Robot Soccer with LEGO Mindstorms

Henrik Hautop Lund **Luigi Pagliarini**

LEGO Lab
University of Aarhus, Aabogade 34, 8200 Aarhus N., Denmark

hhl@daimi.aau.dk
http://www.daimi.aau.dk/~hhl/

Abstract. We have made a robot soccer model using LEGO Mindstorms robots, which was shown at RoboCup98 during the World Cup in soccer in France 1998. We developed the *distributed behaviour-based approach* in order to make a robust and high performing robot soccer demonstration. Indeed, our robots scored in an average of 75-80% of the periods in the games. For the robot soccer model, we constructed a stadium out of LEGO pieces, including stadium light, rolling commercials, moving cameras projecting images to big screens, scoreboard and approximately 1500 small LEGO spectators who made the "Mexican wave" as known from soccer stadiums. These devices were controlled using the LEGO Dacta Control Lab system and the LEGO CodePilot system that allow programming motor reactions which can be based on sensor inputs. The wave of the LEGO spectators was made using the principle of *emergent behaviour*. There was no central control of the wave, but it emerges from the interaction between small units of spectators with a local feedback control.

1 Introduction

Before the LEGO Mindstorms Robotic Invention System was to be released on market, we wanted to make a large-scale test of the robot kit. We selected to make a LEGO Mindstorms robot soccer play with a distributed behaviour-based system to be demonstrated in Paris during the soccer World Cup France'98 at RoboCup'98. Robot soccer has been defined as a new landmark project for artificial intelligence [3], and its characteristics fitted our purpose. In contrast to previous artificial intelligence challenges such as computer chess, robot soccer is a dynamic and physical game, where real time control is essential. Further, where a game like chess might allow extensive use of symbolic representation, robot control put emphasis on embodiment and many aspects of this prohibits the use of symbolic representation. In general, participating in robot soccer is believed to provide both students and researchers with knowledge about the importance of embodiment and the problems that ungrounded abstractions might lead to [4].

M. Asada, H. Kitano (Eds.): RoboCup-98, LNAI 1604, pp. 141–151, 1999.

However, we also found a number of problems that had to be solved before robot soccer could be made appealing for a public audience. Robot soccer is a very young research field, so the performance of the robot soccer players might not look impressive enough from a public audience's point of view. Even though, we expected a huge public interest in our LEGO Mindstorms robot soccer demonstration[1], so it was important to alleviate this problem. The robot game had to be put into the right context. From an aesthetic point of view, some robot soccer players might be looked at as essentially cubic, metallic devices that move around in a pen and push a ball — it might not appear to be much like soccer to the public audience if the audience is not told in advance to look at this as soccer. Therefore, in our robot soccer game, we put much more emphasis on making a context that immediately would allow the public audience to recognise the game to be a soccer game. This was done by making a whole stadium (which we named Stade de Victor LEGO) out of LEGO with light towers, rolling commercials, and almost 1500 LEGO spectators who made the "wave", by providing sounds related to the game (tackling, kicking, spectator noise, etc.), and by giving the robot soccer players a face. Indeed, in the developing phase, we had a graphical designer to make huge colour drawings of possible scenarios, we had a technical designer to make appealing facial expression of the robots, and we made different scripts for games (how to enter the field, how to sing the national anthems, how to get into the kick off positions, what play strategies to use, etc.).

2 RoboCup and LEGO Mindstorms Robot Soccer

We wanted to construct LEGO Mindstorms robot soccer players to play a demonstration tournament during RoboCup'98. First, we held an internal fully autonomous LEGO robot soccer competition. In this tournament, our students were allowed to use nothing else than LEGO Mindstorms and LEGO sensors, which include light sensors, angle sensors, temperature sensors, and switch sensors. However, one has to think very carefully about the set-up to make an impressive robot soccer game with only these sensors. Since the initial results of the fully autonomous robot soccer experiment were not impressive enough to bring to RoboCup'98, we ran another experiment in parallel. In this experiment, we wanted to increase the sensing capabilities of the LEGO robot soccer players. This was done by using the approach taken in the RoboCup Small League, namely to use an overhead camera. Here, the idea is to increase the sensing capability by having a camera that can overview the whole scene (i.e. the field), interface this with a host computer, and then have the host computer to transmit information to the robots on the field.

