Synthetic Movies for Computer Vision Applications

A. Santuari, O. Lanz, and R. Brunelli[1]
[1] ITC-irst, POVO (TN), ITALY

ABSTRACT
This paper presents a real time graphical simulator based on a client-server architecture. The rendering engine, supported by a specialized client application for the automatic generation of goal oriented motion of synthetic characters, is used to produce realistic image sequences for extensive performance assessment of computer vision algorithms for people tracking.

KEY WORDS
Animation, Virtual Reality, Tracking, Intelligent Ambience

1 Introduction

Vision is human most important sensory channel and a detailed understanding of its inner workings is one of the great challenges of modern science. Artificial vision research tries to match human visual competence by developing algorithms able to infer from bi-dimensional images, or image sequences, the contents of the corresponding 3D static or dynamic scenes. Extensive evaluation of algorithm performance is necessary but in many cases extremely costly to perform. Let us consider a simple algorithm to detect all persons within a camera field of view, providing a binary image distinguishing moving people from static background. Precise evaluation of the accuracy of the algorithm would require manual annotation of each single pixel for every image of the captured sequences. Reliable estimation of the performance under different operating conditions requires the use of many sequences, further increasing the cost of performance evaluation. A similar situation can be found in the development of systems based on the *learning by example* paradigm. This approach has demonstrated its validity in several applications but its feasibility rests on the availability of a set of examples covering the range of operating conditions of the system. Again, gathering a representative set of examples may be too costly, ruling out the use of otherwise powerful techniques.

In this paper we focus on the possibilities offered by current computer graphics techniques to generate realistic synthetic imagery supported by extensive ground truth information at frame rate. Among the many topics currently investigated by the computer vision community, the development of indoor surveillance systems is of increasing scientific and practical interest. The evaluation of people tracking algorithms, especially in the case of crowded environments, is particularly difficult and has not been addressed so far in a principled way. This paper presents a graphical simulator whose architecture has been designed to improve the work-flow of algorithm development and evaluation for *third generation surveillance systems* based on distributed multi camera systems for museum monitoring. The architecture of the system and the rendering strategies are presented in Section 2. Animation techniques are described in Section 3 while first results and foreseen extensions are reported in Section 4.

2 Rendering Architecture

One of the trends in the development of novel surveillance systems is represented by the use of multiple sensors cooperating in monitoring multi room environments. The design of computer vision algorithms within this framework is particularly challenging both on the theoretical side (effective integration of multiple information streams) and on the practical side (setting up large sensor networks and evaluating algorithm performance). The proposed simulator derives its architectural features from the necessity of mimicking the structure of such surveillance systems:

- network wide operation, supporting multiple connections with several image processing systems (emulating a set of intelligent active cameras).
- management of virtual environments comprising static structures and moving *objects* such as people;
- generation of believable motion patterns for people groups;
- synthesis of image sequences providing realistic input to image processing algorithms;
- efficient image generation to effectively support tight develop test refine work-flows;
- automatic generation of ground truth data upon which basing automatic evaluation of algorithms.

The adopted solution is based on the client-server architecture schematically reproduced in Figure 1. The server manages the environment and provides clients with access to dynamic objects, currently characters and cameras. Each client can then take control of an available camera to get a personalized view of the environment or join the simulation from the point of view already controlled by another client.

Synthetic characters can also be controlled by a client and the next section will provide a detailed description of the currently available module for the control of groups of

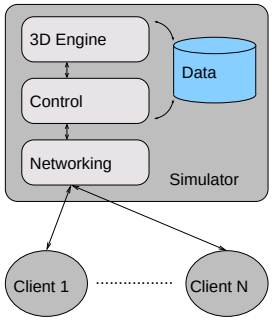

Figure 1. System architecture

characters visiting a museum. Each client can then request from the server the images of the scene, their depth map (that can be used to simulate other kind of sensors) and a segmented image where each person is represented with a unique color (see Figure 2). This provides the necessary ground truth to evaluate the performance of people tracking algorithms. The feature set of the rendering engine has

Figure 2. The information generated by the simulator: the visual stimuli, a depth map, and the 'oracle'.

been driven by the requirement of providing a useful substitute of real camera signals to image processing algorithms. In particular the chosen scenario requires:

- effective management of complex geometry to simulate multi room environments with many characters;
- easy control of character animation to create believable human motion;
- realistic lighting of static and dynamic geometry.

