

Vieillissement interactif pour la synthèse d'image réaliste des environnements urbains

Imanol Muñoz Pandiella

▶ **To cite this version:**

Imanol Muñoz Pandiella. Vieillissement interactif pour la synthèse d'image réaliste des environnements urbains. Graphics [cs.GR]. Université de Limoges; Universitat de Girona, 2017. English. ⟨NNT : 2017LIMO0089⟩. ⟨tel-01707902⟩

HAL Id: tel-01707902

https://theses.hal.science/tel-01707902v1

Submitted on 13 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Thèse de doctorat

Université de Limoges

Universitat de Girona

Thèse en cotutelle internationale entre

Université de Limoges

École Doctorale Sciences et Ingénierie pour l'Information, Mathématiques (ED 521)

Institut de Recherche XLIM - UMR CNRS 7252

Universitat de Girona

Escola de Doctorat

Departament IMAE

Centre de Recerca ViRVIG

Thèse pour obtenir le grade de

Docteur de l'Université de Limoges

en informatique graphique

et le grade en

Doctor per la Universitat de Girona

Présentée et soutenue par

Imanol Muñoz Pandiella

Le 5 décembre 2017

Interactive Weathering for Realistic Image Synthesis of Urban Environments

Thèse dirigée par Stéphane MÉRILLOU, Carles BOSCH et Xavier PUEYO
et co-encadrée par Nicolas MÉRILLOU

JURY :

Rapporteurs	
M. Pierre POULIN	Professeur, Université de Montréal
M. Livio DE LUCA	Directeur de Recherche CNRS
Examinateurs	
M. Jean-Michel DISCHLER	Professeur, Université de Strasbourg
M. Carlos ANDUJAR	Maitre de conférences, Universitat Politècnica de Catalunya
Directeurs	
M. Stéphane MÉRILLOU	Professeur, Université de Limoges
M. Xavier PUEYO	Professeur, Universitat de Girona

INTERACTIVE WEATHERING FOR REALISTIC IMAGE SYNTHESIS OF URBAN ENVIRONMENTS

IMANOL MUÑOZ PANDIELLA

PHD THESIS, UNIVERSITÉ DE LIMOGES, UNIVERSITAT DE GIRONA

December 2017

This thesis is licensed under the **Attriution-NonCommercial-NoDerivs 3.0 France License**.
To view a copy of this license, visit:
http://creativecommons.org/licenses/by-nc-nd/3.0/fr/



Als meus,
per l'alè necessari a cada passa.

(cc) BY-NC-ND

(cc) BY-NC-ND

Las cosas hermosas, las obras de arte, los objetos sagrados, sufren, como nosotros, los efectos imparables del paso del tiempo.

MATILDE ASENSI

(cc) BY-NC-ND

(cc) BY-NC-ND

Acknowledgements

First of all, I would like to thank Xavier Pueyo for motivating me during this thesis and giving me a lot of support for its achievement. Moreover, I would also like to thank Carles Bosch for his invaluable aid at each step of this thesis and his giant effort to make me succeed. I would also thank Stéphane and Nicolas Mérillou for their help and work during the thesis and for the warm care of me during my stays in Limoges as well as their encouragement.

I am also very grateful to the people at the Graphics Group of Girona, for the treatment that I have received during these years. Specially, I would like to thank Gustavo Patow who first encouraged me to do a PhD and who has collaborated with good ideas and hours of work to improve my work. Moreover I would like to thank the several office colleagues I had, particularly Lien, Raisel, Fran and Oriol for the coffee moments and for backing me. Moreover, I would also thank the lunch group, Roberto, Yeimy and Robert, who has helped me to disconnect with interesting conversations.

I also want to thank the people of XLIM, specially those members of the graphics group, for their welcome during my stays in Limoges. Specially I thank my PhD colleagues, for the good moments I spent with them. Thanks also for the people outside the university who I met there, specially the spaniards and dentists group, for the good moments and company during my time in Limoges. It has been very nice to have been in the Limousin region and to have experienced their culture and traditions. I will always have a bit of it with me.

Furthermore, I would like to thank Holly Rushmeier for her welcome, help and work during my stay at Yale University. I would also like to thank Julie Dorsey, the people who I have work with, specially Goze and the colleagues of the Yale Graphics group: Justin, Zach, Alex and Victoria. Moreover, I would like to thank the people I have met in New Haven, specially in Hellatight house, who has helped me to better understand the American culture and its people.

I do not want to forget to thank the Barcelona team of the ViRVIG Research Center. They are the first who introduced me in Computer Graphics and who encouraged me to start a research career. Specially, I want to thank Pere Brunet, Isabel Navazo and Eva Monclús. Moreover, I would like to thank the office colleagues that I have had there during my membership and during the days that I have worked there in my PhD.

Finally, I would like to specially thank Claudia, whose affection and understanding have accompanied me so many times along this thesis, and to my family, specially brothers and parents, for their support and cheering during my work in spite of not understanding so much its purpose. I do not either forget my friends from Barberà del Vallès and Barcelona, who made me have very good moments of disconnection during these years.

This thesis was funded by Doctoral grant of region Limousin and was partially funded by the ANR under ref. ANR-16-CE33-0001 (France), TIN2014-52211-C2-2R project from Ministerio de Economía y Competitividad (Spain) and by MPCUdG2016/125 project from Universitat de Girona (Spain).

(cc) BY-NC-ND

(cc) BY-NC-ND

Contents

I Introduction 1

I.1 Motivation 1

I.2 Contributions 2

I.2.1 Secondary Contributions 4

I.3 Manuscript Outline 5

II Previous Work 7

II.1 Weathering in CG 7

II.1.1 Centered on Effects 8

II.1.2 Multi-effect Techniques 15

II.2 Estimation of Weathering Factors 20

II.2.1 Solar Radiation Estimation 20

II.2.2 Wind Estimation 22

II.3 Weathering Effects in Cultural Heritage 24

II.3.1 Cultural Heritage Analysis 24

II.3.2 Computer Graphics Analysis Tools 26

II.4 Conclusions 26

III Solar Exposure 29

III.1 Introduction 29

III.2 Overview 30

III.3 Solar Radiation 31

III.3.1 Computing Daylight Radiance 32

III.3.2 Annual Exposure 34

III.4 Sky Visibility Estimation 35

III.4.1 Image-Space Horizon-Based Ambient Occlusion 37

III.4.2 Image-Space Horizon-Based Sky Visibility 38

III.5 Surface Solar Exposure 42

III.6 Results and Discussion 46

III.6.1 Results 46

III.6.2 Discussion 52

(cc) BY-NC-ND

III.7 Conclusions 53
IV Pollution 55
IV.1 Introduction 55
IV.2 Overview 56
IV.3 Pre-process 58
IV.3.1 Wind Map Generation 58
IV.3.2 Pollution Map Generation 61
IV.4 Weathering Factors 63
IV.4.1 Wind Accessibility (A_w) 63
IV.4.2 Insolation (I) 63
IV.4.3 Rain Accessibility (A_r) 65
IV.4.4 Sky Accessibility (A_s) 67
IV.4.5 Porosity (p) 67
IV.5 Deposition Model 67
IV.5.1 Particle Deposition (C_D) 67
IV.5.2 Chemical Reactions (H) 70
IV.5.3 Washing Effects (W) 70
IV.5.4 Final Render (R) 71
IV.6 Results and Discussion 71
IV.6.1 Tests 71
IV.6.2 Results 80
IV.6.3 Discussion 85
IV.7 Conclusions 86
V Stone Weathering 89
V.1 Introduction 89
V.2 Overview 91
V.3 Model Generation 92
V.3.1 Capturing Models 92
V.3.2 Pre-processing the Model 92
V.4 Original Surface Estimation 95
V.5 Weathered Zones Detection 98
V.5.1 Feature Filtering 98
V.5.2 Detecting Scaling Weathering Effects 100

(cc) BY-NC-ND

V.5.3 Segmentation of Weathered Regions 102
V.5.4 Mapping the Regions 103
V.6 Results and Discussion 104
V.6.1 Results 104
V.6.2 Discussion 109
V.7 Conclusions 110
VI Conclusions and Future Work 111
VI.1 Conclusions 111
VI.2 Future Work 112
VI.2.1 Solar Radiation Estimation 113
VI.2.2 Urban Pollution Weathering Model 113
VI.2.3 Stone Weathering Analysis 113

(cc) BY-NC-ND

(cc) BY-NC-ND

List of Figures

I.1 Examples of weathering effects in urban environments. 1

II.1 Results of different scratch techniques. From left to right, images from Merillou et al. [MDG01a], Bosch et al. [Bos+04], Raymond et al. [RGB16] and Werner et al. [Wer+17]. 8

II.2 Results of different cracks techniques. From left to right, images from Desbenoit et al. [DGA05], Iben and O'brien [IO06], Glondu et al. [Glo+12] and Pfaff et al. [Pfa+14] works. 9

II.3 Different terrains generated taking into account erosion processes. From left to right, images from Crespin et al. [Cre+14] and Cordonier et al. [Cor+16] [Cor+17]. 10

II.4 Results of different matter deposition techniques. From left to right, images from Dorsey et al. [DPH96], Gu et al. [Gu+07] and Bosch et al. [Bos+11]. 11

II.5 Examples of chemical weathering effect simulations. From left to right, destructive corrosion by Mérillou et al. [MDG01a], patina by Dorsey and Hanrahan [DH96], fading by Kimmel et al. [Kim+13] and pollution deposition by Mérillou et al. [Mer+10]. 12

II.6 Different scenes after a matter dissolution process produced by chemical reactions. From left to right, images generated by Dorsey et al. [Dor+99], Jones et al. [Jon+10], Mérillou et al. [Mer+12] and Xue et al. [XDR11]. 13

II.7 Examples of different biological weathering effects. From left to right, organic material growth by Hadrick et al. [Ḧ17], senescense by Kider et al. [KRB11] and sking aging by Iglesias-Guitian et al. [IG+15]. 15

II.8 Different weathering effects simulated using γ-ton technique [Che+05]. From left to right, we can observe corrosion, erosion and stain bleeding. 15

II.9 Different examples of weathering effects transferred from a captured object. From left to right, patina from Lu et al.[Lu+07], pottery decay from Mertens et al. [Mer+06] and corrosion from Wang et al. [Wan+06]. 17

II.10 Different images generated using image-based weathering methods. From left to right, cracks from Bellini et al. [BKCO16], lichen growth from Iizuka et al. [Iiz+16] and corrosion from Xue et al.[Xue+08]. 19

II.11 Solar radiation models are commonly used to render skies. From left to right, images generated using Preetham et al.'s model [PSS99], Bruneton and Neyret model [BN08] and Hosek and Wilkie model [HW12]. 20

(cc) BY-NC-ND

II.12 Solar radiation models have several applications, some of them are shown in this figure: from left to right, urban global illumination by Argudo et al. [AAP12], cloud rendering by Dobashi et al. [Dob+00], solar exposure on cities by Liang et al. [Lia+14] and interior daylight building illumination Walter et al. [Wal+05]. 21

II.13 CFD techniques are used to simulate flows in a liquid or gaseous state. From left to right, examples of simulations from Foster and Fedwick [FF01], Goswami et al. [Gos+10] and Li et al. [LWK03]. 23

II.14 Amsterdam ArenA stadium is an example of a wind simulation study in urban physics developed by Van Hooff and Blocken [HB10]. To the left, it is shown the 3D model while, to the right, the results of the simulation. 24

II.15 Example of weathering analysis and mapping on a tomb in Petra (Jordan) from the work of Fitzner et al. [FHB00]. To the left, it shows a picture of the tomb and, to the right, the corresponding analysis. 25

II.16 Computer Graphics is very useful in several Cultural Heritage applications. From left ro right, several examples are shown: relief extraction from Chen et al [Che+11] and Zatzarini et al. [ZTS09], enhancing carving from Lawonn et al. [Law+17] and improving visualization from Kolomenkin et al. [KST08]. 27

III.1 Overview of our system. A directional solar exposure map is precomputed over the specified time period and under the given conditions (bottom row). At run-time, we evaluate visibility in screen-space by combining a global and local view of the scene (top row). Surface solar exposure is finally computed by sampling the map within the sky regions visible from each point (right). 30

III.2 Sky exposure with a turbidity value of 2 and at a specific instant (June 23rd 12:00) for different latitudes. 34

III.3 Annual exposure maps (diffuse on top and direct at bottom with their respective scales) obtained using different configurations: latitude 0^oN turbidity 2.5 (left), latitude 30^oN turbidity 2.5 (middle) and latitude 30^oN turbidity 10 (right). Red arrows mark the solstice sun band. 36

III.4 The SVF estimates the portion of visible sky from a point occluded by the surrounding geometry (left). It evaluates the visible rate of the sky hemisphere (right). 36

III.5 Overview of ISHBAO technique. For each pixel the neighboring area is sampled in several directions. For each direction θ several horizon angles $h_i(\theta)$ and a tangent angle $t(\theta)$ are found. These angles are then used to estimate the ambient occlusion term. 38

III.6 View-dependent sky visibility. Left: Large-scale visibility is computed in screen space using a zenithal orthographic view, accounting for non-heightfield features. Right: Fine-scale visibility is evaluated from the current camera position. Bottom: Both results are combined. 39

III.7 Visibility obtained using the opening angle $h'(\theta)$ (right) and without using it (left), computed using 8 directions (N_{dir}) and 8 samples per direction (S_{dir}). 41

(cc) BY-NC-ND

III.8 Sky visibility comparison between ground truth (top) generated in several hours and our method (centre) rendered in less than 15ms, including difference images (bottom). Ground truth is based on ray casting using 128 samples per point. Our method uses 16 directions (N_{dir}) and 16 samples per direction (S_{dir}). Difference images show the difference of visibility percentage. The City Blocks model uses our large-scale visibility term (centre left), while the Mediterranean and Urban Sprawl models use our combined visibility (centre middle and right). 43

III.9 Computing surface solar exposure. The input exposure map is sampled within the visible regions obtained at each point. The normal map provides the surface orientation for the final weighting with the cosine term. 44

III.10 Sampling of the solar exposure map. We take samples along each visibility direction, adapting the number of samples to the visible region (red stars). We resort to mipmap pre-filtering to retrieve the exposure of each sample, whose size is fitted to the visible region (orange boxes). 45

III.11 Solar exposure computed in the *Urban Sprawl* (top) and *Mediterranean* (bottom) models using 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples for the exposure map (n_s). Latitudes on each row: 0^oN (up) and 30^oN (down) for *Urban Sprawl* and 0^oN (up) and 60^oN (down) for *Mediterranean*. Turbidities on each column: 2.5 (left) and 10 (center) for each model. Right column shows close-ups for the red boxes. 47

III.12 Visibility and solar exposure (30^oN latitude and a 2.5 turbidity) for the *Mediterranean* (left) and *Urban Sprawl* (right) models, with their respective close-up's below, using different parameters: top, 8 directions (N_{dir}), 8 samples per direction (S_{dir}) and a maximum of 8 samples in the exposure map (n_s); bottom, 16 directions(N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples in the exposure map (n_s). 49

III.13 Solar exposure comparison between ground truth (top) generated in 50 hours and our method (center) rendered in less than 30ms, including difference image (bottom). Ground truth is based on ray casting, which samples the sky each hour using 1 ray for direct irradiance and 128 for indirect irradiance per pixel. Our method uses the precomputed map and samples visibility using 16 directions (N_{dir}) and 16 samples per direction (S_{dir}). 50

III.14 Comparison between existing techniques and our method at 30^oN latitude and a 2.5 turbidity. Left: Heliodon image generated in 345 sec., sampling every 15 minutes with 1 meter resolution; Right: image generated with Autodesk Revit 2017 and Autodesk Insight 360 tool in 15 sec., sampling every hour with 0.45 meter resolution; Center: our solution generated in 0.07 sec. using per-pixel resolution (average in this view: 0.2 meters) using 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples in the exposure map (n_s). 51

III.15 Frames extracted from an animation (a block moves through the scene) demonstrating how our technique handles dynamic scenes with interactive solar exposure. 53

(cc) BY-NC-ND

IV.1 The air pollution process can be divided in three steps (bottom): deposition, chemical reaction and washing. Each step is influenced by external factors shown in the top (pollution sources, wind, sun and rain) and its combination shown in the middle (humidity and incoming pollution). 57

IV.2 The D2Q9 Lattice-Boltzman model cell. Each of the arrows corresponds to lattice direction vectors e_i and its length shows its magnitude. 59

IV.3 Velocity magnitude (blue represents at different time steps for a simulation of a map of the city of Girona. Wind entering from the bottom simulation boundary and leaving on top. Red color represents positions where velocity is maximum while blue shows the zones where it is zero. 60

IV.4 Given a map of the city with the pollution sources (traffic roads) marked in red (top-left), we compute a wind map using a 2D Lattice-Boltzman simulation (top-right) which we use to simulate the propagation of pollution along the city (bottom-left). Each bounce of the pollution particles is stored in the pollution map (bottom-right). 62

IV.5 Estimation of the different factors that affect (a) final render: (b) wind accessibility A_w, (c) sky accessibility A_s, (d) rain accessibility A_r, (e) insolation I, (f) pollution $(C_{map} * C_{height})$, (g) porosity p and (h) texture and shading. 64

IV.6 Rain accessibility sampling. Global scale sampling is performed in the zenithal view along a set of directions (top). Local sampling uses the same set of directions but defined on the rain cone and projected onto the current view (middle). Each local sample is projected back to the zenithal view to evaluate the occlusion on the same space (bottom). The color code of the samples (bottom) corresponds to its original direction in the zenithal view (top). Note that points falling in the upper part belong to samples taken in the building's roof in spite of the original direction in the zenithal. Although not shown, jittering can be optionally applied on both sampling directions and positions. 66

IV.7 Pollution process: (a) city clean of pollution, (b) city with the incoming pollution due to dry and wet deposition, (c) city with pollution deposited after the chemical reaction, and (d) city with pollution cleaned by rain and wind. 68

IV.8 (a) C_{height} function used to model the distribution of the deposition along the facade. (b) Example of dry deposition on a facade using this function (right) over the original distribution (left). 69

IV.9 Comparison of different pollution maps (b) and their corresponding dry pollution deposition (d) to an isotropic pollution deposition (c) generated after changing the closing environment. In (a), we show the initial configuration: buildings (black), pollution sources (red), wind direction (green arrows) and the camera position (yellow triangle) used in (c) and (d). 73

IV.10 Comparison of different pollution maps (b) and their corresponding dry pollution deposition (d) to an isotropic pollution deposition (c) generated after changing the alignment and separation of the buildings of a city. In (a), we show the initial configuration: buildings (black), pollution sources (red), wind direction (green arrows) and the camera position (yellow triangle) used in (c) and (d). 74

(cc) BY-NC-ND

IV.11 Pollution deposition comparison between different combinations of the steps of our pollution model with respect to an isotropic pollution deposition model. Left column shows the pollution layer and right column the shaded render. 76

IV.12 Pollution deposition comparison between an isotropic pollution deposition model (first row) and our model using only the particle deposition step (C_D) taking into account the dry deposition (second row), the wet deposition (third row) and both of them (fourth row). Left column shows the pollution layer and right column the shaded render. 77

IV.13 Pollution deposition comparison between an isotropic pollution deposition model (first row) and our model using only the chemical reactions (H) step taking into account the estimated humidity (second row), the resulting humidity after being dried by the sun (third row), the resulting humidity after being dried by the wind (fourth row) and the resulting humidity after being dried by the sun and the wind (fifth row). Left column shows the pollution layer and right column the shaded render. 78

IV.14 Pollution deposition comparison between an isotropic pollution deposition model (first row) and our model using only the washing step (W) taking into account the washing produced by the rain (second row), the washing produced by the wind (third row) and the washing produced by the rain and the wind (fourth row). Left column shows the pollution layer and right column the shaded render. 79

IV.15 Evolution of pollution on a urban scene along time using our method. (a) the initial clean buildings, (b) pollution effects start to be visible, (c) the pollution on the surfaces is quite noticeable and (d) the buildings are very polluted. 81

IV.16 Our technique (right) is able to reproduce the behavior of real life pollution weathering effects (left). Pollution distribution and modified material appearance are comparable in a very plausible way. 82

IV.17 Examples generated using our technique (bottom) and their corresponding clean references (top) from different points of view, which shows how our technique is suitable for different scales and how pollution integrates well with buildings details. 83

IV.18 Users can change interactively the importance of each external factor in the final render. Here we show how the final appearance changes by modifying the importance of each factor: insolation (left), wind (center) and rain (right). 83

IV.19 Examples of polluted areas over a city. Top rows: views of buildings without pollution (top), with pollution (middle) and pollution layer of our simulation (bottom). Bottom rows: closeups comparing clean (top) and polluted surfaces (middle) and the corresponding pollution layer (bottom). 84

IV.20 Example of deposition patterns produced by on-surface water flows. 86

(cc) BY-NC-ND

V.1 Overview of our technique: First, we capture a scene using photogrammetry and we generate the corresponding 3D model. Then, we smooth this model in a preprocess in order to reduce acquisition noise. After that, on the one hand, we perform a geometrical analysis to identify the features related to scaling weathering effect. On the other hand, we estimate the unweathered surface and we segment model regions according to their distance to it. Finally, we map the weathered zones on the model using the obtained segmentation and the detected weathering features. 91

V.2 Our method requires an isolated 3D manifold model to work properly. Generating it consists on taking pictures, loading them to Autodesk ReMake, which generates a 3D reconstructed scene and, then, separating the target model and filling its holes. 93

V.3 We perform a smoothing filter to reduce the acquisition noise of the reconstructed model. Here it shows a comparison before and after applying the filter (top) and zoom-in's showing the obtained color-coded curvature in each case (bottom). . 94

V.4 Segmentation of the original mesh. After the click of the user (left), a segmentation process is applied in function of the faces normal (center) until a final segmentation is obtained (right). Note that center image shows a schematic side view of the object and that red mean a selected face. 96

V.5 The iterative refinement of the surface estimation consists of erasing the outliers and estimating original surface again using RANSAC until no outliers are detected. 97

V.6 In order to detect the loss of material in the model, we first need to estimate the corresponding unweathered surface. This is done (a) by performing an initial segmentation starting from an unweathered point marked by the user. Then, we perform a RANSAC process to obtain the unweathered surface estimation (b). This process is iteratively refined removing outliers from segmentation and computing RANSAC again, until no outliers are segmented (c) and the definitive unweathered surface is found (d). We use red color to indicate the segmented faces and green color to show the estimated surface. 99

V.7 Scaling weathering effects are related to the geometrical features of the model. In order to detect them, we use the mean curvature (left) to identify the vertex where it is minimum and maximum (center) and we process it taking into account the highest and lowest values and filtering the small features (right). 100

V.8 We detect the scaling features looking for opposite features in the near vertices after projecting them to the original surface (center). Then, we use the difference of distances to the original surface (center) as an indicator of scaling subtype of the surviving vertices (right). 101

V.9 In order to detect regions affected by the scaling weathering effect, we first need to identify the different regions of the model. Our region segmentation uses a map of the distance between the model and the estimated unweathered surface (left) and computes the region segmentation in image-space (right). 102

(cc) BY-NC-ND

V.10 The segmented regions (top-left) need to be culled in order to obtain a map of weathered zones (right). We perform it by discarding the regions in which mean distance to the original surface, previoulsy computed in the depth map (bottom-left) is lower than $dist_{min}$. 103

V.11 From the surviving regions of the previous culling process (left), we need to tag them according to the sub-type of scaling effect present on them (right) taking into account the scaling weathering features. 105

V.12 Model 1. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment. 106

V.13 Model 2. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment. 107

V.14 Model 3. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment. 107

V.15 Model 4. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment. 108

V.16 Model 5. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment. 108

(cc) BY-NC-ND

(cc) BY-NC-ND

List of Tables

III.1 Performance of our technique for different models. The *Samples* column specifies the number of visibility directions [N_{dir}] $\times$ (samples per direction [S_{dir}] $+$ maximum solar exposure samples [n_s]). Note that for a square solar exposure texture, as used in the examples, the latter equals to N_{dir} in the worst case (full visibility). Timings in fps were obtained with $60Hz$ vsync enabled on an 800×600 viewport. The range of values means performance obtained for far and close views, respectively. 48

IV.1 Frames per second (fps) for rendering a $800x600$ screen (left) and iterations per second (ips) during the wind and pollution map generation using 100K particles and a 800x580 map representing the city (right). 80

V.1 Performance of our technique for different models. For each model, we show the number of pictures used in the reconstruction, the number of triangles of the reconstructed 3D model and the performance in seconds of each step of our technique: smoothing the model (Section V.3.2), estimating the unweathered surface (Section V.4), detecting the scaling weathering features (Sections V.5.1 and V.5.2) and mapping the weathered regions (Sections V.5.3 and V.5.4). . . . 106

(cc) BY-NC-ND

(cc) BY-NC-ND

Abstract

Being able to capture and synthesize realistic materials is an important research topic in Computer Graphics. In order to render realistic images it is not enough to represent materials reliably, but we also need to represent its temporal components. Simulation of weathering effects is the research area that pursues the incorporation of decay to virtual models. After studying the previous work, we have found that there is a lack of estimation methods for some important environmental parameters and a need of global models that consider the interactions of the environmental factors in large scenes. Moreover, we have observed that the improvement of current techniques need to incorporate real effects characteristics to avoid the current disconnection from the real processes. But, this needs to be first analyzed and understood. In this thesis, we aim to improve weathering simulation in Computer Graphics in these three fronts.

Concerning weathering factors estimation, we first propose to deal with an undervalued factor in weathering literature which is very important in aging processes: the sun. More specifically, we propose an interactive technique that estimates solar exposure on detailed urban scenes. Given a directional exposure map computed over a given time period, we estimate the sky visibility factor that serves to evaluate the final exposure at each visible point. This is done using a screen-space method based on a two-scale approach, which is geometry independent and has low storage costs. Our method results in an interesting technique not only in weathering simulations, but also in architecture, in sustainable building design and in the estimation of buildings' energy performance.

After that, we introduce a global model to simulate the changes of appearance of buildings due to pollution deposition. This process is very noticeable on urban scenes and, so far, no technique has dealt with it using a global approach. We propose a physically-based technique that considers pollution effects to depend on three main factors: wind, rain and sun exposure, and that takes into account three intervening steps: deposition, reaction and washing. With a low-cost pre-computation step, we evaluate the pollution distribution through the city. Based on this distribution and the use of screen-space operators, our method results in an efficient approach that generate realistic images of polluted surfaces at interactive rates. In addition, the pre-computation demands a reduced amount of memory to store the pollution map and, as it is independent from scene complexity, it can suit large and complex models adapting its resolution.

Finally, in order to improve the understanding of weathering processes, we present a method to identify and map "scaling" weathering effects on stone monuments with very little user interaction. Our method first captures and pre-processes real 3D models of weathered stone monuments using standard photogrammetry techniques. From these models, we then estimate the original shape of the surface using the RANSAC algorithm. With this information, we perform a geometrical analysis to detect the features affected by this weathering effect and compute their characteristics. Then, we map the regions that have suffered scaling using the detected features and a segmentation based on the distance between the mesh of the decayed object and the unweathered surface. Our technique results can be very useful to understand the level of weathering of a monument and to trace the weathered parts through time automatically.

(cc) BY-NC-ND

Keywords: Computer graphics, realistic rendering, weathering effects, natural phenomena

(cc) BY-NC-ND

Résumé

Etre capable de capturer et de synthétiser des comportements réalistes est un sujet de recherche important en Informatique Graphique. Mais, pour produire des images réalistes, il ne suffit pas de représenter les matériaux de manière fiable, il faut également prendre en compte leurs évolutions dans le temps. La simulation des effets de vieillissement est le domaine de la recherche qui s'intéresse à la prise en compte de la détérioration des modèles virtuels. Après avoir étudié les travaux précédents dans ce domaine, nous avons constaté qu'il y a un manque d'estimation pour certains paramètres environnementaux importants ainsi qu'un besoin de modèles globaux qui tiennent compte des interactions des facteurs intervenant dans de grandes scènes. De plus, nous avons observé que l'amélioration des techniques actuelles nécessite d'intégrer les caractéristiques physiques réelles pour éviter la déconnection des processus réels, ce qui doit donc être analysé et compris. Dans cette thèse, nous visons à améliorer la simulation du vieillissement en Informatique Graphique sous les trois angles suivants.

Concernant l'estimation des facteurs de vieillissement, nous proposons d'abord de traiter un facteur sous-estimé dans la littérature mais qui est très important dans les processus de vieillissement : le soleil. Plus spécifiquement, nous proposons une technique interactive qui estime l'exposition solaire sur des scènes urbaines détaillées. Compte tenu d'une carte d'exposition directionnelle calculée sur une période donnée, nous estimons le facteur de visibilité du ciel qui sert à évaluer l'exposition finale à chaque point visible. Ceci est fait en utilisant une méthode espace-écran basée sur une approche à deux échelles, qui est indépendante de la géométrie et qui a un coût bas en stockage. Notre méthode aboutit à une technique intéressante non seulement pour les simulations du vieillissement, mais également en architecture, dans la conception de bâtiments durables et dans l'estimation de la performance énergétique des bâtiments.

Ensuite, nous introduisons un modèle global permettant de simuler les changements d'apparence des bâtiments liés aux dépôts de pollution. Ce processus est très visible dans des scènes urbaines et, jusqu'à présent, aucune technique n'a été proposée utilisant une approche globale (sur une ville entière). Nous proposons ici une technique basée sur la physique considérant que les effets de la pollution dépendent de trois facteurs principaux : l'exposition au vent, à la pluie et au soleil. Cette approche prend en compte trois étapes : le dépôt, la réaction et le lavage. Avec un pré-calcul à bas coût, nous évaluons la distribution de la pollution dans la ville. Puis, par le biais d'opérateurs espace-écran, nous proposons une méthode efficace capable de générer des images réalistes en temps interactif. De plus, le pré-calcul n'exige qu'une quantité réduite de mémoire pour stocker la carte de pollution et, comme la méthode utilisée est indépendante de la complexité de la scène, elle peut s'adapter à des modèles complexes et de grande taille en ajustant la résolution de la carte.

Finalement, afin de mieux appréhender les processus de vieillissement, nous présentons une méthode permettant de prendre en compte l'effet d'écaillage sur les monuments en pierre avec très peu d'interaction de l'utilisateur. Notre méthode capture et prétraite des modèles 3D réels de monuments en pierre altérés par le passage du temps en utilisant des techniques photogramétriques standard. Nous estimons leurs formes originales en utilisant l'algorithme

(cc) BY-NC-ND

RANSAC. A partir de ces informations, nous effectuons une analyse géométrique permettant de détecter les zones affectées. Ensuite, nous projetons les régions qui ont subi cet effet d'écaillage en utilisant les caractéristiques détectées et une segmentation basée sur la distance entre le maillage de l'objet vieilli et la surface non altérée. En plus de la contribution consistant à produire des images de synthèse précises d'objets virtuels vieillis, les résultats de notre technique peuvent être très utiles pour comprendre le niveau de l'altération d'un monument et de faire un suivi des parties altérées à travers le temps automatiquement.

Mots-clés: informatique graphique, rendu réaliste, effets de vieillissement, phénomènes naturels

(cc) BY-NC-ND

Resum

Ser capaç de capturar i sintetitzar material de forma realista és un tema important de recerca en Informàtica Gràfica. Però, per tal de generar imatges realistes no n'hi ha prou amb representar materials de forma fiable, sinó que també és convenient representar el seu component temporal. La simulació d'efectes d'envelliment és el camp de recerca que persegueix la incorporació del deteriorament dels materials en models virtuals. Després d'estudiar els treballs previs, hem trobat que hi ha un buit en l'estimació d'alguns paràmetres mediambientals importants en l'envelliment i una necessitat de models globals que considerin les interaccions dels factors que intervenen en grans escenes. D'altra banda, hem observat que les tècniques actuals es basen principalment en aproximacions en certa manera desconnectades dels processos reals. La seva millora requereix incorporar característiques dels processos reals d'envelliment que cal analitzar i comprendre primer. En aquesta tesi, ens proposem millorar la simulació d'envelliment de materials en Informàtica Gràfica en aquests tres fronts.

Pel què fa a l'estimació dels factors de deteriorament, primer proposem ocupar-nos d'un factor infravalorat en la literatura d'aquest camp i que és molt important en els processos d'envelliment: el sol. Més específicament, proposem una tècnica interactiva que estima l'exposició solar en escenes urbanes detallades. Donat un mapa d'exposició direccional calculat al llarg d'un període de temps determinat, estimem el factor de visibilitat del cel que serveix per avaluar l'exposició final en cada punt visible. Això es fa mitjançant un mètode que treballa en l'espai de pantalla basat en un enfocament a dos nivells que és independent de la geometria i que té un baix cost d'emmagatzematge. El nostre mètode resulta en una tècnica interessant no només en simulacions d'envelliment, sinó també en arquitectura, en disseny de construcció sostenible i en l'estimació del rendiment energètic d'edificis.

A continuació, introduïm un model global per simular els canvis d'aspecte dels edificis a causa de la deposició de la contaminació. Aquest procés és molt notable en les escenes urbanes reals i, fins ara, cap mètode conegut l'ha tractat amb un enfocament global. Proposem una tècnica basada en la física que considera que els efectes de la contaminació depenen de tres factors principals: el vent, la pluja i l'exposició al sol, i que té en compte els tres processos que hi intervenen: deposició, reacció i netejat. Amb una precomputació de baix cost, avaluem la distribució de la contaminació a través de la ciutat. Basat en això i l'ús d'operadors de l'espai-pantalla, el nostre mètode resulta en un enfocament eficient capaç per generar imatges realistes amb ràtios interactius. A més, el pre-procés requereix un espai reduït de memòria per emmagatzemar el mapa de contaminació resultant i, atès que és independent de la complexitat de l'escena, pot adaptar-se a models grans i complexos ajustant la resolució del mapa.

Finalment, per millorar la comprensió dels processos d'envelliment, presentem un mètode per identificar i projectar l'efecte d'"escata" en monuments de pedra amb una interacció molt reduïda de l'usuari. El nostre mètode primer captura i pre-processa models reals 3D de monuments de pedra deteriorats pel pas del temps usant tècniques fotogramètriques estàndard. Després d'això, fa una estimació de la forma original de la superfície utilitzant l'algoritme RANSAC. Amb aquesta informació, realitzem una anàlisi geomètrica per detectar les característiques afectades per l'efecte envelliment i les calculem. A continuació, projectem les regions que han patit l'efecte

(cc) BY-NC-ND

del temps usant les característiques detectades i una segmentació basada en la distància entre la malla de la superfície deteriorada i la malla de la superfície original abans de l'envelliment. A més de la contribució per produir imatges de síntesi precises d'objectes virtuals envellits, els resultats d'aquesta tècnica poden ser molt útils per entendre el nivell d'envelliment d'un monument i observar de manera automàtica la seva degradació al llarg del temps.

Paraules clau: informàtica gràfica, representació realista, efectes d'envelliments (weathering), fenòmens naturals.

(cc) BY-NC-ND

Chapter I. Introduction

I.1 Motivation

One of the ultimate goals of Computer Graphics is to be able to render virtual worlds making them indistinguishable from the real one. To achieve that, Computer Graphics community has focused its efforts on techniques permitting to model complex objects and the interactions between light and matter. Nowadays, thanks to the availability of acquisition and design tools, we are able to capture or model objects with complex shapes and high quality materials. Moreover, there are several techniques to handle not only reflection and transmittance, but even complex light/matter interactions like sub-surface scattering and participating media.

Despite of this, when striving for realism, high quality modeling is not enough because real objects also have a temporal dimension. Over time, aging and weathering effects significantly alter their appearances, from shape and color to light/matter specific responses. These changes are caused either by natural processes or due to human factors. Figure I.1 shows some examples of results of these weathering effects in real-life. It is very important to notice that in order to synthesize them in a virtual world, these processes affect every single task of the traditional image synthesis pipeline, from geometry to rendering. Therefore, handling these temporal changes in appearance is a key factor to achieve realistic rendering of virtual models.

Nowadays, the main strategy in the video-game or animation movie industries for realistic scene modeling is to incorporate weathering effects by relying on the manual work of a team of artists,



Figure I.1: *Examples of weathering effects in urban environments.*

(cc) BY-NC-ND

who adjust and modify textures and/or geometry of virtual models. Thanks to it, several types of high quality imperfections are modeled, but following only aesthetic criteria is often disconnected from real life effects. In current state-of-the-art approaches, the developed techniques are based on texture-transfer of these effects from existing images to virtual objects, or on the simulation of the physical underlying phenomenon, which also consumes large amounts of resources.

The simulation of weathering effects has mainly been addressed by an effect-centered approach, i.e., for each considered weathering effect, we try to figure out where it would appear and simulate it locally. The main problem of this approach is that, in order to achieve a multi-effect final image, we must perform different weathering algorithms on the same geometry several times. Moreover, most of the current weathering techniques are not suitable for large models, such as cities. They require high computational costs and important levels of memory consumption. Finally, another drawback of current techniques is that they are not fully connected with real-life processes. They mainly rely on the final aspect without taking into account the inner structures of the objects. This is due to a lack of a profound study of real-life processes and a dearth of analysis of real weathered objects, which often result in strong simplifications of the phenomena and the resulting synthesized images.

In this thesis, we face some of the stated drawbacks of current weathering techniques. First, we propose a change in the common schema to consider the weathering effects as the result of global processes influenced by interrelated factors. After that, we propose new, fast and reliable techniques to estimate those weathering factors. More specifically, we propose a new method to estimate solar exposure and a fast implementation to simulate wind. Then, we propose a new model to compute changes in appearance of buildings caused by pollution deposition following the previously commented schema. Finally, we propose a new method to identify semi-automatically weathering effects on real objects as a first attempt to identify weathering characteristics that will open a new research line to improve weathering simulations.

In conclusion, the state of the art of weathering techniques in Computer Graphics can still be extended in different directions. Our research, throughout the development of this thesis, has focused on going a step further by improving the analysis and synthesis of weathering effects with fast and low-cost solutions.