Our set-up included an NTSC video-camera which was connected to a hardware vision system that extracted the position of the players and the ball in the robot soccer field. This information was processed to a host computer that would communicate information to the robot soccer players, and the information

[1] Indeed, our LEGO Mindstorms robot soccer demonstration was broadcasted to an estimated 200-250 million television viewers world-wide.

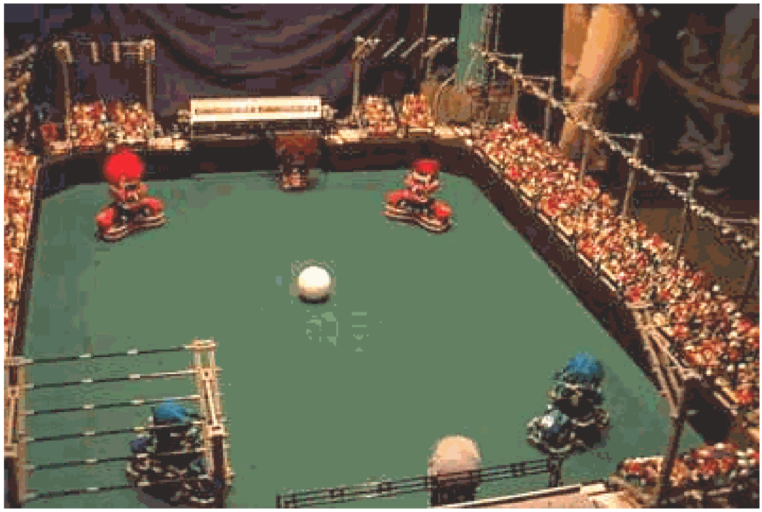

Fig. 1. The LEGO robot soccer set-up. There is one goalkeeper and two field players on each team (one red and one blue team). The stadium has light towers, scanning cameras that project images to large monitors, scoreboard, rolling commercials, and almost 1500 small LEGO spectators that make the "Mexican wave". ©H. H. Lund, 1998.

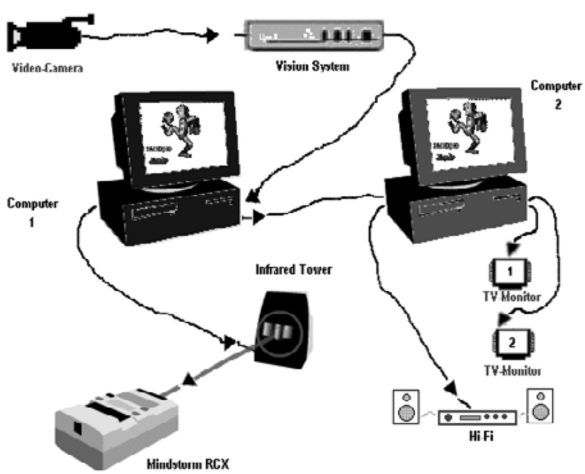

Fig. 2. The image from the video-camera was fed into a hardware vision system that extracted the position of players and ball in the field. This information was first sent to computer 1 that contained part of the distributed behaviour-based controller and communicated information via infra-red communication towers to the LEGO Mindstorms RCX (Robot Control System). Secondly, the information was forwarded to computer 2 that steered a couple of cameras from which images were sent to monitors, and it held the sound processing system. ©H. H. Lund and L. Pagliarini, 1999.

was passed on to a second computer that controlled sound and small cameras (for projecting the game from different angles to big screens) during the games. The set-up is shown on Figure 2. Again, it is important to note how we here use both monitoring and sound for making the robot soccer games more lively and pleasing for the public audience.

Each team consisted of one goalkeeper and two field players. The goalkeeper was controlled with a LEGO CodePilot, while the two field players were constructed around the LEGO Mindstorms RCX (Robot Control System). Each player had two independent motors to control two wheels to make the robot move around on the field, and one motor to control movement of the robot's mouth (so that it could "sing" the national anthem and "shout" when scoring a goal). A player had three angle sensors to detect the motion of wheels and mouth. All parts of the robots except for batteries and coloured hair to indicate the team were original LEGO elements (LEGO Dacta, LEGO Mindstorms, LEGO Technic).

