Real-Time Modelling and Rendering of Sprayed Concrete

G. Vélez[1], L. Matey[1,2], A. Amundarain[1], F. Ordás[3] and J.A. Marín[3]

[1]CEIT, Spain
[2]Tecnun (University of Navarra), Spain
[3]Fundación Santa Bárbara, Spain

ABSTRACT

This paper presents a new real-time method to model and render how sprayed concrete is spread on a surface. The method not only models and renders deposits sprayed from any angle, any distance and with any concrete flow, but it is also able to compute the amount of deposited volume taking into account the percentage of material that rebounds. The proposed method has been developed for a real-time training simulator for concrete spraying machinery, where most of the algorithm is parallelised and computed in the GPU, leaving the CPU free for other computations. In this research, the method has been validated for its use on plain surfaces and tunnel walls, but it can be extended to other types of surfaces.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

1. Introduction

Nowadays simulators are gaining importance in the training of civil machinery operators due to the reduction in cost of these tools. The main advantages of simulators are the economical benefit thanks to the savings in materials and the practical absence of safety risks. Other simulators for civil work have been successfully implemented such as [VSA*09] and [PCGFGMD09]. In a similar way, tools for training operators of concrete spraying machinery are required. Shotcrete, or sprayed concrete, is used in tunnels and other underground constructions to preserve the stability of the rock. The aim of the operator is to cover the tunnel walls with a uniform concrete layer of a specific thickness.

This research project is part of a program to develop a training simulator for a concrete spraying system. Figure 1 shows the entire spraying system, which consists of a carrier vehicle and a telescopic spraying arm. The nozzle, from which the concrete is sprayed, is at the end of the telescopic arm and is guided manually by a machine operator. The spraying system is specially designed for its use in mines, tunnels and pits.

The training simulator must reproduce the concrete spraying system with a high degree of realism, providing complete training on each of the different spraying modes, maintenance operations, and diagnose and solution of breakdowns. The training simulator's architecture consists of an instructor station and up to eleven trainee stations. The instructor station is able to edit, visualise and evaluate the training exercises that run in the trainee stations. Each station is composed of a standard PC, forming a network where the instructor station can control remotely any other station.

Figure 1: *Concrete spraying system.*

The main difficulty in the development of this training simulator is the creation of a realistic model of shotcrete behaviour for a real-time environment. More specifically, there is a necessity to model how the concrete is spread throughout the wall. The complexity of a continuum approach has led to the use of discrete methods for shotcrete modelling, like the Discrete Element Method (DEM) [CS79]. The first to introduce DEM in shotcrete research were Puri and Uomoto [PU99], who modelled the fresh concrete as a two phase granular material, composed of gravel and mortar particles. This method has recently been employed to simulate the workability of fresh concrete [GYSF10]. DEM produces accurate physical simulations but it is computationally expensive. Thus, it is not the best solution for a real-time application.

In [GM01], the theoretical shotcrete layer thickness is calculated by overlapping a spray profile in parallel horizontal path lines. The spraying distance, the nozzle velocity and the perpendicularity of the nozzle to the rock are maintained constant throughout the entire simulation, which are limitations that a training simulator cannot assume. The way concrete spreads on the wall can also be modelled using a spraying density function, but such a function can only be represented analytically in the simplest cases [RNR09]. In all the other cases, some tests must be performed to estimate the parameters of the model. Then, the parameterised model can be used for making thickness predictions.

[BT09] proposed using an adhesion calculation system that refreshes the data stored in a height map. However, their focus is on the visual appearance and not on the development of a realistic model, while in a training simulator both are necessary.

This paper presents a new method that generates the surface of a shotcrete layer by adding spray deposits. The method divides the tunnel walls in a grid, and casts a ray from the nozzle to each cell, computing how much concrete is deposited. The data is stored in a height map, which is used to render the concrete surface. This method can run in a real-time application such as a training simulator.

The paper is organised as follows. In the second section, the modelling of the proposed method is described, and in the third section how this model is rendered is explained. The forth section shows the results and the last section presents the conclusions.

2. Modelling

This section explains how the proposed method obtains a three-dimensional model of the shotcrete layer's surface. The section is divided into three main parts. The first one describes the calculation of the shape of each static deposit that forms the shotcrete layer profile, the second explains how the deposited volume is estimated, and the last one details the implementation of the described model.