The rendering engine relies the OpenGL API, for which very efficient hardware implementations exist, and manages complex geometry using Binary Space Partition (BSP) Trees [1] and precomputed Potentially Visible Sets (PVS).

While fast, plain OpenGL based rendering does not provide realistic images. The available lighting model is simple and, among other limitations, does not take into account environment diffusions. In order to generate high quality images, the rendering engine exploits a precomputed global illumination for the static environment made available through a set of *lightmaps*, mapped by OpenGL onto surface textures, modulating their lighting. Moving objects must instead be shaded dynamically using OpenGL lights. The maximum number of supported lights depends on the implementation and the rendering engine keeps a sorted list of lights for each volume cell generated by the BSP tree, so that only the most relevant lights are activated for lighting (and for shadowing).

As the basic OpenGL primitives are polygons, all scene geometries are defined by their polygonal approximations. Character animation, an important feature of the proposed simulator, would then require the position of all moving vertices as a function of time. As this would be impractical, animations are broken down into cyclic actions that are in turn specified through a limited set of snapshots, the key frames, with all intervening positions interpolated by the system. The flexibility of the system in specifying composed actions can be further increased using the technique of *skeletal animation*. Each model is represented by a hierarchical skeleton: each bone has specific rotation constraints and is further characterized by an influence sphere which identifies the vertices that will follow the movements of the bone. This system requires the interpolation of rotations that can be accomplished neatly using quaternions and spherical linear interpolation [8]. In the described sys-

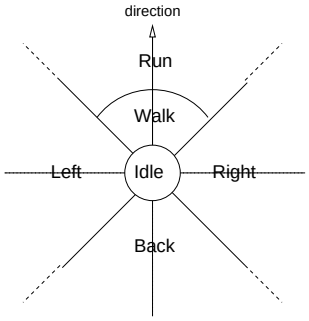

Figure 3. Character is moving upwards: animation is chosen based on speed direction and magnitude

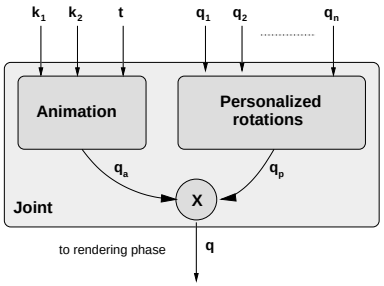

Figure 4. Pipeline for the computation of joint rotation. The additional degree of freedom granted by personalized rotations permits the introduction of new character poses.

tem, characters are animated through a hybrid technique used in current video games. The approach tries to gain the speed of vertex animation approaches and the flexibility of skeletal animation techniques. Each character is represented by a very simple skeleton with three bones con-

nected to the head, the torso, and the legs. The flexibility of skeletal animation is exposed to client applications, allowing them to specify personalized rotations for the joint (as described in Figure 4). This is particularly useful in the simulation of a museum environment as the character attention may be directed to a given point, such as an exhibit, through a combined rotation of head and torso (see Figure 5). In the current system, character position is not

Figure 5. A character pose can be controlled through the rotation of his joints.

controlled by the rendering engine itself but is provided by a specialized client application. This implies that the rendering engine must deduce from character orientation and speed the appropriate animation class among the available ones: run, walk, stride left, stride right (see Figure 3). As the kinematics of the character may suddenly result in the choice of a new animation, the rendering engine *smooths* the transition between the animations to avoid annoying artifacts. The transition between animation classes is obtained by creating temporary additional key-frames and interpolating towards the character configuration specified by the new motion class (see Figure 6). A major source of dif-

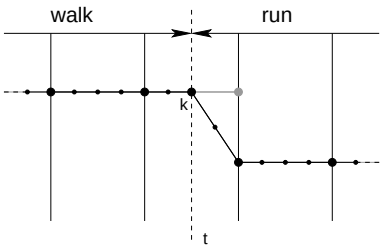

Figure 6. Interpolated transition between two animation classes

ficulties in the development of computer vision algorithms for surveillance is given by dynamic lighting conditions, including shadows due to moving objects such as people and vehicles [2, 6]. Fast but geometrically accurate rendering of shadows, coherent with environment global illumination, is one of the key features of the developed rendering engine. Geometrical accuracy of characters shadows has been obtained generating shadow volumes using a non standard OpenGL perspective projection matrix, obtained by moving to infinity the *far* clipping plane of the standard matrix [9].