I.2 Contributions

The work presented in this thesis extends the state of the art in rendering weathering effects by the following contributions:

- **New weathering challenges.** After analyzing the challenges of current weathering techniques, we have identified new challenges that relate to large scenes such as urban scenes. Instead of focusing on simulating the final effects independently, we propose to consider the city as a whole and to study the relations between weathering effects and the internal and environmental factors that cause them. We propose to first determine and detect those weathering factors and, then, we introduce a model that describes the relations of these factors with the specific weathering effect. The techniques presented in this thesis follow this change of philosophy. This analysis and the new schema have been published in:

(cc) BY-NC-ND

Weathering of Urban Scenes: Challenges and Possible Solutions.
Muñoz-Pandiella, I.; Bosch, C.; Mérillou, N.; Pueyo, X.; Mérillou, S. and Patow, G.
Eurographics Workshop on Material Appearance Modeling '16, Dublin, Ireland.
June 2016. ISSN 2309-5059.
DOI 10.2312/mam.20161248
URL https://diglib.eg.org/handle/10.2312/mam20161248
Publisher The Eurographics Association

- **Interactive solar exposure estimation.** Solar exposure is a key factor in several weathering effects. Despite of this, it has been commonly ignored in weathering simulations. This lack of consideration is related to the computational cost of typical solutions to estimate it. We present a technique to estimate it in real-time handling complex scenes and without requiring high memory consumption. This technique is a clear improvement over previous ones to detect weathering factors and it is even suitable for urban planning and engineering areas. The results presented in this work have been published in:

  Real-Time Solar Exposure Simulation in Complex Cities.
  Muñoz-Pandiella, I.; Bosch, C.; Mérillou, N.; Pueyo, X. and Mérillou, S.
  Computer Graphics Forum, Volume -, Issue -, pages -.
  2017. ISSN 1467-8659.
  DOI 10.1111/cgf.13152
  URL http://dx.doi.org/10.1111/cgf.13152

  A preliminar version of this work was presented at:

  Interactive Solar Irradiance Estimation in Urban Scenes.
  Muñoz-Pandiella, I.; Bosch, C.; Mérillou, N.; Mérillou, S. and Pueyo, X.
  XXV Congreso Español de Informática Gráfica 2015
  July 2nd 2015
  Benicàssim, Spain.

  This work was also a *Computer Graphics Forum* invited paper at:

  Real-time Solar Exposure Simulation in Complex Cities.
  Muñoz-Pandiella, I.; Bosch, C.; Mérillou, N.; Pueyo, X. and Mérillou, S.
  XXVIII Eurographics Symposium on Rendering 2017
  June 19th 2017
  Helsinki, Finland.

- **Interactive pollution simulation.** We present a technique that models changes in appearance of cities due to pollution. In this method, we are able to estimate pollution propagation and deposition and synthesize realistic renderings at interactive rates. We take into account the pollution sources, wind simulations, rain and solar exposure estimations. Although other methods have addressed this problem before, our technique is the first that reproduces global changes of appearance in cities, thanks to our schema that considers the city as a whole. In addition, it avoids memory consumption pre-computations that is suitable for large scale and complex scenarios. This work has been submitted and is currently under review:

(cc) BY-NC-ND

Urban Weathering: Interactive Rendering of Polluted Cities.
Muñoz-Pandiella, I.; Bosch, C.; Mérillou, N.; Pueyo, X.; Mérillou, S. and Patow, G.
Minor Revisions at IEEE Transactions on Visualization and Computer Graphics.

- **Fast and simple scaling detection.** In order to connect better the synthesized weathering effects and the real ones, for instance to transfer some of their properties from examples, it is important to provide automatic tools to detect and analyze those effects. In the last part of the thesis, we present a method to detect the scaling effects on stones. We propose to digitalize real models and to analyze their geometry to detect the inner structures that determine the presence of this common weathering effect. Moreover, we present a model to classify and tag the affected zones. This technique is the first effort to connect weathering effects with the inner structures of the weathered objects.This work was developed with the Yale Graphics Group during an internship at Yale University under the supervision of Professor Holly Rushmeier and it was presented in:

Towards Semi-Automatic Scaling Detection on Flat Stones.
Muñoz-Pandiella, I.; Akoglu K.; Bosch, C. and Rushmeier, H.
Eurographics Workshop on Graphics and Cultural Heritage '17, Graz, Austria. September 2017.
ISSN 2312-6124.
DOI 10.2312/gch.20171291
URL https://diglib.eg.org/handle/10.2312/gch20171291
Publisher The Eurographics Association

I.2.1 Secondary Contributions

Besides the main contributions, during the development of this thesis other smaller contributions have been done:

- **Doctoral Consortium Eurographics 2016.** This thesis participated in the Doctoral Consortium carried out during the Eurographics 2016 conference. There, we presented the work in progress and ideas to a group of experts and a discussion of them was conducted. Moreover, a Poster was presented in the conference poster session:

Real-Time Weathering of Urban Materials Muñoz-Pandiella, I.
Doctoral Consortium at Eurographics 2016, Lisbon, Portugal.
May 2016.

- **Interactive urban wind estimation.** Wind is an important factor related to several weathering effects. However, due to the expensive cost of its simulation, it has commonly been simplified or not taken into account. At the beginning of the second part of the thesis, we present a technique that improves the existing wind simulations by means of a model simplification and a GPU implementation. This technique was used in the contribution "Interactive pollution simulation" as it is explained in Section IV.3.1. Moreover, the model adaptation and results were presented in:

(cc) BY-NC-ND

Interactive Low-Cost Wind Simulation For Cities.
Rando, E.; Muñoz-Pandiella, I.; and Patow, G.
Eurographics Workshop on Urban Data Modeling and Visualisation '16, Liège, Belgium.
December 2016. ISSN 2307-8251.
DOI 10.2312/udmv.20161421
URL https://diglib.eg.org/handle/10.2312/udmv20161421
Publisher The Eurographics Association

I.3 Manuscript Outline

The content of this manuscript is organized as follows:

- **Chapter II: Previous Work.** In the second chapter we overview the state of the art in weathering techniques. Then, we introduce techniques that estimate environmental factors that affect weathering processes, like solar radiation and wind. Finally, we overview the existing approaches to capture real-life weathered objects and to analyze and synthesize these weathering effects.

- **Chapter III: Real-time Solar Exposure Estimation.** In this chapter we improve the detection of weathering factors introducing a technique to estimate solar exposure in real time. Our screen-space solution is inspired by Ambient Occlusion techniques and it is able to handle complex scenes requiring a simple pre-computation.

- **Chapter IV: Screen-space Pollution Weathering.** In the fourth chapter, we enrich the existing weathering techniques by presenting a method to render pollution deposition on cities. Our technique uses as input factors wind, rain, sun and pollution sources. Then, a pollution model is simulated in screen-space and, finally, the appearance of the urban elements is modified accordingly. This pollution simulation shows the global model referred to in the contribution "New Weathering Challenges", thus you will find a detailed description of this contribution at Chapter IV.

- **Chapter V: Stone Weathering.** In this chapter we present a first step towards the automatic detection of weathering effects in stones. We capture models using photogrammetry techniques and we perform a geometric analysis to detect and classify the scaling effect on stone monuments and buildings.

- **Chapter VI: Conclusions and Future Work.** We conclude the thesis by summarizing the main findings we have obtained along this thesis. We also highlight all the contributions of this thesis into the field of weathering effects in Computer Graphics and, finally, we talk about future work that could be addressed to extend the methods presented in this thesis.

(cc) BY-NC-ND

(cc) BY-NC-ND

Chapter II. Previous Work

In this chapter, we present the state of the art about the simulation of weathering effects in Computer Graphics. First of all, we summarize the different existing techniques to simulate and render these effects. We also classify them depending on the types of effects they handle and the type of approach they develop. Later, we do a review of some existing techniques, related to Computer Graphics, that handle the estimation of key factors in several weathering effects. More specifically, we review solar radiation estimation techniques which is a key contribution of this thesis. Moreover, we also propose an introduction to some important techniques in wind simulation. Finally, we introduce the most common techniques to capture and analyze real objects in Computer Graphics and Cultural Heritage with the purpose of detecting and measuring weathering effects.

II.1 Weathering in CG

When we talk about weathering effects, we are referring to the changes in the appearance of the objects as a result of their interaction with the surrounding environment. These effects appear after a process of change and its rate and extent depend on the shape of the object, its exposure to the surrounding environment (including here its use by humans) and the properties of the material which it is composed of. In order to increase the realism of virtual objects, the Computer Graphics community has dedicated much effort to simulate and reproduce these effects. In 2008, Merillou and Ghazanfarpour [MG08] presented a survey that gathers and classifies most important weathering techniques. In the same year, Dorsey et al. [DRS08] published the excellent book *Digital Modeling of Material Appereance*, which is a reference in material modeling and dedicated a chapter to define and classify weathering effects and to review most important existing techniques. More recently, Frerichs et al. [FVG15] presented a survey of object deformation and decomposition that reviews the state of the art in several effect-centered weathering techniques. Following Dorsey et al. classification, we can divide the existing techniques in two groups: on one hand there are techniques that focus on the study of one specific weathering effect while, on the other, we find generic methods that handle a set of effects. The following subsections are dedicated to detail each group.

(cc) BY-NC-ND

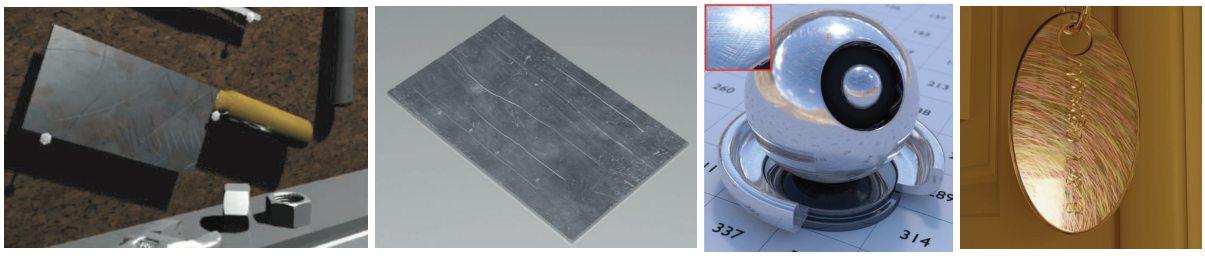


Figure II.1: *Results of different scratch techniques. From left to right, images from Merillou et al. [MDG01a], Bosch et al. [Bos+04], Raymond et al. [RGB16] and Werner et al. [Wer+17].*

II.1.1 Centered on Effects

When focusing on a specific weathering effect, we can observe how its final appearance is strongly associated with the mechanisms that have originated it. We can classify these effects in three wide categories of mechanisms: mechanical, chemical and biological. Mechanical processes entail changes in shape, matter loss or material deposition, chemical reactions involve modifying the nature of the surface and biological effects result from the action of life forms.

Mechanical effects

Mechanical mechanisms of weathering commonly result in shape deformations, in the disintegration of the original material into fragments or in the deposition of other materials on its surface. These two variants are also combined in some cases.

Dust accumulation is a very common change in the appearance of objects, and affects their reflection properties. It is caused by the accumulation of microscopic particles of material that are transported by the air. Blinn [Bli82] first addressed it taking only into account the interaction between light and dust particles. He focused on rendering clouds of particles using layers and a model to simulate the single scattering of light inside the layers. But, this method can be also used to render dust on surfaces. Later, Hsu and Wong [HW95] proposed to use dust sources as an analogy of light sources. They determined that the accumulation of dust was related to the cosine of the angle between the dust source and the surface normal and they proposed to use accessibility in order to take into account external factors such as wind. During rendering, they introduce a Perlin noise perturbation in the distribution and they linearly interpolate the dust value between a fully covered reference and the original surface. More recently, Sun et al. [Sun+07a] developed an acquisition system to measure time-varying BRDFs. They use the captured data to fit analytic BRDF functions being able to model BRDF changes by dust accumulation. Lately, Guo and Pan [GP14] proposed a framework to simulate and render objects covered by dust layers that vary spatially. In their model, they simulate dust accumulation taking into account material stickiness and surface exposure and inclination while, in the rendering step, they use the Kubelka-Munk reflectance model [KM31].

Scratches and impacts are common deformations of the surface of real objects. While scratches are made by tools (such as etching or brushing) or everyday use, impacts deformation results from the repetitive collision of other objects during long periods of time. The first attempts to specifically deal with scratches were made by Mérillou et al. [MDG01b] and Bosch et al. [Bos+04]. They distinguished between microscratches modeled with anisotropic BRDF models

(cc) BY-NC-ND



Figure II.2: *Results of different cracks techniques. From left to right, images from Desbenoit et al. [DGA05], Iben and O'brien [IO06], Glondu et al. [Glo+12] and Pfaff et al. [Pfa+14] works.*

and isolated scratches described by a texture coupled with a specific BRDF model. To generate the profiles associated to these scratches, while Mérillou et al. is based on real measurements, Bosch et al. use a physical approach to derive them from scratching tools, penetration forces and material properties. Lately, several efforts have been done in order to improve the rendering of microscratches. Dong et al. [Don+15] predicted the appearance of metal surfaces by deriving microfacet distributions from measured surface topography. Later, Yan et al. [Yan+16] proposed a more efficient method that considers specular surfaces as a four-dimensional position-normal distribution and approximates them using millions of 4D Gaussians. Raymond et al. [RGB16] proposed a method that is able to take into account the inter-reflections inside a scratch taking advantage of the regular structure of microscratches by using a multi-scale texture that stores 2D scratches computed using ray-tracing. Finally, Werner et al. [Wer+17] presented a method to compute the iridescence of the scratches by representing surface roughness as a set of line segments and using non-paraxial scalar diffraction theory to render it. In impacts simulation, we find the work of Paquette et al. [PPD01] in which they propose to use a tool to define impact zones where the impacts are automatically computed by refining adaptively the surface and displacing intersected vertices. More recently, Schafer et al. [SKS13] proposed a method that allows to simulate sculpting operations or deformations resulting from collisions in real time. The modifications are performed on GPU thanks to adaptive mesh refinements and local displacement operations. In Figure II.1, several examples of these techniques are shown.

Cracks and fractures have been widely studied by the Computer Graphics community. They represent the separation of a material following specific paths and, sometimes, into multiple pieces due to the supported stress. For a complete analysis of existing techniques we point the reader the survey by Muguercia et al. [MBP14], we would like to introduce some of the most important work. First, Hirota et al. [HTK98] proposed a model to simulate crack and fracture patterns in mud. Their model consists in a mass-spring system that simulates the contraction and expansion of the drying process and cracks are produced when the springs extend more than a fixed distance. Gobron and Chiva [GC01a] proposed a solution to 3D surfaces using a multi-layer system that takes into account curvature and thickness of layers. Later, Iben and O'Brien [IO06] simulated the evolution of the cracks over time through stress fields and a triangle discretization of the surface. Desbenoit et al. [DGA05] proposed to extract crack patterns from photographies, mapping them automatically on the surface of the model and generating fractures carving out a procedural swept volume. Valette et al. [VPL06] simulated cracks in a volume of cubic cells that represent matter and pores, that are transferred to any surface thanks to a specific parametrization. More recently, Wicke et al. [Wic+10] proposed to use a finite element simulation to reproduce fractures and to keep high tetrahedron quality using a conservative local re-meshing algorithm. Glondu et al. [Glo+12] generated fractures using an

(cc) BY-NC-ND



Figure II.3: *Different terrains generated taking into account erosion processes. From left to right, images from Crespin et al. [Cre+14] and Cordonier et al. [Cor+16] [Cor+17].*

example-based method that matches the statistics of fracture patterns in a photography. Finally, Pfaff et al. [Pfa+14] proposed an adaptive method to propagate fractures in thin sheets that refines the mesh dynamically and guides the simulation. In Figure II.2, several examples of these techniques are shown.

Peeling is a weathering effect that affects thin layers attached on surfaces (e.g., paint). After long periods of exposure to several factors, the attached layer loses its adhesion, peels and falls off. Gobron and Chiva [GC01b] simulated it using an extension of their cracks work [GC01a]. They propose to use a precomputed crack distribution and a hybrid cellular automata that is able to compute full and partial peeled zones. However, they do not take into account the physical characteristics of materials and stresses. Later, Paquette et al. [PPD02] introduced a technique that addressed the previous lack of physical properties. They proposed to use a 2D grid to represent material properties and a set of crack segments, and simulated the peeling process propagating cracks onto the surface.

Erosion is the process of degradation of surfaces caused by external factors, that results in a loss of material of this surface. Although sometimes this effect is involved with chemical and biological processes, the main components use to be mechanical. First, Musgrave et al. [MKM89] proposed a method to generate terrain models taking erosion into account. They proposed to represent the surface with a height map and apply thermic stress and hydraulic erosion to it. In a similar line, Nagashima [Nag98] proposed a physically-based iterative method that takes into account water flows and rainfalls. Later, Ito et al. [Ito+03] presented a fully 3D erosion model of jointed rocks represented by voxels. They applied an iterative process that removes voxels and computes the falling process of the remaining ones. Benes and Arriaga [BA05] studied the generation of table mountains. Their technique represent the terrain as a height field and simulate erosion transforming rocks into sand. Liu et al. [Liu+05] presented a method that simulates erosion in real time. First, they compute flows using a 2D implementation of Navier-Stokes equations and, then, they handle flow-surface interactions with an erosion-deposition model based on simple rules. Benes et al. [Ben+06] also simulated the hydraulic erosion with a 3D model based on Navier-Stokes equations. They also compute the reallocation of the eroded material following a transport equation. Stava et al. [Sta+08] proposed an interactive method to model terrains using hydraulic erosion. They proposed a divided terrain and computed erosion at each portion independently in GPU using physics-based algorithms. More recently, Crespin et al. [Cre+14] presented an erosion simulation that relies on a generalized map representation of the geological layers and a model based on user vertex displacements and a collision detection system. Cordonnier et al. [Cor+16] proposed a method to generate terrains

(cc) BY-NC-ND



Figure II.4: *Results of different matter deposition techniques. From left to right, images from Dorsey et al. [DPH96], Gu et al. [Gu+07] and Bosch et al. [Bos+11].*

using as input an uplift map. Then, an erosion simulation is computed on a planar graph that is converted to an elevation model representing the terrain. Finally, Cordonnier et al. [Cor+17] presented a framework that computes the temporal evolution of a terrain taking into account the interrelations between vegetation and erosion. They use a spatio-temporal simulation in which geomorphological and ecological events modify a layered model that represents terrain materials and vegetations. In Figure II.3, several examples of these techniques are shown.

Matter depositions correspond to the attachment of dirtiness on a material surface. This matter uses to reach the surface mixed in a flow that later evaporates, leaving behind dirty stains. Dorsey et al. [DPH96] were the first to deal with this phenomenon. In order to model the water flow, they developed a particle model where each particle represents a drop of water. The simulation is controlled by forces, material properties and external factors, and a set of equations rules the interaction between drops and surface. Later, in order to improve the realism of the rendering of deposited material, Gu et al. [Gu+07] studied the interactions between light and stains on transparent surfaces. They proposed an analytical model that follows radiative transport theory and takes into account the measured spatially-varying thicknesses and scattering parameters of glass stained with different materials. More recently, Bosch et al. [Bos+11] proposed a method to transfer flow stains from a photography to a synthetic model. They proposed to extract parameters and detail maps using an inverse simulation approach. In Figure II.4, several images are shown generated using these techniques.

Chemical effects

The main characteristic of weathering effects caused by chemical processes is that the original material is modified, becoming a new material with different composition and physical characteristics. Commonly, the resulting material properties are altered and become more sensitive to weathering effects. Figure II.5 shows several chemical weathering effects generated using techniques introduced below.

Tarnishing is a type of corrosion that appears on some metals, showing a thin layer that covers the surface, following a very slow rate of development. Miller [Mil94] proposed a method that is able to represent it. They proposed to modulate light reflection taking into account accessibility computed using tangent spheres.

Patina is a weathering effect that occurs mostly in metals, although it also can affect wood

(cc) BY-NC-ND



Figure II.5: *Examples of chemical weathering effect simulations. From left to right, destructive corrosion by Mérillou et al. [MDG01a], patina by Dorsey and Hanrahan [DH96], fading by Kimmel et al. [Kim+13] and pollution deposition by Mérillou et al. [Mer+10].*

and stone. After a chemical reaction, a film is created covering the surface, that protects the underlying material (avoiding its weathering). It has been introduced by Dorsey and Hanrahan [DH96]. They represent the surface as a set of layers and proposed to use procedural operators to simulate patina taking into account object geometry and external factors. The visualization consists in modifying an adapted BRDF from Kubelka-Munk model [KM31]. Later, Chang and Shih [CS00] proposed to simulate patina using a set of L-system based rules guided by an environmental tendency factor and object geometry.

Destructive corrosion is a chemical attack that affects materials by atmospheric agents. The process forms a complex system of layers that increases the porosity of the material and involves a loss of material. Usually, it finishes when the reactant is exhausted and the material destroyed. First, Mérillou et al. [MDG01a] proposed a model in which this simulation is inspired by physics. They store the simulation in a spreading map that they use during the rendering step to modify the BRDF of the materials. Subsequently, Chang and Shih [CS03] developed an L-system-based simulation to generate rust on metals in seawater. They combined this system with a simple current model to compute rust diffusion. Lately, Wojtan et al. [Woj+07] presented a method that simulates acidic corrosion. They discretized the chemical equations using finite differences and used the results to modify the surface represented as a level set.

Matter dissolution is not always a mechanical process. Several chemical reactions can affect stones and other construction materials resulting in a loss of matter. The first authors to study these reactions on stones are Dorsey et al. [Dor+99]. They proposed to use a slab structure as a volume aligned with the object surface. The simulation consists in estimating cycles of penetration of water and dissolution is driven by a differential equation. They are able to achieve very realistic results although it requires a high computational cost. Later, Jones et al. [Jon+10] simplified the weathering of concave stone, considering animation and game industry. They proposed a simple technique that simulates the matter dissolution caused by the interaction between stone and moisture taking into account the mean curvature and a voxel representation of the model. Xue et al. [XDR11] proposed an image-based technique to simulate smoothing and roughening of stones due to chemical interactions. They compute a patch segmentation considering the weathering tendency of each pixel, they then compute smoothing and roughening on every patch, and finally apply a spheroidal erosion technique to the silhouette. More recently, Mérillou et al. [Mer+12] proposed a technique to compute the decay produced by salts on buildings. They simulate crystallization depth of salts taking

(cc) BY-NC-ND

into account the exposition to external factors and the properties of the material. The final degradations are visualized by means of displacement maps. In Figure II.6, several examples of these techniques are shown.

Fading is a change in appearance that occurs caused by UV light that generates a chemical reaction in which organic compounds are oxidized and degraded. Dorsey et al. [Dor+99] also studied it in the previously stated work. They proposed to take into account the computed composition of minerals near the surface and simulate the translucency and coloration by accounting for the scattering of light inside the stone using a subsurface Monte Carlo ray tracer. More recently, Kimmel et al. [Kim+13] proposed a framework to simulate spectral changes generated by absorbed radiation. They use the Kubelka-Munk model [KM31] to compute the absorbed radiation and a physicochemical approach that takes into account the kinetics of pigments and dye.

Efflorescence is the weathering process through which the internally held salt of a porous material, after being dissolved in water, migrates to the surface and leaves a coating when the water is evaporated. Shahidi et al. [SMG05] proposed a physically-inspired method to simulate it. They generate a 3D texture that stores the affected zones by means of a random-walk algorithm simulating the transport and crystallization of salt.

Pollution deposition is an important weathering effect in cities. It is responsible of the appearance of most stains on facades and monuments. Although it could be considered as a mechanical effect, the key aspect of its development is the chemical reaction present in interaction between the surface and pollutants. Mérillou et al. [Mer+10] studied the formation of black crusts due to pollution deposition. They proposed a physically-inspired method where they identify the affected regions taking into account external factors and a geometry accessibility term. The final render consists in a modification of the BRDF of the material in function of a time rate.

Biological effects

Biological processes can alter objects and their materials over time. For example, when organisms colonize the surface of an object, they change its appearance showing patterns on its surface. Moreover, these patterns can vary over time due to the lifecycles of the responsible organism. For living beings these biological processes are even more important, since its own life cycle alters their own appearance. Figure II.7 shows examples of techniques that deal with

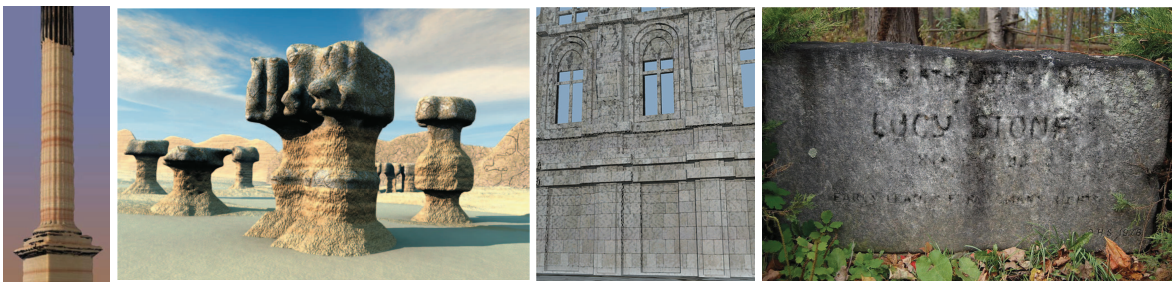


Figure II.6: *Different scenes after a matter dissolution process produced by chemical reactions. From left to right, images generated by Dorsey et al. [Dor+99], Jones et al. [Jon+10], Mérillou et al. [Mer+12] and Xue et al. [XDR11].*

(cc) BY-NC-ND

each of the subtypes mentioned below.

Organic material growth on surfaces is very common in exterior scenes. This is a process that depends on external factors like moisture, light and nutrients. Although it mainly changes the appearance of surfaces, in some cases its presence can result in important changes in objects shape. Desbenoit et al. [DGA04] introduced a method called Open Diffusion Limited Aggregation to compute the growth of lichens on surfaces. In their simulations lichens grow considering environmental conditions and competing for favorable resources. The rendering step is done by instantiating stored models of lichens. Later, Fan et al. [FLW13] studied the ring rot formation in fruits. They propose to use a numerical reaction-diffusion model guided by fruit and ambient properties to compute the bacterial colony and the concentric ring pattern. More recently, Hadrich et al. [Ḧ17] proposed a method to model climbing plants. They represent plants by a particle system that respond to external factors and internal properties during its growth, which is linked to a Lagrangian fluid dynamics simulation.

Senescence is the gradual deterioration of lifeforms in which the organism increases mortality after maturation. It is an important indicator of aging in all living beings and it usually results in changes in both shape and color. The first author dealing with its consequences was Blinn [Bli78], who proposed to represent wrinkles using bump mapping. More recently, several works on fruit senescence have been proposed. Kider et al. [KRB11] developed a technique that handles changes in shape and color. They propose a reaction-diffusion model that generates growth patterns of bacteria and nutrient depletion. After that, they use these points and a set of springs to simulate changes in shape due to water loss. Liu et al. [Liu+12] proposed a framework based on finite element methods to deal with fruit decay. They solve the diffusion process with a tetrahedral representation of the fruit using a moisture value integrated in a Lagrangian dynamic equation. As for the skin, they represent it with a triangle mesh coupled with the volume by a tracking force. Jeong et al. [JPK13] studied the deformation produced by drying leaves. They represent the leaf with a Delaunay triangulation discretized through the vein structure and its Voronoi diagram, and use this structure to simulate water flow inside the leaf and its shrinkage when it dries.

Skin aging is one of the most important senescence indicators in humans. Wu et al. [Wu+99] studied it first. They proposed a three-layer representation of skin and muscles connected by springs. A finite-element method is used to simulate wrinkles that are stored in a texture. In order to improve realism, Boissieux et al. [Boi+00] introduced a volumetric representation of the skin by means of a layered structure where each layer is set with its own properties. Later, Golovinsky et al. [Gol+06] proposed to use a statistical method to synthesize skin imperfections from a previously captured atlas. They captured high-resolution face geometries and separated details from the surface storing them into displacement maps using a texture analysis framework. More recently, Iglesias-Guitian et al. [IG+15] presented a biophysically-based model to reproduce optical changes in skin appearance. Their model consists in a multi-layered model that is able to reproduce changes in reflection, scattering and absorption of the skin. They modify the biological properties (structure and composition) of each layer, fitting profiles from sparse measured data. Finally, Frerichs et al. [FVG16] developed a technique to simulate livor mortis coloration changes in skin. Their model consists of a two-layer triangle mesh representing skin and a tetrahedral mesh where blood dynamics are simulated. Rendering is done by combining the different layers using a diffusion approximation approach.

(cc) BY-NC-ND



Figure II.7: *Examples of different biological weathering effects. From left to right, organic material growth by Hadrick et al. [H+17], senescense by Kider et al. [KRB11] and sking aging by Iglesias-Guitian et al. [IG+15].*

II.1.2 Multi-effect Techniques

Real scenes are very complex. They are composed by many different materials exposed to several weathering effects. In order to compute them, if we rely on effect-centered techniques, we would be forced to use several different techniques. This is not a suitable option because most of the weathering processes are interrelated. However, simulating the whole weathering process is very time-consuming and very complicated due to the fact that all of the underlying processes are still not understood. In spite of this, there are several groups of techniques that try to handle weathering processes in a generic way.

The first group is composed by general techniques that try to simulate several weathering effects in textures and geometry following different approaches. In the second group we introduce the techniques that use sources to emit weathering particles and compute the degree of weathering taking into account the accessibility with regard to the source. In the third group, we find techniques that capture an object and modify its texture to simulate weathering. And, finally, the last group is formed by techniques that apply weathering simulations directly in images.



Figure II.8: *Different weathering effects simulated using γ-ton technique [Che+05]. From left to right, we can observe corrosion, erosion and stain bleeding.*

(cc) BY-NC-ND

General techniques

The pioneer work in this field was proposed by Becket and Badler [BB90] based on real world observations. They proposed a rule-based system to generate weathered textures relying on fractal subdivision techniques and simple distribution models. Despite the fact that they are able to simulate a wide range of effects, their results are not realistic due to the lack of physical validation and image analysis.

Later, other general techniques have been developed that also deal with geometry changes. Cutler et al. [Cut+02] proposed a procedural approach to generate solid models. The designed operators sculpt the model and can be applied to represent weathering effects. They use a tetrahedral representation to deal with different layers in the volumes and they can apply powerful simulation tools such as finite element methods or particle systems. They can represent deformations, displacements of elements and, even, biological growth.

More recently, Clement and Paquette [CP10] designed a framework to apply weathering effects to objects from an exemplar image. Their framework takes into account object properties and can be used to apply weathering to different objects in a similar way or to apply different weathering effects to a unique object. The technique uses a color-independent synthesis process to produce different colorations of the same weathering effects and bump mapping to represent complex BRDFs.

Using weathering sources

An important group of global techniques is characterized by the use of an abstraction of weathering processes as particles emitted from weathering sources. These techniques are able to deal with several weathering effects that are linked to the accessibility of the surface with respect to external factors.

Wong et al. [WNH97] were the first on using this analogy. They proposed to compute a map that expresses the potential of presence of surface imperfections. This tendency map is generated taking into account the accessibility to a set of weathering sources defined in the scene and refined using the curvature of the surface. The weathering sources follow the light source analogy by relating flux density to weathering tendency. The rendering step is done using the tendency map and applying suitable noises to it.

Later, Chen et al. [Che+05] proposed a technique inspired by photon mapping, that is able to achieve multi-effect representations. They propose to use a specific type of particles called γ-ton, representing weathering characteristics. In a first step, these particles are thrown through out the scene from weathering sources and a γ-ton map is generated accounting for the interactions between particles and surfaces. During the second step the weathering effect is generated by modifying the properties and geometry of surfaces. As the technique is interactive, it allows to generate results with multiple grades of weathering. Figure II.8 shows examples of the several effects that the γ-ton method can generate.

More recently, some extensions have been done to the work of Chen et al. First, Xia et al. [Xia+11] used it to simulate weathering effects in videos. They compute a depth map of the video frame and extract the objects with a depth-based method. Then, they reconstruct the 3D object and use the γ-ton technique to compute a weathering map that is transferred to the original video frame by a blending operation. Later, Gunther et al. [GRG12] proposed to take advantage

(cc) BY-NC-ND



Figure II.9: *Different examples of weathering effects transferred from a captured object. From left to right, patina from Lu et al.[Lu+07], pottery decay from Mertens et al. [Mer+06] and corrosion from Wang et al. [Wan+06].*

of GPU parallelization to be able to simulate γ-ton propagation and aging at interactive frame rates. Moreover, they proposed to simulate the final weathering effect in shaders using simple rules and weathering textures that can be combined with the original surface properties during the composition stage.

Weathering of captured objects

Although the realism of synthesized images is increased after applying weathering simulations, it is still far away from the apparences due to real processes. As an alternative to expensive physical simulation, several works have focused on capturing objects with weathered textures in order to use real information to synthesize weathered textures. In Figure II.9, several examples of these techniques are shown.

Some techniques were focused on BRDF changes of weathering processes. Gu et al. [Gu+06] proposed a system to acquire, model and render time-varying surfaces. First, in order to cover several weathering effects, they generate a 26-element database of appearance changes at different moments using a multi-light source, multi-camera dome. Then, they propose a "Space-Time Appearance Factorization" as a model that separates temporally-varying effects from spatial variations. It estimates a temporal characteristic curve only depending on the effect and a static base texture. Finally, to render the effects, they propose a data-driven BRDF guided by the curve, the static texture and the evolution rate, and offset of different data locations. This method can be used with any kind of time-varying surfaces, but it only takes into account shading changes.

In a similar approach, Wang et al. [Wan+06] proposed a method that is able to model time-variant surface appearance after capturing just one instant. The main difference is that instead of capturing weathered textures at different instants, they use multiple samples at only one instant. They rely on the idea that in a simple instant, different degrees of the same weathering effect are present. After the user marks the lowest and highest weathered points on the surface, the system is able to construct a neighborhood graph, called appearance manifold, that approximates the subspace of weathered surface appearances for a material. Thanks to it, a weathering degree map is built taking into account the distances in the graph of each point to the lowest and highest weathered areas. This construction pairs the weathering degree with the appearance state taking into account spatial and temporal characteristics of the material. The final texture

(cc) BY-NC-ND

synthesis is performed based on the previous pairing and an interpolation to preserve coherence between different frames. It allows to weather or un-weather objects texture in a realistic way requiring only one capture and simple user interaction.

Another group of works that capture real objects focus on studying their weathering changes by relating them to geometrical properties of the objects. Mertens et al. [Mer+06] proposed a method to transfer geometry-dependent textures from one object to another. Given an acquired object with texture, they propose to analyze the geometrical properties of the mesh that are potentially relevant to texture variation (i.e., curvature, accessibility, ...), and they compute the correlation between features and texture using Canonical Correlation Analysis. The most correlated features form the guidance map for the texture synthesis step. Before this step, they introduce a feature matching step between the acquired and target meshes, that transfers multi-scale characteristics from the source to the target mesh. Finally, the texture synthesis is addressed in two possible ways. If the correlation between mesh and texture is not too strong, they use an adapted version of Turk's non-parametric texture synthesis method [Tur01]. Otherwise, they propose to adapt Heeger and Bergen's model [HB95].

Looking for physical validation on weathered texture synthesis, Lu et al. [Lu+07] developed a technique that studies the role of context parameters in weathering effects. Their work relies on capturing progressively-varying textures and shapes at different time steps while controlling other external factors that influence the changes. They analyze the relation between changes and shape of the object, trying to identify relevant context parameters that control these changes. Once this relation is identified, they compute a spatio-temporal texture that contains the time-varying appearance history of the captured object paired with the context features related to it. Then, they synthesize the new spatiotemporal texture for the target mesh by transferring to it the input texture patterns using texture synthesis techniques. This process is guided by the context parameters of the target mesh in order to model the variations of the texture that depend on these parameters. The technique is fully automatic for a whole object, and allows the transfer of textures to partial regions with user interaction. Moreover, they propose a framework to apply several weathering effects, that combines the current effect with those applied before.

Image-based weathering

Finally, there is a group of techniques that study weathering effects using image-based techniques. This kind of techniques, although not physically-based, provides realistic results and gives the possibility of modifying real-life images. Figure II.10 shows examples of images generated with techniques described below.

The first image-based multi-effect weathering work was an extension of the appearance manifold technique [Wan+06], developed by Xue et al. [Xue+08]. Given an input image, the user first segments the weathered regions of the object. Then, the image is analyzed in Lab color space. The chroma channels are denoted as weathering components and an appearance manifold is built from them, while the luminance is used to generate a shading map. Then for each pixel they decompose the weathering degree and shading, projecting the pixel in the appearance manifold to obtain a weathering degree map. Finally, they can edit the weathering of the image using a pixel-walking scheme over the appearance manifold obtaining the weathering component for the new weathering degree and the shading map for the luminance component.

Later, Rosenberger et al. [RCOL09] proposed a technique to synthesize new textures from an

(cc) BY-NC-ND

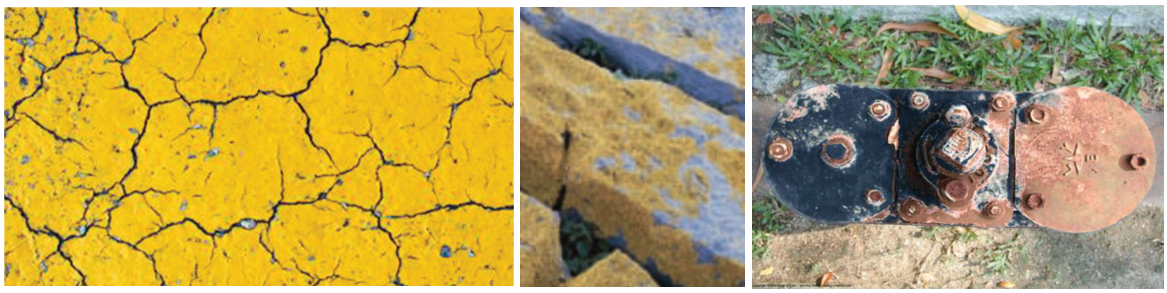


Figure II.10: *Different images generated using image-based weathering methods. From left to right, cracks from Bellini et al. [BKCO16], lichen growth from Iizuka et al. [Iiz+16] and corrosion from Xue et al.[Xue+08].*

exemplar using control maps. The technique is based on the observation that heterogeneous textures with several weathering effects can be seen as a set of layers, each one with a more homogenous texture. First, they decomposed the exemplar image in layers. Then, they use a novel algorithm based on shape synthesis to generate a new combination of layers visually close to the exemplar ones. This is done using a bidirectional measure of similarity between shapes of the layers based on their boundaries. Finally, they convert the new layers set in a texture, thanks to a transfer process.

Lately, Iizuka et al. [Iiz+16] presented an approach for generating detailed appearance weathering effects in a single image. Instead of considering weathering as a time-varying reflectance transition, they model it as a feature propagation from most weathered regions to the rest of the image. After the user indicates the less and most weathered points of the image, they compute the weathering degree map using Radial Basis Functions. Then, they generate a weathering exemplar by extracting the most weathered regions using graph cut and filling the existing holes. From this exemplar, they find optimal patches using a global energy optimization guided by the weathering degree map. For the final render, they propose to use a weathered texture that covers the non-weathered regions, that are generated using the found optimal patches and the shading map. The technique can be used with automatic or manual weathering map updates that represent the weathering simulations.

Bellini et al. [BKCO16] proposed a technique to synthesize automatically time-varying weathered textures exploiting the fact that textures follow repetitive patterns while weathering effects are unique. First, they compute a weathering degree map dividing the image in patches and estimating the prevalence of each patch taking into account their luminance and gradient. Then, they compute an intact texture. The de-weathering and weathering processes are performed using the intact texture by blending patches from the original texture. The first one is guided by the weathering degree map and the second by iteratively using random points where they increase the weathering degree map.

