_{i+1} + 1 = 1 - t_{i+1}\frac{\mathrm{d}}{\mathrm{d}t}D_{i+1,j}(0)/D_{i+1,j}(0) > 0$$

Thus, the upper bound of $k_{i+1,j}$, $\mathrm{e}^{\gamma_j t_{i+1}+1}$, is greater than 1 and proper $k_{i+1,j}$s can be set for course $i + 1$.

The above discussion provides us with a coherent way to assign the values of $k_{i,j}$s so that the whole motion is continuous and integral.

## 4. IMPLEMENTATION

Besides the point-to-nail constraint, orientation constraints, point-to-path constraints, and so on may also be worked out successfully using sequential-goal constraints. Using the orientation constraint, animators can easily and precisely control the orientation of a modelled element during its travel and avoid unnecessary oscillation. The orientation constraint also provides a useful mechanism for generating more dramatic action, such as turning a somersault somewhere in the motion. In sequential-goal constraints, the point-to-path constraint, i.e. a path constraint of timing, can be transformed into a sequence of point-to-nail constraints. That is, employing the path function, animators can set up a number of intermediate goals along the designated path and then have the objects meet them exactly and punctually in sequence.

In animation, the modelled elements usually interact with other models or with the environment to produce natural and realistic visual effects. The ability to ensure exactness and punctuality makes the sequential-goal constraints technique compatible with other modelling techniques. For example, if we know when and where an object modelled by the kinematic method will be, an object modelled using sequential-goal constraints can interact with it properly.

We used the sequential-goal constraints technique to produce an animated sequence on a Silicon Graphics 4D/310 workstation; the sequence is illustrated in Plates 1–8. In this sequence, three rods fly through the air along individually designated paths and finally hit a target. Each rod also penetrates a red ball at the beginning of the motion.

The motion of each rod was modelled with a sequence of point-to-nail constraints and a sequence of orientation constraints. The point-to-nail constraints control the path of the rods and the orientation constraints control the orientation of the rods during the motion. There were 25, 16 and 17 transient constraints used for the paths of these three rods, respectively. A kinematic model was used for the motion of the red balls. The penetration of the rods through the balls shows that sequential-goal constraints can easily be used in conjunction with other modelling techniques.

## 5. CONCLUDING REMARKS

The sequential-goal problem arises naturally when generating complicated or dramatic animated sequences. We propose the sequential-goal constraints technique to remedy the deficiency of the dynamic constraints technique in solving the sequential-goal problem. Our technique not only preserves all the advantages of dynamic constraints but also provides the capability to guide constrained elements to meet goals exactly and punctually while ensuring that the whole motion is continuous and integral. With this technique, animators can generate animated sequences easily and intuitively by proper specification of transient constraints and respective routeing times. In addition, the exactness and punctuality of element motion obtained using sequential-goal constraints make it wasy to combine objects modelled using sequential-goal constraints with objects modelled using traditional methods.

## REFERENCES

1. John Lasseter, 'Principles of traditional animation applied to 3D computer animation', *Computer Graphics*, **21**, (4), 35–44 (1987).
2. James K. Hahn, 'Realistic animation of rigid bodies', *Computer Graphics*, **22**, (4), 299–308 (1988).
3. Ronen Barzel and Alan H. Barr, 'A modeling system based on dynamic constraints', *Computer Graphics*, **22**, (4), 179–188 (1988).

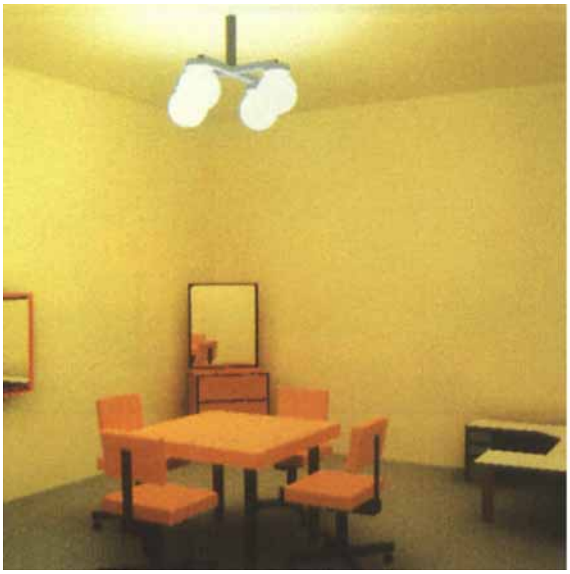

*Plate 1(a) (Pattanaik and Mudur). A three-dimensional environment with spherical light sources without a participating medium*