2.1. Estimation of the shotcrete layer profile

In our method, similarly to [GM01] [RNR09], the shotcrete layer profile is calculated as a sum of static spray profiles. This idea has also been used in agricultural engineering to calculate the herbicide distribution on crop fields [TZ00]. At each time step, the corresponding spray deposit is calculated and added to the previously calculated ones to form the current layer.

Static spray distribution models are represented as a function describing the height of the deposit in two-dimensional space [RMDB97]:

$$h = \begin{cases} f(x,y) & \text{if } x,y \in dom(f) \\ 0 & \text{if } x,y \notin dom(f) \end{cases} \quad (1)$$

The domain of function f, dom(f), is the stationary nozzle spray cone covering area. The way the static spray deposit is calculated is the main innovation of this method.

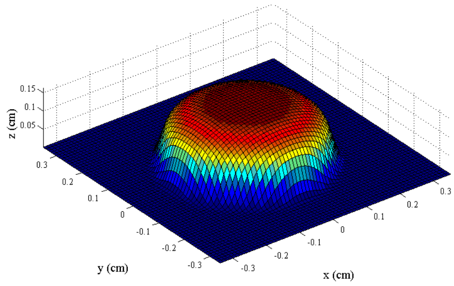

Figure 2: *3D representation of a shotcrete deposit generated from an experimental sample.*

As a first step to get the proposed static spray distribution model, some tests were run under ideal working conditions: with the nozzle perpendicular to a plain surface and situated at a distance of 1.5 metres. The measurements concluded that in these working conditions, with the nozzle perpendicular to a plane surface, the deposits can be represented as a surface of revolution, as shown in Figure 2. Only the half cross section of one of these sample deposits needs to be stored, as the rest of the deposit shape can be reconstructed using this information. Additionally, with our algorithm, any other deposit shape will be predicted using this half cross section.

From the measurements, a table is obtained that relates the angle to the nozzle direction vector, β, and the height of the sample deposit, h_s. Figure 3 shows the measured relationship between β and h_s. Linear interpolation has been used to obtain the height for all possible angles.

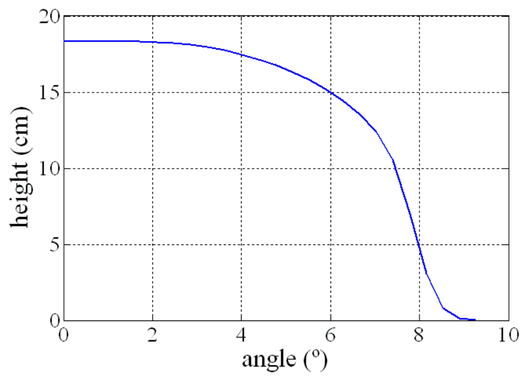

Figure 3: *Angle between the nozzle direction and the ray (β) vs deposit height (h_s).*

In the algorithm, the surface that is going to be shotcreted is divided into a grid that forms the height field that will define the shotcrete layer's surface. In the i^{th} time step, a ray from the nozzle tip to the centre of each cell of the grid is casted. The angle β_i is calculated as depicted in Figure 4, using as input the total height of the cell at that time instant, $h_{CTi} = \sum_{n=0}^{i-1} h_{Cn}$, where h_{Cn} is the height's growth at the n^{th} time instant.

Depending on the angle between the nozzle direction vector and the ray, a certain amount of concrete volume is deposited on the cell. This volume is equal to the multiplication of the cell's area (A_{Ci}) and the increase in the cell's

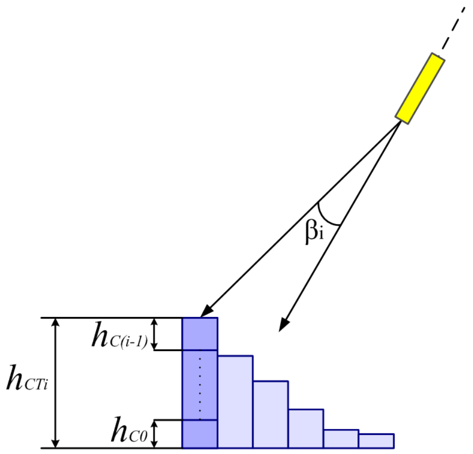

Figure 4: *Angle between the nozzle direction vector and the ray* (β_i)*.*

height (h_{Ci}). Assuring volume conservation, this increase in the cell's height can be estimated as:

$$h_{Ci}(\beta) = h_S(\beta) \cdot \frac{A_{Si}}{A_{Ci}} \tag{2}$$

where A_{Si} is the area on the sample deposit. In order to obtain this area, A_{Si}, we propose projecting the cell area on a plane perpendicular to the nozzle direction as shown in Figure 5. If a line from the nozzle to each of the vertexes of the cell is traced, the projected area is delimited by the intersection points of the four lines with the plane, as seen in Figure 5.