Finally, Zhou et al. [Zho+17] presented a method to synthesize heterogeneous textures that can be used for weathering simulations. Given an input texture, they generate a guidance map formed by a scalar progression channel that captures low-frequency spatial changes and a direction field that stores the dominant local orientation. Then they use this computed guidance map to refine target guidance maps by introducing into them procedural noise that follows the statistics of the acquired ones. The target guidance maps can be user defined or generated

(cc) BY-NC-ND

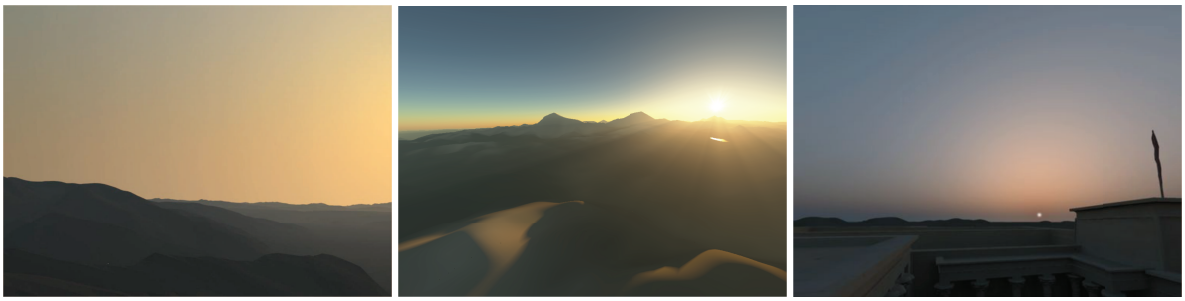


Figure II.11: *Solar radiation models are commonly used to render skies. From left to right, images generated using Preetham et al.'s model [PSS99], Bruneton and Neyret model [BN08] and Hosek and Wilkie model [HW12].*

automatically taking into account the geometry of a target 3D model.

II.2 Estimation of Weathering Factors

There is a set of external factors that have a high importance in the development of weathering effects. Not only they influence the rate of progress of weathering processes, but they can even propagate the effects through the scene. Although their study could be considered out of the scope of Computer Graphics, there are some techniques that deal with them, providing tools to improve the computation and visualization of weathering effects. In the following subsections, we will introduce briefly the most important ones.

II.2.1 Solar Radiation Estimation

The sun is probably the most important source of energy we have available. It is responsible of the light that illuminates our world and, also, it is one of the most important sources of energy at our disposal. Computer Graphics community has focused its study in two big areas: on one hand, modeling solar radiation distribution on the sky and, on the other hand, studying its effects in the scene.

Solar radiation models

Solar radiation modelling became popular in Computer Graphics after the pioneer work of Preetham et al. [PSS99]. In their work, they used the well-known Perez model [PSM93] to compute the non-uniform variations of radiation over the sky dome. The Perez model computes the luminance of clear skies using an analytic formula that computes the radiation of circumsolar and horizon regions and superimpose them over an isotropic background. After computing several reference simulations at different solar elevations and sky conditions using the Nishita model [Nis+96], Preetham et al. replaced the tabulated values in the Perez model by a set of linear functions that depend on turbidity. Moreover, they introduced interpolated chromatic channels to compute sky color distributions.

More recently other works have been developed. Bruneton and Neyret [BN08] proposed a method to render the atmosphere in real time, that handles multiple scattering after the precomputation of the light transport equation. This precomputation is based on a formulation valid for all view and sun directions and, thus, it can compute the sky from all viewpoints (ground

(cc) BY-NC-ND



Figure II.12: *Solar radiation models have several applications, some of them are shown in this figure: from left to right, urban global illumination by Argudo et al. [AAP12], cloud rendering by Dobashi et al. [Dob+00], solar exposure on cities by Liang et al. [Lia+14] and interior daylight building illumination Walter et al. [Wal+05].*

to space). Later, Hosek and Wilkie [HW12] extended the work of Preetham et al. They introduced modifications in the fitting function coefficients that results in an improvement of sunsets and high turbidity setups. Moreover, they introduced ground albedo in the model and adapted it to handle spectral data. More recently, Satilmis et al. [Sat+17] presented a method to compute a sky model using a machine learning approach. Their Artificial Neural Network is able to work with analytic sky models and captured environment maps after being trained with a sparse set of samples. Figure II.11 shows several skies rendered using some of these models.

Additionally, some validations of these models have been developed. Kider et al. [Kid+14] presented a very wide comparison study between several radiation models. They presented a framework to capture real radiation map and compared it against implementations of several models. Recently Bruneton [Bru17] provided a qualitative and quantitative evaluation of clear sky models. Moreover, he compares these Computer Graphics models against a reference model from the physics community.

Solar radiation applications

Several research areas have addressed urban solar radiation in their applications. It has been used for instance to estimate indoor illumination with daylight. Dobashi et al. [Dob+94] proposed a technique that computes natural illumination through transparent planes. It takes into account non-uniform luminous sky distributions and glass distribution of transparency according to incident angles. More recently, Fernandez and Besuievsky [FB12] proposed a method that solves inverse lighting design problems taking into account the combination of natural and artificial light in indoor scenes taking advantage of scene coherence and using a heuristic for optimization. Solar radiation has also been studied in global illumination techniques. Walter et al. [Wal+05] proposed a scalable framework to compute realistic illumination by grouping and approximating efficiently many point lights. Their technique handles several light sources including sun and sky. Argudo et al. [AAP12] developed a technique to compute solar global illumination in urban scenes thanks to an interactive GPU ray-tracing implementation. Moreover, it has been studied the contribution of the sun in atmospheric rendering techniques. Dobashi et al. [Dob+00] studied how sun light interacts with animated clouds. Riley et al. [Ril+04] proposed a multiple-model lighting system to capture multiple atmospheric phenomena, like rainbows and halos, caused by scattering of sun light in the atmosphere. Related to these, Sun et al. [Sun+05] proposed a real-time technique to study interaction of sun light in participating media such as fog and mist. They proposed an analytic integration of single scattering light transport equations.

(cc) BY-NC-ND

Other techniques face the problem from an engineering point of view, trying to estimate solar exposure on urban surfaces. This is an important topic due to the current interest of several areas in energy efficiency, but a realistic estimation in a complex scene is still a challenging task. Main techniques sample the geometry in several points and interpolate the surface in between, avoiding the details smaller than the sampling resolution. One of this tools, Heliodon [Mer+10], estimates solar exposure using ray casting for the direct component and an estimation of the sky visibility for the diffuse component. Also, we can find in this group ArchSim in the DIVA suite [DRM12], which simulates the solar exposure using a simulation based on Radiance [War94], and the popular commercial software Autodesk Revit with its energy analysis tool called Autodesk insight [Aut].

Moreover, we would like to mention the work of Liang et al. [Lia+14]. In their work, they proposed a technique based on surface mapping techniques that transform 3D urban scenes in 2D maps and use a GPU-accelerated computation. Finally, Freitas et al. [Fre+15] presented a very complete state-of-the-art review of techniques for estimating solar radiation in urban environments. The main conclusion of their review was the need of a compromise between accuracy and computational costs.

Figure II.12 show some examples of these applications.

II.2.2 Wind Estimation

Wind simulation is a well-known and long-studied problem. It has been well studied in many fields, including meteorology, climate, energy, human comfort studies and even urban planning. In Computer Graphics wind simulation has been used for the animation of plants (grass, leaves, trees) and water (rain, sea storms). Scientific Visualization has also largely dealt with the involved fluid dynamics.

Computational Fluid Dynamics

For an accurate simulation of flows, we have to address ourselves to physically-based techniques. Computational Fluid Dynamics (CFD) is the field of fluid mechanics that uses numerical analysis and data structures to solve fluid flow problems. Main alternatives inside this field are Navier-Stokes equations (NS), Smoothed Particles Hydrodynamics (SPH) and Lattice-Boltzmann Method (LBM). Figure II.13 shows examples of liquid and gaseous simulations using techniques introduced below.

NS is a reformulation of Newton's Second Law that describes the motion of fluid substances. After defining initial and boundary conditions, the equations can be discretized with finite differences and a grid, and solved using numerical methods. They have been traditionally used for accurate simulations but their complexity makes them infeasible for fast simulations of large scenarios. Examples of implementations of these equations can be found in the works of Foster and Metaxas [FM97], Foster and Fedwick [FF01], Sussman [Sus03] or Layton and Van de Panne [LP02], among many.

SPH techniques simulate fluids avoiding the grid and using a particle-based system. The fluid properties are interpolated among the particles. One problem is that it simulates compressible flows, although some implementations have adapted it to achieve near-incompressibility. Examples of techniques that use SPH are the works of Stam and Fiume [SF95], Muller et al. [MCG03]

(cc) BY-NC-ND

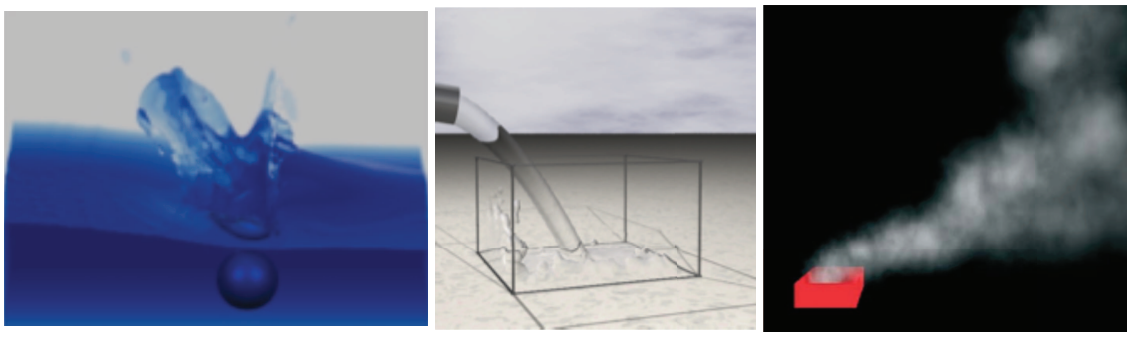


Figure II.13: *CFD techniques are used to simulate flows in a liquid or gaseous state. From left to right, examples of simulations from Foster and Fedwick [FF01], Goswami et al. [Gos+10] and Li et al. [LWK03].*

and Goswami et al. [Gos+10].

LBM is a group of techniques that has achieved more interest recently. It is based on statistical physics and, instead of solving Navier-Stokes equations, they solve the discrete Boltzmann equation to simulate fluid flows and collisions. It uses a grid to discretize the fluid scenario and directions to represent where fluid molecules can travel along. It is efficient and suitable for parallelization. Examples of LBM techniques are those presented by Li et al. [LWK03] and Tölke [Töl08].

Urban Physics

In this thesis, we have focused on the study of wind carried out by urban physics research. Understanding urban dynamics is an important research subject that also relates to weathering processes. Blocken [Blo15] presented a study that discussed the importance of Computational Fluid Dynamics in the study of urban physics. Moreover, he discussed the existing possibilities and limitations of CFD and gave some clues to achieve accurate and reliable simulations.

Several urban physics techniques have studied the effects of wind in urban environments using CFD. Blocken et al. [BJH12] performed CFD simulations to observe pedestrian wind comfort and wind safety in urban areas. Van Hooff and Blocken [HB10] presented a CFD approach for simulating urban wind flow and indoor natural ventilation of a very detailed model using a high-resolution grid. Figure II.14 shows a result of their simulation. Moreover, Ramponi and Blocken [RB12] studied the cross-ventilation of generic isolated buildings. Furthermore Blocken et al. [BCS07] investigated the characteristics of wind speed in the passages between parallel buildings. Besides, Gousseau et al. [Gou+11] researched how pollutants disperse inside a city represented by a high resolution grid. Finally, Janssen et al. [JBC07] modeled moisture and heat transfer in buildings under winds. These represent only a small set of examples among many others.

Although high-quality results can be achieved. With these methods, the current challenge in wind simulation techniques consists in achieving interactive simulations but accurate enough to be considered reliable, as stated by Blocken [Blo14].

(cc) BY-NC-ND

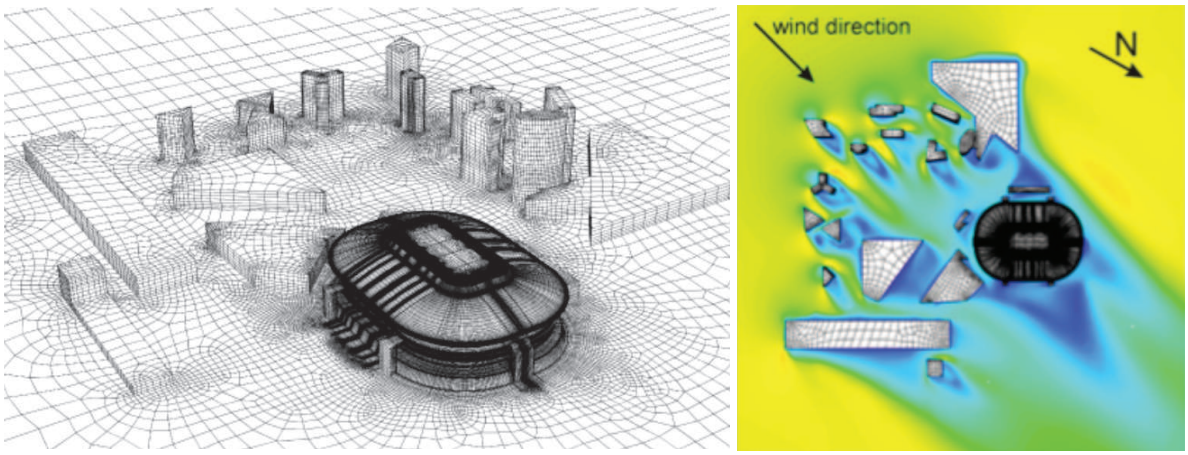


Figure II.14: *Amsterdam ArenA stadium is an example of a wind simulation study in urban physics developed by Van Hooff and Blocken [HB10]. To the left, it is shown the 3D model while, to the right, the results of the simulation.*

II.3 Weathering Effects in Cultural Heritage

One of the disciplines that have studied weathering effects widely is Cultural Heritage. The need of preserving the existing cultural assets has involved a large work from the Cultural Heritage community to better understand the associated weathering processes. Although Computer Graphics would not seem directly related to this research area, it has proved to be a very useful tool to help curators in their work. In the following lines, we will review briefly Cultural Heritage and Computer Graphics works related to legacy preservation with a special focus on the analysis of weathering damage.

II.3.1 Cultural Heritage Analysis

In Cultural Heritage several efforts have been made to better understand the weathering effects that legacy suffers in order to develop correct preservation strategies. One of these efforts consists of mapping Visual Weathering forms. This method, which has been widely used in the literature [FHK95][FHK97][FHB03], is an established non-destructive procedure for in situ studying material damages and providing relevant information on the evolving weathering mechanisms. The resulting maps are very useful to assess the present damages and develop preservation strategies for the future. Figure II.15 shows an example of weathering mapping performed by Fitzner et al. [FHB00].

Since this mapping process is very subjective, it is difficult to have a precise idea of the state of a monument. To solve that, Fitzner et al. [FHB00] introduced damage indices as a tool for the quantification and rating of stone damages. These indices are based on the affected area and the depth of the loss. In the same line, Warke et al. [War+03] tried to simplify the prediction of the behavior of the building stone decay by using the TNM Staging System analogy. This system that uses the depth, area and spread of a decay, helps the conservator to decide if conservation is still possible. Lately, Smith and Prikryl [SP07] examined the medical analogy for stone weathering. They observed the potential of this analogy as it provides understanding of the background of the buildings and they stated the importance of formalizing the condition assessment. More recently, in order to reduce the ambiguity of deterioration mapping, Delgado

(cc) BY-NC-ND

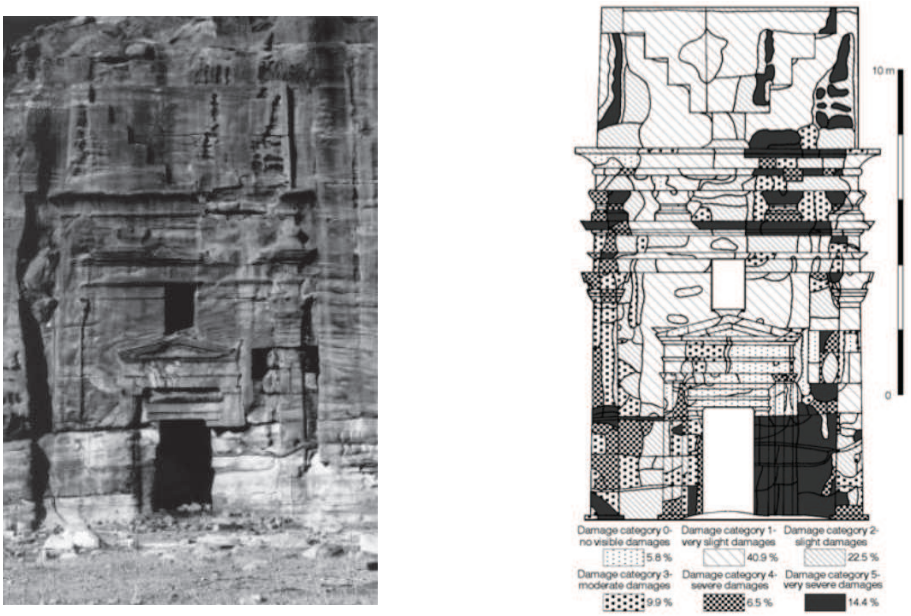


Figure II.15: *Example of weathering analysis and mapping on a tomb in Petra (Jordan) from the work of Fitzner et al. [FHB00]. To the left, it shows a picture of the tomb and, to the right, the corresponding analysis.*

[DR15] proposed a new mapping strategy that divides the monument into entities and takes into account its structure to assign a level of priority of restoration.

In the last years, some efforts have promoted the use of 3D models in archaeological heritage. De Reu et al. [Reu+13] indicated the limitations of using 2D drawing for weathering mapping and proved that using 3D models is a cost-effective improvement. In a similar line, Stefani et al. [Ste+14] developed a toolkit that simplifies the mapping task. They use a connection between the 3D model and a 2D representation using the NUBES web platform. Finally, Koutsoudis et al. [Kou+14] analyzed the quality of reconstructions of photogrammetry algorithms and they found that high-quality results can be achieved if the number of images used is large enough.

One of the most important works in this field is the glossary presented by ICOMOS ISCS (International Council on Monuments and Sites - International Scientific Committee for Stone) [ICO08], a global non-government organisation dedicated to promoting the application of theory, methodology and scientific techniques to the conservation of Cultural Heritage. During the last decades, rather than focusing on analysis tasks, ICOMOS ISCS has collected and combined definitions that are in use for the visual weathering forms of monuments to obtain a generalized glossary together with its translation in different languages. This glossary provides the explanations for different types of visual weathering forms. Its importance lies in the fact that the mapping process is a manual work based on observations and that it carries the possibility of getting different mappings for the same monument. This possible disparity makes it difficult to track weathering effects over time and to compare the monitoring of a same monument made by different people. By contrast, ICOMOS ISCS glossary is a step forward towards the needed standardization of criteria in weathering analysis in Cultural Heritage. In the same line several works have been developed from multidisciplinary teams. Doerr [Doe05] proposed an ontology that makes it posible to exchange and integrate information between different sources of Cultural

(cc) BY-NC-ND

Heritage knowledge. More recently, Cacciotti et al. [CBV15] proposed a computer-readable model based on an ontological decay representation in order to improve identification of the relations between factors and effects. Finally, Messaoudi et al. [Mes+17] developed a model to correlate an ontological model and a semantic-aware 3D representation in order to help in the recording ant integration of multidisciplinary observations in conservation projects.

II.3.2 Computer Graphics Analysis Tools

Computer Graphics can provide useful tools to reduce the subjectivity of the current mapping methods for weathered objects. They can provide for instance a geometrical analysis, not only to help in the assessment of weathering in Cultural Heritage, but also to understand how weathering processes affect these assets and ultimately to use this information to improve its simulation.

Although to the best of our knowledge there is no technique that identifies and maps automatically weathering effects, there are techniques that try to handle this kind of problem. One group is formed by techniques that try to identify and map reliefs and details on surfaces. We note the work of Zatzarinni et al. [ZTS09] in which they compute the height function over the base, without a previous computation of the base surface, and they use this height to extract relief and details. Similarly, Chen et al. [Che+11] proposed a technique to visualize and edit reliefs previously extracted by a height threshold from smooth surfaces reconstructed using normal smoothing and Poisson reconstruction. There is another group of techniques where the geometry the mesh is analized to enhance or recover features of the models. In that group, we remark the work of Kolomenkin et al. [KST08]. They proposed a new class of view-independent curves, that describe places with the strongest inflections on the surface, and they use it to enhance the features during the visualization. With a similar goal, Lawonn et al. [Law+17] presented a technique to extract carvings on stones. Their algorithm is based on an adaptation of the Frangi filter and an ordering of the detected structures by saliency. They used these structures in the final visualization, enhancing the shading of carvings in the models. In this group, there is also the work of Nespeca and De Luca [NDL16]. They proposed to use the point cloud of a reconstructed wall to obtain different quantitative information from the model. After an analysis, they use this information to guide segmentations and to reconstruct surfaces and volumes. Finally, Peteler et al. [Pet+15] proposed an image-based approach to measure the degradation of a scene using depth maps of the model reconstructed at different time steps. Figure II.16 shows several cases where Computer Graphics are used in Cultural Heritage problems by means of techniques introduced above.

There are several other techniques for geometrical analysis used in Cultural Heritage. They are mainly focused in collection analysis, interpretation, monitoring, restoration and perception. For a complete review, we point the reader to the survey by Pintus et al. [Pin+16].

II.4 Conclusions

In this chapter, we have first described a large number of different works and approaches that focus on the simulation and rendering of weathering effects in Computer Graphics. We have shown that a group of these techniques focuses on simulating a single effect, while others try to simulate several effects at the same time in a more general approach. However, a common limitation of current techniques is that, even for general purpose ones, they are often focused

(cc) BY-NC-ND



Figure II.16: *Computer Graphics is very useful in several Cultural Heritage applications. From left ro right, several examples are shown: relief extraction from Chen et al [Che+11] and Zatzarini et al. [ZTS09], enhancing carving from Lawonn et al. [Law+17] and improving visualization from Kolomenkin et al. [KST08].*

in the effect instead of the causes of that effect. This results in a collection of models oriented towards the final result, which avoids in-depth analysis of the weathering processes. Thus, several involved factors are not taken into account and models can hardly be extended to other effects. For this reason, we consider there is a lack of techniques that face weathering problems from a global point of view, which would allow generating models that can be easily adapted to face different weathering processes.

Then, we have presented a review on weathering factors estimation. We have focused our summary on solar radiation and wind estimation. We have shown how several efforts have been carried out to achieve precise estimations. And how the main problem of existing techniques that estimate solar radiation is the lack of methods that are able to perform detailed estimations without requiring huge memory and/or computational cost. When we analyze the current techniques that deal with wind simulation, we observe that they experience the same problem than solar radiation ones. Despite of being able to achieve precise estimations, they need high amounts of resources. Therefore, we consider that the reason why weathering techniques do not use realistic estimations of these factors is related to their computational limitations. In this review, we would have liked to incorporate also an introduction of techniques that deal with pollution in urban environments. But, to the best of our knowledge, there are no techniques that deal with pollution distribution at a urban scale outside of CG. Such situations make it necessary to develop new techniques that reduce the computational and memory costs of weathering factors estimation without compromising the precision of the results in order to make this estimation useful for weathering simulations.

Finally, we have reviewed the role of weathering effects in Cultural Heritage. In this context, we have made a review of techniques that deal with weathering effects in legacy including those that use Computer Graphics in their studies. Then, we have reviewed Computer Graphics techniques that perform geometric analysis that can be used in Cultural Heritage applications. We have observed that the main problem in the preservation tasks is the lack of standard and automatic methods to detect and map weathering processes. Current assessments involve a high level of detailed surveying that leads to considerable demands of time and expertise. Moreover, although current geometrical analysis techniques of Computer Graphics can help to enhance

(cc) BY-NC-ND

weathering effects, they are not able to detect weathering processes automatically. Hence, there is an important line of research to improve the weathering detection in real objects. This would not only help curators to decide better preservation strategies, but it would help the Computer Graphics community to better understand weathering processes and, thus, to improve their simulations.

(cc) BY-NC-ND

Chapter III. Solar Exposure

III.1 Introduction

The realism of virtual models is directly related with the presence of weathering effects on them. Therefore, the Computer Graphics community has developed several efforts to improve the synthesis of weathering effects. But these efforts, instead of focusing on accurate weathering simulations, have only tended to study how to obtain plausible results. One of the causes of this situation is the lack of practical methods to perform these simulations or to efficiently estimate the environmental factors that affect weathering processes. These factors are complex and their existing estimation techniques require high computational costs. This is the reason why, as we have stated in Chapter II, it is very common to find gross simplifications in their estimation.

One of the factors that suffer more the lack of a proper study in this context is solar exposure estimation. While the amount of radiation received by a surface is a key factor in several weathering processes, its estimation is usually very simplified. At best, it is considered taking only into account the accessibility to the sun. But, reality is much more complex than that. It depends not only on the sun, but on the whole sky hemisphere, whose behavior along the sky dome is anisotropic. Moreover, weathering effects are affected by different regions of the radiation spectrum, which also varies along the sky hemisphere.

Solar radiation has been studied in other areas of Computer Graphics, like in global illumination techniques or in the modeling and simulation of natural phenomena. Furthermore, solar radiation is an important subject in various other fields such as renewable energy studies, climatology and urban planning. To the best of our knowledge, the existing techniques in such areas that estimate solar exposure do not allow the complex geometries of real buildings and they need to follow different simplification strategies. Moreover, their efficiency is directly related with the large number of samples that are often required. Thus, they are very time-consuming when dealing with large cities and high resolutions. These two main drawbacks make them impractical for realistic weathering simulations.

In order to try to tackle this problem, in this chapter we will present our solution to compute solar radiation and its interaction with surfaces. Our solution is a new technique for estimating detailed solar exposure on complex urban models. We represent solar radiation as a directional exposure map integrated over a desired time period. We then evaluate the sky visibility using a two-scale screen-space approach, and we use this information to compute the final exposure

(cc) BY-NC-ND

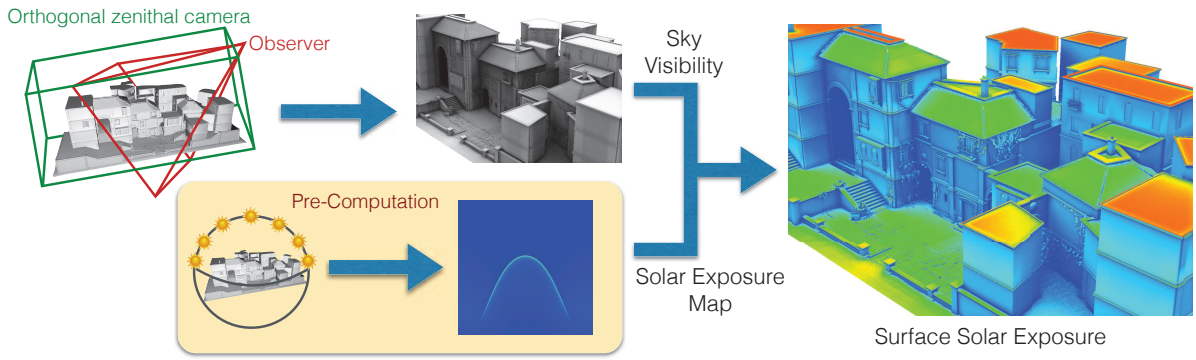


Figure III.1: *Overview of our system. A directional solar exposure map is precomputed over the specified time period and under the given conditions (bottom row). At run-time, we evaluate visibility in screen-space by combining a global and local view of the scene (top row). Surface solar exposure is finally computed by sampling the map within the sky regions visible from each point (right).*

on surfaces. The proposed technique is not only designed for weathering simulations, but it is also useful in the sustainable development of urban environments. The main contributions of our technique can be summarized as follows:

- Our method relies on screen-space operations and a pre-process that stores a single, global exposure map. As a result, our method is geometry independent, has low storage requirements and offers interactive and detailed feedback on complex scenes.
- The method computes sky visibility inspired by screen-space ambient occlusion techniques. This results in a fast sky visibility estimation that compares well against ground truth.
- We take advantage of a pre-filtering strategy so as to reduce the number of samples on the exposure map. This provides a fast and smooth evaluation minimizing the introduced error.
- Our technique is amenable to dynamic modifications of the scene, useful in preliminary stages of applications where considerable accuracy is needed, particularly for preview purposes.

The remainder of this chapter is organized as follows. In the next section, we present an overview of our pipeline. The following sections concentrate on estimating the sky directional exposure map and the screen-space visibility based on two levels of accuracy: both global and local scales. Finally, we describe the solar exposure evaluation step and we present our results and discussion.

III.2 Overview

The overview of our method is depicted in Figure III.1. We start with precomputing a directional exposure map by integrating daylight irradiance over the desired (typically a year) time period (see bottom-left box). Using Hosek and Wilkie technique [HW13], the map is computed according to parameters such as the scene latitude and mean sky turbidity. This is done for each direction inside the full sky dome and stored in a map.

At run-time, we evaluate the visibility of the scene using a two-step screen-space approach.

(cc) BY-NC-ND

Using an adapted screen-space solution, large-scale visibility is computed from a global view of the scene captured from the zenith (green frustum in left-top part). This visibility is then combined with a fine-scale term computed from the current viewpoint using common screen-space ambient occlusion (red frustum). In both cases, visibility is only evaluated for points visible from the current camera, providing the required accuracy for each view without storing or precomputing anything. Thanks to this two-scale process, we achieve a good match compared to ground truth visibility but at interactive rates.

Finally, we use the visibility information to sample the directional exposure map only in the visible regions of the sky dome. We compute the final solar exposure at each surface point (right-most image) integrating the performed samples. This sampling is developed taking advantage of mipmap prefiltering, which reduces the number of required samples.

All these results are obtained in an interactive framework that provides the required accuracy from each view and only entails two geometry-independent textures being pre-computed (i.e. the exposure map and a global depth map). Our technique belongs to the category of interactive simulation approaches, which allows accuracy and building geometries to be changed on-the-fly. In addition, thanks to our two-scale process, we obtaine a good match (compared to ground-truth visibility and solar exposure) over detailed scenes, and achieve a high level of realism when compared to existing solutions.

III.3 Solar Radiation

Correctly modeling solar radiation has been a matter of study for physically-based rendering of outdoor scenes as well as indoor illumination with daylight. But, it also has implications in many other fields outside of Computer Graphics, like urban climate or photovoltaics simulations. In Computer Graphics, two main strategies have been used. On one hand there is a group of techniques that simulate light scattering phenomena in the atmosphere while, on the other, we find techniques that are based on an analytical representation. The latter is mainly based on fitting a set of coefficients from functions to data captured or obtained from former simulations. Our method belongs to the second group.

When we analyze the solar radiation that reaches a surface point, we can split it into the sum of three components [Mer+10]. The first component is known as direct or beam radiation. It corresponds to the radiation that arrives to the surface directly from the sun that has not been absorbed or scattered by the atmosphere. The second component is the sky diffuse radiation. Although its origin is the solar beam, it reaches the surface after scattering in the atmosphere and sky clouds. Finally, the third component is named as reflected radiation. It encloses the direct and diffuse radiation that arrives to the considered surface point after reflecting in other points of the surface. The behavior of these components is affected by several parameters. On a global scale, the radiation is controlled by the geometry of the earth, the relative location of the sun and the current atmospheric transmittance conditions. By contrast, on a local scale the value of the incoming radiance depends on the surface orientation, the ground elevation and the albedo.

Analyzing these components we see that direct radiation provides the main radiation contribution and it is the most straightforward to compute. By contrast, reflected radiation has a high computational cost while it is usually considered to have a lesser impact. On the other hand, the diffuse component has a significant importance while it is considered as the largest potential

(cc) BY-NC-ND

source of computational error [NMK08]. In this work, we rely on the Hosek and Wilkie skydome model [HW13], as it significantly reduces the error related to the diffuse component when compared to the Preetham et al. model as shown in Kider study[Kid+14]. In the proposed technique we neglect the reflections from the ground or neighboring surfaces because their contribution is considerably low.

III.3.1 Computing Daylight Radiance

Kajiya [Kaj86] described the rendering equation that models the radiance reflected by a given point x on a surface. It is based on the law of conservation of energy and describes the outgoing radiance (L_o) as the sum of the reflected portion of emitted radiance (L_e) plus the reflected radiance, which can be described as the integration of the incoming radiance (L_i):

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(x, \omega_i, \omega_o) L_i(x, \omega_i)(\omega_i \cdot n) d\omega_i, \tag{III.1}$$

where f_r is the material BRDF, ω_o and ω_i are the outgoing and incoming directions, n is the surface normal and Ω is the upper hemisphere.

Without the presence of artificial lighting and neglecting reflections between surfaces, the incoming radiance only depends on the daylight illumination. Assuming no ground reflection, we can decompose daylight effects into two main components: the direct beam coming from the sun, L_{dir}, and the sky diffuse term due to light scattering on the atmosphere, L_{dif}. The incoming radiance L_i at a given point x is thus described by:

$$L_i(x, \omega_i) = L_{dir}(\omega_i) + L_{dif}(\omega_i), \tag{III.2}$$

where ω_i is the incident direction defined over the hemispherical domain Ω. It is important to note that both terms represent spectral quantities that change over time. The spectral and temporal component is not considered in the equations to simplify its reading. Moreover, the considered directions ω_i and ω_o are not depending on the spatial position x. We can dissociate this relation due to the relative distance of the sky and the sun from the scene.

We assume the direct component is originated from the centre of the solar disk. Thus, we describe the direct component by a Dirac delta function over the sun direction, ω_{sun}. In this direction, the incoming radiance can be computed as the extra-atmospheric radiance reaching the atmosphere, L_{ea}, weighted by the atmospheric transmittance τ. The latter oversees light attenuation resulting from absorption and scattering. Preetham et al. provide measured spectral data for both factors [PSS99]:

$$L_{dir}(\omega_i) = \tau L_{ea} \ , \ \text{if } \omega_i = \omega_{sun}. \tag{III.3}$$

Several models have been proposed to approximate the effect of the diffuse component. Despite diffuse radiation follows an anisotropic distribution, existing methods rely on both behaviors, isotropic and anisotropic, to describe the sky [NMK08]. Classical anisotropic luminance models, such as the CIE standard or the Perez model, assume the sky is composed of three main terms: a circumsolar region, a horizon band and an isotropic backscattering term. As previously mentioned, our work is based on Hosek's model [HW13] which uses the Perez model to describe the anisotropic behavior of the sky. In it, each of these terms is controlled by a set of parameters (a through e). Although in the original Perez model these parameters are later fitted to measured

(cc) BY-NC-ND

data [PSM93], our work follows Preetham et al. and Hosek and Wilkie approaches of fitting them to simulations [PSS99] [HW12] [HW13]. Moreover, there is an additional parameter to consider different atmospheric conditions called turbidity. This parameter is defined as the ratio of the additional optical thickness of the considered atmosphere to the optical thickness of an ideal atmosphere with pure air. It is computed as follows:

$$T = \frac{t_m + t_h}{t_m} \tag{III.4}$$

where t_m is the vertical optical thickness of the molecular atmosphere and t_h is the vertical optical thickness of the considered atmosphere.

The Perez model was originally restricted to luminance, but Preetham et al. extended it using the same formulation to describe chrominance variations relating the coefficients with turbidity for each color component [PSS99]. The classical Perez equation to describe sky luminance is the following [PSM93]:

$$f(\theta_z, \gamma) = (1 + a\exp(\frac{b}{\cos\theta_z}))(1 + c\exp(\frac{d}{\gamma}) + e\cos^2\gamma) \tag{III.5}$$

where a and b control the magnitude/sign and shape of the horizon band, c and d the intensity and width of the circumsolar region, and e the relative intensity of backscattering received by the earth's surface. These parameters are fit to measured data [PSM93] or simulations [PSS99]. Moreover, θ_z is the angle from the zenith and γ the angle from the sun direction. The function f models the relative luminance of the sky with respect to the zenith luminance, $L_i(\omega_z)$. Zenith luminance can be found in tables or can be based on formulas parametrized by sun position and turbidity [PSS99]. After normalizing and weighting by it, the final diffuse term becomes:

$$L_{dif}(\omega_i) = L_i(\omega_z)\frac{f(\theta_{z,i}, \angle(\omega_i, \omega_{sun}))}{f(0, \theta_{z,sun})}. \tag{III.6}$$

Hosek and Wilkie extended the Preetham et al. formulation to consider spectral bands by introducing slight formula modifications and fitting it for each wavelength separately [HW12] [HW13]. More specifically, they extended the model with more parameters (a through i) and using Bezier curves for calculating them. The calculation process is performed for each given wavelength taking into account the turbidity value of the sky. The computed parameters are stored in tables and used in the radiation distribution. The final equation is the following:

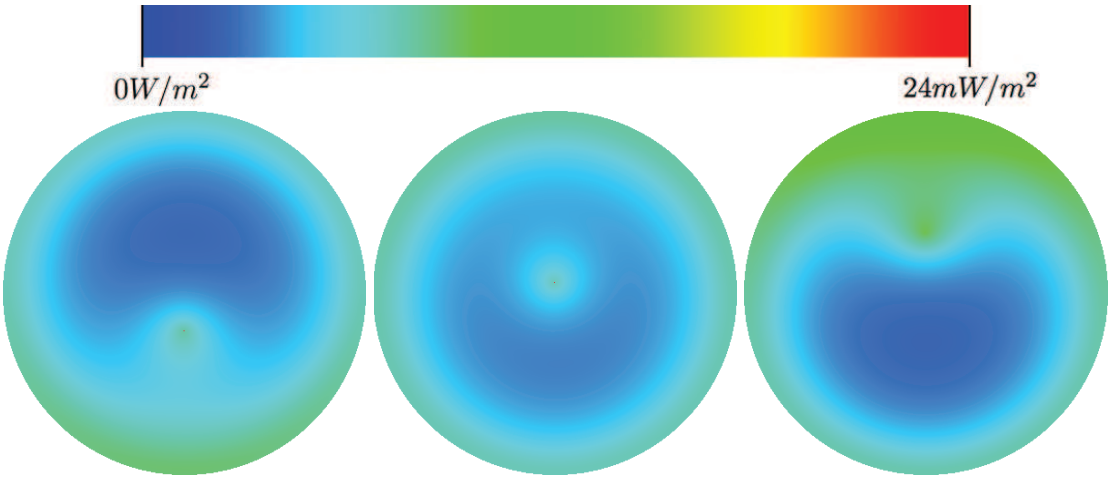
$$f(\theta_z, \gamma) = (1 + a\exp(\frac{b}{\cos\theta_z + 0.01}))(c + d(\exp(e\gamma)) + f\cos^2\gamma + g\chi(h, \gamma) + i\cos^{\frac{1}{2}}\theta_z) \tag{III.7}$$

$$\chi(G, \alpha) = \frac{1 + \cos^2\alpha}{(1 + G^2 - 2G\cos\alpha)^{\frac{3}{2}}} \tag{III.8}$$

Moreover, they modified the final function avoiding the use of the zenith value to simplify the process. Instead of considering $L_i(\omega_z)$, they propose to use the expected value of spectral radiance in a random point of the hemisphere, L_{iM}. The final diffuse term is then calculated using:

$$L_{dif}(\omega_i) = L_{iM}f(\theta_{z,i}, \angle(\omega_i, \omega_{sun})) \tag{III.9}$$

(cc) BY-NC-ND



(a) *Latitude* 0^oN **(b)** *Latitude* 30^oN **(c)** *Latitude* 60^oN

Figure III.2: *Sky exposure with a turbidity value of 2 and at a specific instant (June 23rd 12:00) for different latitudes.*

Figure III.2 shows three distributions of solar exposure at a specific instant and turbidity for different latitudes using Hosek and Wilkie model. The three terms of the diffuse component can be specially appreciated in the middle subfigure. Moreover, it is easy to see the difference of energy in the sun direction (direct component) with respect to the rest of the hemisphere (diffuse component).