Since we wanted to use LEGO Mindstorms only for the robot soccer demonstration, we had to communicate from host computer to robots with infra-red transmission. This is notoriously slow and unreliable, so most other RoboCup teams use a radio-link. Because of our infra-red link, the game would necessarily be a bit slow. But we solved a great deal of the problems by making a distributed behaviour- based system, in which only high levels of competence were implemented in the host computer, and low levels of competence ran directly on the robots.

When opening the mouth, a small bar on the back of the robot would move

Fig. 3. The LEGO Mindstorms robot soccer players have an appealing design with big eyes, a moving mouth, coloured hair, etc. ©H. H. Lund, 1998.

down on the ground, which made the robot tip slightly forward. This was made in order to make a shout look even more lively. However, the functionality turned out to provide a nice, unintentional use. The border of the field had a chamfered edge to allow the ball not getting stuck up against the wall. However, because of the slight slope, the robot soccer players could get stuck at the edge, since the distance between the wheels touch point on the ground and the base of the robot was quite small. The problem was solved by allowing the robot soccer player to scream after being stuck for a short time. Apart from opening the mouth, this would make the bar on the back go down on the ground and tip the robot, so that it would actually get free.

The LEGO Mindstorms robot soccer players were programmed using the Mindstorms OCX together with MicroSoft Visual Basic. The control system was a distributed system between the robots and the host computer. The host computer ran a program that collected co-ordinates from the hardware vision system, processed this information and sent angles to turn out via the infra-red transmitters. The robots had a control system that allowed them to collect the appropriate data, and react accordingly. In a sense, we implemented a control system that we can term a *distributed behaviour-based system*, after behaviour- based systems such as Brooks' subsumption architecture [1]. Some levels of competence were placed on the robot, while other higher levels of competence were placed on the host computer. The distributed behaviour-based system is described in full details in [7]. The division of levels of competence and their distribution on host computer and robot(s) is shown in Table 1.

Table1. The distributed behaviour based control system of the LEGO robot soccer players. The low levels of competence must necessarily be performed on the

	Competence	Level	Computational unit
Level 4	Position Planner	High level	Host computer
Level 3	Action selection	High level	Host computer
Level 2	Turn (and Scream)	Low level	Robot
Level 1	Forward	Low level	Robot
Level 0	Stop and wait	Low level	Robot

computational device that has direct access to sensors and actuators (here the robot), while higher levels of competence can be implemented in external computational devices (here the host computer).

3 The LEGO Stadium

In order to put the robot soccer play into a stimulating context, we built a whole LEGO stadium (see Figure 1). The stadium had light towers (with light) in each corner, and these towers also hold infra-red transmitters that could transmit information from a host computer to the RCXs. In one end, there was a scoreboard that could be updated when a goal was scored via an interface with the LEGO Dacta Control Lab (see Figure 4). Over each goal, there was a rolling commercial sign that held three commercials that were shown in approximately 30 seconds each before the sign would turn to the next commercial. The control of the two rolling commercial signs was made with the LEGO CodePilot. A camera-tower with a small b/w camera was placed in one corner. The camera (controlled from a CodePilot) could scan to the left and the right of the field, while displaying the image to the audience on a large monitor. Another camera was placed over the sideline on one side and should scan back and forth following the ball (see Figure 4). Also this camera image was displayed on a large monitors, and its control was made from LEGO Dacta Control Lab.

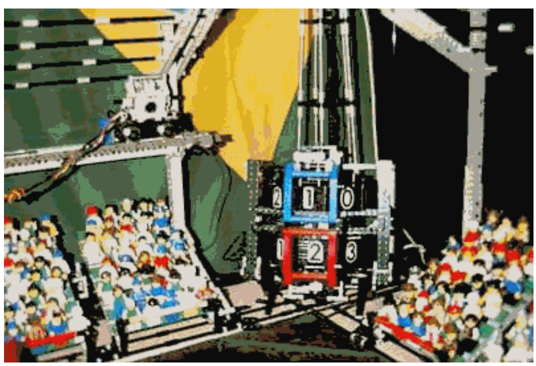

Fig. 4. The scoreboard and one of the small cameras. The camera runs up along the sideline, while projecting the images to a large monitor. ©H. H. Lund, 1998.