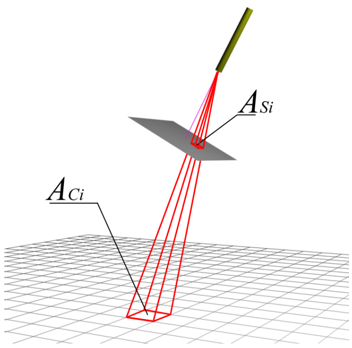

Figure 5: *Projection on a perpendicular plane of the area that covers the ray.*

The volume deposited in A_{Si} equals the volume deposited in A_{Ci}, as Figure 6 shows. Each ray deposits a certain amount of material, and this amount depends on where this ray is situated in the spray cone. If the ray is outside the spray cone, because the angle between the nozzle direction vector and the ray is bigger than the spray angle, it will not deposit anything. There are different models of spraying nozzles, with different spray angles, and it is possible to simulate them by changing the table displayed in Figure 3. This table can be obtained by means of experimental measurements, as is the case in this research project.

For every time step that a cell is sprayed, its height grows, changing the shape of the shotcrete layer. The proposed algorithm is employed to estimate this height growth. This

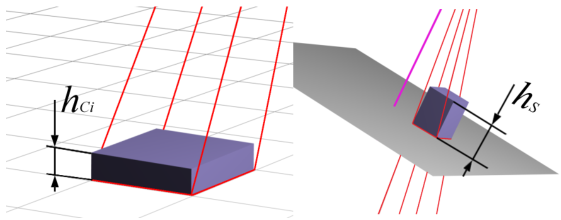

Figure 6: *Volume conservation.*

model can be extended to be used on different types of surfaces, such as tunnel walls. In this case, the height of each cell grows following the normal vector of the tunnel's excavated profile on that point, and the sprayed area of each cell is approximated by simplifying that surface piece to a plane. Figure 7 shows a tunnel section where the points that delimit

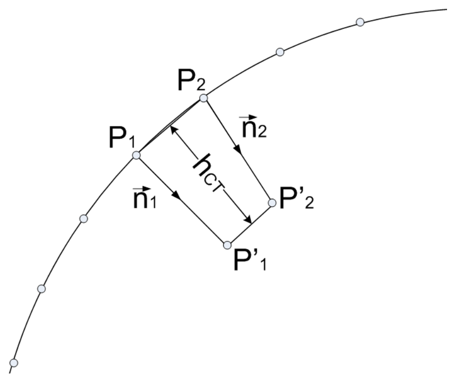

Figure 7: *Tunnel section with a sprayed cell.*

the cell's top face are a translation of the cell's base points following a normal vector. This way:

$$P'_n = P_n + \hat{n}_n \cdot h_{CT} \tag{3}$$

where $\hat{n}_n$ is the unitary normal vector. Note that this sprayed area, A_{Ci}, decreases as the cell's total height increases, since cells grow on the concave part of the tunnel. This model can easily be applied to other types of surfaces, it is only necessary to know the normal vector and the Cartesian coordinates of the surface points that define the height field's grid.

2.2. Deposited volume

One of the required features for a concrete sprayer training simulator, is the control of the output concrete flow of the spraying machine during the simulation. Using equation (2), all the sprayed deposits have the same volume, which means that the concrete flow remains constant throughout the simulation. To break this limitation, we just add a new term to the equation:

$$h_{Ci}(\beta) = \frac{V_{Di}}{V_S} \cdot h_S(\beta) \cdot \frac{A_{Si}}{A_{Ci}} \tag{4}$$

where V_{Di} is the volume we want to obtain for this static deposit, and V_S is the volume we obtain using directly (2). The volume of each static deposit defines the current concrete flow, so:

$$V_{Di} = Q_i \cdot t_i \tag{5}$$

where Q_i is the output flow and t_i is the time the step takes.

However, in a real application, not all the sprayed material is stuck on the surface, some percentage is lost due to the rebound effect. The machine operator should minimise the amount of material rebounded, because a waste of material implies a waste of money. The skill of the trainee minimising the rebounds is one of the main evaluation factors of the simulator's training exercises, so it is necessary to estimate correctly this rebounded volume.