III.3.2 Annual Exposure

In many applications, being able to estimate the amount of solar energy reaching a surface over a specific period of time is essential. These computations quantify the incoming energy and provide information for analysis and design. This term is often referred to as solar exposure, and expressed as MWh/m^2. Conventional quantities relate to daily, weekly, or monthly estimations. However, due to the yearly cyclic behavior of the sun, annual estimations are specially relevant for long-term decisions such as building design or weathering simulation.

At this point, we are interested in the distribution of energy over the sky hemispherical domain Ω. Thus, we propose to compute an annual map that stores the hemispherical distribution of incoming energy during a year. This energy will be later combined with visibility information to obtain the final exposure at each surface point. The distribution of energy over the sky depends on the sun path over the specific time period (i.e., a year), the atmospheric conditions (i.e., turbidity value) and the latitude that our model occupies. In our examples, we use a fixed turbidity value representing the average conditions at a specific location and for an explicit period of time. Our technique can achieve a more accurate computation in case different turbidity information is available during the studied period for that location.

In order to estimate an annual map, we need to integrate Equation III.2 over a year for each hemispherical direction ω_i. On the one hand, we compute the diffuse radiation. For each pixel of the map, we compute the corresponding sampling direction ω_i and we sample the sky radiance distribution at uniform time steps within the daily hours of the period. On the other hand, the direct component is computed and also baked into the map to store its temporal behavior over

(cc) BY-NC-ND

the same period. Due to the discrete nature of the sun model, in this case we rely on computing sun direction at each time step and splatting its energy over the map on the corresponding pixels.

At each step, the radiance of the sun and the sky is obtained using Hosek and Wilkie model [HW13], then integrated as previously explained. Our directional exposure map is finally obtained by weighting these values with the corresponding solid angle, i.e., the sky patch or the sun radius in each case. The cosine of the angle formed by each surface normal and sampling direction ω_i will be computed at run-time using the scene information. This term is needed for computing the final exposure, as it depends on surface orientation (see Section III.5).

Besides time integration, we also integrate quantities from the available spectrum. Although Hosek and Wilkie approach is limited to a small range of the light spectrum, one could focus on any specific part of the spectrum if necessary. Focusing on specific ranges such as visible, uv, or infrared light, might be of special interest for applications like weathering or plant growth simulation. The map obtained is ultimately represented as a floating-point texture using the classical Blinn-Newell parametrization of the hemisphere. We chose this parametrization because it is proper for the kind of sampling based on the directions that we use, as detailed in Section III.5.

Figure III.3 shows annual exposure maps for different latitudes and turbidities. Please note that we use different scales for the direct and diffuse components to depict the maps. From centre to right, it is easy to observe how the contribution of the sky diffuse term increases for high turbidity values. From left to centre, the sun path, as expected, is greatly influenced by latitude changes. Also, notice how the energy attributable to the direct term, concentrates at the declination limits of the sun band (see red arrows in bottom-middle image of Figure III.3). This happens because during solstice the variation of the sun path is less than during equinox.

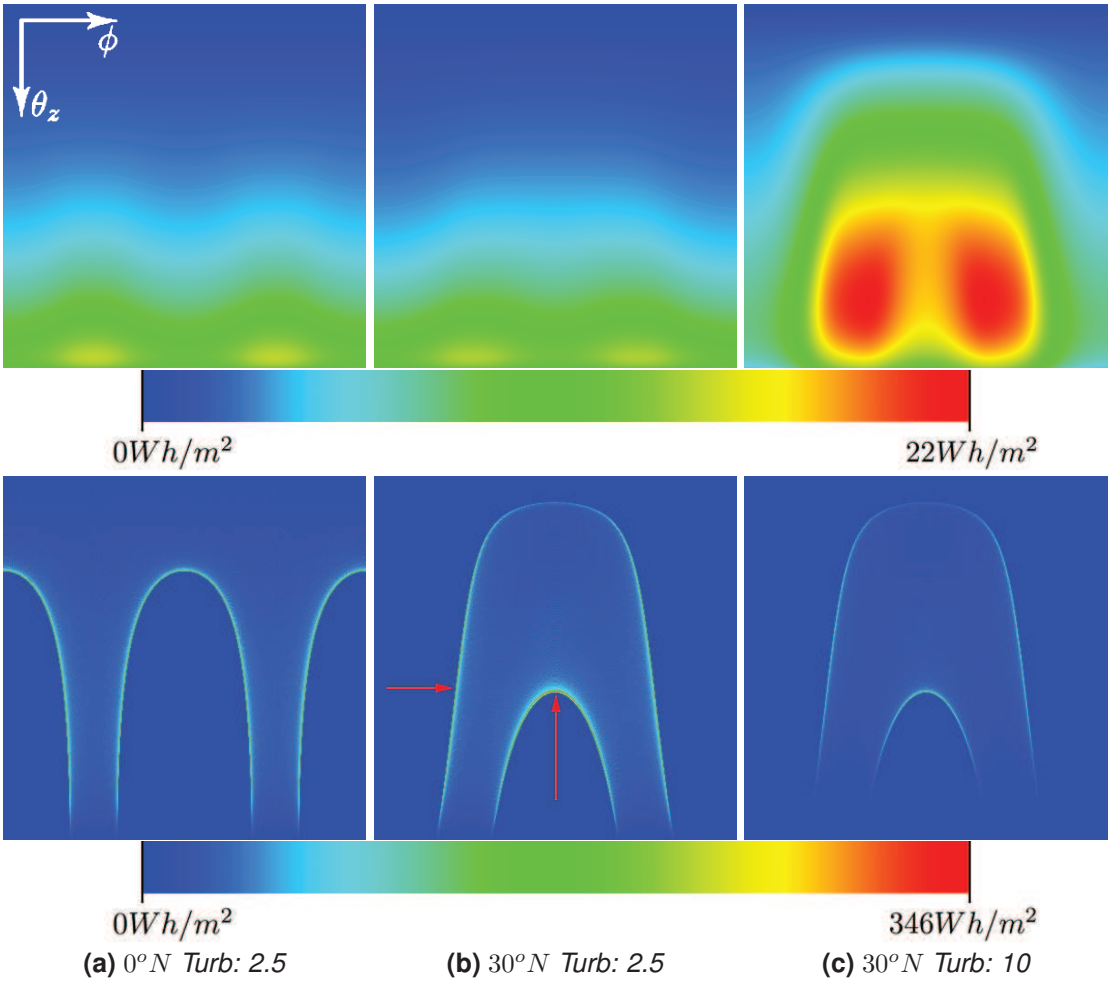
III.4 Sky Visibility Estimation

After computing the exposure map, we need to determine the contribution this map makes at each surface point. Sky visibility is a key mechanism in estimating solar exposure on surfaces. Thanks to it, we are able to estimate the visible regions of the sky from a surface point and, thus, the amount of sky incoming energy at that point. We compute it by evaluating, for every direction of the sky hemisphere, whether an object of the scene is blocking it. This parameter, that is only geometry dependent, is commonly called the Sky View Factor (SVF).

The SVF was introduced in a pioneering paper from Johnson and Watson [JW84]. It is defined as the fraction of radiant flux leaving a planar surface that is being intercepted by the sky. Figure III.4 shows a visual representation of it. This factor is closely related to the well-known Ambient Occlusion (AO) term [MFS09] used to estimate part of the rendering equation (Equation III.1) in Computer Graphics. In this equation, AO is defined as the amount of environmental occlusion from a surface point. So, the main difference is that while AO computes local accessibility, SVF only computes sky accessibility. This difference means that, to make this complementary relationship true, AO needs to be adapted to take into account the sky hemisphere instead of the surrounding hemisphere at each point.

Since we want to be able to process large city models at interactive rates, we resort to screen-space techniques, which are independent of scene complexity and can be easily evaluated with a simple depth map of the scene. Screen Space Ambient Occlusion (SSAO) is a well-known

(cc) BY-NC-ND



(a) 0^oN *Turb: 2.5* **(b)** 30^oN *Turb: 2.5* **(c)** 30^oN *Turb: 10*

Figure III.3: *Annual exposure maps (diffuse on top and direct at bottom with their respective scales) obtained using different configurations: latitude* 0^oN *turbidity 2.5 (left), latitude* 30^oN *turbidity 2.5 (middle) and latitude* 30^oN *turbidity 10 (right). Red arrows mark the solstice sun band.*

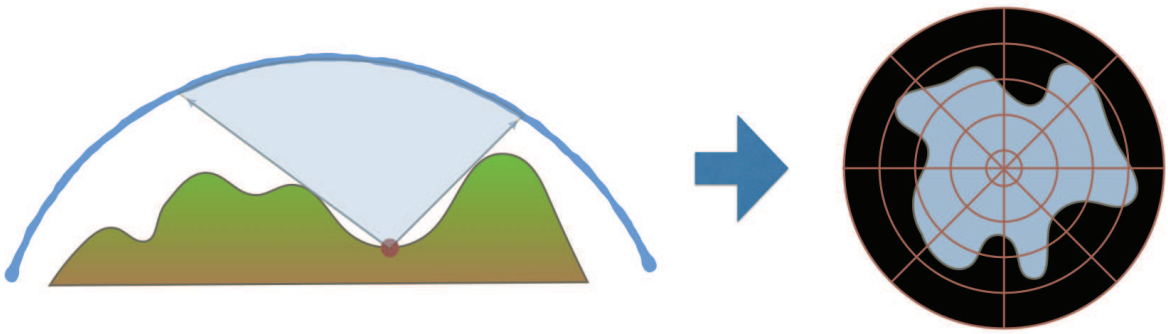


Figure III.4: *The SVF estimates the portion of visible sky from a point occluded by the surrounding geometry (left). It evaluates the visible rate of the sky hemisphere (right).*

(cc) BY-NC-ND

technique for computing visibility in screen-space, but it is commonly restricted to visibility from nearby surfaces that are directly observed from the camera's point-of-view. In our method, we evaluate occlusion using a similar approach but resort to a two-scale procedure that evaluates the sky visibility more precisely.

In the next subsections, we first introduce the screen-space ambient occlusion technique in which our work is inspired and, then, we present our approach and the two scales our technique consists of.

III.4.1 Image-Space Horizon-Based Ambient Occlusion

As mentioned before, due to the close proximity between SVF and AO, we propose handling the computation of SVF using an adapted version of screen-space AO. In their pioneering paper, Zhukov et al. [ZIK98] introduced ambient obscurance, replacing the visibility function with a distance-based falloff function. A large number of studies concerning both the approaches of obscurance and AO [MFS09] [McG+11] have since followed. After the intial screen-space approaches appeared [Mit07] [SA07], a number of new techniques improving performance and quality [BSD08] [RGS09] [BS09] were developed. We build our SVF estimation chiefly on the horizon-based ambient occlusion method presented by Bavoil et al. [BSD08].

Image-Space Horizon-Based Ambient Occlusion (ISHBAO) is a real-time algorithm proposed by Bavoil et al. [BSD08] to estimate AO in screen-space. It relies on approximating global accessibility by computing the horizon line of the normal-oriented hemisphere. They follow an approach that considers AO as the shadow contribution from an environment light and try to solve the following AO integral:

$$AO = 1 - \frac{1}{2\pi} \int_{\Omega} V(\omega_i) W(\omega_i)\, d\omega_i \tag{III.10}$$

where V is the visibility function over the normal-oriented unit hemisphere Ω and W is a linear attenuation function that considers the distance from surface to occluder.

In their technique, they propose to use the depth map generated by the camera as a continuous height field where they estimate occlusion. Following that, they asume that there exists a horizon angle below which any traced ray would be occluded. Thus, by finding this horizon angle the visibility test can be avoided. They propose to find it in image-space by sampling several directions around each pixel. Moreover, they parametrize the integral using spherical coordinates with the zenith axis aligned with the Z axis in eye-space (i.e., pointing to the camera), the azimuth angle θ around the Z axis and the elevation angle α. Following these considerations, Equation III.10 can be reformulated as:

$$AO = 1 - \frac{1}{2\pi} \int_{\theta=-\pi}^{\pi} \int_{\alpha=t(\theta)}^{h(\theta)} W(\omega_i) \cos(\alpha)\, d\alpha\, d\theta \tag{III.11}$$

where the tangent angle $t(\theta)$ is the signed elevation angle α of the surface tangent vector and the horizon angle $h(\theta)$ is the maximum elevation $\alpha \geq t(\theta)$ for which ω_i is occluded for all $\alpha < h(\theta)$.

Moreover, they propose to use a linear attenuation function $W(\theta) = \max(0, 1 - r(\theta)/R)$ where $r(\theta)$ is the distance from the current point to the occluding horizon point and R is the radius of influence. Then, they reformulate Equation III.11 to:

$$AO = 1 - \frac{1}{2\pi} \int_{\theta=-\pi}^{\pi} (\sin(h(\theta)) - \sin(t(\theta)))W(\theta)\, d\theta \tag{III.12}$$

(cc) BY-NC-ND

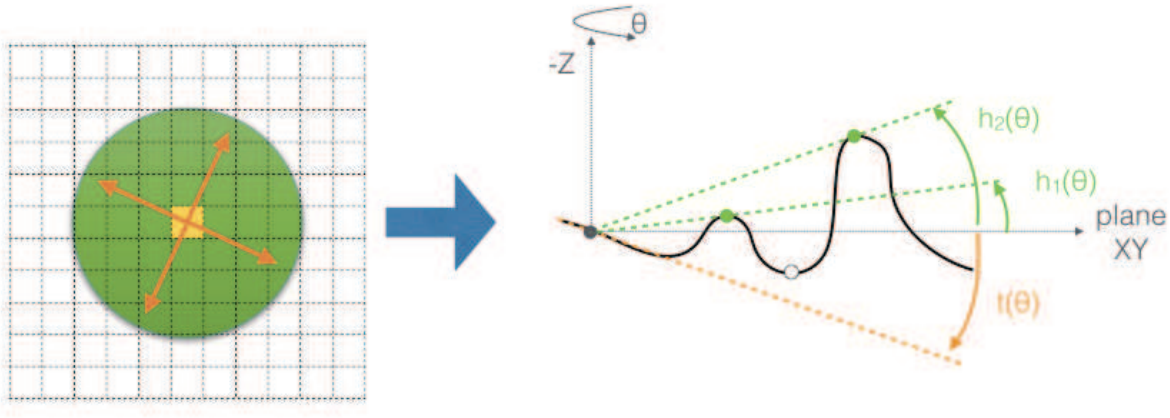


Figure III.5: *Overview of ISHBAO technique. For each pixel the neighboring area is sampled in several directions. For each direction θ several horizon angles $h_i(\theta)$ and a tangent angle $t(\theta)$ are found. These angles are then used to estimate the ambient occlusion term.*

In order to improve the quality of the results, instead of using a unique horizon angle they propose to find several horizon angles inside the radius of influence R and compute the influence of each of them approximating better the occluding geometry. More specifically, they propose to sample N_d directions θ in image-space around the current pixel. For each direction, they traverse the depth map sampling N_s horizon angles $h_j(\theta)$ and skipping those that do not increase the last accepted horizon angle or that are outside the radius of influence R. By doing this, they obtain a set of horizon angles where $h_j(\theta) \in [t(\theta), h(\theta)]$ and $t(\theta) = h_0(\theta) \leq h_1(\theta) \leq h_2(\theta) \leq ... \leq h_{N_s}(\theta) = h(\theta)$. This incremental approach yields the following computation of AO:

$$AO = 1 - \frac{1}{2\pi} \int_{\theta=-\pi}^{\pi} \sum_{j=1}^{N_s} W(\omega_{i,j})(\sin(h_j(\theta)) - \sin(h_{j-1}(\theta)))\, d\theta \tag{III.13}$$

Figure III.5 shows a visual overview of the ISHBAO technique. On the left it shows how the technique samples the neighboring area for each pixel, while on the right it depicts how it finds the horizon and tangent angles.

III.4.2 Image-Space Horizon-Based Sky Visibility

Screen-space methods typically consist of two stages: a rendering step that computes the depth map of the scene from the current camera position, followed by a second step that performs the computations using this depth map to reconstruct world-space positions of each pixel. Along with the depth, a normal map is often retrieved to determine surface orientation that can be used during the computations. In our solution an additional map is required. It takes into account the occlusion produced by the geometry that is outside the frustum of the camera. Figure III.6 overviews our approach. In a first step, we evaluate large-scale occlusion from the sky using the additional map that consists in a zenithal view of the city, which encompasses a larger portion of the scene. This information is then combined with standard ISHBAO from the camera's point-of-view to include fine-scale visibility.

(cc) BY-NC-ND

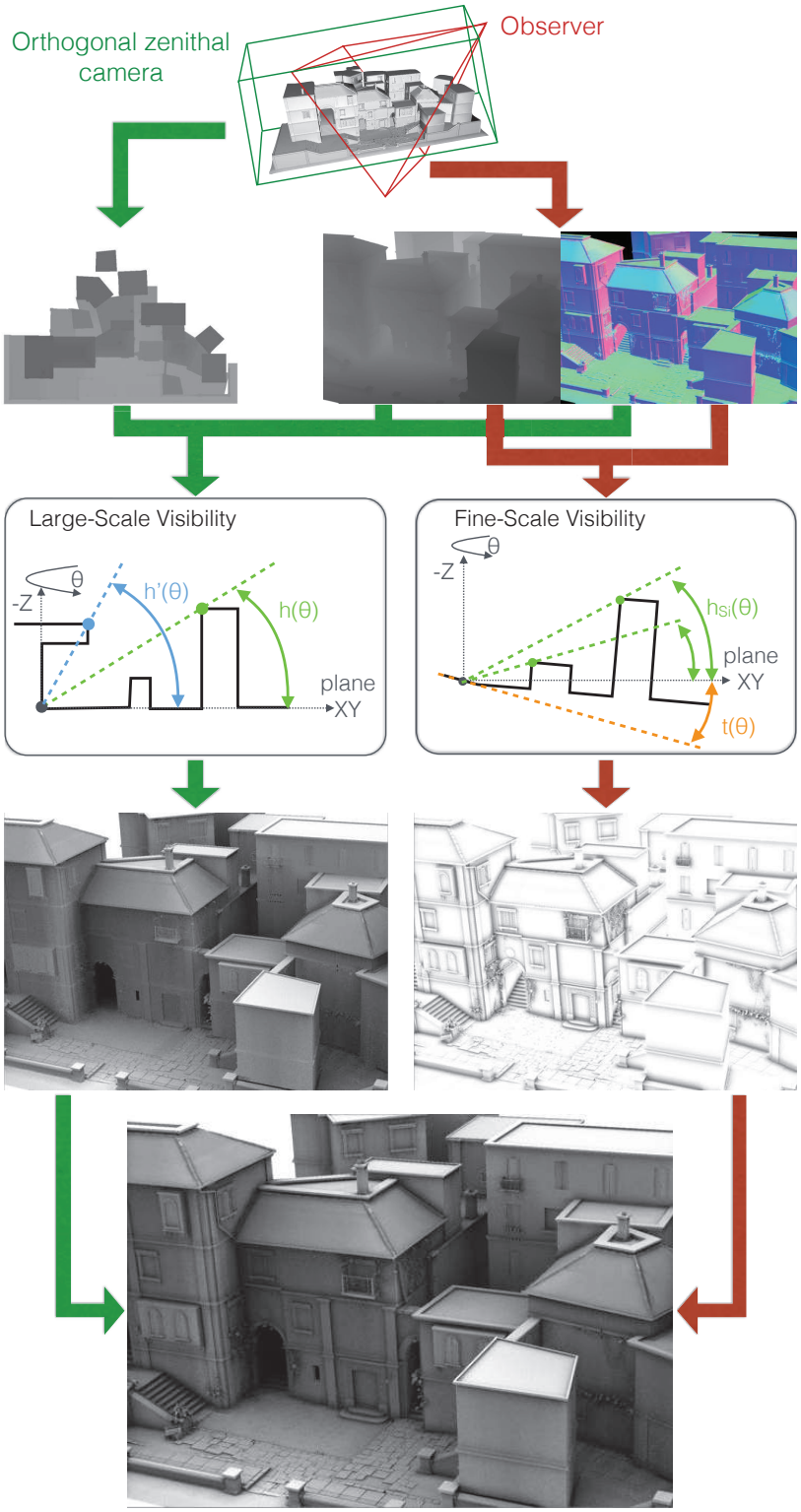


Figure III.6: *View-dependent sky visibility. Left: Large-scale visibility is computed in screen space using a zenithal orthographic view, accounting for non-heightfield features. Right: Fine-scale visibility is evaluated from the current camera position. Bottom: Both results are combined.*

(cc) BY-NC-ND

Large-scale Visibility

Relying on information recovered from the current view is usually enough for evaluating local visibility at every point, but for global visibility we need additional information. In our case, we include an additional view from the zenith using an orthogonal projection. This view is adjusted to encompass a large portion of the city model and its depth map is generated as a pre-process, independent of the current camera position. As it is view independent, this map must only be generated again if the geometry of the scene changes. To compute global visibility, this depth map is evaluated on-the-fly, but only for the points visible from the observers view. That is, we project each visible point P_m from the observers view to the zenithal view P_z and evaluate visibility there (see input maps in Figure III.6). Once the pixel position is recovered from the depth map in eye-space, the projection from observer's view to zenithal view is performed using the ModelView matrices $\mathbf{M}$ of both cameras:

$$P_z = \mathbf{M}_z \mathbf{M}_m^{-1} P_m \tag{III.14}$$

where P_m is the current point in eye-space, P_z is the point in the zenithal view space, M_m is the ModelView matrix of the observer camera and M_z is the ModelView matrix of the zenithal view camera. In this way, computations are only evaluated for visible points, thus avoiding the need to precompute and store them in the model.

Once at the zenithal view, we sample the corresponding depth map in several directions (N_{dir}), similar to ISHBAO. For each direction θ in image-space, we take multiple samples (S_{dir}) and we compute an elevation angle α at each of them (see Figure III.6). This angle is computed as:

$$\alpha = \frac{(P_z - S_{dir_i}).z}{|(P_z - S_{dir_i}).xy|} \tag{III.15}$$

The highest angle, also called horizon angle $h(\theta)$, represents the occluded portion of the environment for that particular direction. It can be defined as $max(0, \alpha(S_{dir_i}, i = 1..N_s))$. Furthermore, as it is performed in ISHBAO, we apply a random rotation to all directions at each point and a jittering of the samples to reduce aliasing artifacts.

Two common problems arise when using the previous approach. The first is due to sampling, which might lead to certain features being missed between samples during evaluation. Although this is a common problem in SSAO techniques, it becomes more important here due to the large portion of the scene that a small number of samples may cover. Artifacts produced by this problem are especially noticeable on corners of buildings. There, it tends to underestimate occlusion for directions opposing the surface normal. To solve this, we discard those directions based on their cosine angle.

$$h(\theta) = \begin{cases} 0, & \text{if } \theta \cdot \vec{n} < 0 \\ h(\theta), & \text{otherwise} \end{cases} \tag{III.16}$$

The other issue is that relying on a single depth map inherently limits us to height-field models as seen from the current (zenithal) view. This often results in facades or other elements being completely darkened due to salient roofs or balconies (see Figure III.7 left). Ideally, one should resort to multiple views or depth peeling (multiple depths) to handle those situations [RGS09]. However, this would be computationally expensive. Here, we rely on a simpler approach based

(cc) BY-NC-ND

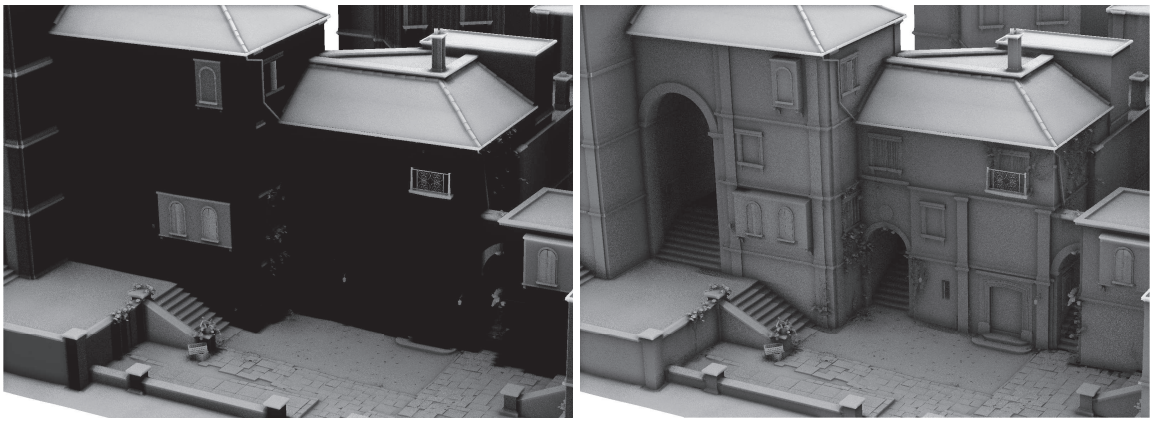


Figure III.7: *Visibility obtained using the opening angle $h'(\theta)$ (right) and without using it (left), computed using 8 directions (N_{dir}) and 8 samples per direction (S_{dir}).*

on computing a second elevation angle $h'(\theta)$ that we call the opening angle. For each sample direction, the opening angle is evaluated right before the first sample is found being lower than the current point (i.e., farther in depth). This term is combined with the subsequent elevation angle in order to obtain the final visible region for the current direction: $[h'(\theta), h(\theta)]$ (see Figure III.6, left diagram and Figure III.7 right).

Since we can not detect intercepting features causing complete occlusion, this approach might lead to underestimated occlusion. However, in practice it works well in most common situations, as shown in Figure III.8 in the comparison with ground truth generated in a 7 hour average computation using ray casting. Additionally, a user-provided threshold can be employed to discard openings far from the current point.

Note that the Sky View Factor for a specific direction can be obtained using:

$$GVF(\theta) = \sin h'(\theta) - \sin h(\theta). \tag{III.17}$$

The average of this term over all θ directions would then give us the final GVF term:

$$GVF = \frac{1}{N_{dir}} \sum_{i=1}^{N_{dir}} \sin h'(\theta_i) - \sin h(\theta_i) \tag{III.18}$$

For isotropic sky models, this term can simply be used as a weighting factor of the uniform diffuse exposure. But, for anisotropic models, we rather need to use the specific visible regions to explicitly sample and weight the exposure map, as detailed in Section III.5.

Fine-scale Visibility

The occlusion estimated using the method above, lacks enough detail for an appropriate estimation. This is mainly due to the resolution used for the zenithal depth map and the differences in viewing angles. To incorporate fine-scale visibility, we resort to ISHBAO computed from the camera view, which perfectly suits our needs in this case.

Similar to our large-scale approach, we sample the depth map from the current view in different directions, each time computing the horizon angle with respect to the viewing plane (XY in

(cc) BY-NC-ND

Figure III.6). For each direction, the samples here are combined using a distance function $W(s_i)$ as explained in Section III.4.1. Moreover, we use a radius of influence R, also mentioned in Section III.4.1, to discard samples of neighboring pixels that after being recovered its position this is far away from the current position. This culling process is needed in the local scale to avoid over-oclusion of non-captured discontinuities in the depth map. But, it may not be used in the large-scale because it will not improve possible discontinuities in the height field such as bridges. Moreover, the tangent plane at the current point defines a second angle $t(\theta)$ from which the actual horizon angle should be defined (like in ISHBAO). The local visibility term for a specific direction is then computed as:

$$LVF(\theta) = \sum_{i=1}^{S_{dir}}((\sin h_{s_i}(\theta) - \sin t(\theta)) \cdot W(S_i)). \tag{III.19}$$

As before, the final visibility term is obtained by averaging $LVF(\theta)$ over all θ directions:

$$LVF = \frac{1}{N_{dir}} \sum_{i=1}^{N_{dir}} \sum_{j=1}^{S_{dir}}((\sin h_{s_j}(\theta_i) - \sin t(\theta_i)) \cdot W(S_j)). \tag{III.20}$$

The combination of the fine-scale occlusion term with the large-scale term should, ideally, be done by merging their visible regions, which would require handling their differences in sampled directions. A much simpler approach to compute the final sky visiblitiy (SVF) consists of weighting the factor computed by both scales:

$$SVF = GVF \cdot (1 - LVF) \tag{III.21}$$

which, in practice, proved to be adequate (see Figure III.8).

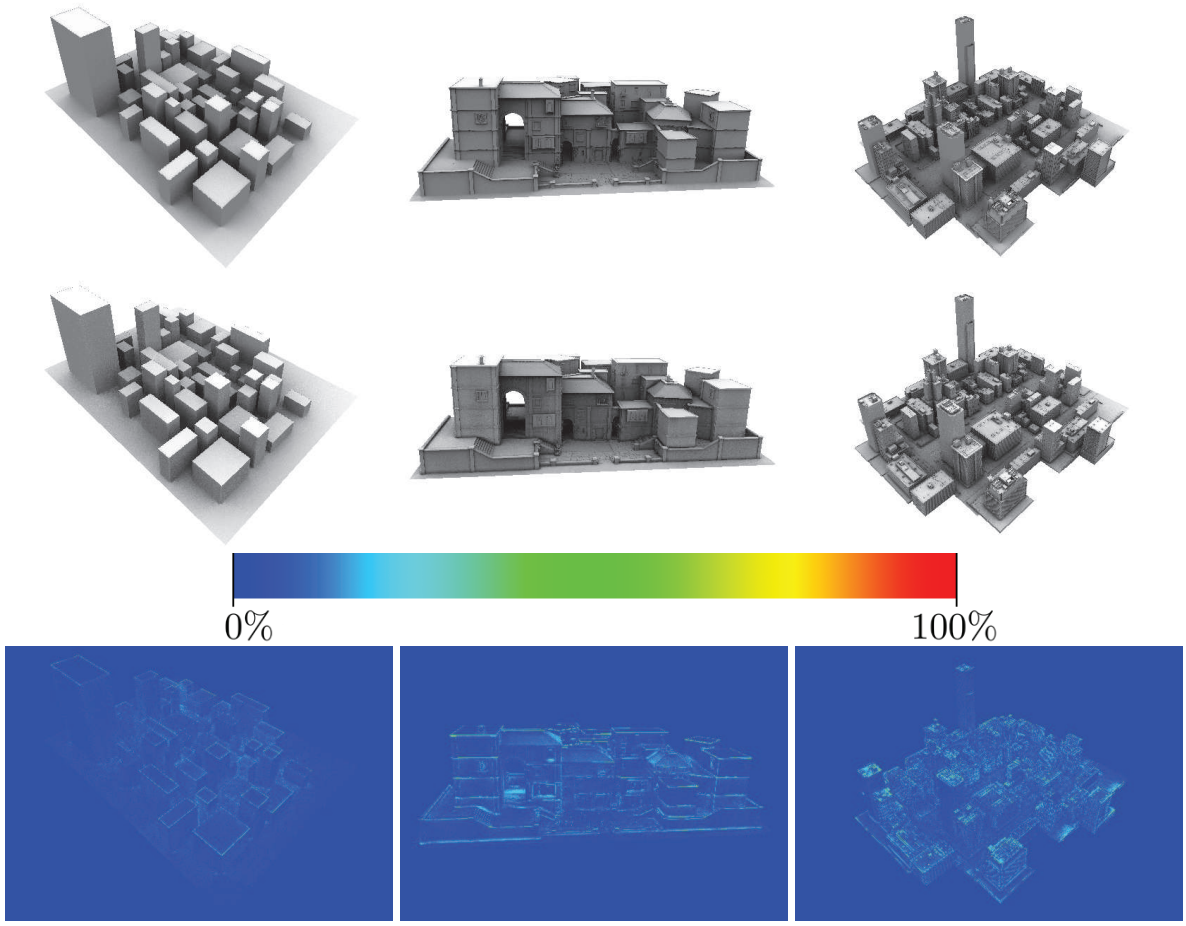
III.5 Surface Solar Exposure

After evaluating the visibility, we are ready to compute the final solar exposure at each surface point. Instead of using directly the computed SVF, we propose to restrict the exposure to the sky visible regions. Thus, we propose to use the large-scale visibility step which gives us the visible regions of the exposure map that will contribute to this final term. Inside these regions, we take a set of samples to recover the corresponding values, and weight their contribution according to the cosine term based on the current normal (see Figure III.9). To avoid using a lot of samples, we resort to prefiltering the map by means of mipmapping, as we detail next.

It is important to note that the cosine term cannot be baked into the map. It changes for each surface point and direction and, thus, it should be computed at each sampling direction. However, thanks to mipmapping prefiltering, we perform only one sample for several directions. So, we consider this term constant within the corresponding region of each mipmap sample. Fortunately, the smooth condition of the prefiltering strategy minimize the error introduced. This problem would also affect the alternatives. Using other pre-integration approaches, such as summed-area tables, would still require an appropriate cosine weighting within subregions.

Recall, from Large-Scale Visibility in Section III.4, that a visible region over a specific direction is defined by a pair of angles $[h'(\theta), h(\theta)]$. We propose to consider these angles as the limits of the

(cc) BY-NC-ND



(a) *City Blocks* **(b)** *Mediterranean* **(c)** *Urban Sprawl*

Figure III.8: *Sky visibility comparison between ground truth (top) generated in several hours and our method (centre) rendered in less than 15ms, including difference images (bottom). Ground truth is based on ray casting using 128 samples per point. Our method uses 16 directions (N_{dir}) and 16 samples per direction (S_{dir}). Difference images show the difference of visibility percentage. The City Blocks model uses our large-scale visibility term (centre left), while the Mediterranean and Urban Sprawl models use our combined visibility (centre middle and right).*

(cc) BY-NC-ND

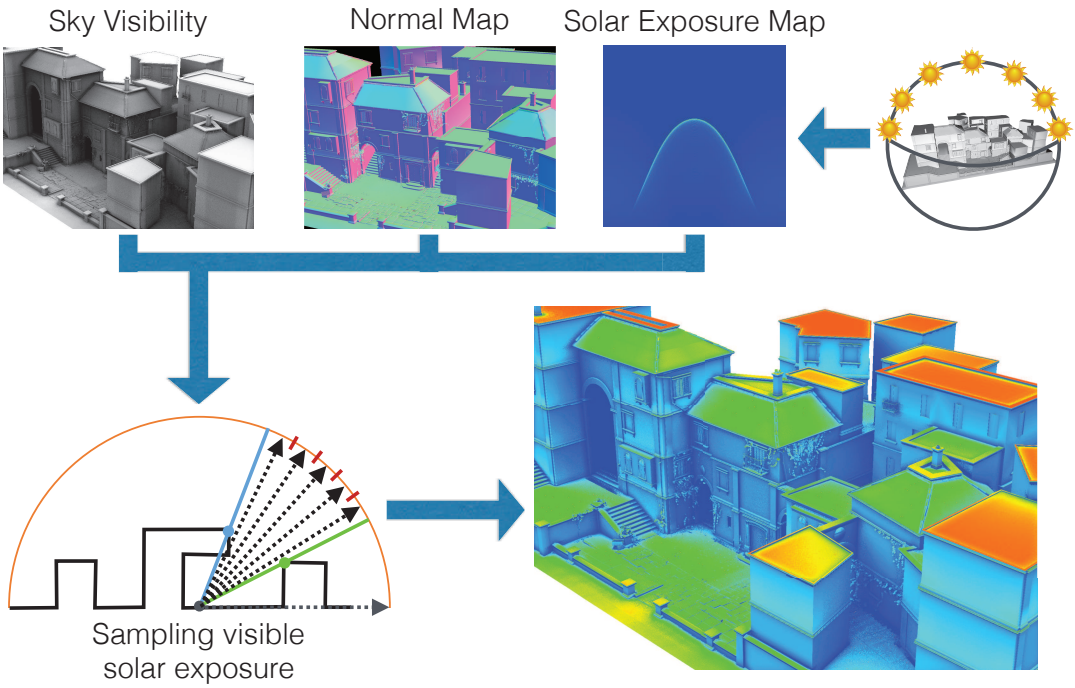


Figure III.9: *Computing surface solar exposure. The input exposure map is sampled within the visible regions obtained at each point. The normal map provides the surface orientation for the final weighting with the cosine term.*

sampling domain in a specific direction. Within this range, we take uniform samples along the current direction. We adapt the number of samples according to:

$$n_s = \max(\frac{2 * (h'(\theta) - h(\theta)) * t_{height} * N_{dir}}{\pi * t_{width}}, 1). \tag{III.22}$$

Here, t_{height} and t_{width} are the dimensions of the exposure map and N_{dir} is the number of visibility directions (see vertical slices in Figure III.10). According to this, n_s is proportional to N_{dir} and it depends on both the visible region and the aspect ratio of the exposure map.

Our sampling strategy consists of uniformly sampling the visible region corresponding to each direction and appropriately fitting the samples within this region (see Figure III.10). This strategy guarantees a higher number of samples in the directions with a bigger visible region. For the last sample, we then adjust its position and size according to the remaining portion. Compared to a classical overlapping of samples, this approach does not overestimate exposure. For a full coverage of the visible region, additional samples could always be taken into the remaining portion.