4 Emergent Behaviour for the Spectator Wave

When wanting to make a spectator wave (a "Mexican wave") at our stadium, we identified the spectator wave at stadiums to be an instance of emergent behaviour, rather than being with central control [5]. The "Mexican wave" that is made when spectators stand up and sit down does not have central control. The wave is initialised when a couple of spectators anywhere on the stadium decide to make the stand up + sit down movement, and some nearby spectators go with. There is no central control to tell the individual spectator to do a specific thing at a given time, rather it is an emergent behaviour.

Emergent behaviour is an interesting phenomenon that can be observed in natural systems. We define emergent behaviour as being the behaviour of a system that is the product of interaction between smaller sub-systems. The emergent

behaviour is of higher complexity than the sum of the behaviours of the smaller sub-systems. The reason that behaviour of higher complexity than the sum can emerge is the interaction between the sub-systems.
Emergent behaviour is known from flocks of birds, schools of fish and herds of land animals. When observing a flock of birds, we will notice that there is no apparent leader in the flock and there appears to be no central control of motion. The motion of the flock might seem complex and at times random, but on the other hand, it also appears synchronous. The motion of a flock of birds is an example of emergent behaviour and can be modelled as such. Reynolds [9] has made an impressive study of the general motion of flocks, herds, and schools in a distributed behavioural model with the goal of using this to model flocking in computer graphics. Recently, similar models have been used in the Disney movie *Lion King* for a wild-beast stampede and to produce photo-realistic imagery of bat swarms in the feature motion pictures *Batman Returns* and *Cliffhanger*. Reynolds calls his simulated bird-like organisms *boids* (bird-oids). The boids are controlled by three primary rules:

1. Collision Avoidance: avoid collision with nearby boids
2. Velocity Matching: attempt to match velocity with nearby boids
3. Flock Centering: attempt to stay close to nearby boids

The three rules are local in the sense that a boid only has knowledge about nearby boids and there is no global knowledge like size or centre of the flock. For instance, Flock Centering is achieved by having boids to perceive the centroid of nearby boids only. This actually gives the advantage of allowing for bifurcation: the flock can split around an obstacle in the moving direction, since the boids only tend to stay close to nearby flock-mates.
In general, the phenomenon of emergent behaviour is fundamental in a number of artificial life systems. Artificial life tries to synthesise life with a bottom-up approach by using small building blocks that emerge to a complex system by their interaction.
When using emergent behaviour in real world models, there are a number of pitfalls that we have to be aware of. When we look at Reynolds' boid model, we notice that only local knowledge of neighbours is used, so the model might appear appropriate for control tasks for autonomous agents in the real world. However, it is not clear how to obtain even the local knowledge that is available for the simulated boids. For instance, we have to solve the question of how to measure *nearby* (distance and direction). This demands an advanced sensor that can measure distance and direction, and at the same time identify an object as being a neighbour (and, for instance, not an obstacle). The task is worsened further by the demand for doing this in real time with moving objects.
There are other significant differences between a simulation model and a real world implementation that we have to take into account. For instance, the actuators will produce friction and there will be a whole range of noise issues that makes it very difficult to transfer an idealised model from simulation to the real

world. In some cases, it will be possible to transfer models from simulation to reality [8,6,2]. This is done by very careful building of a simulator that models the important characteristics of the real device and the way that real world noise interferes with this device.

In our emergent behaviour model, we work directly in the real world, so we avoid the problems of difficulties in transfer from an idealised model to the real world. Our model therefore has to work with the noise, friction, etc. that exists in the real world.

5 The LEGO spectator wave

Apart from robot soccer players, cameras, rolling commercials, and scoreboard, we had placed almost 1500 small LEGO spectators on the grandstands. Our idea was to have all these spectators make the "Mexican wave" as we see soccer fans make at real World Cup matches.