There are four major factors that influence the amount of concrete rebound [Mel01]: nozzle angle to substrate, accelerator dosage, nozzle distance to substrate and area of application in the tunnel.

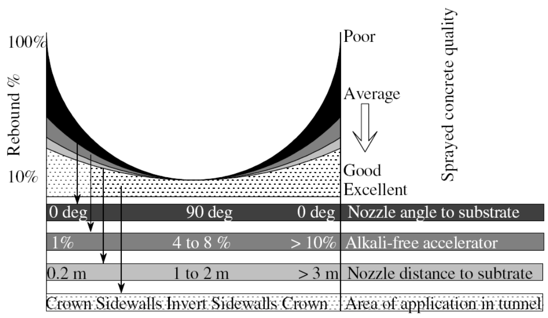

Figure 8: *The principal spraying factors that influence rebound [Mel01].*

Figure 8 displays a graph created by [Mel01] that shows these factors influence, which has also been used in other research projects [BT09] [RR07] [MNR05]. The most significant factor is the nozzle angle to substrate, which is the angle that forms the vector that defines the nozzle direction and the normal vector of the surface at that point. This angle should always be maintained near 90°. The next two factors are the accelerator dosage and the nozzle distance to substrate. The accelerator is one of the shotcrete components, that it is used to enhance its performance in shortening the setting time. These two factors do not influence the rebounds if they are maintained at optimal working values. Finally, the last factor, the area of application, is equivalent to the influence gravity has on the process. Analysing Figure 8, it can be noticed that the greatest amount of rebound is at the crown, which is the highest part of the tunnel, and in this position, the nozzle is oriented upwards, against gravity.

In our model, the rebound percentage is calculated using these curves. Each time step, the values of these four factors are calculated and then, the influence of each factor is estimated by looking up the curves. Finally, the deposited volume is calculated as the difference between the output volume and the rebounded volume, obtaining the following equation to compute the height growth of each cell at each time step:

$$h_{Ci}(\beta) = \frac{(100 - P_{Ri})}{100} \cdot \frac{Q_i \cdot t_i}{V_S} \cdot h_S(\beta) \cdot \frac{A_{Si}}{A_{Ci}} \qquad (6)$$

being P_{Ri} the percentage of rebounded material.

2.3. Implementation

A great advantage of this algorithm is that the new height of each point only depends on the nozzle emplacement and its previous height. Therefore, the new heights can be calculated using the GPU, exploiting the parallelisable features of the algorithm.

The model proceeds iteratively, using the output texture from the previous step as input texture. Thus, in our implementation of the model, the height field data is stored in two textures and a ping-pong technique is used to access to them. The dimension of these textures depends on the requirements of the simulation. More precise simulations are achieved by using bigger textures; however the computational costs also increase.

Additional information must be transferred to the shader to fulfil the simulation step, both the information that describes the shape of the surface where the concrete is sprayed and the information of the sample deposit that describes the spray distribution.

The information describing the shape is transferred using a texture where the data of the excavated profile is stored. Information describing the tunnel's geometry and the normal vector of each point is stored in the texture. The size of this texture depends on the irregularities of the profile. In irregular profiles more data is needed to describe it, therefore bigger textures are necessary.

The second data, the information of the sample deposit, is also transferred using a texture. For computational efficiency, the height of the deposit is transmitted relative to the cosine of the angle instead of the angle. Thanks to this difference, several cosine computations are avoided in the shader.

The implemented algorithm can be summarised as follows:

1. Calculate the volume of the current static spray deposit.
2. Calculate the plane perpendicular to the nozzle direction.
3. Cast a ray to each cell of the height field and calculate the cosine of the angle between the nozzle direction vector and the casted ray.
4. Project the area of each cell on a plane perpendicular to the nozzle direction.
5. Compute the equation (6) to calculate the height growth of each cell, and store it in the output texture.
6. Swap input and output textures.

The algorithm is computed each time a frame is drawn. The first two points are computed in the CPU, and the rest in the GPU.

3. Rendering

For rendering the concrete layer a square mesh is defined, the resolution of the mesh is equal to the resolution of the texture where the height field is defined. The height field information is updated in each simulation step, and in each texel the displacement of the corresponding vertex is stored.