In order to select the appropriate level b within the mipmap, we take into account the limits of the current sample in elevation and the height of the texture:

$$p_{height} = \frac{2 * (h_{sup}(\theta) - h_{inf}(\theta)) * t_{height}}{\pi} \tag{III.23}$$

$$b = max(0, \log_2(p_{height})), \tag{III.24}$$

(cc) BY-NC-ND

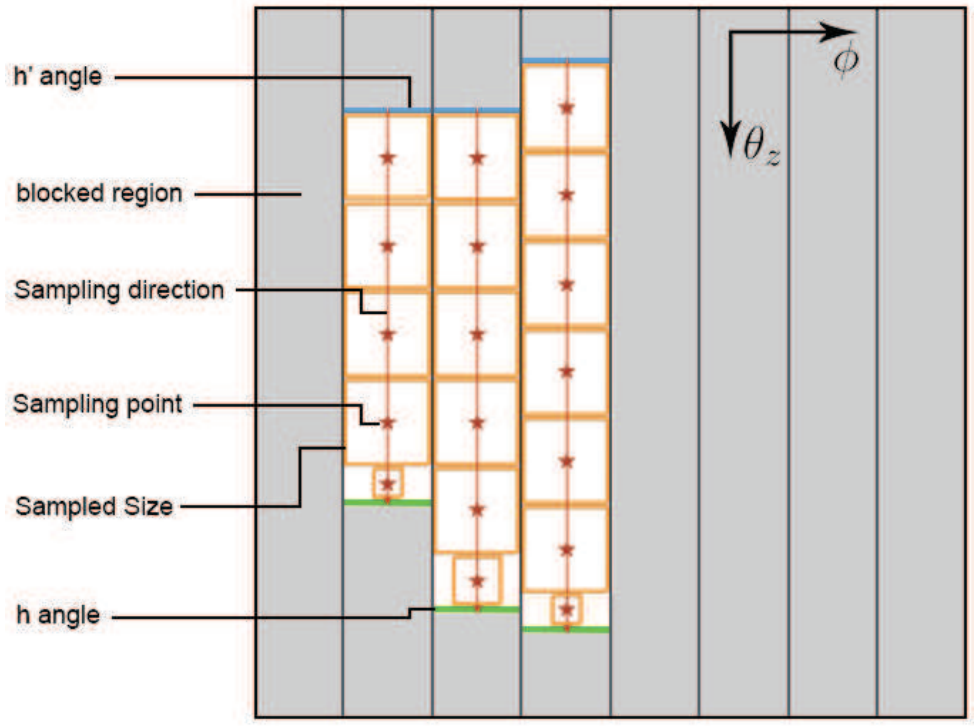


Figure III.10: *Sampling of the solar exposure map. We take samples along each visibility direction, adapting the number of samples to the visible region (red stars). We resort to mipmap pre-filtering to retrieve the exposure of each sample, whose size is fitted to the visible region (orange boxes).*

(cc) BY-NC-ND

where $h_{sup}(\theta)$ and $h_{inf}(\theta)$ define the limits of the current sample in elevation (see Figure III.10), and p_{height} represents the height of the sample in pixels.

After retrieving the values from the mipmapped texture, they represent averaged values. Hence, to obtain the total exposure, we subsequently weight them according to the number of pixels (n_p) subtended by the mipmap patch:

$$n_p = {p_{height}}^2 \tag{III.25}$$

The value obtained from each sample is finally weighted by the cosine term and accumulated to obtain the final exposure of the surface point. Figure III.10 shows an example of the sampling performed over three visible regions obtained from a surface point. Note how the number of samples and their size is adapted to each region.

At this point, we are only considering the large-scale visibility regions in the solar exposure estimation. Although the ideal solution is to combine visible regions of both scales and sample the exposure within it, we opt for a simple combination that proved to be enough for including geometry details in our solar exposure estimations, as stated before. We propose to simply weight the estimated exposure with the fine-scale visibility factor LVF:

$$SolarExposure = (1 - LVF) \sum_{i=1}^{N_{dir}} \sum_{j=1}^{n_s(\theta_i)} H_{filt}(\omega_j(\theta_i), b_j(\theta_i)) n_{p,j} \max(0, \omega_j(\theta_i) \cdot \vec{n}) \tag{III.26}$$

where H_{filt} is a filtered value of the exposure map for direction ω_j and mipmap level b_j, $\omega_j(\theta_i)$ corresponds to the j-th in direction θ_i and $\vec{n}$ is the normal of the current pixel.

III.6 Results and Discussion

III.6.1 Results

We have tested our approach on three different models: one simple heightfield model (*City Blocks*) and two more complex city models (*Mediterranean* and *Urban Sprawl*). Table III.1 shows the performance we obtained for an unoptimized code on a MacBook Pro 15" with an Intel Core i7 2.6GHz, 16GB memory and an NVIDIA GeForce GT 750M with 2GB memory. It is noticeable that even when using a considerable number of samples, our model still exhibits interactive performance. Also, thanks to our screen-space approach, the performance does not change considerably with scene complexity. The timings for the pre-computation step, generating a one year integration exposure map with an unoptimized code, are 5 seconds for the direct beam part and over 3,700 seconds for the diffuse part. The extra memory costs of our method, with respect to traditional SSAO methods, are a float texture of 512×512 pixels for the exposure map, which takes up to 1 MBytes, and a 32-bit texture of 800×600 pixels for the zenithal depth map with a cost of 1.92 MBytes. In order to display our results, the visibility shown in the figures represents the Image-Space Horizon-Based Sky Visibility SVF. Moreover, all the exposure maps used in the results presented here account for a one year integration period and the exposure value represented by the corresponding color map.

Figure III.11 shows the final exposure computed on the *Urban Sprawl* and *Mediterranean* models for different latitudes and turbidities. At latitude 0^oN (top images), variations on turbidity are less noticeable. Only some small differences can be observed in regions with less direct beam impact (mainly facades oriented less towards the sun path). At latitudes 30^oN and 60^oN (bottom

(cc) BY-NC-ND

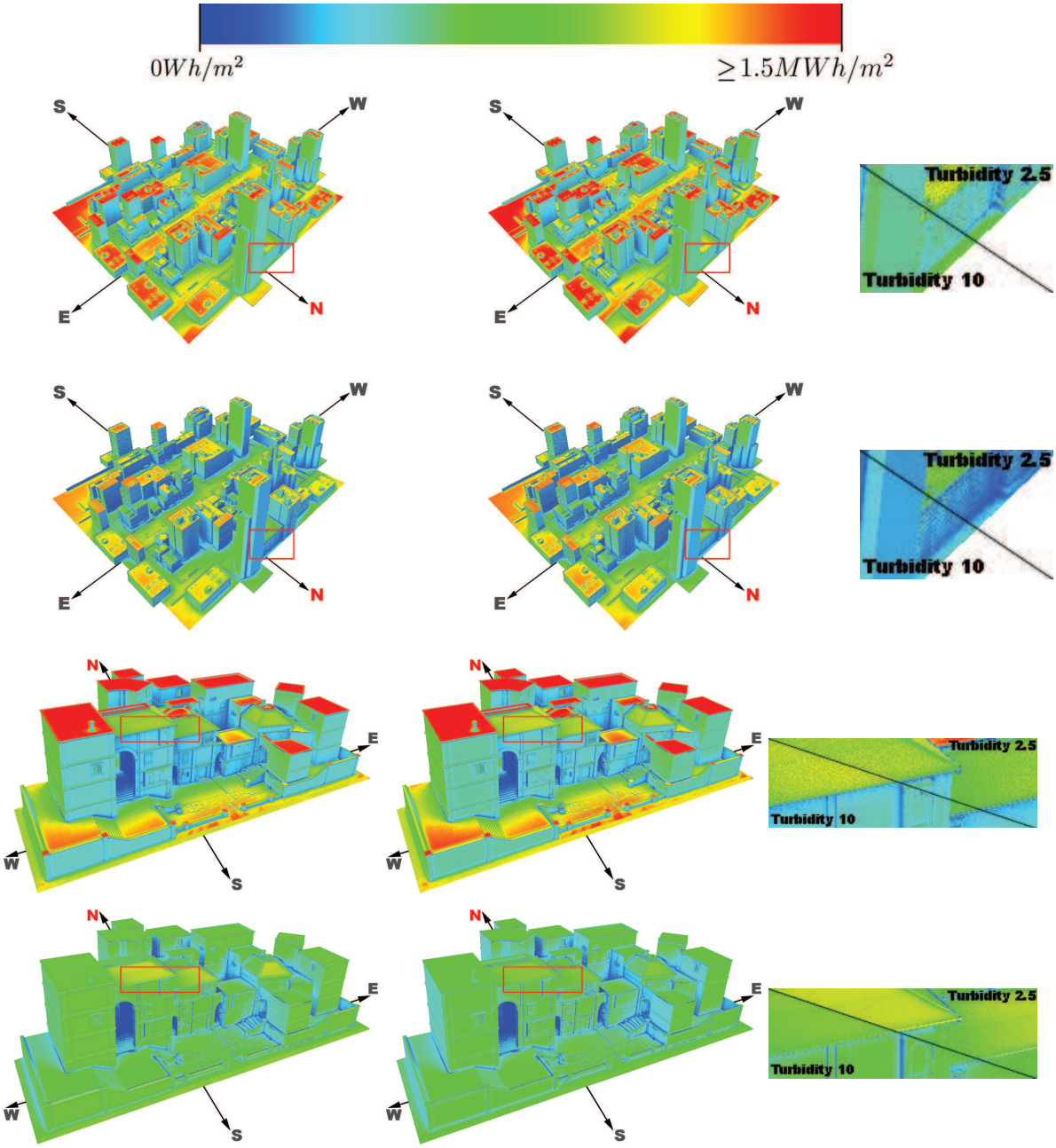


Figure III.11: *Solar exposure computed in the Urban Sprawl (top) and Mediterranean (bottom) models using 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples for the exposure map (n_s). Latitudes on each row: 0^oN (up) and 30^oN (down) for Urban Sprawl and 0^oN (up) and 60^oN (down) for Mediterranean. Turbidities on each column: 2.5 (left) and 10 (center) for each model. Right column shows close-ups for the red boxes.*

(cc) BY-NC-ND

Model	Size	Samples (worst case)	Large Scale Visibility (fps)	Large+Fine Scale Visibility (fps)	Solar exposure (fps)
City Blocks	4K Faces	8 $\times$ (8+8)	60-39	60-25	45-12
City Blocks	4K Faces	16 $\times$ (16+16)	42-15	33-10	25-4
Mediterranean	604K Faces	8 $\times$ (8+8)	60-34	60-21	37-11
Mediterranean	604K Faces	16 $\times$ (16+16)	34-12	30-9	15-5
Urban Sprawl	777K Faces	8 $\times$ (8+8)	57-27	50-18	36-10
Urban Sprawl	777K Faces	16 $\times$ (16+16)	27-10	24-8	16-4

Table III.1: *Performance of our technique for different models. The* Samples *column specifies the number of visibility directions [N_{dir}] $\times$ (samples per direction [S_{dir}] + maximum solar exposure samples [n_s]). Note that for a square solar exposure texture, as used in the examples, the latter equals to N_{dir} in the worst case (full visibility). Timings in fps were obtained with $60Hz$ vsync enabled on an 800×600 viewport. The range of values means performance obtained for far and close views, respectively.*

images), as the exposure associated to the orientation changes regarding to the sun path, it becomes more noticeable (again on facades and pitched roofs). One can see how regions less affected by the direct beam increase their exposure when turbidity is increased (see facades north oriented at 30^oN) thanks to an increase in the diffuse component. Similarly, the exposure is reduced in regions affected by the direct beam (see roofs at 60^oN).

While variations on turbidity are intuitive (decreasing on direct and increasing on diffuse components), the influence of the latitude (for a same polar orientation) is clearly noticeable and proves to be very important. The combination of both a high latitude and a high turbidity leads to a low exposure (bottom-right results of *Mediterranean* model in Figure III.11 for example). Also, the behavior of common urban features, such as porches, is well handled.

Figure III.12 shows different views of the visibility and solar exposure computed on the models *Mediterranean* and *Urban Sprawl* for a fixed latitude and turbidity. We can see that visibility and exposure tend to be correlated in some situations (like for some orientations in latitudes near to 0^o), but not for the rest. In particular, this lack of correlation is very noticeable on some planar horizontal surfaces or on vertical walls oriented differently. This is caused by the anisotropic behavior of radiation, which not only depends in the amount of visible sky but also in the incoming radiation present in that part.

Moreover, the same Figure III.12 shows the influence of the sampling configuration used for computing sky visibility and solar exposure. One can notice that even with a low quality configuration, although our results present more noise, our technique is still capable of accounting for small geometric features such as vegetation that grows on the facade (see close-up in the bottom left part of Figure III.12). However, for smoother results the high-quality configuration proves to perform better, increasing slightly the computational time.

In Figure III.8 and Figure III.13 we compare our technique against a ground truth solution based on ray casting. Figure III.8 compares the visibility term, while Figure III.13 compares the final solar exposure. In the latter, the ray casting approach directly samples the sky model without resorting to our precomputed map. Main differences between the two methods are mostly due

(cc) BY-NC-ND

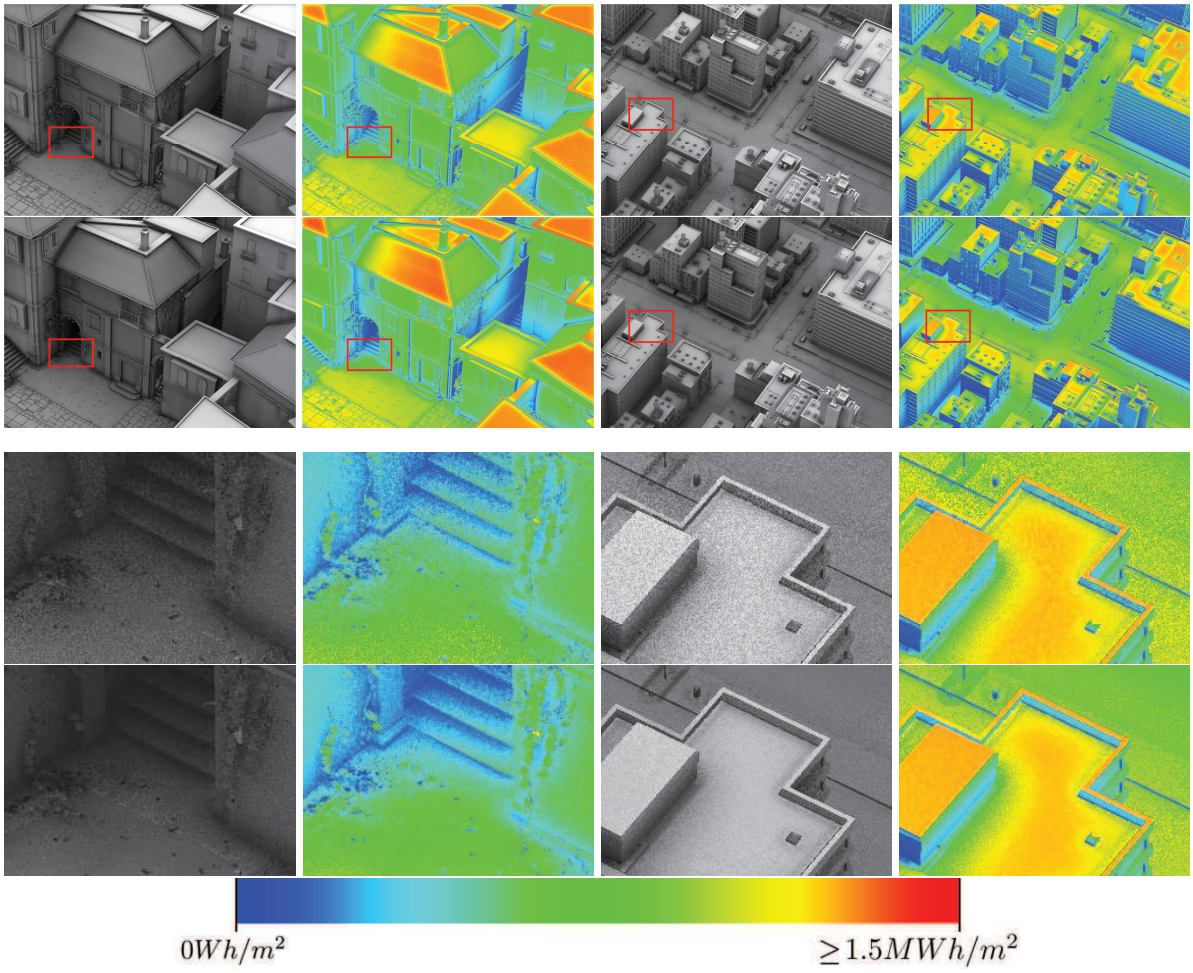


Figure III.12: *Visibility and solar exposure (30^oN latitude and a 2.5 turbidity) for the Mediterranean (left) and Urban Sprawl (right) models, with their respective close-up's below, using different parameters: top, 8 directions (N_{dir}), 8 samples per direction (S_{dir}) and a maximum of 8 samples in the exposure map (n_s); bottom, 16 directions(N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples in the exposure map (n_s).*

(cc) BY-NC-ND

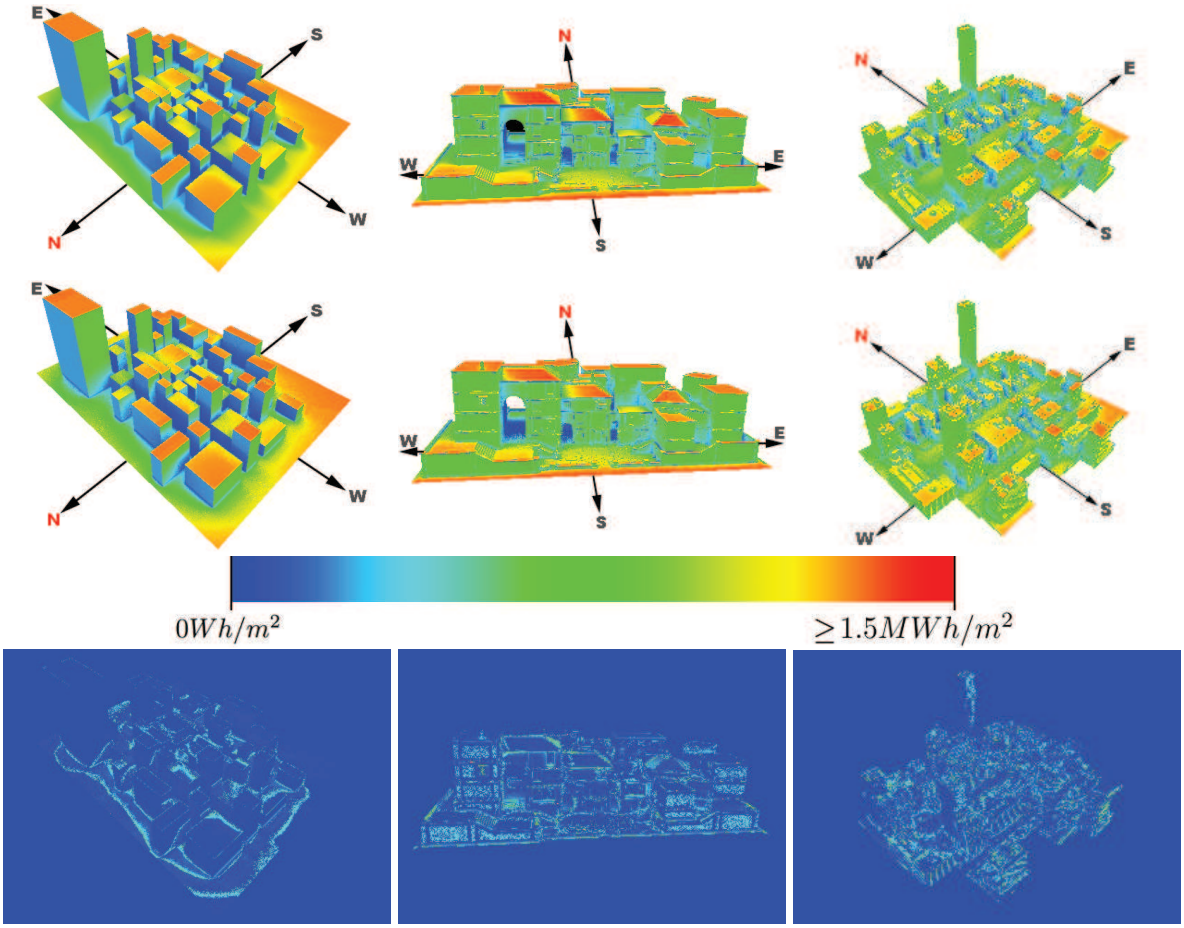


Figure III.13: *Solar exposure comparison between ground truth (top) generated in 50 hours and our method (center) rendered in less than 30ms, including difference image (bottom). Ground truth is based on ray casting, which samples the sky each hour using 1 ray for direct irradiance and 128 for indirect irradiance per pixel. Our method uses the precomputed map and samples visibility using 16 directions (N_{dir}) and 16 samples per direction (S_{dir}).*

(cc) BY-NC-ND

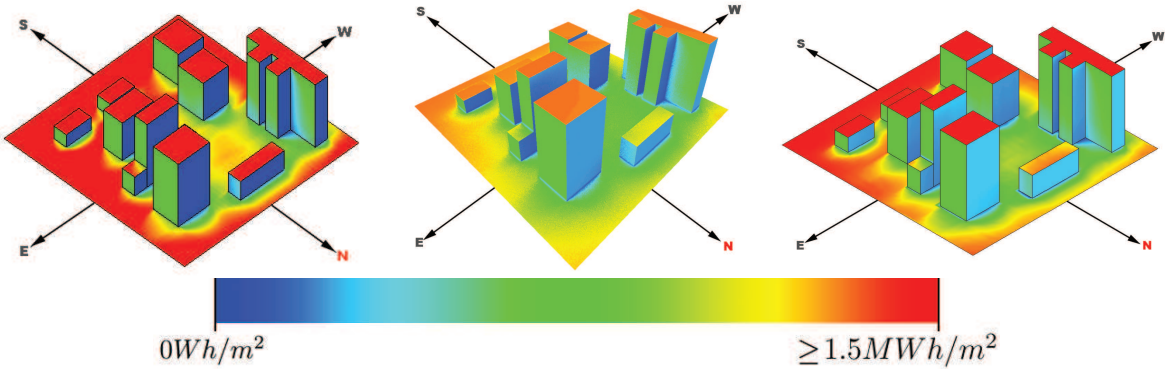


Figure III.14: *Comparison between existing techniques and our method at 30^oN latitude and a 2.5 turbidity. Left: Heliodon image generated in 345 sec., sampling every 15 minutes with 1 meter resolution; Right: image generated with Autodesk Revit 2017 and Autodesk Insight 360 tool in 15 sec., sampling every hour with 0.45 meter resolution; Center: our solution generated in 0.07 sec. using per-pixel resolution (average in this view: 0.2 meters) using 16 directions (N_{dir}), 16 samples per direction (S_{dir}) and a maximum of 16 samples in the exposure map (n_s).*

to slight differences in visibility angles, which cause the exposure to slightly shift over surfaces (see Figure III.13 left), or slightly differ on facades with salient elements such as roofs, balconies or porches (see Figure III.13 middle and right). These differences have different causes. First, some differences are inherent to screen-space techniques, like the height field assumption and the resolution of the depth map that can imply skipping geometry elements. Second, our sampling and filtering strategy tends to reduce high frequency results which fortunately are not very common in exposure results. Finally, our approximation of the opening angle involves the introduction of errors in visibility estimation that is carried in the exposure estimation. Our method, however, provides a good match compared to ground truth visibility and exposure overall, achieving such results in real time.

Finally, Figure III.14 shows a comparison between existing tools and our method, specifically comparing our solution against *Heliodon2* and *Autodesk Revit 2017* with Insight 360 plugin. We have used a simple model, similar to *City Blocks*, due to the limitation to load complex geometry into *Heliodon2*. Both tools compute the solar exposure in several points of the geometry, they store it in memory and, finally, interpolate between these sampling points. The main difference between them is the computational model. On the one hand, *Heliodon* uses its own algorithm to compute the direct beam based on ray-tracing, while the diffuse component is approximated using an isotropic atmospheric transmittance value weighted by the SVF, the latter being computed using a projection of the neighbor buildings. On the other hand, *Autodesk Revit 2017* uses a ray-casting solution with the Preetham model, combined with measured information of the weather conditions in near locations. Notice how the obtained results differ between the three tools as they rely on different sky models, but the patterns of solar exposure distribution fit well in all cases. In comparison, a clear advantage of our technique is that it requires much less computing time.

(cc) BY-NC-ND

III.6.2 Discussion

Our screen-space approach for estimating visibility provides interactive results with a good match compared to ray-casted visibility, as presented in Figure III.8. The large-scale visibility step is able to handle non-height field configurations, although it does have limitations with estimated opening angle accuracy. Features not captured by the zenithal depth map will not be detected by this stage, although the fine-scale visibility step based on the observer's depth map tends to compensate for this. When those features are not captured by either view, our technique is not able to consider them. A possible solution would be to provide additional views or resort to depth peeling [RGS09].

The combination of the visibility information from the two scales based on a final weighting is fast and provides reasonable results in most situations. Ideally, as already mentioned, the visible regions from the two scales should be intersected on a common frame and used during the computation of the exposure. Despite the required computational effort, such a strategy should compensate for over-occluded places.

As for surface exposure, our pre-filtering strategy requires a few sets of samples to be taken for each direction. Moreover, they are automatically fitted according to the visible regions. Our technique provides interactive results with a good comparison with respect to a ground truth generated in a 40 hour average using ray-casting, as shown in Figure III.13. Moreover, Figure III.14 shows how our technique obtains good results in much shorter computation times when compared to existing solutions: we achieve two orders of magnitude speed-up with respect to *Autodesk Revit* and three orders of magnitude speed-up with respect to *Heliodon*. In addition, we can handle dynamic scenes as shown in Figure III.15. This option is very useful in urban planning and solar engineering in order to evaluate specific configurations of buildings or elements. An example of the relevance of the proposed technique is in the design and location of solar panels or other solar devices either for capturing and transforming solar energy or to light indoor environments with daylight. Another strategy, which could improve the results and reduce the computation time even further, would be to perform an adapted version of importance sampling, along the lines of Colbert and Krivanek's work [CK07], by considering not only visibility, but also the cosine and the directional exposure terms in order to guide the sampling.

Our pre-computed exposure map relies on the Hosek and Wilkie's model to capture different sky conditions which, similar to Preetham et al., is restricted to continuous conditions of clear or, at most, overcast skies. Yearly conditions can be adjusted through the turbidity parameter to approximate varying conditions of cloudiness over time. For partially cloudy skies, however, one should preferably rely on measured data. Our generation is easily adaptable to use measured data whether directional radiation measures are available. But, while directional information of the direct component is very common in solar radiation databases, it is rather hard to find directional measurements of the diffuse component.

Last but not least, if changes in sky conditions or scene latitude are needed, one might pre-compute several maps to be interchanged at run-time. A better solution would be to develop specific analytical models able to capture the integrated temporal behavior or, at least, to approximate the diffuse component given its lower contribution, in order to easily evaluate them interactively.

(cc) BY-NC-ND

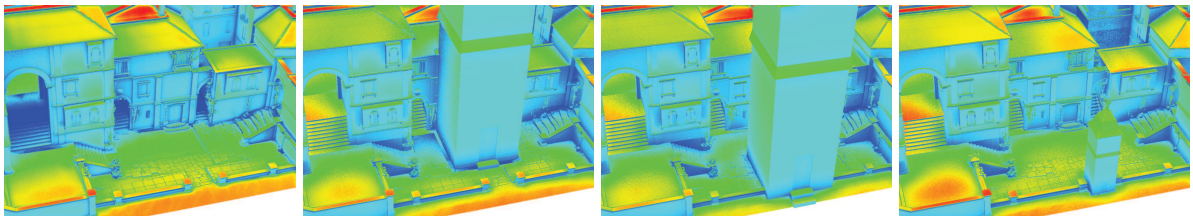


Figure III.15: *Frames extracted from an animation (a block moves through the scene) demonstrating how our technique handles dynamic scenes with interactive solar exposure.*

III.7 Conclusions

In this chapter, we have presented an interactive method, based on sky maps integrated over arbitrary periods of time and under different sky conditions, to compute and evaluate solar exposure on complex urban scenes. Our system, based on a screen-space approach, provides interactive feedback and is thus amenable to dynamic modifications of the scene. We have demonstrated that our two-scale fast visibility estimation, besides its limitations, is fast and compares well against ray-casting. Moreover, we have demonstrated that our pre-filtering strategy is able to reduce considerably the number of needed samples and it reproduces high-quality results that compare well against ray-casting. Furthermore, we have proved how our technique speeds up the computational time against commercial software in 2-3 orders of magnitude while it obtains a comparable radiation distribution. Finally, we believe our technique can be useful in various scientific areas such as architecture and energy efficiency, in particular for interactive preview purposes prior to more accurate and costly simulations for final planning decisions. Moreover, due to its accuracy and the low computational cost, we consider our technique to be suitable as a realistic estimator of the solar exposure factor for weathering systems.

(cc) BY-NC-ND

(cc) BY-NC-ND

Chapter IV. Pollution

IV.1 Introduction

As mentioned before, weathering effects are ubiquitous phenomena in modern cities, where buildings age and deteriorate over time because of their interactions with the environment. In Computer Graphics, relevant work has been done to produce impressive images of virtual urban environments including weathering effects. Some of the approaches introduced in Chapter II reproduce realistic images considering local weathering effects [Che+05][DRS08][MG08][Bos+11], where they apply the results to different predetermined parts of a building to confer it a weathered aspect. But, so far no technique has been proposed to deal with weathering effects using a global approach. In other words, there is a lack of techniques that deal with urban weathering estimating first the weathering factors and, then, reproducing the effects using a holistic model.

In this chapter, we use this principle by focusing on the accumulation of pollution on buildings facades. Pollution is a very visible weathering effect in modern cities, specially concerning older buildings. Its accumulation on facades is very characteristic, and in some cases it provides important visual cues about the building itself, like its age. Although the accumulation of pollution could be studied locally, the distribution of pollutants inside a city is never an isolated, facade-by-facade effect, but it is a more general phenomenon that affects whole neighborhoods. Pollution weathering is mainly due to the transport of pollution agents (chemical elements, particles, ...) from pollution sources to the surface of the buildings. These main sources are chimneys (industries, houses and buildings) and traffic jam contamination, that supply pollutants to the environment. These pollutants are distributed through the city thanks to the action of wind and rain. During this distribution process, pollutants interact with building surfaces until they are deposited on them.

As far as we know, no technique has been proposed to deal with this aging process on an urban environment as a whole, taking into account the delicate inter-relations between all these elements at a large scale. Here we propose a technique that focuses on modeling changes in appearance due to pollution soiling at a urban scale, based on a fast but physically-inspired approach. We consider pollution effects to depend on three main factors: wind, rain and sun exposure, and we take into account its three intervening steps: deposition, reaction and washing. This is combined with a low-cost pre-computation step that evaluates the propagation of pollution through the city. The global character of the proposed technique is provided by taking into account how the characteristics of the urban environment affect the incidence of

(cc) BY-NC-ND

natural phenomena on the surfaces. In addition to well known visibility techniques and insolation computation, we deal with wind propagation and pollution particles deposition thanks to a fast fluid dynamics proposed approach. We base our solution on the behavior described by physical models but fast enough to offer results at interactive rates.

In summary, our contribution is a low cost, physically plausible pollution distribution simulator and deposition model, which allows interactive computations of weathering effects on a large-scale urban level, including building self-interactions. Our interactive technique is well suited for a wide range of applications, ranging from urban simulations and refurbishing planning to entertainment applications (video games, virtual scenarios, ...). The main contributions of our framework can be summarized as follows:

- We provide, for the first time, a framework to deal with the effect of weathering agents that act globally and integrates them to produce local effects.
- Based on our model and the use of screen-space operators, our method results in an efficient approach able to generate realistic images of urban scenes combining the intervening factors at interactive rates.
- Our pre-computation demands a reduced amount of memory to store a preliminar pollution map and it can suit large and complex models by means of adapting its resolution.
- We propose solutions for two main problems: the high cost of fluid dynamics simulations and the lack of models for both the transport of pollution at a city scale and for the deposition of particles on surfaces.
- Our framework can be easily adapted to model several aging processes and to improve other local aging techniques.

The remainder of this chapter is organized as follows; in the next section, we present an overview of our pipeline. Section IV.3 concentrates on our pre-computation stage. First, we describe our fast fluid dynamics simulation and, then, we present how we model and store the global transport of pollutants over the whole scene. In Section IV.4, we evaluate the weathering factors to determine their effect in the deposition process. Then, in Section IV.5, we present the deposition model built from the pre-computed pollution map and the evaluation of the intervening factors and we present our rendering formulation that produces the final changes in appearance. Finally, we present our results and discussion.

IV.2 Overview

Fundamentals of air pollution principles that lead to black crust formation are summarized in Figure IV.1. In urban environments, the concentration is high in black carbon particles and sulfurous gases, coming from vehicle exhausts or heavy industries. This concentration is however not uniformly distributed because wind propagates particles and gases from their sources and rain makes environmental pollution to fall inside the scene (see Figure IV.1 left). Air pollution is deposited onto surfaces, where it reacts with construction materials by creating a gypsum grid that traps dust and particles, leading to the formation of a black crust. This chemical reaction is enhanced under high humidity conditions (see Figure IV.1 middle). Humidity presence is related to water depositions not dried by wind or solar radiation. After black crust formation, affected surfaces are also washed when directly exposed to strong rainfalls or by means of the action of long-term wind erosion (see Figure IV.1 right). The competition between these behaviors as well as inner material properties induce a strong spatial variability.

(cc) BY-NC-ND

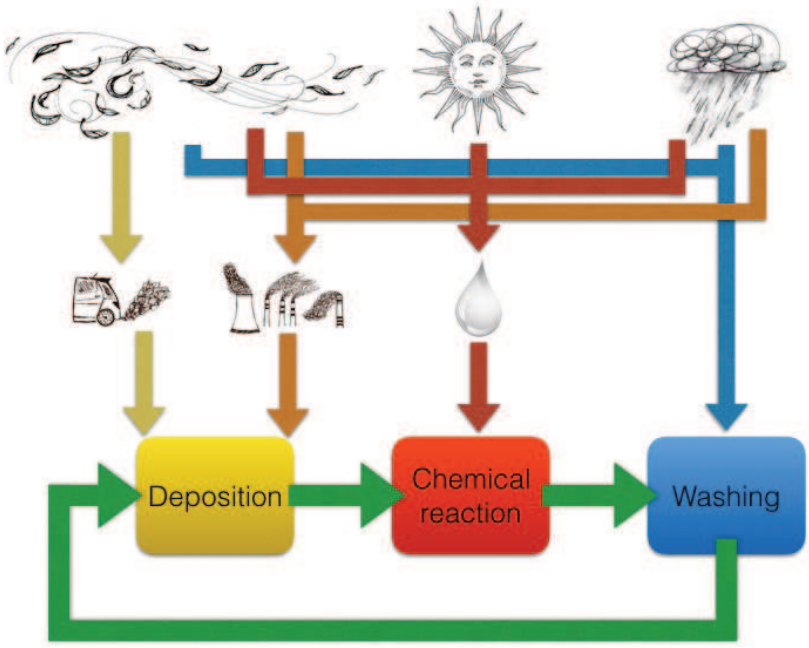


Figure IV.1: *The air pollution process can be divided in three steps (bottom): deposition, chemical reaction and washing. Each step is influenced by external factors shown in the top (pollution sources, wind, sun and rain) and its combination shown in the middle (humidity and incoming pollution).*

Our method uses these principles to model pollution at a urban scale using a fast but physically-inspired approach. We consider pollution deposition to depend on the three environmental factors depicted above: wind, rain and sun exposure. After the pre-computation of the pollution distribution along the city, we model the three intervening steps: deposition, reaction and washing. We treat them as independent steps to simplify the process and to avoid recomputing pollution deposition at subsequent steps. Moreover, we focus on facades while getting rid of roads and sidewalks because these parts are affected by very complicate processes due to people and vehicles interactions. In our examples, we deal only with traffic as the main pollution source but our algorithm is also well suited for other pollution sources.

First of all, we pre-compute (in a few seconds) a physically-based 2D pollution map that models the pollution distribution through the blocks of the city. This is done by our wind simulation, transports pollution from a set of predefined sources (e.g. high traffic streets) to the facades based on a 2D slice of the city defined by the user. We run a fast Lattice-Boltzmann fluid simulation in CUDA over this slice followed by a particle-based dispersion of pollution using an Euler's integration approach. An appropriate scattering function is then proposed to model the interaction between pollution particles and the facades during this simulation.

After this pre-computation, we simulate the pollution accumulation at run-time by propagating the 2D pollution map over the facades and using a set of screen-space operations. These operations allow us to first estimate the related factors and model the local deposition of pollution. Then, these operations determine the chemical reactions of the particles with the surfaces and, finally, clean the deposited pollution on exposed surfaces. All such physically-inspired behaviors are incorporated into a final expression that describes plausible appearance of polluted surfaces.

(cc) BY-NC-ND

This results in an efficient approach that is able to quickly simulate changes of appearance due to pollution on urban scenes. In addition, despite our approach requires a pre-process, it avoids time and memory-consuming pre-computations and is suitable for large-scale and complex scenarios.

IV.3 Pre-process

As introduced above, the transport of pollutants is pre-computed based on a 2D wind simulation followed by the proper particles' propagation and scattering on buildings. In the next subsection, we briefly introduce below how we proceed to compute the wind map resulting from the 2D simulation. Afterwards, we present our approach to simulate the propagation of the pollution and its interaction with the buildings. Thanks to it, we generate a 2D pollution map that contains the pollution distribution along the city. It will be later used by our deposition model to compute the final polluted appearance (see Section IV.5). The computation of this pollution map is split into two steps: wind simulation and pollution propagation. These steps can be executed independently (first compute the wind map and, then, use it to propagate pollution) or interleaved (use different intermediate wind maps for each pollution propagation step). The latter allows us to simulate real complex behaviors by changing the wind direction during the simulation.

IV.3.1 Wind Map Generation

The objective of this part is to perform an approximate, but reliable simulation of the wind flow in an urban environment at interactive frame rates, allowing the dynamic change of wind parameters. To achieve these objectives we approximate the wind in an urban landscape as a 2D grid, that allows us to represent the fluid as a layer with respect to the ground. Moreover, we simulate its behavior thanks to a technique based on the Lattice-Boltzmann Method (LBM) which describes the behavior of a gas on a microscopic level using kinetic theory. More specifically, it consists of a regular lattice that represents the fluid in discrete locations and a simulation of how the fluid flows on it. We have chosen LBM because it is highly parametrizable, fast and it converges into the continuous equations of Navier-Stokes. Although this subsection includes several LBM details, for a more detailed explanation of Lattice-Boltzmann methods, we point the reader to the book of Mohamad [Moh12].

The LBM methods work on a lattice. Several variations of LBM exist depending on the grid dimensions and number of lattice vectors used. In our case we use the $D2Q9$ model, meaning that we do a 2D simulation where we consider 9 neighbors for each cell, including the cell itself (see Figure IV.2). In our model the lattice vectors are $e_0..e_8$. At each lattice site $\vec{x}$ and time t, fluid particles moving at arbitrary velocities are modeled by particle distribution functions $f_i(\vec{x}, t)$, that represent the expected number of particles moving along a lattice direction vector e_i. Particles of a lattice are only allowed to move along the lattice velocity vectors. The magnitude of the velocity vectors e_1 through e_4 is 1 lattice unit per time step, while for velocity vectors e_5 through e_8 is $\sqrt{2}$ lattice units per time step (see Figure IV.2). Also, the vector with index 0 has zero length, and indicates particles that are not moving anywhere in the next time step. The number of resting particles may change at each time step due to the fact that some of them may be accelerated by collisions with other particles.