The actual drawing of shadow boundaries relies on a modification of the original stencil algorithm, proposed by Heidmann [3], known as *Carmack's Reverse* that solves the problems of the original formulation whenever the camera is located within the shadow volume [4]. However, determining shadows boundaries does not completely solve the problem. Shadowed regions should be filled in accordance to a global illumination solution for the given environment. While a correct solution is computationally expensive, a good approximation can be obtained relying on the correct contribution of each light to the illumination of static geometry and using OpenGL lights for the shadowing (and self shadowing) of the dynamic characters (see Figure 2). For each light, shadows are generated by substractive blending of light contribution, providing full support for colored lights and shadows. The major drawback is given by the necessity of generating a global illumination solution of the given environment for each of the lights, as the resulting lightmaps are needed to *paint* the shadows on the static environment.

Figure 7. The different steps in the generation of accurate shadows

The rendering engine implemented for the server side of the architecture is then able to generate realistic images for multiple clients, from different viewpoints, managing both static and dynamic geometry. While the engine takes care of the low level actuation of character motion, the generation of believable, purposive, motion for the simulated characters has been assigned to a specialized client of the simulator to be described in the next section. This architectural choice has the advantage that different algorithms for generating complex behaviour can be tested and compared with minimum intervention on the structure of the simulator.

3 Choreographed Animation

Living beings perceive their environment and react according to subjective criteria, making realistic simulation of moving people a difficult task. Although rendering the characters is by itself a complex task, the creation of a suitable choreography of groups moving purposively in a believable way poses additional challenges. The variety of

collective human motion patterns in a complex environment such as a museum are due to many factors among which we can cite:

visual perception, the interpretation of visual stimuli from the environment;

prior knowledge, already acquired information on the environment;

decision capability, goal driven reaction to perceived impressions;

group behaviour, aggregation based on similar interests;

experience accumulation, dynamically reshaping goals.

In order to create believable motion patterns, a set of synthetic characters must be endowed with the same functionalities, albeit in a simplified version. In our implementation, visual perception is simply responsible for filtering out information on the environment that is not available to the characters due to occlusions and limited field of view. The interpretation of perceived stimuli can be bypassed by giving the character direct access to scene geometry, museum exhibit locations and descriptions, exact position and velocities of mates. The strategy used for the simulation of museum visiting groups relies on a hybrid approach integrating second order dynamics for global motion coordination and indirect control of groups through invisible leaders. In fact, while plausible flock behavior naturally emerges from the simple rules proposed in [7], consistent goal oriented behavior cannot be easily created in the same way.

Our approach is related to the concept of *Co-Fields* introduced in [5]. In the original formulation, each moving character (agent) perceives the position of the other agents through a force field and coordinated behavior emerges from the agents following the force field, dynamically reshaping it by changing their position. In our implementation the potential fields do not only depend on time and character position but also on her velocity. Modeling walls and other static and dynamic obstacles with pure positional fields may result in strange motion patterns that prevent the agents from reaching the expected objectives. The shortcoming of a pure positional approach is also evident in the modeling of group dynamics, as people clearly use information on velocity to adapt their trajectories and avoid collisions. Character motion $(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))$ over a potential field $E(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)$ is governed by a second order differential equation:

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) &= \frac{d^2\boldsymbol{x}}{dt^2}(t) = -\boldsymbol{\nabla}_x E(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) \\ \boldsymbol{x}(t+\Delta t) &= \boldsymbol{x}(t) + \dot{\boldsymbol{x}}(t)\Delta t + \tfrac{1}{2}\boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)\Delta t^2 \\ \dot{\boldsymbol{x}}(t+\Delta t) &= \dot{\boldsymbol{x}}(t) + \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)\Delta t \end{aligned}$$

Although the pioneering work of Reynolds [7] addressed the simulation of group behaviours of animals, similar rules can be applied to human characters. In our implementation, the potential field E is defined by the superposition of the following components:

E_C**,** to avoid collisions with static and dynamic obstacles;

E_L**,** driving the members of a group towards their (invisible) leader;

E_F**,** responsible for flock behaviour;

E_V**,** urging match the velocity of nearby members of the same group.