A shader has been programmed to render the concrete layer. The shader receives as input data the squared mesh, the height field texture and the texture where the profile of the tunnel is stored.

First, the real position of the vertex in the tunnel and its normal vector are calculated using the profile texture. Then the vertex is displaced according to the quantity stored in the height field texture following the normal vector's direction. Then a new normal vector is calculated, taking into account the profile of the tunnel and the shape of the concrete layer. This normal vector is necessary to obtain realistic lighting.

If the rendering phase finishes at this point, the concrete will have a soft look, without the roughness that appears in concrete surfaces. This problem is solved using Perlin noise [Per02]. The perlin noise is transferred to the shader and is used to modify slightly the normal vector, achieving a more realistic appearance.

4. Results

The model described above has been implemented in a new training simulator for a concrete spraying machine. Each time step, the rebound percentage and the coefficients of the plane perpendicular to the nozzle direction are obtained. The rest of the model is computed in the GPU using Cg, thereby parallelising the majority of the algorithm..

In order to test the volume conservation of (2), some tests were run with different nozzle distances and different nozzle angles to the surface. All the deposits were sprayed on the same tunnel area. The tunnel measured 6.47 metres high, 8.9 metres wide and 25 metres long. The height field resolution was 512x512 cells.

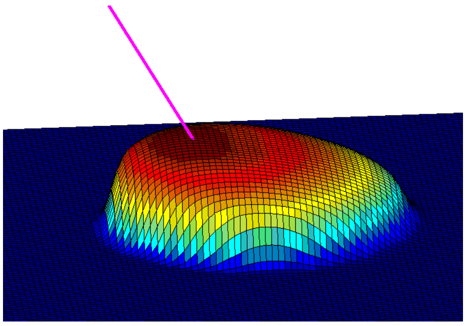

Figure 9: *Deposit sprayed with a nozzle angle of 60°.*

Table 1: *Volume conservation test results*

Nozzle distance (m)	Nozzle angle (°)	Relative error (%)
1	90	4.752
1	60	3.029
1	45	2.868
1.5	45	0.41
2	90	0.2244

Table 1 shows the error in the volume estimation of each static deposit relative to the sample deposit. This sample deposit is the one obtained from the tests, which is also used to generate the curve displayed in Figure 3. In all the studied cases, this relative error is below 5 %. Thus, we can conclude that the volume conservation is fulfilled with the proposed method. It can also be noticed that the relative error decreases as the covering area of the spray cone increases. This is due to the fact that a bigger area involves a greater amount of cells, which reduces the discretisation error.

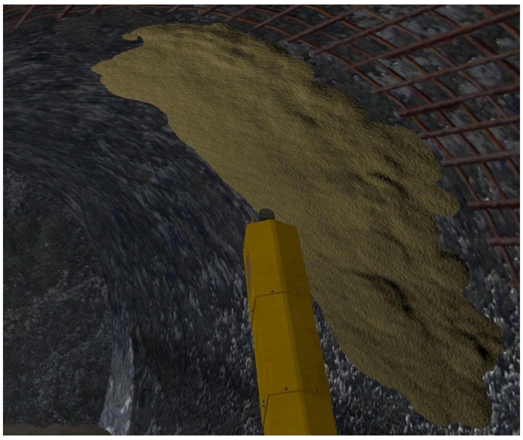

Figure 10: *Sprayed tunnel wall.*

Figure 11: *Sprayed tunnel wall.*

Figure 9 displays a concrete deposit sprayed on a plain surface with a nozzle angle of 60°. The magenta coloured line represents the nozzle direction. The maximum height of the deposit is on the nozzle direction, which is a logical result since it is in the centre of the spray cone where more material is deposited. The shape difference between this and a perpendicularly sprayed deposit can be noticed by comparison with Figure 2. Figure 10 and Figure 11 display screenshots of the training simulator where a sprayed tunnel wall can be seen. We can observe that the proposed model is able to represent accurately how the shotcrete is sprayed onto the wall.

The proposed method has been successfully implemented in a real-time application, where the computation time of the method has been measured. The application, a training simulator, runs on an Intel Core2 Quad Q9550, with 3.25 GB of RAM and a NVIDIA GeForce GTX 275 GPU. Using a texture size of 512x512 for the height field, the computing time of the modelling and rendering is 1.72 ms. With a texture of 1024x1024, the computing time is 6.88 ms. The obtained computing times are suitable for a real-time application, and moreover, the proposed method is much more time efficient than other methods such as DEM [CS79].