Two important physical values can be calculated from the particle distribution functions. On one hand, assuming that all particles have the same mass, we can calculate the density of a cell by

(cc) BY-NC-ND

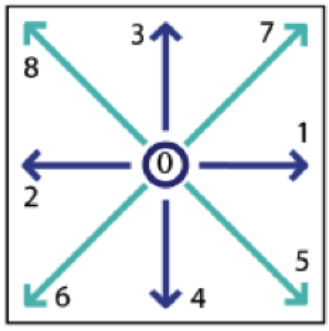


Figure IV.2: *The D2Q9 Lattice-Boltzman model cell. Each of the arrows corresponds to lattice direction vectors e_i and its length shows its magnitude.*

summing up all distribution functions:

$$\rho = \sum_{i=0}^{8} f_i \tag{IV.1}$$

On the other hand, we can compute, for each cell, the velocity and overall direction in which the particles will move:

$$\vec{v} = \frac{1}{\rho} \sum_{i=0}^{8} f_i \vec{e_i} \tag{IV.2}$$

All cells are updated at each time step by simple rules, taking into account the state of the surrounding cells. At each time instant, the simulation consists of two steps: streaming and collision. The former consists in moving particles from one cell to another while the latter only changes the particle distribution of the cell. To model this behavior, the equilibrium distribution function f_i^{eq} must be calculated and the distribution functions accordingly updated. We use the equilibrium distribution function suggested by He and Luo [HL97]:

$$f_i^{eq}(\rho, \vec{v}) = w_i \left[\rho + \rho_0 \left(\frac{3}{c^2}(\vec{e_i} \cdot \vec{v}) + \frac{9}{2c^4}(\vec{e_i} \cdot v)^2 - \frac{3}{2c^2}(\vec{v} \cdot \vec{v}) \right) \right] \tag{IV.3}$$

where weights w_i depend on the length of the velocity vector.

We update the distributions functions of each cell following the streaming and collisions steps. Both are combined into one formula which is known as the Lattice-Boltzmann equation:

$$f_i(\vec{x} + \vec{e_i}, t+1) - f_i(\vec{x}, t) = \omega(f_i^{eq}(\rho, \vec{v}) - f_i(\vec{x}, t)) \tag{IV.4}$$

where $\omega = 2/(6\nu + 1)$ and ν is the kinematic viscosity. The left hand side of this equation accounts for the stream step, and the right hand side represent the collision as combination of the current distribution function and the local equilibrium.

(cc) BY-NC-ND

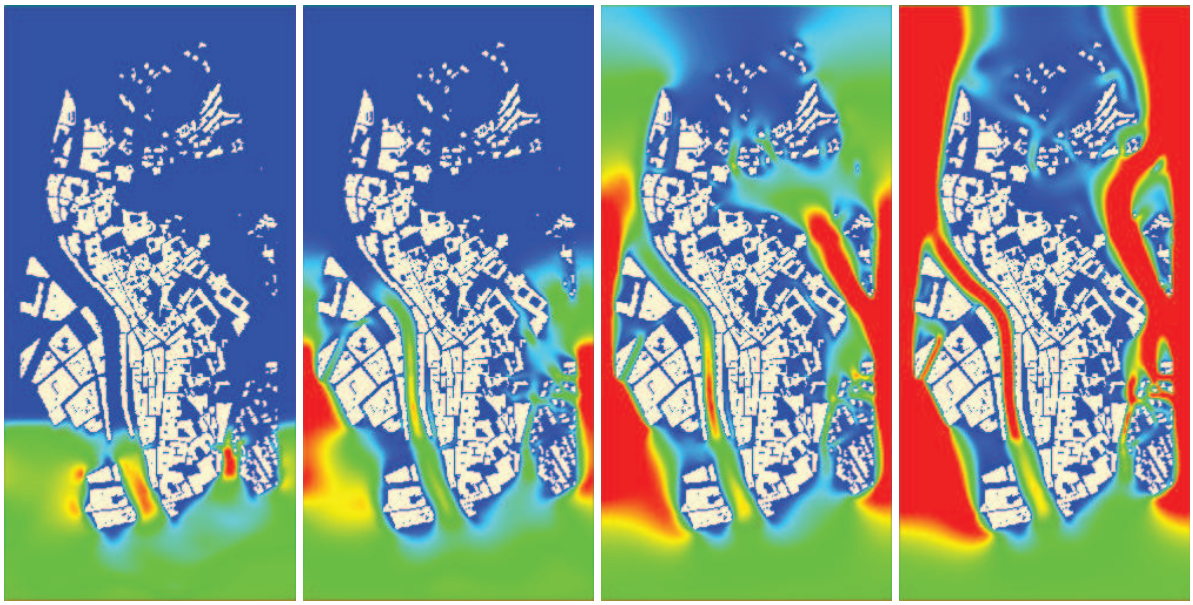


Figure IV.3: *Velocity magnitude (blue represents at different time steps for a simulation of a map of the city of Girona. Wind entering from the bottom simulation boundary and leaving on top. Red color represents positions where velocity is maximum while blue shows the zones where it is zero.*

In a scene, we have two types of boundaries. On one hand, there are the boundary conditions between the simulation space (the empty space between buildings) and the simulation boundaries (the buildings themselves). For them, we have chosen the standard no-slip boundary condition and, for a realistic behavior, we have implemented the so called half-way bounce back condition, which reflects the incoming particle distribution into their opposite directions. In the results presented on this thesis, we have used the second condition. On the other hand, there are the boundaries of the simulation domain that are basically three different conditions: inlet, outlet and simulation sides. At the entrance boundary (inlet), in which we will specify the speed, we have used Zou and He's method [ZH97] to find the density and the cell's initial vectors. At the exit boundary (outlet), where the output speed is not known, we take the values of the cells immediately before and extrapolate their values to calculate the missing directions. These are called open boundary conditions in the literature. For the sides, we use a periodic boundary condition. In this case the implementation connects one side of the simulation with what happens at the opposite side, and vice versa. This simulates that there are no rigid boundaries at sides. Although it is not completely correct if there are buildings close to the sides as their presence can affect the simulation on the opposite boundary, we have observed that it is the most used in the literature and it works well in our experiments taking precautionary empty space.

Figure IV.3 shows the wind simulation process at different time steps. It is important to note that, although we justify the assumptions we do, other simulation configurations (like boundary conditions) could be assumed without refuting our technique.

(cc) BY-NC-ND

IV.3.2 Pollution Map Generation

The second part of the pre-computations deals specifically with the generation of the pollution map. This process consists in simulating the propagation of the pollution from its sources until its deposition on buildings. In cities, the main sources of pollution are chimneys and vehicles. In order to be able to simulate the pollution propagation, our technique requires a wind map and a map of the city with the pollution sources mapped on it. In our experiments, we have considered only high traffic streets in this step and we have integrated the rest of pollution in the ambient pollution term (C_{amb}) defined later. In particular, we consider traffic in an aggregate way and its effect is considered *on average* by using linear pollution sources. However, our technique can be easily extended to model other types of pollution sources like chimneys, which could be modeled as point sources, and to consider different levels of traffic in streets. We do not either deal with sidewalks and roads that are sensible to the effect of other agents like vehicles and pedestrians friction, but rather concentrate on the facades themselves, as pointed out before.

Pollution is generally formed by chemicals, particles of contaminants and other suspended particles (e.g. dust) in the atmosphere. Common gaseous pollutants include carbon monoxide, sulfur dioxide, chlorofluorocarbons (CFCs) and nitrogen oxides produced by industry and motor vehicles. Photochemical ozone and smog are created as nitrogen oxides and hydrocarbons react to sunlight. Particulate matter (PM), or fine dust, is characterized by their micrometer size $PM_{2.5}$ to PM_{10}. Due to this disparity of origins, propagation and deposition of pollutants follow very complicated processes that are related to gas state behavior and its dynamics, chemical reactions and classic mechanics. In order to simplify our experiments, we assume that all the pollutants can be described as pollution particles.

We model the pollutant propagation from their sources (high traffic streets or any other potential source of contaminants) until they reach the final surface where they are deposited. During the simulation, we model the pollutant propagation considering that sources emit a certain amount of pollutant particles (pp) per unit time, and these particles are dragged by our wind map. Then, every time a particle reaches a building, a portion of it (pp^S) remains sticked to the building and the rest (pp^B) bounces being dragged again by the wind map. This remaining portion continues in the simulation as a smaller new particle until it is finally absorbed by a building.

The propagation of the pollution is modeled as follows: each particle (pp) starting at a pollution source follows a dispersion model using Euler's integration approach. Such particles, at time t, will have a displacement Δr:

$$\Delta r = v \cdot t + \frac{1}{2} \cdot a \cdot t^2 \tag{IV.5}$$

where v its velocity at the previous time step, and a the acceleration resulting from the force exerted by the wind at the corresponding particle position. The acceleration a is computed taking into account the area of the pollutant particle and its mass, both obtained from tabulated standard data [Wat].

As stated by Wu [YL+92], the deposition of polluted particles follows three steps: aerodynamic transport, boundary layer transport, and uptake by receptor. The first step was described above, while the two last steps are combined in our model. We perform this simplification because simulating the chemical reactions that occur during the uptake step depends on many parameters demanding this single step a deep specific study. These parameters range from the size of particles to the specific surface material properties, such as roughness, porosity,

(cc) BY-NC-ND

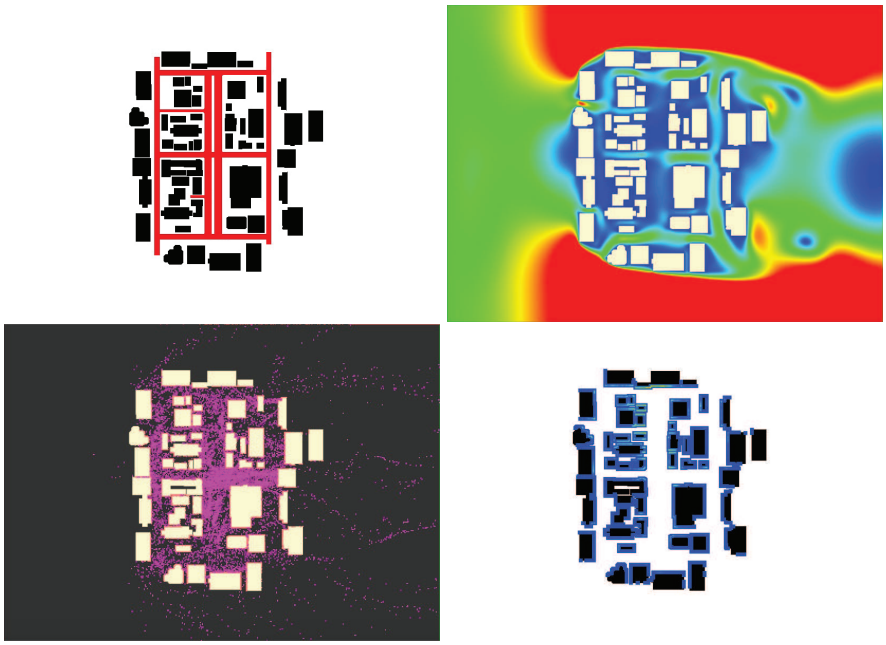


Figure IV.4: *Given a map of the city with the pollution sources (traffic roads) marked in red (top-left), we compute a wind map using a 2D Lattice-Boltzman simulation (top-right) which we use to simulate the propagation of pollution along the city (bottom-left). Each bounce of the pollution particles is stored in the pollution map (bottom-right).*

surface state, as well as meteorological factors like humidity and wash-off [Moh16]. In our model, these last parameters are accounted for at a later stage (see Section IV.5). The modeling of the physical interactions between particles and surfaces is also very complex. However, facing the large amount of different materials in urban scenes which are not available in the 2D simplification, we propose to use a coefficient β as a new empirical and simple virtual material characteristic: β belongs to [0,1] and represents the part of pollutants that stick to the surface. This behavior is described by the following equations:

$$pp^S_{mass} = \beta \cdot pp_{mass} \tag{IV.6}$$

$$pp^B_{mass} = (1 - \beta) \cdot pp_{mass} \tag{IV.7}$$

where pp^S is the sticked portion and pp^B is the bounced one of the considered pollition particles pp. Moreover, the volume of both portions is also scaled by this β coefficient.

In order to generate the final pollution map, when a pollution particle hits a building, we splat the sticked part pp^S of this pollution particle into the 2D pollution map with a gaussian kernel. This splatting process smooths the results making them more similar to gaseous simulations and allows us to propagate pollution to elements not captured in the 2D map (e.g. porches, overhangs and windowsills). The accumulation of pollution in the map is done by means of summing the sticked portions pp^S splatted by the different pollution particles at each pixel. Once the pollution map is generated, we start the rendering and the rest of computations are performed at run-time. From this point onwards, we will refer to this 2D pollution map as C_{map}.

(cc) BY-NC-ND

Figure IV.4 shows the whole process. The top-left image is the layout of the city where black pixels correspond to solid space (buildings) while white and red ones represent the empty space. The difference between white and red pixels is that the latter represent pollution sources. The top-right image shows a wind map in a steady state generated using the method explained in Section IV.3.1. bottom-left image corresponds to a pollution simulation step using the above wind map and pollution sources. The purple points correspond to the pollution particles pp present in the simulation. Finally, the bottom-right image represents a map of the city with the pollution particles splatted on building contours.

IV.4 Weathering Factors

As we mentioned before, there are several factors that influence the deposition of pollution particles and its interaction with surfaces. On one hand there are external factors mostly related to the environment that influence many weathering processes. Among them, our model considers wind, rain and sun. On the other hand, there are the internal properties of the material of the surface that affect the binding of pollutants and its chemical reaction. We only consider material porosity in this group. While the pre-computation stage focuses on simulating the global effect of wind (and pollution) over the city, the run-time stage is in charge of estimating and modeling the influence of the rest of factors. Figure IV.5 shows an example of such estimation in a urban scene. Details of these measures can be found in the following subsections.

In order to achieve interactive results, we focus on a fast and reliable estimation. As proposed for the sun, we compute these factors based on the use of screen-space techniques. More specifically, they are inspired by the screen-space ambient-occlusion technique of Bavoil et al. [BSD08] which has been explained in detail in Section III.4.1. This technique computes the horizon angles (h) and tangent angles (t) for several directions (θ) to approximate ambient occlusions (see Figure III.6). We use the same idea of horizon and tangent angles to approximate the accessibility of the surface to the involved factors.

IV.4.1 Wind Accessibility (A_w)

Wind is a very important element in the pollution soiling process. It brings pollution particles to surfaces, it dries wet surfaces reducing humidity and it cleans polluted surfaces due to erosion. Although we use a physically-based wind simulation to compute the pollution map, it is not directly usable to estimate wind accessibility at pixel level. As a 2D map, it neglects the vertical component of the wind and it does not take into account the small features and details of buildings geometry. Instead, we use the more accurate 2D estimation in the generation of the pollution map and we complete it during run-time with ambient occlusion, by considering that the local influence of wind can come from every space direction (even from vertical directions) due to local turbulent behaviors. This ambient wind term is computed using ISHBAO technique described before (see Section III.4.1).

IV.4.2 Insolation (I)

Sun irradiation on surfaces is a key factor to estimate humidity. Surfaces very insolated dry faster and, consequently, the presence of humidity on them is very low. Moreover, it can also affect other weathering processes besides pollution deposition. It can damage materials, accelerate several chemical reactions and promote biological growth, among others. To account for sun

(cc) BY-NC-ND

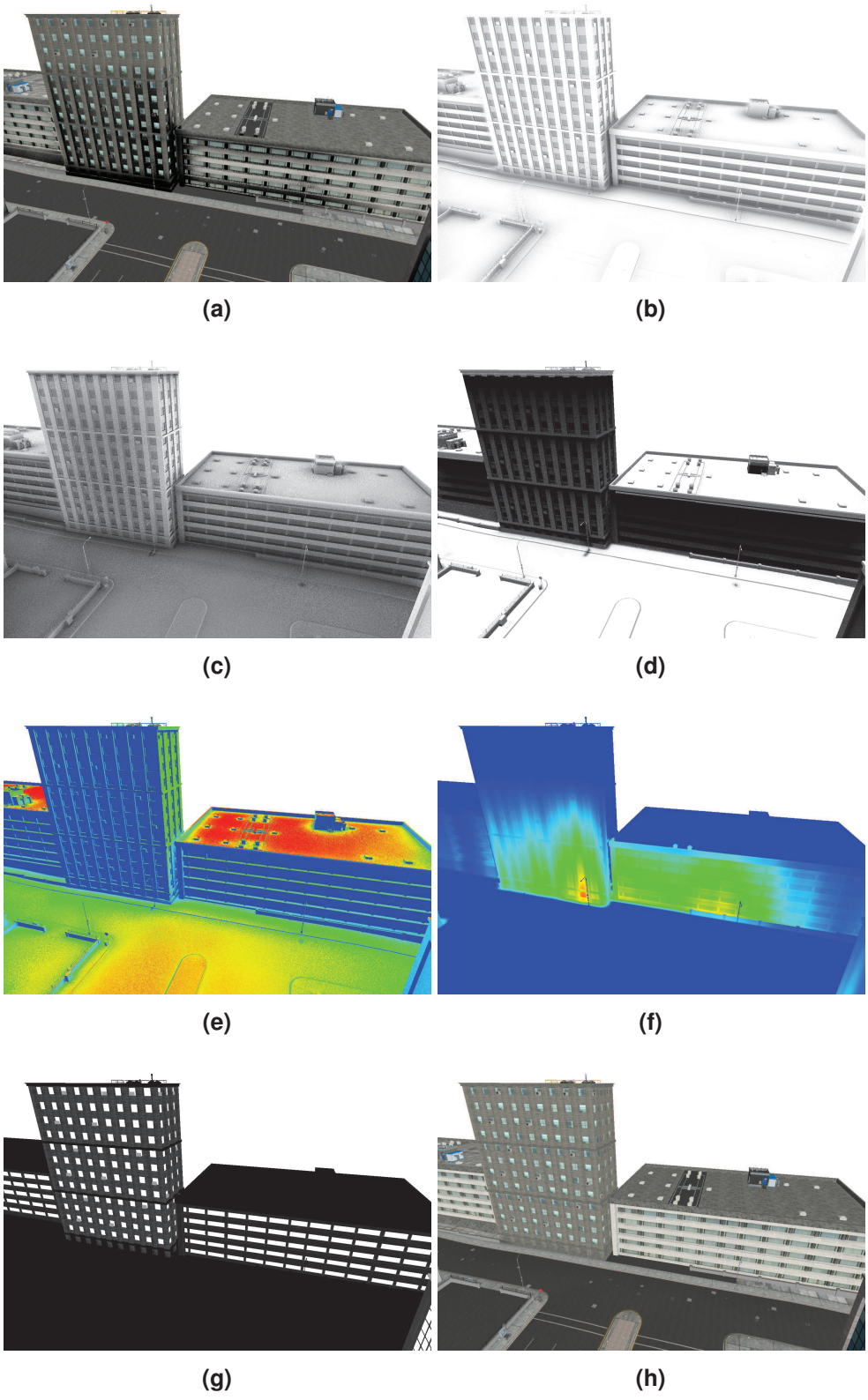


Figure IV.5: *Estimation of the different factors that affect (a) final render: (b) wind accessibility A_w, (c) sky accessibility A_s, (d) rain accessibility A_r, (e) insolation I, (f) pollution ($C_{map} * C_{height}$), (g) porosity p and (h) texture and shading.*

(cc) BY-NC-ND

radiation at a urban scale, we resort to the approach presented in Chapter III. As explained before, our method is a two-scale approach that includes both global and local building features and allows us to estimate long-term solar exposures in real-time. In this case, the obtained exposure measures are further normalized according to the maximum surface exposure, which facilitates its later combination with the other factors.

IV.4.3 Rain Accessibility (A_r)

Rain affects every step of the pollution process. It is responsible of the deposition of atmospheric pollution and of the humidification of surfaces. Moreover, when it falls strongly, its drops mechanically remove pollution previously deposited. In order to estimate rain accessibility, Mérillou et al. [Mer+10] considered a specific cone of directions from which rain impacted the surfaces. With no prior information, this cone is positioned vertically with a 20 degrees aperture. In our case we resort to a similar idea, but adapting the screen-space approach used for the sun instead of relying on ray casting, which would be impractical on a city scale. Compared to the sun, sampling over a cone is more intricate than sampling over the whole sky hemisphere. First, the obtained angles need to be tested against the predefined cone to accordingly restrict the visibility. In addition, at the local scale, we can not use a full directional sampling since the cone projects on a very limited set of directions for views oriented far from the zenith. For this reason, sampling both scales separately and weighting the two terms at the end would lead to noticeable visibility artifacts. To solve that, we combine both samplings during the overall process, as is described next and depicted in Figure IV.6.

At the global scale, visibility is sampled in a similar way than for the sun, i.e., based on the large-scale sky visibility estimation explained in Section III.4.2. We take a set of directions around the current point in the zenithal view, and sample its depth map to determine the corresponding horizon angles. The main difference for rain is that each angle needs to be clamped according to the cone's aperture.

At the local scale, sampling within the cone would imply taking a set of directions within the projection of the cone onto the current view. Instead, we project lines from the cone surface to the view space, directly limiting the sampling within the cone's projection. Afterwards, each depth sample taken in this space is projected back onto the global space (zenithal view), so that its elevation can be computed and tested against the angles found for the global visibility (see Figure IV.6 bottom). As the angles obtained in the global scale are clamped to the cone's aperture, we ensure that, after refining them with those obtained at the local scale, the resulting angle remains inside the cone aperture. This combination process permits to evaluate both visibilities on the same space, which provides higher accuracy than sampling them separately as done for the sun. To properly test each back-projected sample with the corresponding direction, we chose the global direction as the one closest to the sampling point on the zenithal view, rather than the original sampling direction. At the end, the angles obtained for each direction are combined to estimate the rain accessibility term A_r, using the following expression:

$$A_r = \frac{1}{N_{dir}} \sum_{i=1}^{N_{dir}} \left(\frac{\sin h'(\theta_i) - \sin h(\theta_i)}{\sin(\alpha_{conus})} \right) \tag{IV.8}$$

where N_{dir} is the number of sampling directions, $h(\theta)$ the elevation angle at each direction θ, $h'(\theta)$ the opening angle (elevation of salient features as described before) and α_{conus} is the angle of the cone.

(cc) BY-NC-ND

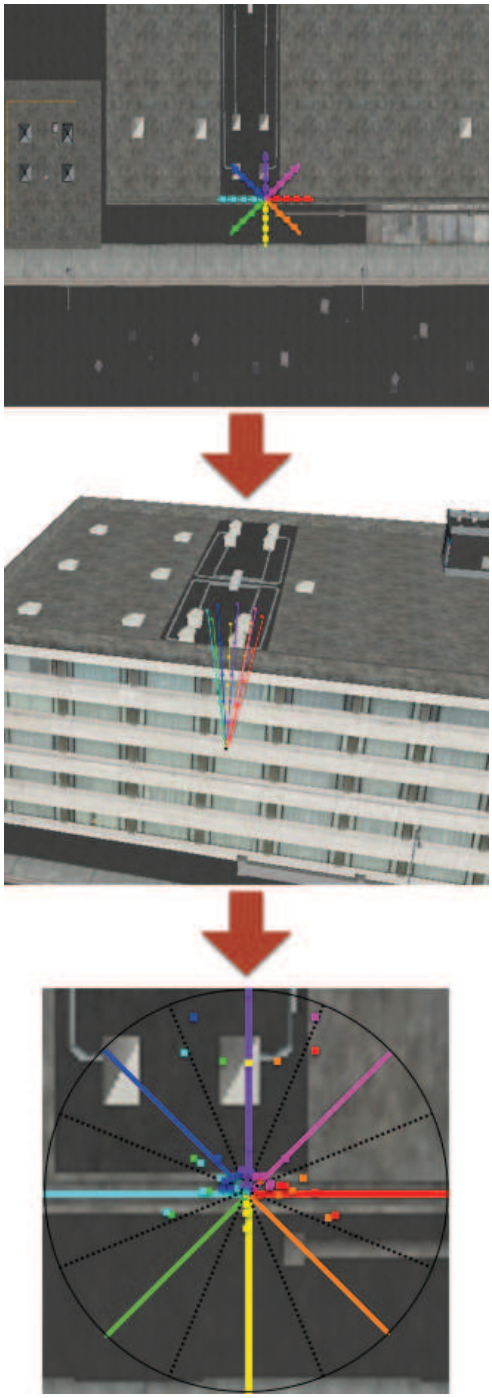


Figure IV.6: *Rain accessibility sampling. Global scale sampling is performed in the zenithal view along a set of directions (top). Local sampling uses the same set of directions but defined on the rain cone and projected onto the current view (middle). Each local sample is projected back to the zenithal view to evaluate the occlusion on the same space (bottom). The color code of the samples (bottom) corresponds to its original direction in the zenithal view (top). Note that points falling in the upper part belong to samples taken in the building's roof in spite of the original direction in the zenithal. Although not shown, jittering can be optionally applied on both sampling directions and positions.*

(cc) BY-NC-ND

IV.4.4 Sky Accessibility (A_s)

One particular case of rain, linked to the wet deposition, is due to the effect of drizzle or fog. In these cases a vertical cone is not a suitable approximation because water condensation can reach surfaces from almost every direction. For this reason, to estimate this misty rain cases we use a full sky accessibility estimation, using the same two-scale accessibility measure proposed for the sun (i.e. $A_s = SVF$ from Section III.4.2).

IV.4.5 Porosity (p)

Porosity is a key factor in the deposition of particles on surfaces [Mer+10]. The more the surface is porous, the more black crusts can develop. This is caused because porose surfaces trap more moisture and, thus, can further develop the gypsum grid. Moreover, the intensity of the washing process also depends on material porosity: porous stones absorb water by capillarity, creating a black crust even in rain-exposed areas. In order to use them in our technique, we require the user to indicate the porosity of the surfaces through material textures. When this porosity texture is not available, we use an empirical constant term associated to each specific material.

IV.5 Deposition Model

The atmospheric pollution weathering process described in Section IV.2 alters facades appearance over time. In order to model such changes of appearance, the goal of our model is to compute a specific concentration of pollution over the facades at run-time, which will serve to modulate the final appearance of the surfaces. Thus, we must estimate first the concentration of polluted particles at each surface point.

In the following subsections we focus on describing how we combine the 2D pollution map and the weathering factors with their corresponding accessibilities into a final pollution concentration C. We compute this term following the steps described in Section IV.2 and Figure IV.1. We can express this computation as the combination of the three main factors modeling each particular stage:

$$C = C_D \cdot H \cdot W \cdot p \tag{IV.9}$$

where C_D models the particle deposition, H is the humidity term in charge of the chemical bounding onto the surfaces, and W represents the washing effects. This expression is finally weighted by the porosity p at each surface point. In a final stage, we use the computed concentration of polluted particles at each surface point C in order to model the changes of appearance that alter facades over time (see Section IV.5.4). Figure IV.7 shows the significance of each of these steps in the final rendering.

IV.5.1 Particle Deposition (C_D)

Polluted particle deposition can be either a wet or a dry process. The wet process consists of deposition via precipitations while the dry process relates to delivering particles to surfaces due to the action of other forces.

(cc) BY-NC-ND

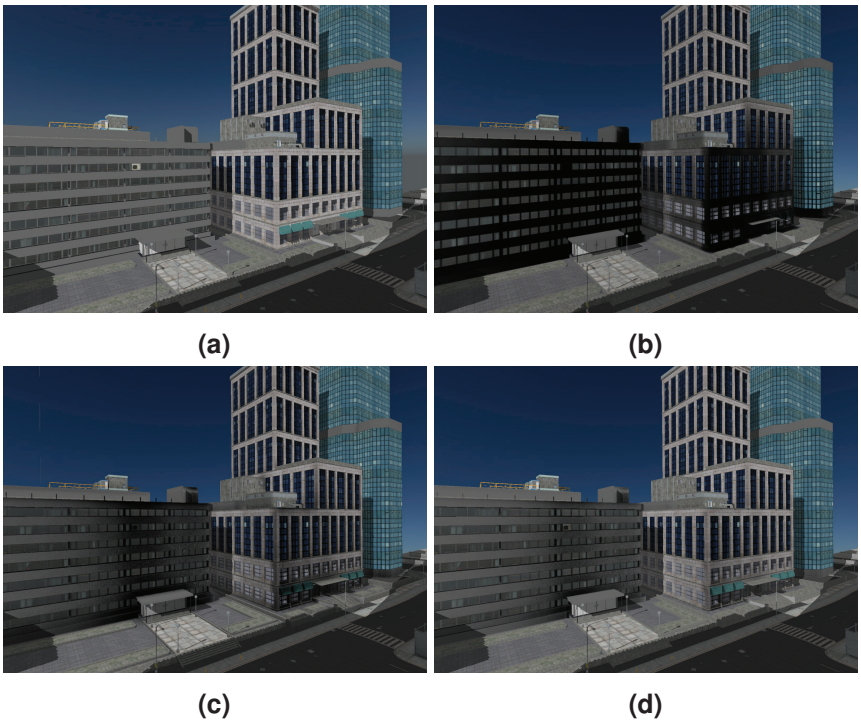


Figure IV.7: *Pollution process: (a) city clean of pollution, (b) city with the incoming pollution due to dry and wet deposition, (c) city with pollution deposited after the chemical reaction, and (d) city with pollution cleaned by rain and wind.*

Wet deposition

Atmospheric pollution [Mer+10] only accounts for the migration of polluted particles over large distances through the atmosphere. While rain fall has a mechanical washing action on polluted surfaces, it also brings down these particles on the overall scene. This process is related to any kind of precipitation: from blizzards to misty rain or fog. According to this, we can consider that all the surfaces in the scene are equally affected by this phenomenon. While horizontal building parts receive more pollutants than vertical ones, they are also more washed by successive precipitations. We simply handle wet deposition, directly in screen-space, as a constant ambient pollution factor C_{amb} weighted phenomenologically by the sky visibility term A_s (see Section IV.4.4), as rain can not bring pollutant particles on non accessible parts. We chose to use sky visibility instead of rain accessibility to be able to handle drizzle and fog rather than only rain fall, as detailed in Section IV.4.4.

Dry deposition

Dry deposition is related to the particles that reach facades through gases. We use the pre-computed 2D pollution map C_{map} explained in Section IV.3.2 to model this particle distribution along the city. At this point, all our computations are in 2D but we know that a vertical gradient exists [Wat]. This vertical distribution is mainly caused by three aspects: the properties of the sources (i.e., direction and force of the emissions), the different polluted particle weights and its interaction with the forces responsible for its movement. A physically intensive simulation is not our objective, so we propose to use an approximative function C_{height} that represents the

(cc) BY-NC-ND

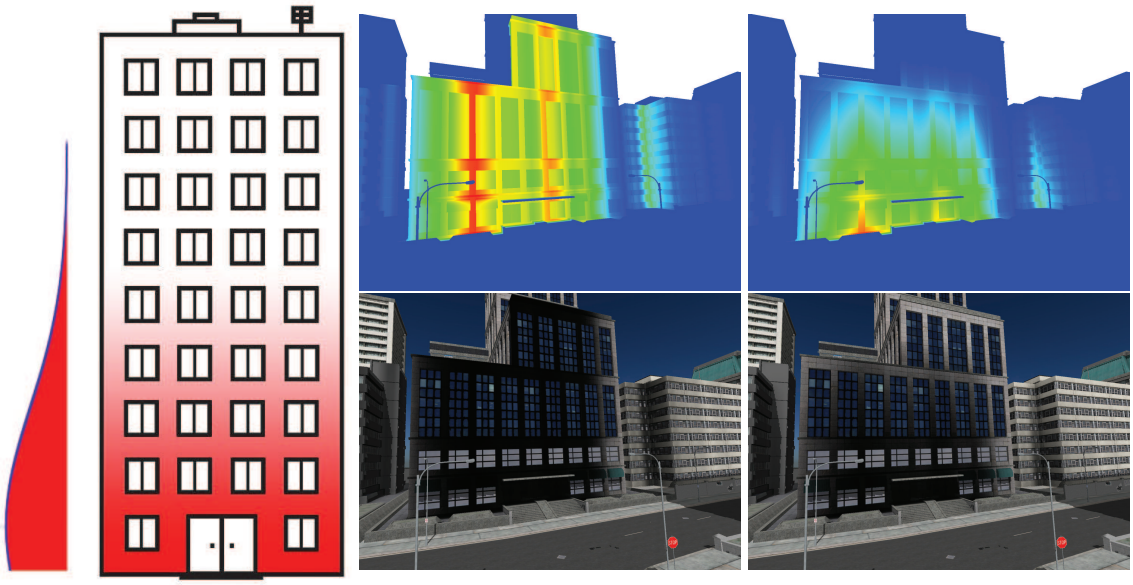


Figure IV.8: *(a) C_{height} function used to model the distribution of the deposition along the facade. (b) Example of dry deposition on a facade using this function (right) over the original distribution (left).*

distribution of the pollution deposition on facades as a relation to ground distance. This fits quite well with our observation in real buildings. Although any type of decreasing function is suitable, we have used a simple linear function:

$$C_{height} = max(0, 1 - h_{pixel}/h_C) \tag{IV.10}$$

where h_{pixel} is the height of the pixel with respect to the ground and h_C is an user defined constant that represents the maximum height that pollution can reach. To understand the importance of this function, we refer the reader to Figure IV.8 where we show the difference between using C_{height} and its absence.

Since the pre-computed pollution map is based on a rough representation of the facades, small features present on them such as windows or balconies are not included in this map. In order to account for them, we further use the local wind accessibility term A_w (see Section IV.4.1) to model how those features affect the deposition of pollution. This is again an empirical physically-inspired parameter permitting a very fast computation compared to alternatives such as a full 3D Lattice-Boltzman Method, too expensive for our purpose.

Integrated deposition

Both types of deposition form part of our model and are integrated as follows:

$$C_D(\vec{x}) = C_{map}(x, y)C_{height}(z)A_w(\vec{x}) + C_{amb}A_s(\vec{x}) \tag{IV.11}$$

For the dry deposition (first term of the expression), we access the 2D pollution map C_{map} to retrieve the propagated pollution, which is subsequently weighted by C_{height} to account for the vertical gradient of pollution on the facades. We then complete the obtained deposition by using the wind accessibility factor A_w in order to take into account how city geometry locally alters

(cc) BY-NC-ND

the wind and hence the deposition. Wet deposition (right term of Equation IV.11), on the other hand, is handled through the constant factor C_{amb} weighted by the sky visibility factor A_s. The latter permits us to avoid the use of complex physical models and parameters while maintaining a physically plausible behavior and exhibiting an intuitive control parameter to the user.

IV.5.2 Chemical Reactions (H)

Particles deposited on surfaces are bounded to them through a set of chemical reactions. Such reactions are boosted by the humidity of affected surfaces. Humidity mainly depends on the accumulated water originated from rainfalls and the overall water concentration in the air. But, it decreases under the drying effects of wind and sun exposure. The latter is particularly due to a raise in the temperature of the surfaces. Although a fluid simulation would be more correct to estimate the zones where the water is accumulated, this kind of simulations are too costly to be done at run-time. Moreover, our solution improves humidity estimation from weathering literature that usually are based on a user's defined map [DGA04].

Given the deposited pollution, we model the humidity term H as its chemical bounding with the surface. We consider the sky accessibility A_s as the source of humidity on the surface to take into account both rain and drizzle. Moreover, we propose to use solar exposure I and wind accessibility A_w as drier agents. We describe this process as a linear combination of the intervening factors:

$$H = k_{hs}A_s - k_{hi}I - k_{hw}A_w \qquad \{0 \leq H \leq 1\} \tag{IV.12}$$

where I is the normalized insolation or solar exposure term computed at the current point (see Section IV.4.2), A_s is the sky accessibility term (see Section IV.4.4), A_w is the wind accessibility term (see Section IV.4.1) and k_{hs}, k_{hi} and k_{hw} are the corresponding weighting factors. Note that humidity needs to be less than one to avoid increasing the pollution at a given point, which means that $0 \leq k_{hi} + k_{hw} \leq k_{hs} \leq 1$.

IV.5.3 Washing Effects (W)

A last phenomenon that guides the concentration of pollution on facades is the washing effect that in our model corresponds to the third term of Equation IV.9. Regions submitted to strong rain impact and water streaming often result in cleaner areas, while those not affected by them are prone to concentrate pollution particles trapped into gypsum, that later results in soiling patterns and black crusts. Although washing is mainly due to the strong impacts of rain droplets, wind also participates in eroding deposited pollution, typically in combination with waterfall.

Despite a physical simulation combining rain and wind would handle better the interactions between droplets and would provide a realistic identification of the impact points and eroded areas, its computation would be too expensive. Therefore, we rely on an approximation based on our screen-space methods to estimate these factors. We propose to estimate the droplets impact areas by estimating rain accessibility A_r (see Section IV.4.3) inside a vertical cone. In addition, we propose to use wind accessibility A_w (see Section IV.4.1) as an indicator of the main erosion points. Our model starts from a situation where no washing effect is applied and, then, we introduce the mentioned washing factors:

$$W = 1 - k_{wr}A_r - k_{ww}A_w \qquad \{0 \leq W \leq 1\} \tag{IV.13}$$

(cc) BY-NC-ND

where k_{wr} and k_{ww} weigh the contribution of rain and wind, respectively. Here again the weights need to fulfill the condition $0 \leq k_{wr} + k_{ww} \leq 1$.

The selection of the weighting factors for the above expressions is left to the user, which thanks to the interactive feedback provided by our approach are easy to fine tune.

IV.5.4 Final Render (R)

Once we have computed the incoming pollution deposited on the current pixel, we must estimate how its appearance changes accordingly. This process commonly involves a change in the reflectance of the surface proportional to the amount of pollution deposited. To compute this change of reflectance, we resort to the following expression that is fast to compute and well-known in atmospheric pollution weathering [PRD98]:

$$R = (R_0 - R_f) \cdot exp(-C \cdot t) + R_f \tag{IV.14}$$

where t is time (in days), C is the concentration of polluted particles (in $\mu g/m^3$), R_0 is the original unaffected reflectance and R_f is the reflectance of a fully polluted zone. The latter term is typically set to black as we do in our experiments. In the original formulation there is also a constant rate for blackening k. Observe that, in our implementation, this constant rate is included into the constant C for simplicity.

IV.6 Results and Discussion

In this section, we present the results and discussion of our method. First of all, we introduce a number of experiments that we have performed to analyze the correctness of our technique. Then, we show how our approach can improve virtual images if it is introduced in a common rendering engine. Finally, we present some discussion describing the main advantages and drawbacks of our technique.

IV.6.1 Tests

This subsection is intended to describe a number of experiments to test the behavior, hence the correctness, of our approach and to analyze their results. We aim with this to highlight the relevance of the pollution map (see Section IV.3.2) and the deposition model (see Section IV.5) of our technique. We also present some tests to help the understanding and the analysis of each step of the model showing how each of the considered weathering factors influences the corresponding step and the final result. One of the main problems testing the correctness of our technique is the lack of ground truth data to compare with. This lack of data is originated because there is not yet physical models well established by experts on urban physics. For this reason, we supply these testing results as a deeper analysis of our technique behavior.