After first trying to make a prototype of the wave with a kind of open-loop con-

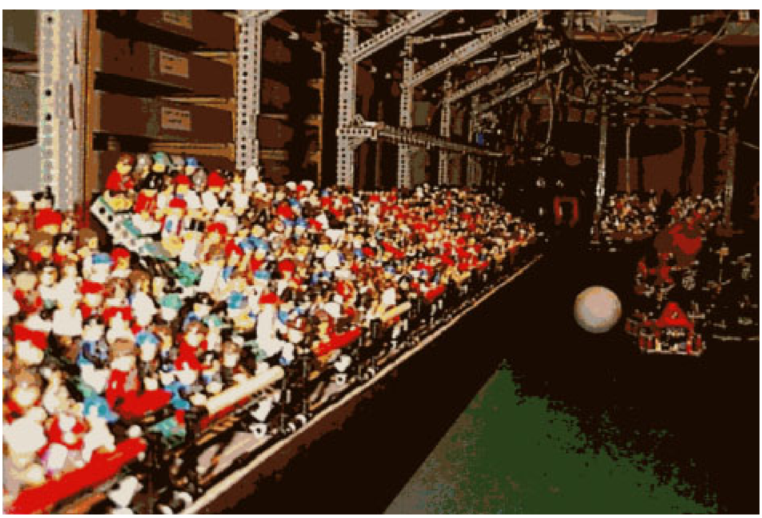

Fig. 5. The LEGO spectator wave. When the switch sensor is released, the next section of LEGO spectators will start to move upwards. ©H. H. Lund, 1998.

trol, the control of the wave was changed to a feedback control, and the idea was to allow the wave to *emerge* from the interaction between the different sectors of spectators that each had their own, local feedback control. In this case, the implementation of a system with emergent behaviour should be possible in the real world, since there would be no demand of advanced sensing. In fact, a switch sensor for each section of LEGO spectators turned out to be enough (see Figure

6). The idea was that the movement of one section should be dependent on sensing what the adjacent section was doing. If a section was moving upwards, then the section to the right should sense this and start to move upwards itself. The section would fall down when reaching the top position (in this way, we used the principle that *what goes up, must fall down*). This was built by placing a switch sensor under each section and connecting this switch sensor to the control unit (a LEGO CodePilot) of the next section. In resting mode, the switch sensor would be pressed by the section of spectators above it, but when this section started to move upwards, the switch sensor would no longer be pressed. This triggered the simple control program in the next section to start moving the section of LEGO spectators upwards. In pseudo-code, the control program of each section could be as follows:

```
Section N control:

            if (Switch(N-1)=false) then turn on motor
            else turn off motor
```

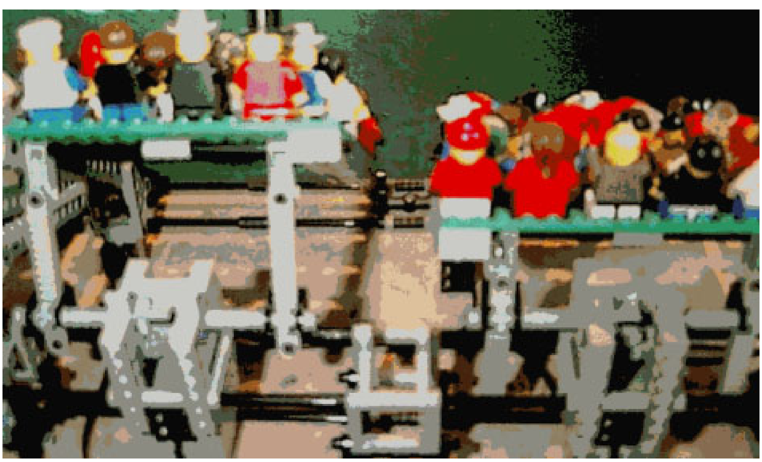

Fig. 6. The movement of a section is triggered by the switch sensor mounted underneath the adjacent section. The left section of spectators has risen, so the switch sensor is no longer pressed. The control of the right section will notice this, and start to move upwards (immediately after this photo was taken). ©H. H. Lund, 1998.

This very simple control allows the wave to emerge when one section is triggered from the external to move upwards. In the actual implementation in the LEGO CodePilot language, it was however necessary to use a timer, since the time slice of the CodePilot is so small, that the above pseudo-code program would result in the section barely moving upwards before the motor would be turned off. So when it was sensed that the switch was no longer pressed, the control would turn the motor one direction for 0.5 seconds and then the other direction for 0.5 seconds. It would have been more sensible to have a switch sensor on the top position that the section should reach, and then base the time of upward and downward movement on the feedback from the top and the bottom sensors. But since the LEGO CodePilot has only one input channel (see Figure 7), we opted for the timing solution. However, it must be noted that this timing is very different from the timing in the first prototype, since here, even though the timing of a section might have been wrong, the section would still lift itself for some time and therefore the next section would be triggered. In a sense, we are setting the time-slice to 0.5 seconds and use the pseudo-code program.