In scenarios with complex geometry and frequent crowding it is important to find plausible collision avoidance policies. If an agent walks away from a wall there is no reason to be repelled by it, independently of how close it is. On the other hand, if two persons move toward each other they have to change their paths in advance. But again, if they walk in the same direction they can stay very close to each other. Positional information with additional velocity data can then be used to implement more realistic collision avoidance strategies. If a polygonal scene model is available, collision detection can be carried out by ray-polygon intersection. Given the impact location $\boldsymbol{x}_o$ of the nearest object and the surface normal versor $\boldsymbol{n}_o$, these behaviors can be simulated by the following field component:

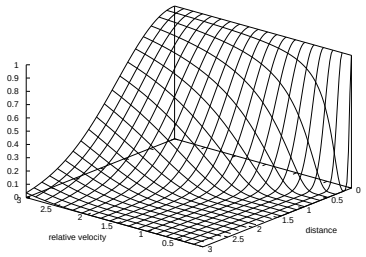

Figure 8. Collision avoidance field $\alpha(t_o)$: the larger the relative velocity the greater the *visibility* of the obstacle.

$$\nabla_x E_C(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) = -\alpha(t_o)\boldsymbol{n}_o, \qquad t_o = \frac{\|\boldsymbol{x}-\boldsymbol{x}_o\|^2}{(\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}_o)\cdot(\boldsymbol{x}-\boldsymbol{x}_o)}$$

and $\alpha(t_0)$ is a positive decreasing function of the expected impact time (e.g. a Gaussian, see Figure 8). Given the positions $\boldsymbol{x}_i$ of members i in the set of n_v^x (close) visible agents A_v^x, flock behaviour can be simulated by introducing the following field:

$$\nabla_x E_F(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) \propto \left(\boldsymbol{x} - \frac{1}{n_v^x}\sum_{i\in A_v^x}\boldsymbol{x}_i\right)$$

so that an agent is attracted by the *center of mass* of nearby group mates with a force proportional to its distance. In a similar way, velocity matching is governed by

$$\nabla_x E_V(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) \propto \left(\dot{\boldsymbol{x}} - \frac{1}{n_v^x}\sum_{i\in A_v^x}\dot{\boldsymbol{x}}_i\right)$$

Although consistent goal-oriented behaviour can be incorporated in agent dynamics through an additional, dynamic *interest* field, group behaviour can be more appropriately maintained through a (invisible) leader following policy. The corresponding potential component is expressed by a radial field centered at the leader position $\boldsymbol{x}_l(t)$:

$$\nabla_x E_L(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) \propto (\boldsymbol{x} - \boldsymbol{x}_l(t))$$

The leaders can then be moved on a graph whose nodes represent physical locations in the museum. However, naive implementation of follow-the-leader behaviour may result in unacceptable motion patterns whenever the character looses sight of the leader. If geometry constraints are ignored, the character may be subjected to acceleration driving her towards the obstacles. On the other side, if the constraints are considered, the character may have no leader to follow in sight. The solution used in the presented system is to endow each character with a memory of the latest positions of the group guide so that motion can be directed by the most recent position of the leader that is visible from the standing point of the moving character $\hat{\boldsymbol{x}}_l(\boldsymbol{x}, t)$:

$$\nabla_x E_L(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) = \delta(\boldsymbol{x} - \hat{\boldsymbol{x}}_l(\boldsymbol{x}, t))$$