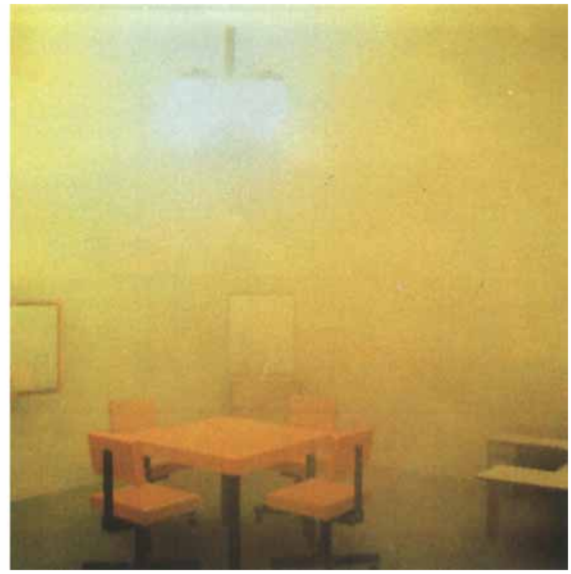

*Plate 1(b) (Pattanaik and Mudur). A three-dimensional environment with spherical light sources with a participating medium*

*Plate 2 (Pattanaik and Mudur). A plant modelled as a participating volume with around 161 spherical volume elements*

*Plate 3 (Pattanaik and Mudur). A gaseous emitting volume*

*Plate 1 (Liu, Ko and Chang)*

*Plate 2 (Liu, Ko and Chang)*

*Plate 3 (Liu, Ko and Chang)*

*Plate 4 (Liu, Ko and Chang)*

*Plate 5 (Liu, Ko and Chang)*

*Plate 6 (Liu, Ko and Chang)*

*Plate 7 (Liu, Ko and Chang)*

*Plate 8 (Liu, Ko and Chang)*

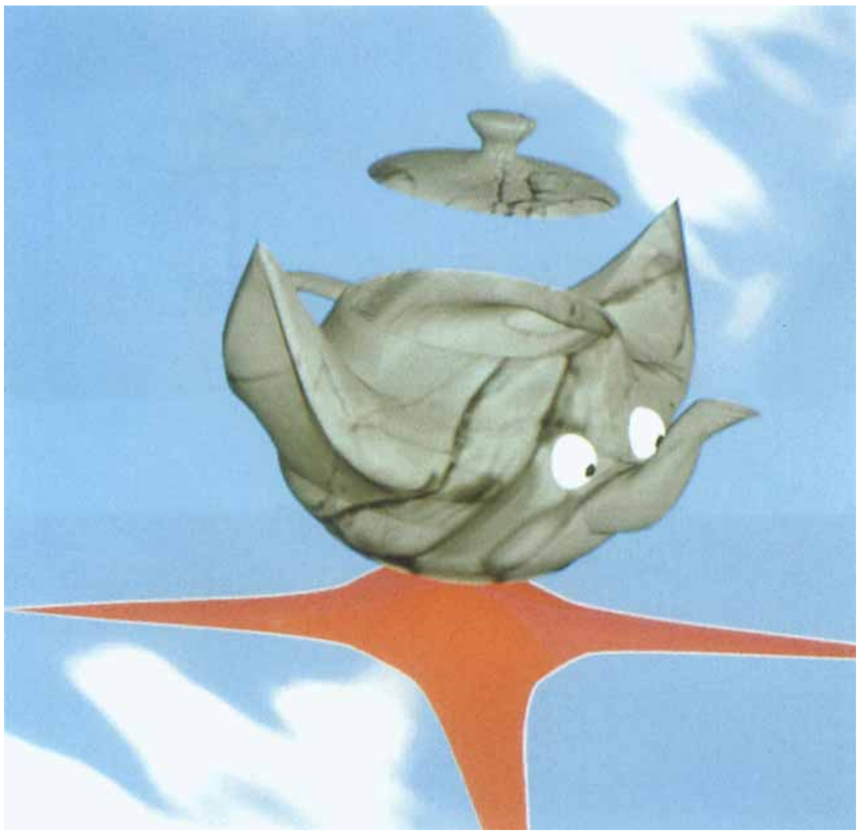

*Plate 1 (Bechmann and Dubreuil)*

4. Paul M. Isaacs and Michael F. Cohen, 'Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics', *Computer Graphics*, **21**, (4), 215–224 (1987).
5. Alan H. Barr, 'Teleological modeling', *ACM SIGGRAPH'88: Developments in Physically-Based Modeling*, 1988.
6. Alan H. Barr, Brian Von Herzen, Ronen Barzel and John Snyder, 'Computational techniques for the self assembly of large space structures', *ACM SIGGRAPH'88: Developments in Physically-Based Modeling*, 1988.
7. Ronen Barzel and Alan H. Barr, 'Controlling rigid bodies with dynamic constraints', *ACM SIGGRAPH'88: Developments in Physically-Based Modeling*, 1988.
8. John C. Platt and Alan H. Barr, 'Constraint methods for flexible models', *ACM SIGGRAPH'88: Developments in Physically-Based Modeling*, 1988.
9. Devendra Kalra, 'A unified framework for constraint-based modeling', *Ph.D. Thesis*, Department of Computer Science, Caltech, 1990. Caltech-CS-TR-90-15.