5. Conclusions

This paper presents a new method to model and render in real-time how shotcrete is spread on a surface. Each time step a static spray deposit is calculated and it is added to the final shotcrete layer. The way the static spray deposit is calculated is the main contribution of the method. The surface is divided into a grid, and a ray is casted from the nozzle to each cell, computing how much concrete is deposited. The data is stored in a height field, which is used to render the concrete surface.

In order to prove the volume conservation of the static spray deposit calculation model, some tests were run with different nozzle distances and different nozzle angles to the surface. The relative error was below 5 % in all cases, satisfying the considered volume conservation hypothesis. Moreover, the method is able to compute in real-time the deposits sprayed from any angle, any distance and with any concrete flow. Thus, the method has been proven valid for its use on a training simulator for concrete spraying machinery. The method is also capable of computing the amount of deposited concrete volume considering the percentage of material that rebounds due to the four most influential factors. Most of the proposed algorithm is parallelised and computed on the GPU, leaving the CPU free for other computations. Finally, the model has been validated for its use on a plain surface and on tunnel walls, but it can also be extended to other surfaces.

Acknowledgments

We want to express our gratitude to Fundación Santa Bárbara for leading this project where we have worked under contract. G. Vélez holds the Asociación de Amigos de la Universidad de Navarra grant.

References

[BT09] BÖRJESSON P., THELL M.: *Shotcrete simulator for education of shotcrete robot operators*. Master of science thesis, University of Gothenburg, 2009.

[CS79] CUNDALL P., STRACK O.: A discrete numerical model for granular assemblies. *Geotechnique*, 2 (1979), 47–65.

[GM01] GIRMSCHEID G., MOSER S.: Fully automated shotcrete robot for rock support. *Computer-Aided Civil and Infrastructure Engineering*, 16 (2001), 200–215.

[GYSF10] GUO Z., YUAN Q., STROEVEN P., FRAAIJ A. L. A.: Discrete Element Method Simulating Workability of Fresh Concrete. In *Proceedings of the 2nd International Symposium on Computational Mechanics and the 12th International Conference on the Enhancement and Promotion of Computational Methods in Engineering and Science* (2010), vol. 1233, pp. 430–435.

[Mel01] MELBYE T.: *Sprayed Concrete for Rock Support*, 9 ed. MBT International Underground Construction Group, Division of MBT, 2001.

[MNR05] MALMGREN L., NORDLUND E., ROLUND S.: Adhesion strength and shrinkage of shotcrete. *Tunneling and Underground Space Technology 20*, 1 (2005), 33–48.

[PCGFGMD09] PLA-CASTELLS M., GARCÍA-FERNÁNDEZ I., GAMÓN M., MARTÍNEZ-DURÁ R.: Interactive earthmoving simulation in real-time. In *CEIG'09* (2009).

[Per02] PERLIN K.: Improving noise. In *SIGGRAPH* (2002), pp. 681–682.

[PU99] PURI U., UOMOTO T.: Numerical modeling - a new tool for understanding shotcrete. *Materials and Structures 32* (May 1999), 266–272.

[RMDB97] RAMON H., MISSOTTEN J., DE BAERDEMAEKER J.: Spray boom motions and spray distribution : Part 2 , experimental validation of the mathematical relation and simulation results. *Journal of Agricultural Engineering Research 66*, 1 (Jan. 1997).

[RNR09] RODRIGUEZ A., NABULSI S., RÍO O.: A method for estimating thickness of sprayed concrete layers from pumped volume. In *Proceedings of the 2nd Thematic Conference on Computational Methods in Tunnelling (EURO:TUN 2009)* (2009).

[RR07] RODRIGUEZ A., RÍO O.: Analysis of real time technical data obtained while shotcreting: an approach towards automation. In *ECCOMAS Thematic Conference on Computational Methods in Tunnelling (EURO:TUN 2007)* (2007).

[TZ00] TIAN L., ZHENG J.: Dynamic deposition pattern simulation of modulated spraying. *Transactions of the ASABE 43*, 1 (2000), 5–11.

[VSA*09] VALERO A., SUESCUN A., AMUNDARAIN A., DE DIOS C., ALIAGA I.: A highly realistic training simulator for the tunneling jumbo sandvik dt1130-c data. In *Proceedings of the 2009 Huntsville Simulation Conference* (2009), pp. 259–265.