The nodes corresponding to the exhibits are given a set of

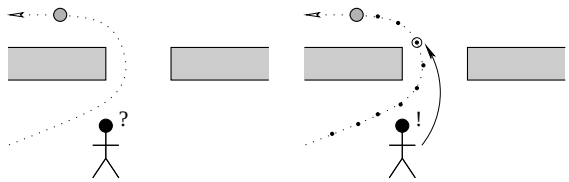

Figure 9. Endowing characters with memory of leaders trajectories eases the implementation of realistic follow-the-leader strategies.

descriptive labels from which an interest score is derived based on the group profile while the arcs are labeled with the corresponding spatial distances. In this case, people motion can be planned in advance, optimizing the fruition of the exhibits (sum of the scores) subject to predefined temporal constraints on the length of the visit. However, such an approach is not completely satisfactory as people do not, usually, optimize their visit this way. Furthermore, the impact of local events (such as overcrowding) on group trajectories cannot be modeled with the necessary spatial detail unless a large number of nodes is inserted into the graph. The proposed solution is based on the idea of *dominant interest field*. Each exhibit in the museum generates an attractive field driving the motion of each group leader. In order to get feasible leader paths, scene geometry information must be incorporated in the interest fields. In the proposed approach fields are generated by propagating *interest rays* from each exhibit that are subject to occlusions by scene obstacles. The rays move linearly in a discretized space and mark each floor cell with the minimum distance to the exhibit, giving rise to the distance field $D_i(\mathbf{x})$ of the ith exhibit. In order to cover the whole environment (in a way not dissimilar to diffuse illumination) each pixel acts as a new source. The following algorithm computes the spatially sampled distance field for a polygonal scene map:

```
set distances of accessible pixels to ∞;
activate source on exhibit location and set distance to 0;
while there is an active source do
    emit light from current source;
    d_s = current source distance;
    foreach illuminated pixel do
        d = d_s + distance from current source;
        if d < current pixel distance then
            set current pixel distance to d;
            label pixel with displacement from current source;
            if current pixel is on a corner then
                activate new source on current pixel;
        end
    end
    turn off current source;
end
```

Algorithm 1: Computation of distance map

The resulting fields can be effectively modulated by the complexity of the path leading to the exhibit as well as the mere distance to be traveled (see Figure 10):

$$E_I^i(\mathbf{x}, \dot{\mathbf{x}}, t) = \frac{\eta_i(t)}{1+D_i(\mathbf{x},t)} \lambda_i(\mathbf{x}, \dot{\mathbf{x}})$$

Here $\lambda_i \in (0, 1]$ accounts for exhibit visibility and for acquired *prior knowledge* about the museum, e.g. by consulting a exhibit plan beforehand. The modulating factor $\eta_i(t)$ defines the group interest in the ith exhibit, which decreases while looking at the exhibit (*experience accumulation*):

$$\eta_i(t + \Delta t) = \eta_i(t) - \nu e^{-\frac{D_i^2(\mathbf{x}_l)}{2\sigma_i^2}} \Delta t$$

where ν characterizes the fruition rate and σ represents the spatial scale from which the exhibit becomes enjoyable. Each group can then be characterized by a specific profile $\eta_i(0)$ that will affect its museum tour. Leader motion is then governed by

$$\mathbf{x}_l(t + \Delta t) = \mathbf{x}_l(t) + \frac{v_l \Delta t}{\|\nabla_x E_I^{\hat{i}}(\mathbf{x}, \dot{\mathbf{x}}, t)\|} \nabla_x E_I^{\hat{i}}(\mathbf{x}, \dot{\mathbf{x}}, t)$$

where v_l denotes constant leader velocity and

$$\hat{i} = \arg\max_i \{E_I^i(\mathbf{x}, \dot{\mathbf{x}}, t)\}$$

represents the current *dominant interest* of the group leader. Due to the consistent formulation, the interest field can directly be used as an additional component of agent dynamics. The implemented system permits the simulation of groups with different interests in museum exhibits as well as different visiting strategies (detailed planning, casual, time limited etc.) by appropriate modulation of interest fields.