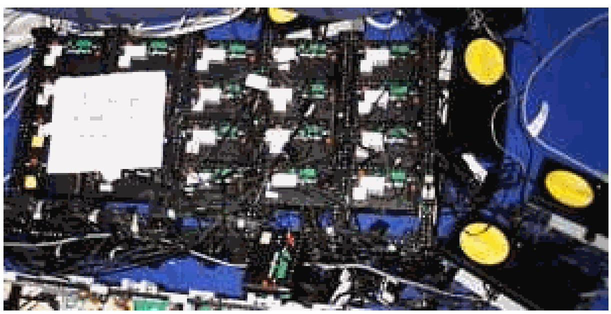

Fig. 7. The mixer. 16 LEGO CodePilots were used to construct the dynamics of the "Mexican wave". Each CodePilot had one switch sensor and one motor connected. ©H. H. Lund, 1998.

The feedback control was used and the wave emerged from the interaction between the simple units. It was run numerous times daily in Paris for a week without any need for refinements apart from a couple of changes of physical aspects (one or twice, a section got stuck).

6 Conclusion

The distributed behaviour-based control system gave a robot soccer play that allowed the LEGO Mindstorms robots to play a good robot soccer game, where goals were scored in most periods. In the demonstration tournament, we played games with five periods of up to two minutes (or until a goal was scored), and on average 3-4 goals were scored in each match. This is far higher average of goals/period than in most other robot soccer games, and it was essential for us to achieve this performance in order to provide a spectacular game for the public audience.

We used emergent behaviour to construct a "Mexican wave" of LEGO spectators for the LEGO robot soccer demonstration. The wave emerged from the interaction between sections of LEGO spectators, each with its own simple control. The control was based on feedback from the local environment. Since sensing was straightforward with one switch sensor for each section, it was fairly easy to implement a real world emergent behaviour. Under other circumstances, it might be more difficult, since more advanced sensing might be necessary, and we cannot guarantee that the desired real world behaviour will emerge. Therefore, the study of real world emergent behaviour is important in order to identify the circumstances that will lead to successful results.

Further information, including photos and videos are available on the LEGO Lab web-site:
`http://legolab.daimi.au.dk/`

Acknowledgements

Jens Aage Arendt designed the LEGO stadium in collaboration with Jakob Fredslund. Students from the LEGO Lab made a highly valuable work in the LEGO robot soccer RoboCup'98 project. Lars Nielsen from LEGO Media worked as a very helpful partner who gave numerous valuable ideas. Ole Caprani participated in a number of inspiring discussions. Thanks to H. Kitano for collaboration on the Paris RoboCup'98 event.

References

1. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics and Automation*, RA-2(1), 1986. 145
2. N. Jakobi. The Minimal Simulation Approach to Evolutionary Robotics. In T. Gomi, editor, *Proceedings of ER'98*, Kanata, 1998. AAI Books. 148
3. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. RoboCup. a Challenge Problem for AI. *AI Magazine*, (spring):73–85, 1997. 141
4. H. H. Lund. Robot Soccer in Education. 1999. submitted to Advanced Robotics journal. 141
5. H. H. Lund, J. A. Arendt, J. Fredslund, and L. Pagliarini. Ola: What Goes Up, Must Fall Down. In *Proceedings of Artificial Life and Robotics 1999 (AROB'99)*, pages I9–I15, Oita, 1999. ISAROB. 146
6. H. H. Lund and O. Miglino. From Simulated to Real Robots. In *Proceedings of IEEE Third International Conference on Evolutionary Computation*, NJ, 1996. IEEE Press. 148
7. H. H. Lund and L. Pagliarini. LEGO Mindstorms Robot Soccer. a Distributed Behaviour-Based System. 1999. To be submitted. 145
8. O. Miglino, H. H. Lund, and S. Nolfi. Evolving Mobile Robots in Simulated and Real Environments. *Artificial Life*, 2(4):417–434, 1995. 148
9. C. W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987. 147