Inside each group of experiments, we have used the same scene and camera configuration. In addition, for easy understanding, the isotropic deposition used as reference image is modeled, on one hand, using a constant dry deposition term (i.e., $C_D = 1 * C_{height}$) for Figures IV.9 and IV.10; and, on the other hand, using a constant dry and wet deposition terms (i.e., $C_D = 1 * C_{height} + C_{amb}$) for Figures IV.11 to IV.14.

(cc) BY-NC-ND

Pollution Map Generation

In order to test correctness and behavior of our method to generate pollution maps (see Section IV.3.2), we have performed two experiments. On the one hand, we have tested the behavior of our simulation taking into account the occlusion between buildings. On the other hand, we have studied how the pollution distribution varies in function of the configuration of the streets in a city (i.e., alignment and spread).

Protection: In Figure IV.9, we show how the pollution map generated changes depending on the buildings position, how these changes affect the pollution deposited by the dry deposition factor and how it compares against an isotropic pollution model. More specifically, we show how the pollution deposited on the surface of a building changes in function of the nearest neighboring buildings. For a completely exposed building, the pollution is deposited along the whole front facade. If it is partially occluded by a building, the pollution is deposited only on the exposed part of the building. Finally, if there is another building which covers the current building, the pollution deposited is negligible.

Layout: In Figure IV.10, we show how the pollution map generated changes depending on the buildings distribution of the city, how the corresponding distribution affects the pollution deposited by the dry deposition factor and a comparison to an isotropic pollution model. More specifically, we show how the pollution deposited on the surface of a building changes in function of the alignment of the buildings of a city and its separation. For a perfectly aligned city (1st row), important amounts of pollution are displaced along the streets without impacting in any building. Moreover, the pollution that reaches the buildings is deposited mainly on front facades. For a more spread but still aligned distribution (2nd row), the pollution distribution is not affected but the amount of pollution on the facades is increased due to the reduction of available facades where pollution can impact. For an unaligned distribution of buildings (3rd row), the pollution particles hit the buildings more easily and, therefore, the amount of pollution deposited is very high and widely extended. If we spread the unaligned distribution of buildings (4th row), there is more empty space to make flow corridors through which the pollution can be transported without impacting on any facade. In this case, the pollution distribution is reduced considerably on the facades not facing the wind direction.

Deposition Model

In order to test the correctness and behavior of our deposition model (see Section IV.5), we have performed four experiments. First, we have analyzed the different possible combinations of the steps that take part of our deposition model. Then, for each of the steps of the model, we have analyzed how the different weathering factors have influence on them by testing different combinations of such factors.

Combination of factors: In Figure IV.11, we show how different combinations of the steps of our deposition model modifies the pollution deposited on surfaces. Starting from an isotropic deposition of pollution, the particle deposition step places the pollution in the zones more exposed to wind and sky. This behavior is very noticeable at the facades in between the two buildings where pollution is less accessible. When we incorporate the chemical bounding step, the parts more exposed to humidity (i.e., less exposed to sun or wind) show higher levels of pollution than the dry zones. This process is very visible when comparing the facades of the tallest building and at the upper parts of the outgoings (protruding elements). If we consider the

(cc) BY-NC-ND

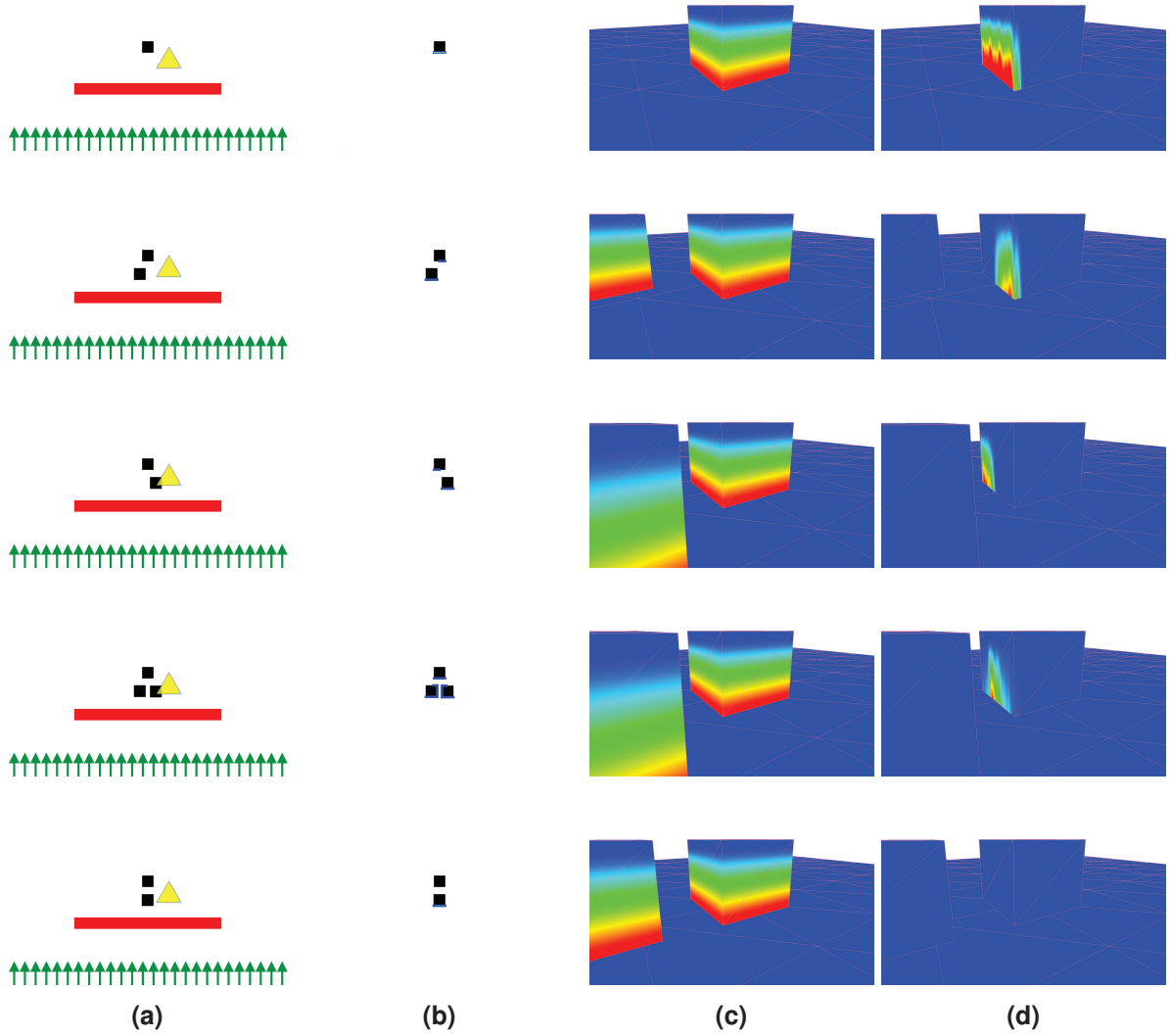


Figure IV.9: *Comparison of different pollution maps (b) and their corresponding dry pollution deposition (d) to an isotropic pollution deposition (c) generated after changing the closing environment. In (a), we show the initial configuration: buildings (black), pollution sources (red), wind direction (green arrows) and the camera position (yellow triangle) used in (c) and (d).*

(cc) BY-NC-ND

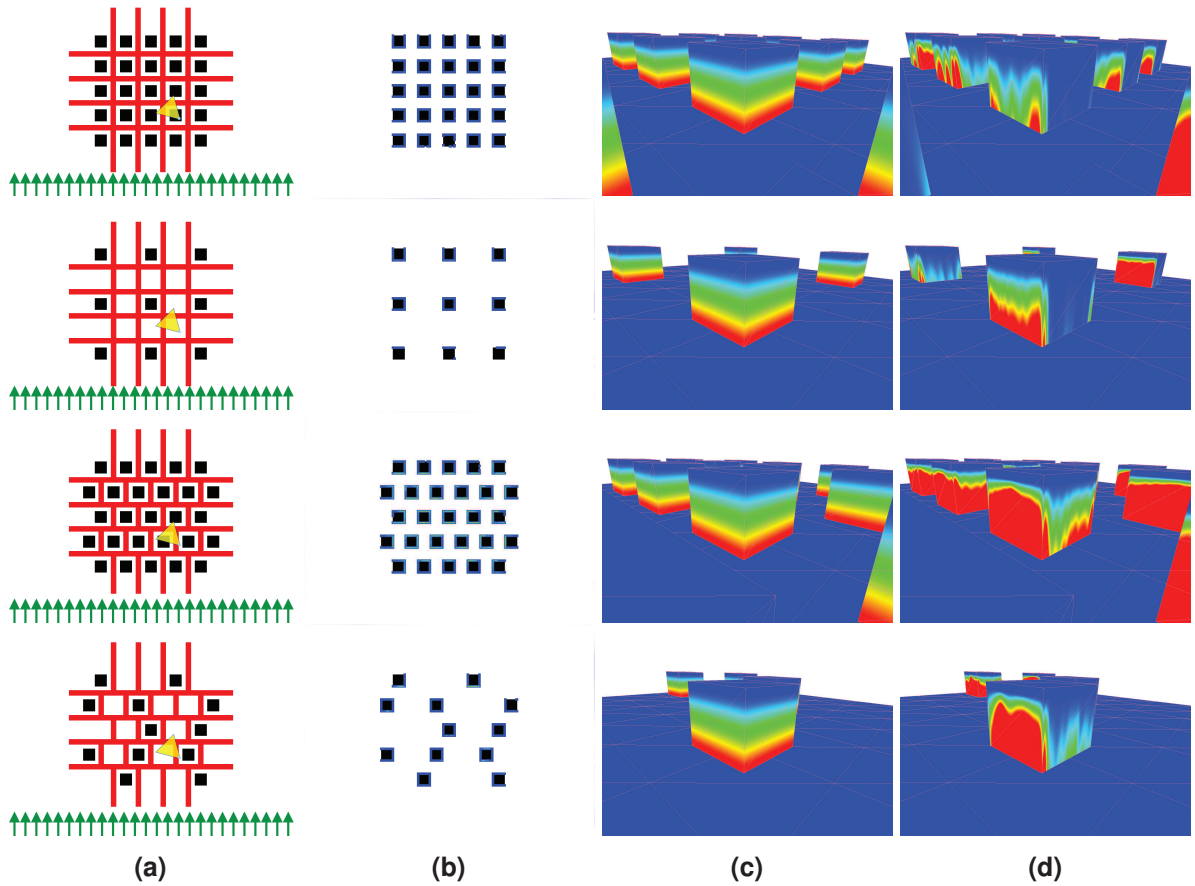


Figure IV.10: *Comparison of different pollution maps (b) and their corresponding dry pollution deposition (d) to an isotropic pollution deposition (c) generated after changing the alignment and separation of the buildings of a city. In (a), we show the initial configuration: buildings (black), pollution sources (red), wind direction (green arrows) and the camera position (yellow triangle) used in (c) and (d).*

(cc) BY-NC-ND

washing process without the chemical bounding, this process cleans the roofs of the buildings and the parts of the facades more exposed to rain drops. This is visible at the corners of the right building and at the outgoings of the tallest building. However, there are no changes between the different humidity zones and the pollution distribution is less realistic. Finally, when we take into account all the steps of our model the obtained results handle better the differences on the pollution distribution and it results in more realistic images.

Particle deposition C_D: In Figure IV.12, we compare an isotropic deposition model to our model using the dry deposition, the wet deposition and both of them taking into account only the particle deposition step. Starting from an isotropic deposition of pollution, we can see how the different factors affect the pollution deposition. On one hand, we incorporate the wind accessibility to the dry deposition term. This results in lower concentrations of pollution in the zones with less wind accessibility like hidden corners of the windows or the lower parts of the facades of the central alley. On the other hand, we incorporate the sky accessibility to the ambient term. This results in a general small deposition with higher concentrations on the roofs with respect to the facades. Finally, when we join both terms we achieve a distribution of the pollution that properly handles the real accessibility to the pollution improving considerably the initial isotropic deposition distribution. Note that these last images are equivalent to the second row of Figure IV.11.

Chemical reactions H: In Figure IV.13, we show how sun and wind modify the humidity of the surface and, therefore, the pollution deposited on it during the chemical reactions step. Starting from an isotropic deposition of pollution, we show how the different components of the chemical step affect the results. First, we use the humidity of the surfaces taking into account the accessibility to the sky. This results in lower concentrations of pollution in the regions with less sky accessibility (facade of right building). Then, we incorporate the reduction of humidity caused by the sun energy. In this case, the sun radiation dries the upper parts of the roofs and protrusions in both buildings. Moreover, this results in a difference of pollution concentration depending on the facade orientation (see front vs side facades of the tallest building). Alternatively, we have incorporated the reduction of humidity caused by the wind. This results in a general reduction of pollution except in the occluded zones like the hidden corners next to the windows in the tallest building. Finally, if we take into account the wind and the sun as humidity reducers, we are able to capture all the changes in the humidity distribution and, with it, obtain more accurate results.

Washing effects W: In Figure IV.14, we show how wind and rain modify the pollution deposited on surface during the washing step. Starting from an isotropic deposition of pollution, the washing process removes pollution from the parts of the building exposed to rain and wind. On one hand, the rain washing reduces pollution deposited on the parts accessible to the rain cone such as roofs and the protrusions of the tallest building. On the other hand, the wind removes pollution in a more general way being more visible in the corners of the buildings and keeping high concentrations of pollution in the nooks of the facades. When we use both factors in the washing process, this results in more realistic images with clean roofs and different pollution levels depending on the accessibility of each point, having more pollution in hidden corners and the central passage and less in the outgoings and more exposed parts.

(cc) BY-NC-ND

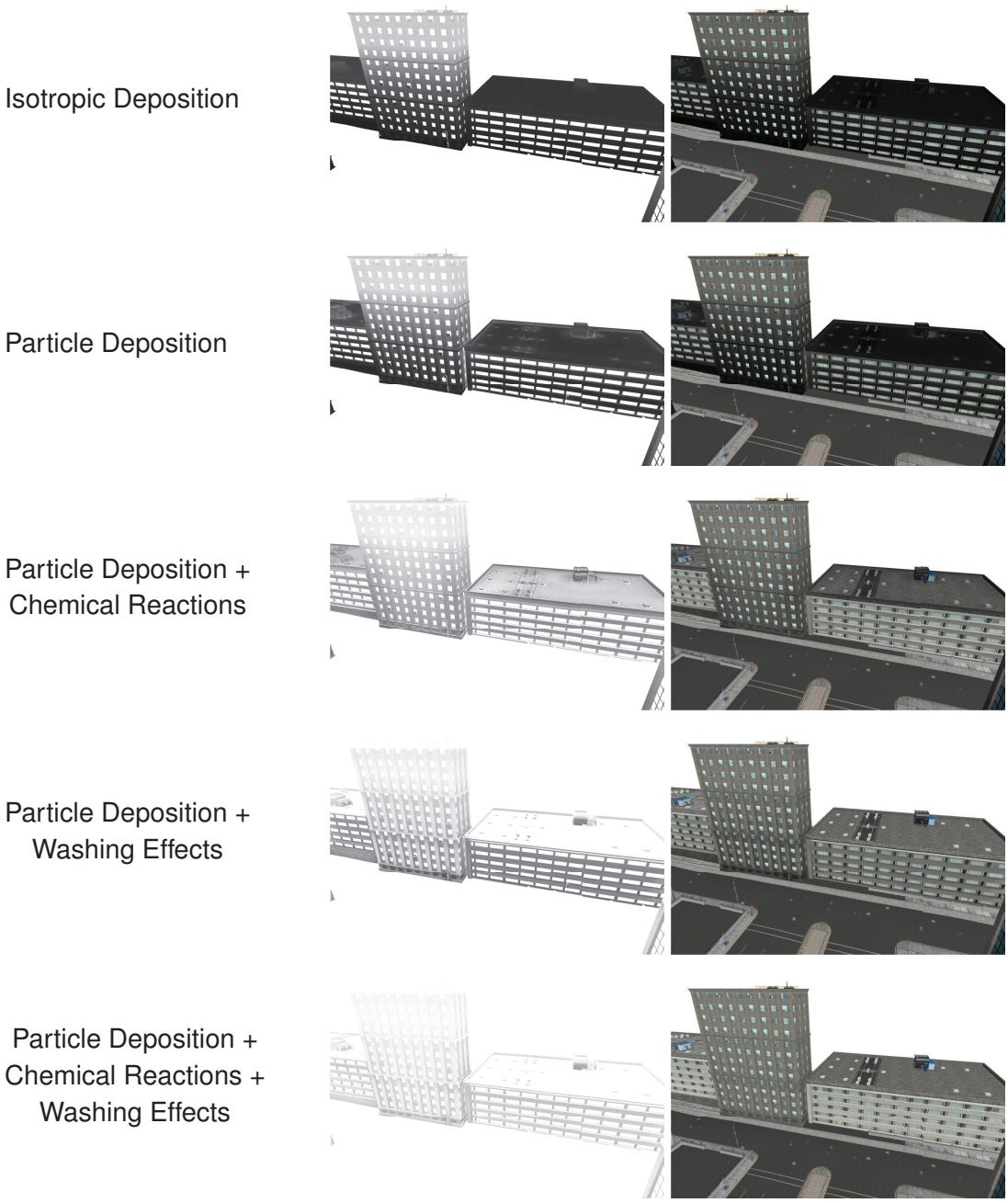


Figure IV.11: *Pollution deposition comparison between different combinations of the steps of our pollution model with respect to an isotropic pollution deposition model. Left column shows the pollution layer and right column the shaded render.*

(cc) BY-NC-ND

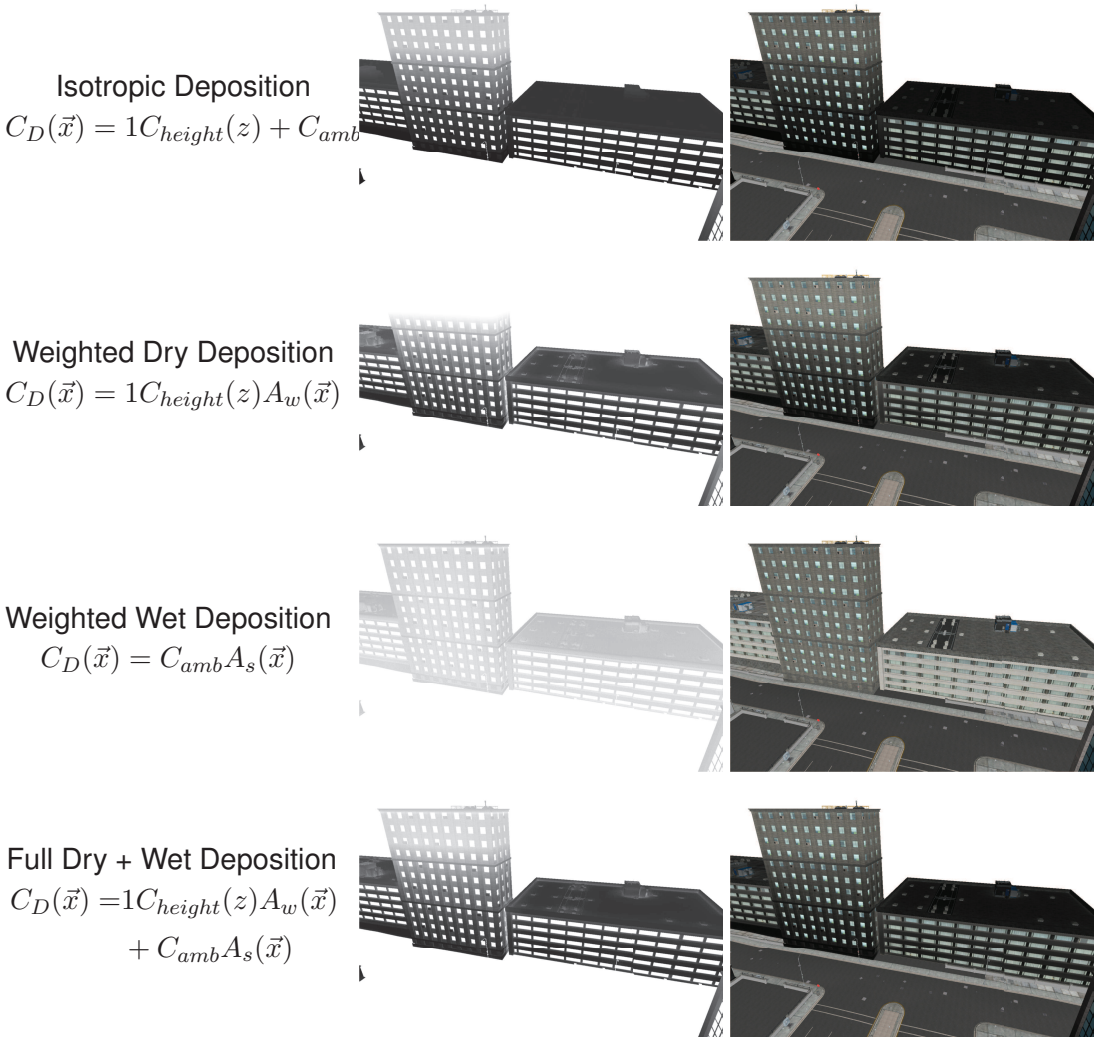


Figure IV.12: *Pollution deposition comparison between an isotropic pollution deposition model (first row) and our model using only the particle deposition step (C_D) taking into account the dry deposition (second row), the wet deposition (third row) and both of them (fourth row). Left column shows the pollution layer and right column the shaded render.*

(cc) BY-NC-ND

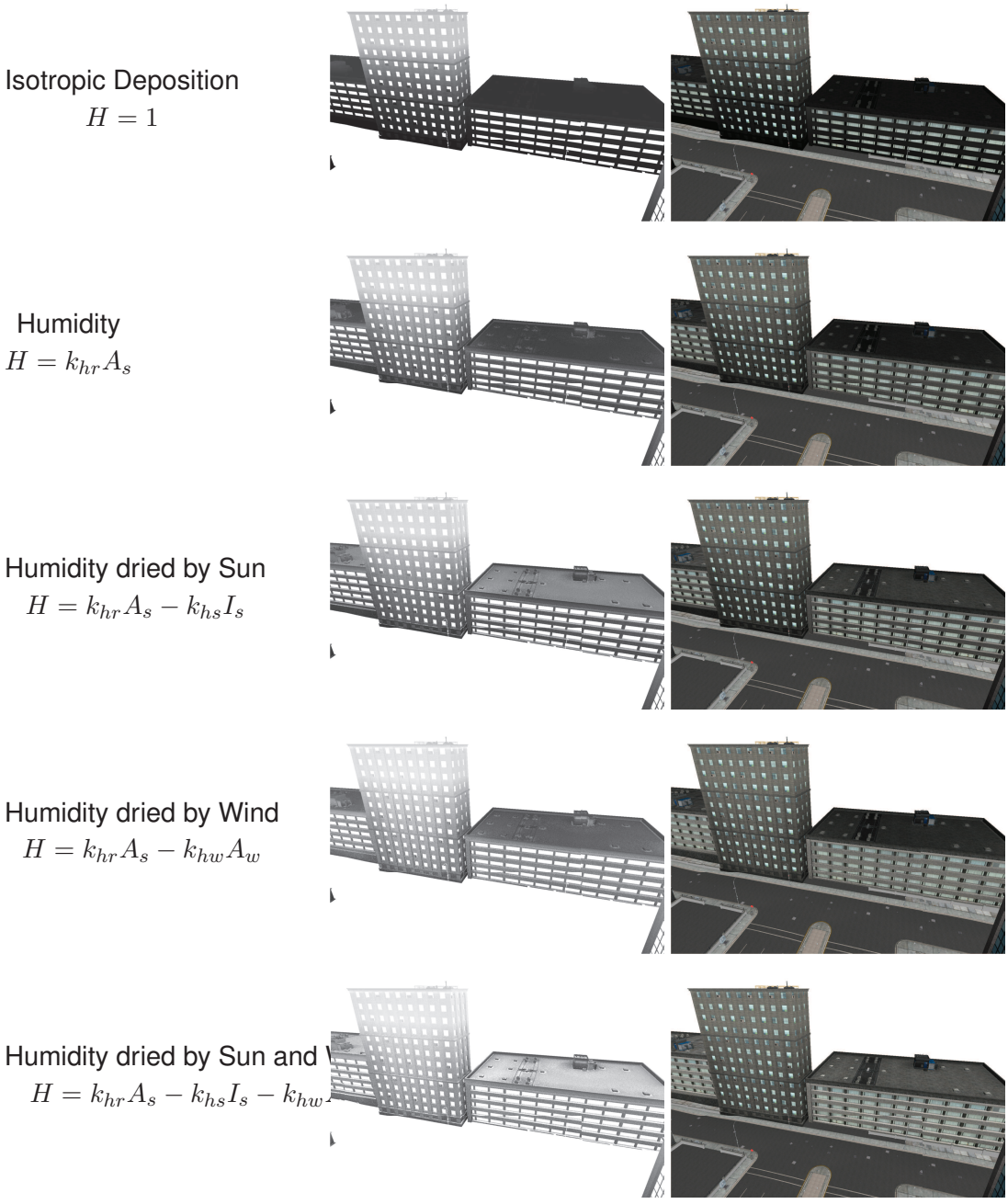


Figure IV.13: *Pollution deposition comparison between an isotropic pollution deposition model (first row) and our model using only the chemical reactions (H) step taking into account the estimated humidity (second row), the resulting humidity after being dried by the sun (third row), the resulting humidity after being dried by the wind (fourth row) and the resulting humidity after being dried by the sun and the wind (fifth row). Left column shows the pollution layer and right column the shaded render.*

(cc) BY-NC-ND

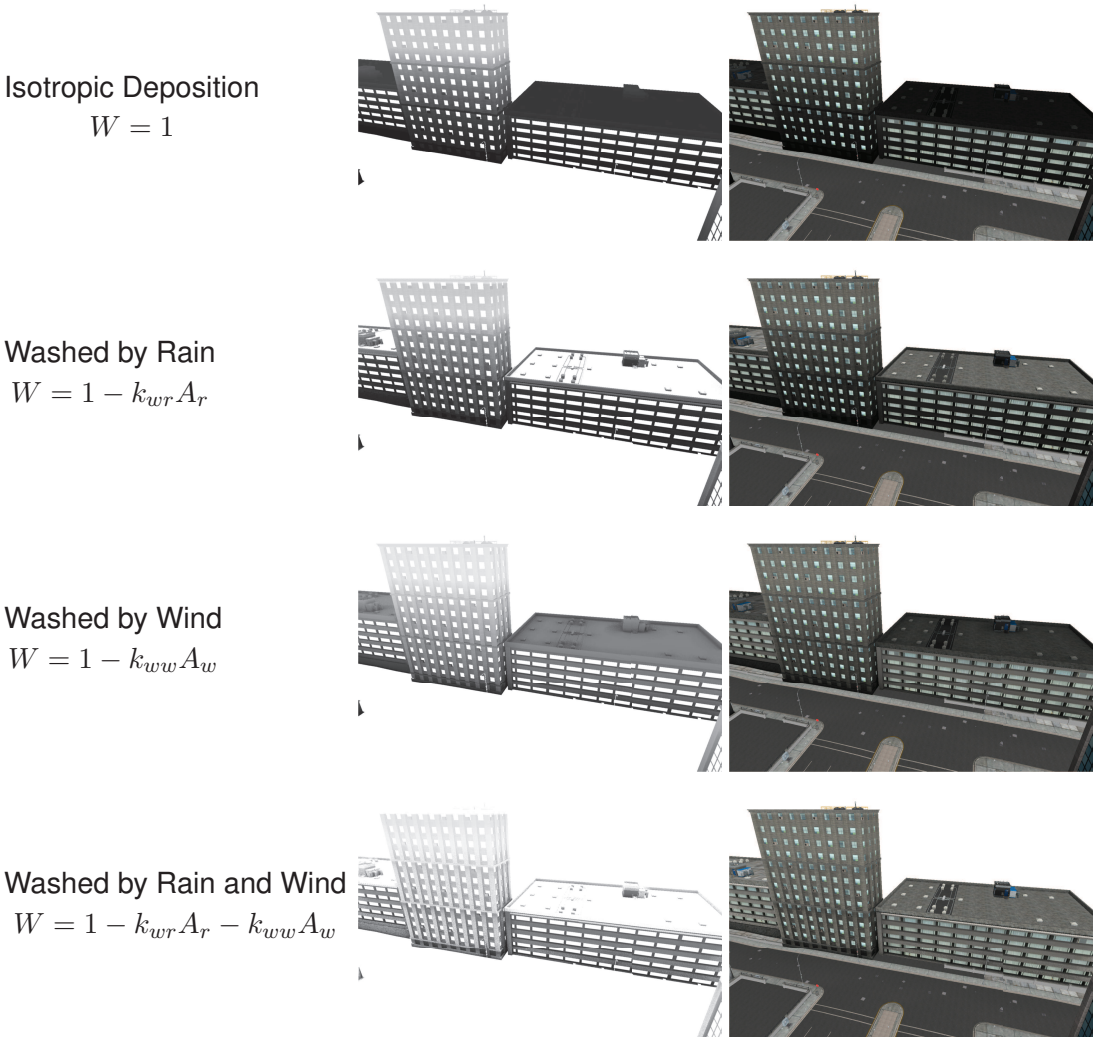


Figure IV.14: *Pollution deposition comparison between an isotropic pollution deposition model (first row) and our model using only the washing step (W) taking into account the washing produced by the rain (second row), the washing produced by the wind (third row) and the washing produced by the rain and the wind (fourth row). Left column shows the pollution layer and right column the shaded render.*

(cc) BY-NC-ND

IV.6.2 Results

We have integrated our model in a basic render engine which uses screen-space shadow mapping and ambient occlusion to illuminate the scene and the Hosek model [HW13] to render the sky. We have used a model of a city with 800K triangles comprising 53 detailed buildings, though we note that our run-time computations should be easily scalable to larger cities or more detailed scenes due to its screen-space nature. All our examples were again executed on a MacBook Pro using a 2.6GHz Intel Core i7 with 16GB of memory and a NVIDIA GT 750 graphic card with 2GB of memory. In all the screen-space accessibility operations we use 8 sampling directions and 16 samples per direction. During the generation of the pollution map, we set the parameter that models the interaction between pollution particles and surfaces as $\beta = 0.6$. In the results showed in this section (except Figure IV.18 where they are augmented/decreased), we use the following configuration of weighting factors for the computation of the pollution concentration: $k_{hr} = 1.0$, $k_{hs} = 0.3$, $k_{hw} = 0.4$, $k_{wr} = 0.6$ and $k_{ww} = 0.5$. Moreover, we model the ambient pollution as an isotropic constant with value $C_{amb} = 0.05$. We also limit the maximum height where pollution is noticeable as $h_C = 15m$.

The memory cost of our technique during rendering of a 800x600 screen is a 512x512 float texture for the exposure map that takes up to 8 MBytes, a 32 bit texture of 800x600 pixels for the zenithal depth map with a cost of 14.64 MBytes, a 128 bit texture with 4 channels of 800x600 pixels for current view's depth and normal map which takes 58.59 MBytes, a 32 bit texture with 4 channels of 800x600 pixels where we store the shading and incoming pollution with a cost of 14.64 MBytes, a 24 bit texture with three channels of 512x256 for the sky environment map that takes up to 3 MBytes, and a 32 bit texture of 800x580 for the precomputed pollution map with a cost of 14.16 MBytes. In our implementation we store the porosity of the materials in the alpha channel of the color textures without additional cost. On the other hand, during the pre-computation of the pollution map, we use a 800x580x8 buffer to store the wind forces that costs 113.28 MBytes, a 800x580 buffer to store the type of each cell with a cost of 442.5 KBytes, a 800x580x2 buffer to store the final wind direction and force that costs 28.32 MBytes and, finally, a buffer to map up to 100K particles with a cost of 36.62 Mbytes. In summary, our technique needs less than 180 MBytes during the pre-computation and less than 115 MBytes during the rendering.

The times of the precomputation and the rendering steps are shown in Table IV.1. Rendering times are again obtained using a 800x600 screen and pre-computation ones using a 800x580 map, where the cell resolution of the pollution map is 0.5x0.5m. The pre-computation times represent how many iterations we perform per second. To obtain steady state maps we need up to a thousand iterations, so we generate a pollution map in 30-40 seconds.

	Rendering			Pre-computation	
	Far Views	Close Views		Wind Map	Pollution Map
fps	21-23	7-9	ips	22-24	80-120

Table IV.1: *Frames per second (fps) for rendering a $800x600$ screen (left) and iterations per second (ips) during the wind and pollution map generation using 100K particles and a 800x580 map representing the city (right).*

Figure IV.7 exposes each of the steps of the pollution process starting from clean buildings (a).

(cc) BY-NC-ND

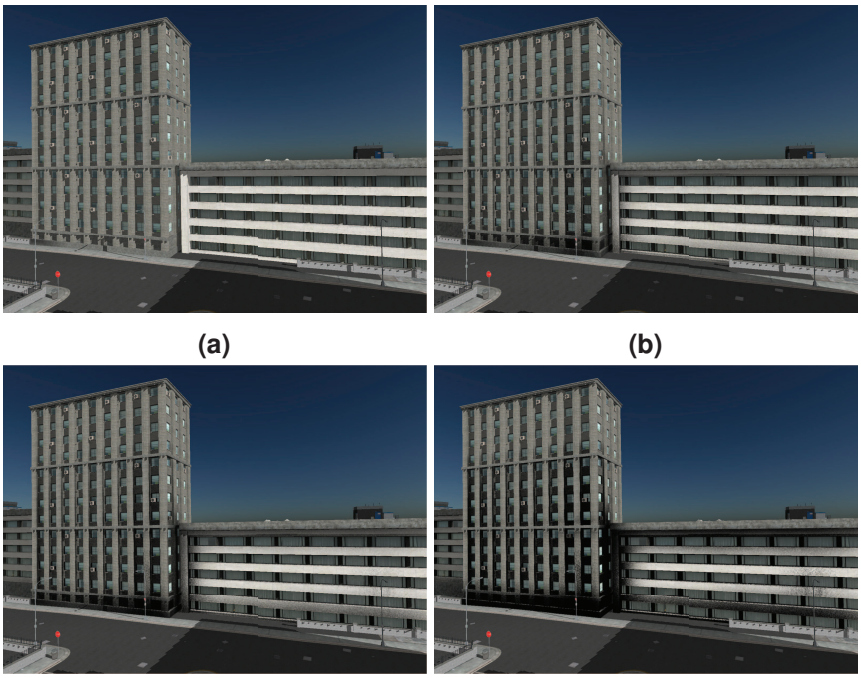


Figure IV.15: *Evolution of pollution on a urban scene along time using our method. (a) the initial clean buildings, (b) pollution effects start to be visible, (c) the pollution on the surfaces is quite noticeable and (d) the buildings are very polluted.*

If we only take into account the incoming pollution (b), this is too pronounced and uniformly distributed. Once we incorporate the chemical reaction produced by the humidity factor into the deposition (c), the distribution improves considerably. One can see how pollution is less concentrated in parts of the facade more exposed to drying factors (sun and wind). Finally, we obtain the definitive concentration after incorporating the washing step (d). Here, we can see how zones more exposed to rain impacts are more cleaned with respect to other surfaces. Such effects are very noticeable at windowsills, at overhangs and at green sunshades.

Figure IV.15 shows how the pollution changes the appearance of two buildings over time. One can see how the pollution is vertically distributed with higher concentrations near the ground. Moreover, pollution is concentrated in zones protected from washing processes (windows on left building and highest balconies on right building) and in wet areas (facades on corridor).

Figure IV.17 shows images from far to very close viewpoints of a city. This demonstrates that our technique is suitable for different view scales. In close images, notice how porosity affects the pollution distribution on the surfaces (compare windows with wall bricks or sunshades) and how our technique integrates smoothly with buildings textures and features. Moreover, one can see how the zones exposed to rain and to humidity reducing factors have a lower level of pollution (see windowsills and overhangs). Against this, the most covered parts become the most polluted (look under the hangovers and balconies).