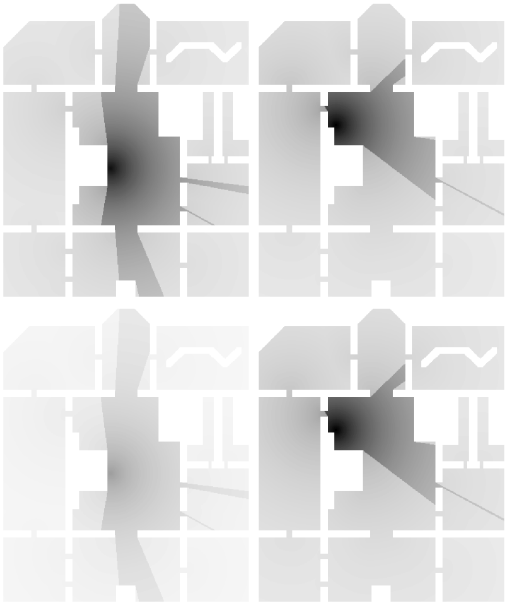

Figure 10. The two columns reports the evolution in time of the co-fields originated from two different exhibits: the darker the field the stronger the attraction of the exhibit. The visitor is at first attracted by the one corresponding to the left. As time pass by, she gets less and less interested in it till her attention is captured by the nearby exhibit to which her attention turns. Field discontinuities are due to the depression of secondary source emission, capturing the effect out-of-sight out-of-interest.

4 Conclusions

This paper has presented a graphical simulator supporting the development of *third generation surveillance systems* based on visual input. Fast image rendering, a flexible client-server architecture, perceptual oracle functionalities to provide extensive ground truth information for algorithm evaluation, and generation of complex movements of people groups are among its most important features. The system is currently employed in the development of algorithms for people segmentation where it is able to provide accurate performance reports (see Figure 11 for an example on the evaluation of a people/background segmentation algorithm). The system will be extended to support characters with an increased number of joints for the development of gesture recognition algorithms.

Acknowledgements

The authors would like to thank F. Bertamini at ITC-irst for kindly providing the information reported in Figure 11.

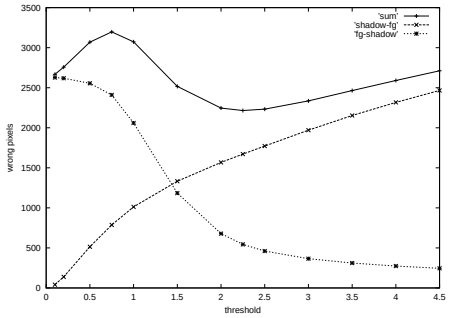

Figure 11. The plot reports the error, in pixel units, made by a simple algorithm for people/background segmentation in an indoor environment. After a first step segmenting the image into background and foreground plus shadows, the algorithm tries to identify shadows using a set of classifiers combined through scoring. Manual computation of algorithm performance at this level of detail would be extremely costly and the accuracy would not reach the level possible with the proposed approach.

References

[1] H. Fuchs, A. Kedem, and B. Naylor. On visible surface generation by a priori tree structures. *Computer Graphics*, 14(3):124–133, July 1980.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis. W^4: Real-Time Surveillance of People and Their Activities. *IEEE Trans. On PAMI*, 22(8):809–830, 2000.

[3] Tim Heidmann. Real shadow real time. *IRIS Universe*, 18:28–31, 1991.

[4] M. J. Kilgard. Robust Stencil Shadow Volumes. In *CEDEC*, Tokyo, Japan, 2001. NVIDIA.

[5] M. Mamei, F. Zambonelli, and L. Leonardi. Co-Fields: A Unifying Approach to Swarm Intelligence. In *Proc. of 3rd International Workshop on Engineering Societies in the Agents' World*, Madrid, Sept. 2002.

[6] A. Prati, I. Mikic, C. Grana, and M. M. Trivedi. Shadow Detection Algorithms for Traffic Flow Analysis: a Comparative Study. In *Proceedings of IEEE Intl. Conference on Intelligent Transportation Systems*, pages 340–345, 2001.

[7] C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–43, 1987.

[8] Jan Svarovsky. Quaternions for game programming. In Mark DeLoura, editor, *Game Programming Gems*, pages 195–199. Charles River Media, 2000.

[9] M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL Programming Guide*. Addison-Wesley, 2001.