Thanks to our intuitive user parameters, our technique allows to easily modify the importance of each external factor at run-time. Thus, users can easily recreate several scenarios: dry weathers, cloudy environments, windy cities, ... Figure IV.18, for instance, shows the behavior of our technique when tuning these parameters. When the user increases insolation importance,

(cc) BY-NC-ND



Figure IV.16: *Our technique (right) is able to reproduce the behavior of real life pollution weathering effects (left). Pollution distribution and modified material appearance are comparable in a very plausible way.*

(cc) BY-NC-ND

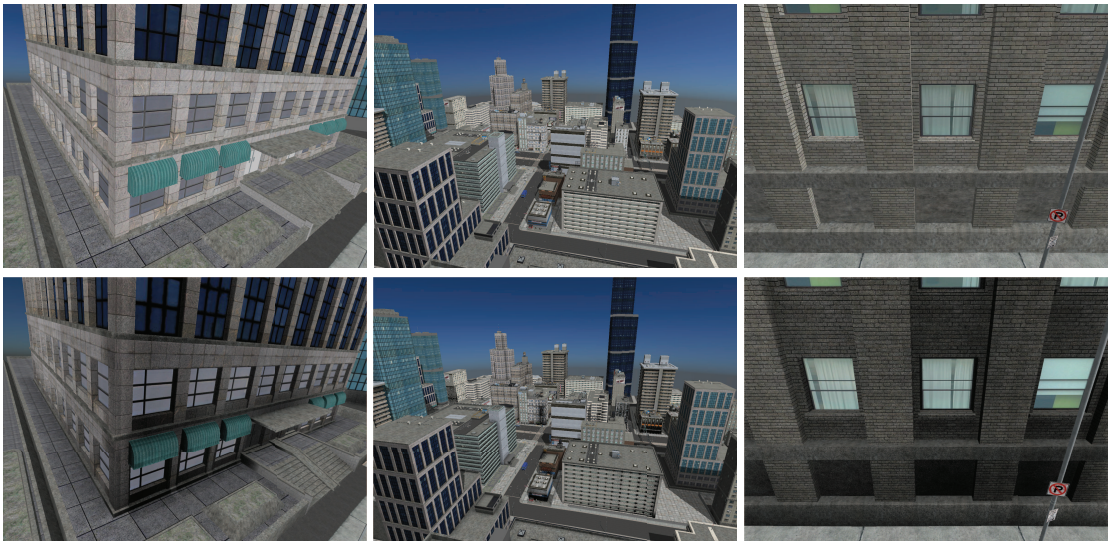


Figure IV.17: *Examples generated using our technique (bottom) and their corresponding clean references (top) from different points of view, which shows how our technique is suitable for different scales and how pollution integrates well with buildings details.*



(a) *Insolation* **(b)** *Wind* **(c)** *Rain*

Figure IV.18: *Users can change interactively the importance of each external factor in the final render. Here we show how the final appearance changes by modifying the importance of each factor: insolation (left), wind (center) and rain (right).*

(cc) BY-NC-ND

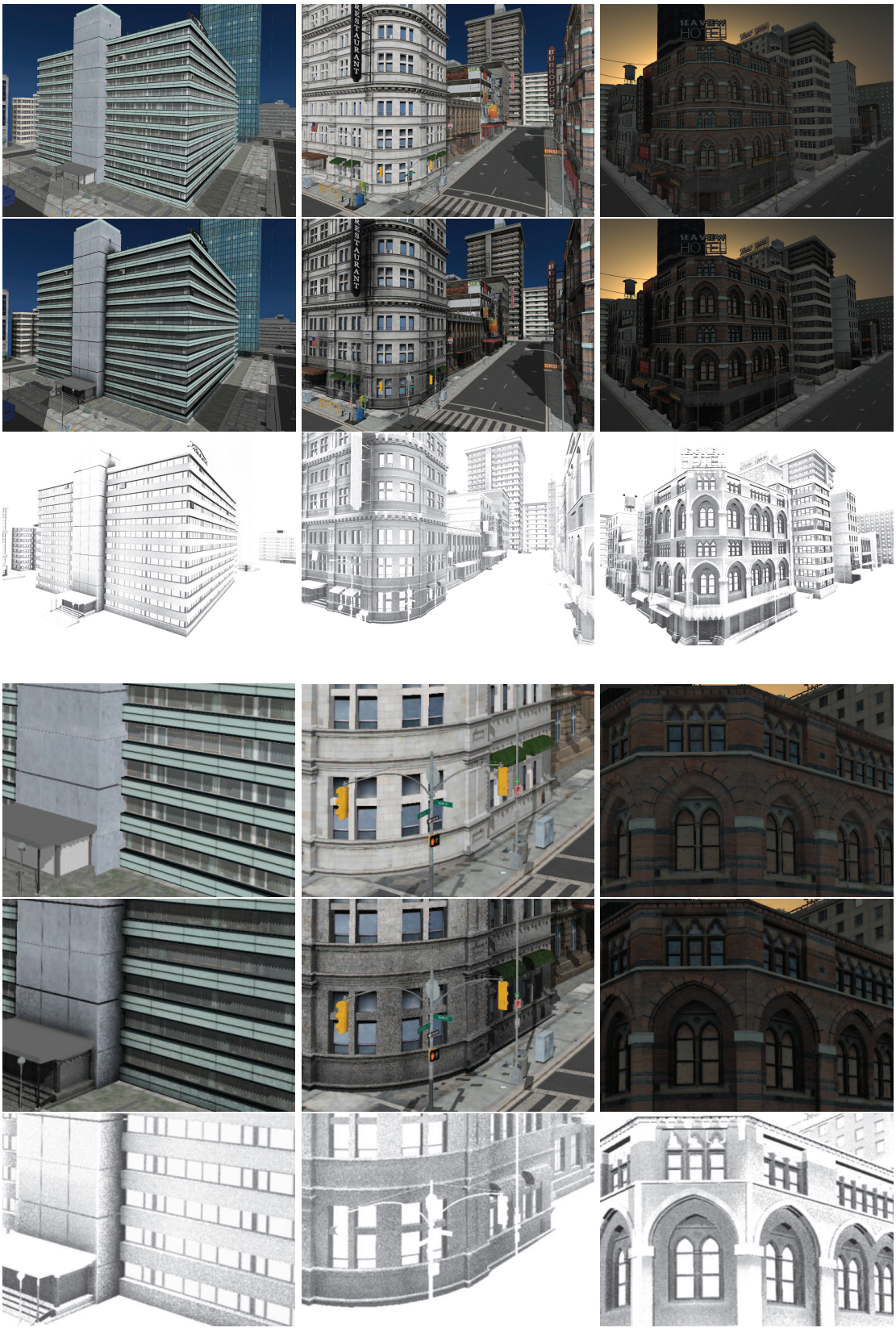


Figure IV.19: *Examples of polluted areas over a city. Top rows: views of buildings without pollution (top), with pollution (middle) and pollution layer of our simulation (bottom). Bottom rows: closeups comparing clean (top) and polluted surfaces (middle) and the corresponding pollution layer (bottom).*

(cc) BY-NC-ND

the facades more exposed to sun irradiance are less affected by pollution due to a reduction of humidity. When the user increases wind weight, the pollution is reduced uniformly except in the most protected zones: e.g. behind building features. Finally, when the importance of rain cleaning is enhanced, the facades are cleaned except where the hangovers protect them from droplets impact.

Figure IV.16 shows how our model is able to simulate plausible pollution deposition on buildings compared to real life situations. Pollution is not only distributed in a comparable way, but the changes on appearance of building materials are modeled in a very realistic way.

Figure IV.19 shows examples of the before and after cases (with its respective closeups) of pollution processes in several buildings of a city. One can see how buildings are polluted in a plausible way. Moreover, the closeups show the detailed pollution distribution and how it merges well with texture details. In order to make easier the judgement of our technique, in this image we also show the pollution layer on the geometry without any shading. It is noticeable how our technique achieves smooth results that increase the realism of the shaded images.

IV.6.3 Discussion

We have proposed for the first time a technique that is able to reproduce global changes of appearance of building facades due to pollution. Our technique is based in a new weathering paradigm that deals with the weathering process considering the scene and the environmental factors as a whole. We achieve that by computing the weathering factors globally and integrating them in a novel model. This allows us to take into account the interrelations between the different factors and to simulate realistically the final effects.

In addition, thanks to our screen-space design, we achieve interactive frame rates coupled with a low cost pre-computed pollution map. Although a full 3D wind simulation would generate a more accurate wind map and in turn a better initial pollution map, it would require much more computational time and memory consumption to store the resulting maps. Instead of this, we rely on a 2D representation that is later propagated into 3D using our height distribution function. Our CUDA implementation of LBM improves current wind simulations in urban scenes and, therefore, the computation of weathering factors. Moreover, we use it to compute our 2D pre-computed pollution map which proves to characterise the influence of buildings situation and alignment along a city and it allows to achieve realistic pollution distributions in comparison with an isotropic value. The height distribution function allows us to model on run-time the characteristic vertical reduction of pollution along the facades with a very low computational cost. A limitation of this approach is that wind flow will not change due to height differences in the buildings. This is however mitigated by the fact that pollution tends to concentrate in the lowest parts of buildings near the streets. Another limitation is that we are not able to take into account the vertical turbulences which are common in urban spaces. This kind of turbulences can only be considered in 3D fluid simulations which require a too high computational and storage cost.

Our deposition model quickly simulates the accumulation of pollution on facades based on the intervening factors. To accurately simulate a similar process, we would need a complex physical model that describes the interactions between pollution and materials. This model would involve a high computational cost and, moreover, the interaction between pollution agents, urban materials and environment factors has not yet been completely studied by physicists working in this field. Consequently, such kind of models are still not available nowadays and,

(cc) BY-NC-ND



Figure IV.20: *Example of deposition patterns produced by on-surface water flows.*

for this reason, there is no ground truth data to compare with. We propose instead a simple linear combination of weathering steps that approximate these complex processes and takes into account an important material property: the porosity. This approximation results in plausible images compared to real life photographs and allows us to achieve interactive frame rates.

Although our results are visually plausible, we do not face the washing and deposition processes typically produced by on-surface water flows. This type of processes are common in some polluted facades (see Figure IV.20) and their presence would further increase the results realism. The lack of this kind of effects could be mitigated using pre-computed textures transferred to the surfaces or developing a screen-space particle simulation in the line of Bosch et al. work [Bos+11]. However, while the former approach would not achieve realistic results, the latter would be too costly to achieve interactive frame-rates. A compromise between both options would be needed to satisfy quality and low cost requirements.

Finally, our model based on simple rules can be easily adapted to reproduce other weathering effects. If other factors are needed, it will require the design of screen-space methods to estimate them. But, once this is done, the design of new rules that describes the weathering steps is relatively straightforward. For this reason, we believe that screen-space approaches such as the one proposed here will open the door to simulate other weathering processes not directly related to pollution, particularly for big scenarios such as urban environments.

IV.7 Conclusions

In this chapter, we have presented an interactive method, based on screen-space weathering factors estimations, to compute and evaluate pollution depositions on complex urban scenes. In order to achieve that, not only we have used the solar exposure estimator presented in the previous chapter, but we have also extended it to deal with other factors estimation such as rain. Moreover, our method relies on a global model that takes into account the interactions between factors at a large, urban scale. Our system only requires a fast pre-computation process to compute a pollution distribution map that does not imply a high memory cost. This pre-computation uses a fast physically-based wind estimation to propagate the pollution particles. Thus, it proves to improve the pollution distribution taking into account the global properties of the city with respect to an isotropic value distribution. Our model is based on simple rules that simulate the different steps of the pollution deposition process by combining the involving factors estimated in run-time with material properties of the buildings. Thanks to it, our technique provides interactive feedback and allows the user to guide the final appearance of the buildings

(cc) BY-NC-ND

modifying several intuitive user-defined values that control the combination rules. Our system is able to simulate several situations and compares well against real images although it is not able to reproduce flow stains. Finally, we believe our technique can be easily adaptable to simulate other weathering processes globally. For this reason, we have confidence that the novel paradigm we have proposed will open a new way of simulating weathering effects taking into account the global conditions of the environment.

(cc) BY-NC-ND

(cc) BY-NC-ND

Chapter V. Stone Weathering

V.1 Introduction

Even though in the previous chapter we claim for the need of global techniques for generating weathered virtual urban environments, the aging effects at a local level need further research in order to be able to obtain really accurate images of weathered surfaces. Actually, the approach of the previous chapter obtains realistic results in a global scale and it links the effect (pollution in that case) with an upper scale. However, there is still a lack of enough local details as we may found in the real world (i.e. in the form of characteristic texture/spatial patterns, etc.). One of the main problems in weathering simulation is related to the complexity of the weathering processes. Furthermore, there is a lack of physical understanding in the appearance and evolution of an important number of effects and, consequently, simulation models are still unavailable in these cases. Thus, the common tendency in the recent years has been to transfer such details from real exemplars [Mer+06] [Lu+07]. The problem with such methods is that the selection of the weathering pattern to transfer is left to the user, which implies knowing which effect is present in the image. To the best of our knowledge, no efforts have been done on identifying weathering effects from exemplars. Although some techniques are able to easily separate weathered and unweathered regions [Wan+06], they are not able to detect which type of effect is present. In this chapter, we thus propose to analyze real weathered objects to identify the type of the present weathering effect. We have focused on flat stones as a starting point due to the clear features of its weathering processes. Studying stones, we will understand the underlying processes and will identify their common appearance characteristics. This analysis not only will improve the synthesis of realistic images, but also will be useful for other areas including weathering studies and Cultural Heritage.

Stone is generally considered to be one of the most durable materials. Nonetheless, it is susceptible to deterioration. In fact, processes that are operating in situ appear as deterioration patterns. They are the visible components of the decay which is a result of the interaction between the materials' intrinsic components and extrinsic factors. Decay patterns provide information in relation to decay mechanisms. In most cases these mechanisms come from a group of processes, i.e., physical, chemical and biological processes, that alter the physical and chemical state of the material in atmospheric conditions. Those processes may work individually, sequentially or simultaneously. In weathering studies the emphasis should be given to relationships between processes, weathering forms, material properties and environmental

(cc) BY-NC-ND

conditions. Holistic weathering research that includes intrinsic and extrinsic influences is vital for the better understanding of decay mechanisms. This kind of analysis is not only useful for the synthesis of weathering effects. The deterioration patterns and their depiction in images or measured drawings are essential tools for the conservation scientists to work with, and constitute the basic units to define the relationships between weathering types and decay mechanisms.

Mapping of Visual Weathering forms is an established non-destructive procedure for in situ studies on material damages and provides relevant information on the evolving decay mechanisms. This helps in assessing the present damages, damages to appear in the near future and the undamaged parts of the building materials. Therefore, priority measures such as which parts are to be conserved first and more intensively can be developed with the help of maps of Visual Weathering types. There are various descriptive schemes for the classification of weathering features on stone, most of which provide a framework for mapping the extent and nature of deterioration [War+03][SP07]. During the last decades, one of the tasks of the ICOMOS ISCS (International Council on Monuments and Sites - International Scientific Committee for Stone) has been to collect and combine the definitions that are in use for the Visual Weathering forms of stone to obtain a generalized glossary. While the “Illustrated Glossary of Stone Deterioration Patterns” [ICO08] of ICOMOS ISCS gives the explanations for different types of visual weathering, the assessments involve a high level of detailed surveying that leads to considerable demands on time and expertise. Moreover, the fact that the mapping process is based on observation and that it is a manual work, carries the possibility of getting different mappings for the same monuments. This possible disparity makes it difficult to track the weathering effects over time and to compare monitoring of the same monument made by different people. Furthermore, this need of expertise during the assessment and the possible disparity of criteria in experts’ mapping demonstrate that weathering processes are not still fully understood. For this reason, we think there is a need of automatic detection techniques to, on the one hand, improve the current understanding of weathering processes and, on the other hand, take advantage of this information in weathering simulation and texture transfer to achieve accurate realistic images.

In this work we focus on the weathering effect known as “scaling”. Following the ICOMOS definition, we consider scaling the decay effect where there is a detachment of a stone as a scale or a stack of scales, not following any stone structure and detaching either as fish scales or parallel to the surface. The plane of detachment is located near the stone surface and its thickness is generally from millimeters to centimeters. There are two sub-types of scaling. On the one hand, the thin flat or curved scales of thickness lower than 5 millimeters organized as fish scales are called *flakes*. On the other hand, the scaling subtype in which the scales are parallel to the stone surface are called *contour scaling* (or simply *scaling* from now on).

In this chapter, we propose a semi-automatic technique to identify, measure and map scaling effects on stone surfaces requiring very little interaction by the user. This identification is performed in 3D by using reconstructed flat objects, which will help in both their detection and measuring. Although we do not obtain interactive results, the key of this chapter is the identification of weathering features that characterize weathering processes and will allow us to develop interactive synthesis techniques in the near future. The main contributions of our framework can be summarized as follows:

- We propose an analysis framework that allows a better understanding scaling weathering effect.
- We introduce a pipeline to process the stone monuments that can be followed by any user

(cc) BY-NC-ND

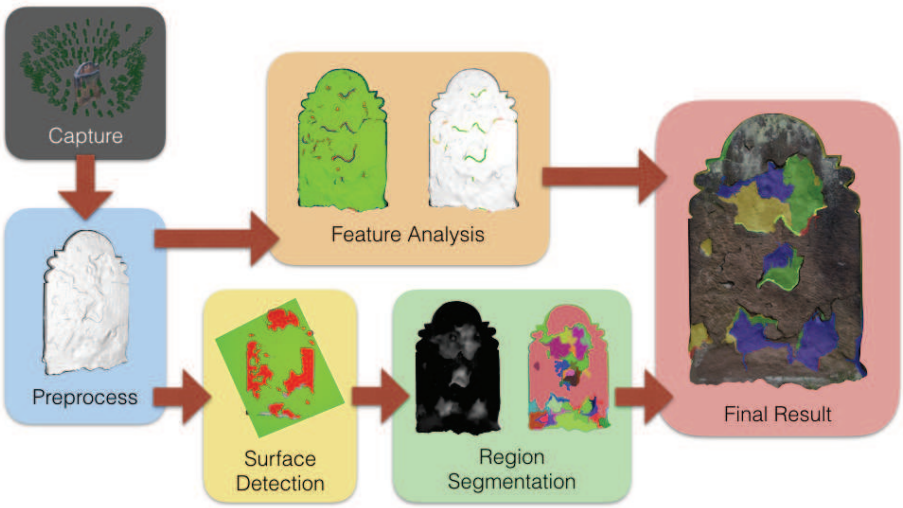


Figure V.1: *Overview of our technique: First, we capture a scene using photogrammetry and we generate the corresponding 3D model. Then, we smooth this model in a preprocess in order to reduce acquisition noise. After that, on the one hand, we perform a geometrical analysis to identify the features related to scaling weathering effect. On the other hand, we estimate the unweathered surface and we segment model regions according to their distance to it. Finally, we map the weathered zones on the model using the obtained segmentation and the detected weathering features.*

without any expert knowledge.

- Our method is able to detect and map scaling weathering effects in a semi-automatic way. Moreover, it can classify the detected effects in two subtypes.
- Although our technique has some limitations, we consider it as a first step towards an objective stone decay mapping.
- Our framework is useful for Cultural Heritage preservation projects and it will be useful in the synthesis of images of scenes with weathering effects.

The rest of this chapter is organized as follows. In the next section we present an overview of our pipeline. The following sections focus on the capturing of the stone models and the processing that the model requires before being analyzed. After that, we describe how we recover the unweathered mesh and how we identify and map the scaling processes. Finally, we present some results and discussion.

V.2 Overview

The overview of our method is depicted in Figure V.1. We start by reconstructing a 3D model of the weathered object using photogrammetry techniques. After that, we need to process the reconstructed scene to isolate the model in a manifold mesh. Then, we apply a preprocessing to the model in order to reduce the typical noise that appears in the reconstruction. We propose using an adaptive anisotropic Gaussian filter. Once the model is processed, we estimate the unweathered surface using a RANSAC technique. In this estimation, we use the point cloud of a segmentation started at a point on an unweathered zone of the surface provided by the user. Then, we perform a geometrical analysis of the curvature of the mesh to find

(cc) BY-NC-ND

characteristic features of scaling weathering effects. Finally, we map the weathered regions using an adapted image-space segmentation based on the distance between the mesh and the estimated unweathered surface; and we classify them using the detected weathering features. This results in a set of weathered regions and features that are classified according to the corresponding sub-type of scaling.

V.3 Model Generation

V.3.1 Capturing Models

There are several methods to capture a real-world scene and generate a 3D model out of it. Although 3D scanners are able to reconstruct models achieving a very high quality, they are expensive and require trained people to use them. For this reason, photogrammetry techniques are taking a more prominent role. They only require a digital camera and basic training to achieve good results. To simplify the capturing step, thus we use photogrammetry.

We use a digital camera to take a set of pictures around the object we want to reconstruct. In order to assure a high quality reconstruction, we take the pictures on overcast days. This makes us to reduce the possible discontinuities in the illumination of the model. Moreover, to guarantee a complete high quality reconstruction, we take around 150-200 pictures per object following circles around the object at different heights. Once all the images are taken, we generate a 3D model from them using Autodesk ReMake. The process consists of providing the pictures to the software and after an automatic process we obtain a 3D reconstruction of the scene. Since our method works better with an isolated manifold model to work, we use the provided software tools to remove the parts of the scene not corresponding to the target model and to fill the holes of the unreconstructed parts. Figure V.2 illustrates this process. Finally, we calibrate the scale of the reconstructed model using measured distances between points on the real model.

V.3.2 Pre-processing the Model

One inherent problem of the acquisition step is the presence of noise in the reconstruction process, and photogrammetry tends to produce models with noise. Such noise complicates the analysis and understanding of the surface and, consequently, the detection of weathering effects. For this reason it is very important to reduce its presence as much as possible.

Smoothing is a good strategy to reduce the presence of noise in a mesh. In general, smoothing techniques use a constant smoothing term over the whole surface which can deform the mesh and remove important information. To avoid that, an adaptive method is needed that preserves the features of the original mesh reducing the noise elsewhere. The challenge is to decide the degree of smoothing at each point. To handle it we perform an adaptive and anisotropic Gaussian filter based on the Ohtake et al. technique [OBS02] to smooth the normals of the mesh. After that, we use these smoothed normals to subsequently smooth the vertices of the mesh.

First, we compute the best smoothing scale for each face normal by analyzing the neighboring area. We compute the best standard deviation σ of the Gaussian filter around each normal using the following equation:

$$\sigma_{best} = \arg\min_{\sigma} \frac{c}{\sigma^2} + \epsilon^2(\sigma) \tag{V.1}$$

(cc) BY-NC-ND

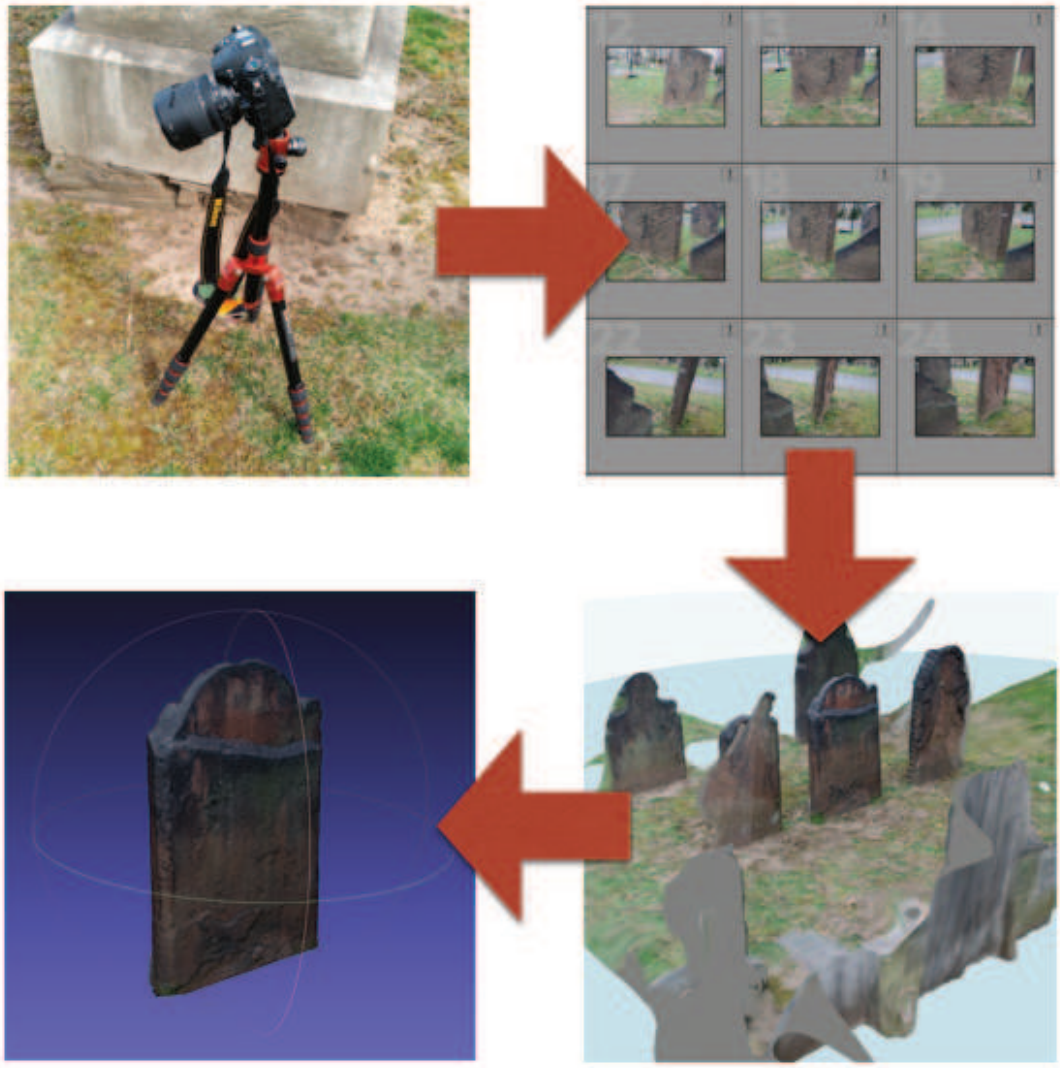
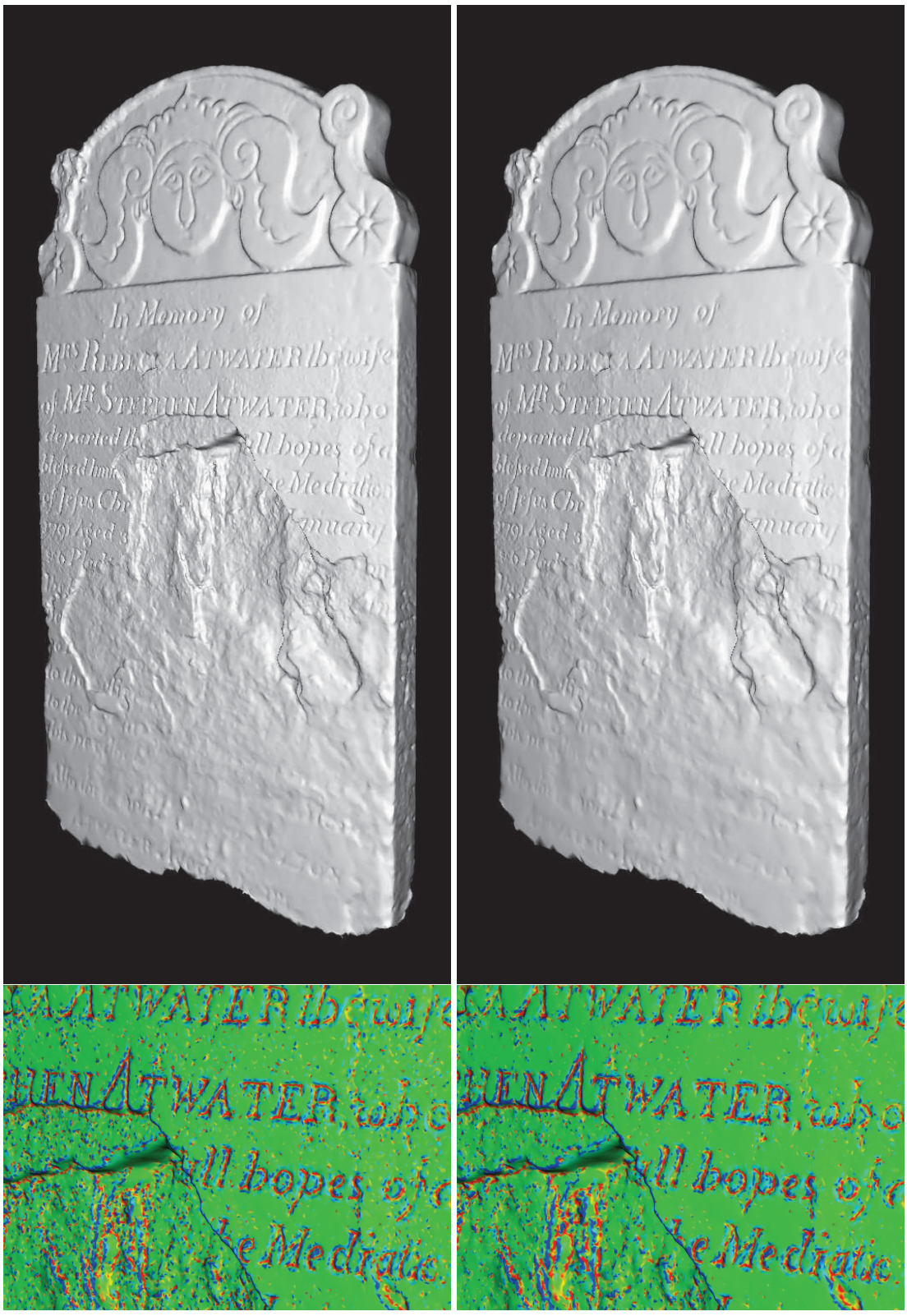


Figure V.2: *Our method requires an isolated 3D manifold model to work properly. Generating it consists on taking pictures, loading them to Autodesk ReMake, which generates a 3D reconstructed scene and, then, separating the target model and filling its holes.*

(cc) BY-NC-ND



(a) *Pre-Filter* **(b)** *Post-Filter*

Figure V.3: *We perform a smoothing filter to reduce the acquisition noise of the reconstructed model. Here it shows a comparison before and after applying the filter (top) and zoom-in's showing the obtained color-coded curvature in each case (bottom).*

(cc) BY-NC-ND

where $c = 4.0 \times 10^{-3} l^2$, l is the arithmetic mean of the edge lengths of the mesh and ϵ is the local variance of the smoothed normal using a Gaussian filter with a σ standard deviation. To find σ_{best}, we use 10 uniformly spaced values of σ from $\sigma_{min} = 0.5l$ to $\sigma_{max} = 5.0l$.

Then, the smoothed normals n'_σ are computed as:

$$n'_\sigma = \frac{\sum w_j n_j}{|\sum w_j n_j|} \tag{V.2}$$

where n_j are normals of the neighbor triangles at a geodesic distance $d \leq 4\sigma_{best}$ and w_j are the weights applied onto the normals which take into account the area of the corresponding triangle T_j and a Gaussian function $K_{\sigma_{best}}$ based on d:

$$w_j = A(T_j) K_{\sigma_{best}}(d). \tag{V.3}$$

.

More specifically, the Gaussian function $K_{\sigma_{best}}$ is defined as follows:

$$K_{\sigma_{best}}(x) = \frac{1}{\sigma_{best}\sqrt{2\pi}} \begin{cases} e^{\frac{x^2}{2\sigma_{best}{}^2}}, & \text{if } |x| \leq 2\sigma_{best} \\ \frac{1}{16e^2}(4 - \frac{|x|}{\sigma_{best}})^4, & \text{if } 2\sigma_{best} < |x| \leq 4\sigma_{best} \\ 0, & \text{otherwise} \end{cases} \tag{V.4}$$

where $e = 2.71828...$ is the base of natural logarithms. Note that $K_{\sigma_{best}}$ is a Gaussian-like function with the tails splined to zero.

Once the normals of all faces of the mesh have been smoothed, we smooth the mesh by updating the vertex position as proposed by Sun et al. [Sun+07b]. According to this, we update each vertex position x' following the equation:

$$x'_i = x_i + \frac{1}{|F_V(i)|} \sum_{k \in F_V(i)} n'_\sigma(k)(n'_\sigma(k) \cdot (C_F(k) - x_i)) \tag{V.5}$$

where x_i is the original position of the vertex, $F_V(i)$ is the set of surrounding faces of the vertex i, $n'_\sigma(k)$ is the smoothed normal of the face k and $C_F(k)$ is the center of the face k. Figure V.3 shows the results of applying this smoothing to an example model. We can observe how the level of noise is considerably reduced while the features of the mesh are still preserved.

V.4 Original Surface Estimation

As mentioned before, the scaling weathering process is characterized by the loss of material on the surface. For this reason, in order to be able to detect this loss we first need to estimate the shape of the model before the decay process.

First we detect what parts of the surface have not suffered any weathering. To achieve that, we ask the user to mark a unique point on the surface which has not suffered any weathering process. Taking as input the face where this point lies, we proceed to segment the mesh by extending the segmentation taking into account the normal of the face and the mean normal of the already segmented region following the next algorithm:

function SEGMENTBASESURFACE($face$)

(cc) BY-NC-ND

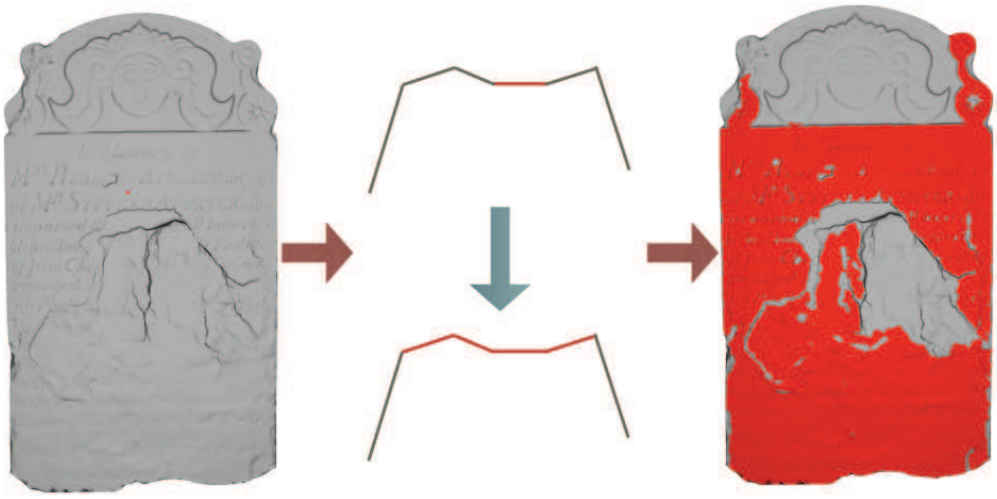


Figure V.4: *Segmentation of the original mesh. After the click of the user (left), a segmentation process is applied in function of the faces normal (center) until a final segmentation is obtained (right). Note that center image shows a schematic side view of the object and that red mean a selected face.*

$segmentation \leftarrow face$
$visitedFaces \leftarrow face$
$candidates \leftarrow face.neighbors()$
$meanNormal \leftarrow segmentation.meanNormal()$
while $!candidates.empty()$ **do**
  $f \leftarrow candidates.pop()$
  if $!visitedFaces.contains(f)$ **then**
    $visitedFaces \leftarrow visitedFaces \cup f$
    if $dot(f.normal(), meanNormal) > threshold$ **then**
      $segmentation \leftarrow segmentation \cup f$
      $meanNormal \leftarrow segmentation.meanNormal()$
      $candidates \leftarrow candidates \cup f.neighbors()$
return $segmentation$

Figure V.4 illustrates this segmentation process (center) showing the resulting segmentation (right) after a user click (left).

Once the base surface is segmented, we use the vertices of the segmented faces to estimate the original surface S_{base}. Although several approaches could be suited to estimate a shape, we use a method based on an efficient RANSAC technique proposed by Schnabel et al. [SWK07]. This method is fast, it handles high levels of noise and it can reconstruct typical hand-made shapes. This technique computes a set of primitive shapes given a point-cloud with its associated normals. At each iteration of their algorithm, the primitive with maximal score is searched using the RANSAC paradigm: we generate candidates of all the primitive shapes sampling randomly minimal subsets of the point cloud, then we chose the candidate with the highest score and we only accept it if the probability that no better candidate was overlooked during the sampling is

(cc) BY-NC-ND

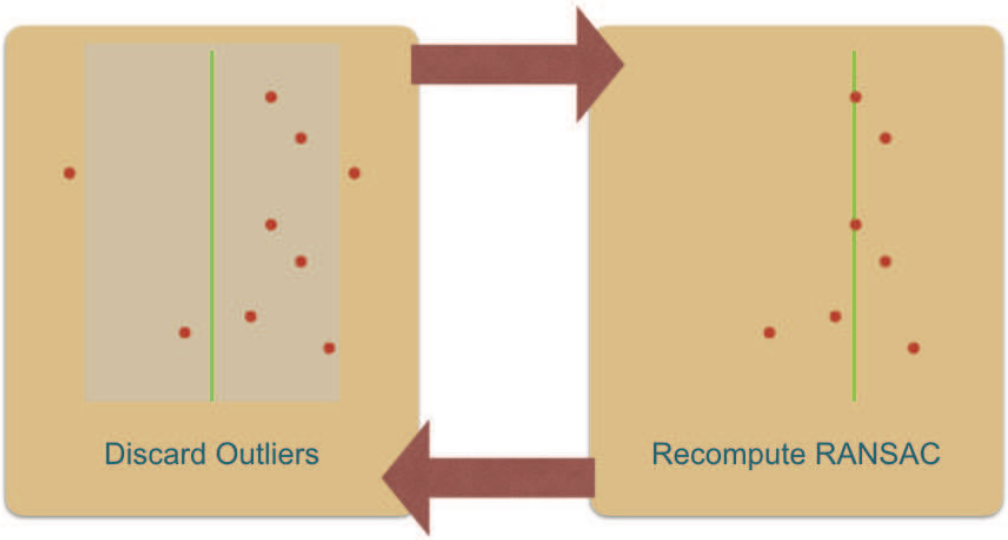


Figure V.5: *The iterative refinement of the surface estimation consists of erasing the outliers and estimating original surface again using RANSAC until no outliers are detected.*

high enough. In the current implementation we only consider plane shapes in RANSAC because our experiments has focused on flat stones.

Figure V.6 shows the result of the RANSAC method in the subfigure (b). As the reader can observe, the detected shape does not fit perfectly the original surface S_{base}. This is because although the RANSAC method is able to handle a high percentage of outliers, its recovered surfaces can be shifted for points that are not on the original surface S_{base}, but not far enough away to be considered outliers.

In order to solve that, we propose to use an iterative algorithm where we erase the near outliers from the point cloud. The objective of this iterative algorithm is to improve the surface estimation by recomputing RANSAC after removing the outliers detected on the previous RANSAC surface estimation. Figure V.5 illustrates this idea. More specifically, given a first RANSAC reconstruction, at each iteration we erase the points which are farther than a user-defined threshold $\Delta_{outliers}$ and we re-estimate the surface until all the points are inside another user defined threshold $\Delta_{surface}$. The resulting surface S_{base} will be considered as the unweathered surface of the model from this point on. This process is described in the following algorithm:

function FINDSURFACE($pointCloud$)
  repeat
    $shape \leftarrow EfficientRANSAC(pointCloud)$
    $outliers \leftarrow false$
    for all $p \in pointCloud$ **do**
      if $p.distance(shape) > \Delta_{surface}$ **then**
        $outliers \leftarrow true$
        $break$
    if $outliers$ **then**
      for all $p \in pointCloud$ **do**

(cc) BY-NC-ND

if $p.distance(shape) > \Delta_{outliers}$ **then**
$pointCloud.remove(p)$
until $outliers = 0$
$S_{base} \leftarrow shape$
return S_{base}

Figure V.6 shows the result of this iterative process in subfigure (d) and the corresponding segmentation in subfigure (c). Although the refined segmentation is good enough to be used with RANSAC to estimate the original surface S_{base}, it still selects parts of the weathered region of the model. For this reason, we avoid using this segmentation in the next steps of our technique in which we use the estimated surface S_{base}.

V.5 Weathered Zones Detection

After estimating the original surface S_{base}, we can analyze the differences between the mesh and the reconstructed surface in order to find loss of material during the scaling weathering process. We propose to classify these differences into weathering zones by taking into account the characteristics of the detached regions trying to figure out where the main features are placed and by mapping the affected regions consequently.

V.5.1 Feature Filtering

As mentioned previously, the scaling weathering process is related to the layers of the stone. After the layers fall, they typically show a staircase effect in the margins of the lost area. Moreover, we can use the difference of depth in these staircase steps as an indicator of the subtypes of scaling. For this reason, it is important to detect these steps in the surface.

We propose to use the mean curvature as an indicator of the features of the mesh that can be considered as scaling features. First, we compute a local mean curvature at the vertices of the mesh. Then, we process this curvature to consider only the vertices where the curvature is maximum or minimum. More specifically, we tag as maximum curvature vertices (CV_{max}) those in which the curvature is greater or equal to the percentile 85. On the other hand, we tag as minimum curvature vertices (CV_{min}), those in which the curvature is lower or equal to the percentile 15. These values give the best results according to our experiments.

We need to find the steps associated with the layers, but since the current approach only takes into account the value of the mean curvature, we can be still considering the remaining noise in the model. To avoid that, we propose filtering the noise, untagging all the vertices tagged as minimum or maximum curvature vertices when the number of connected vertices with the same tag is lower than a user defined threshold tv_{min}. In our experiments, we have seen that setting $tv_{min} = 5$ is enough to achieve good results. This simple idea proved to be very useful to remove the noise while preserving the main features where the curvature is minimum and maximum. We use these tagged features to identify the scaling effects. Figure V.7 illustrates this process, showing, on the left, the mean curvature; on the center, the vertices where curvatures is maximum or minimum tagged; and, on the right, the tagged features after applying the proposed filter.

(cc) BY-NC-ND

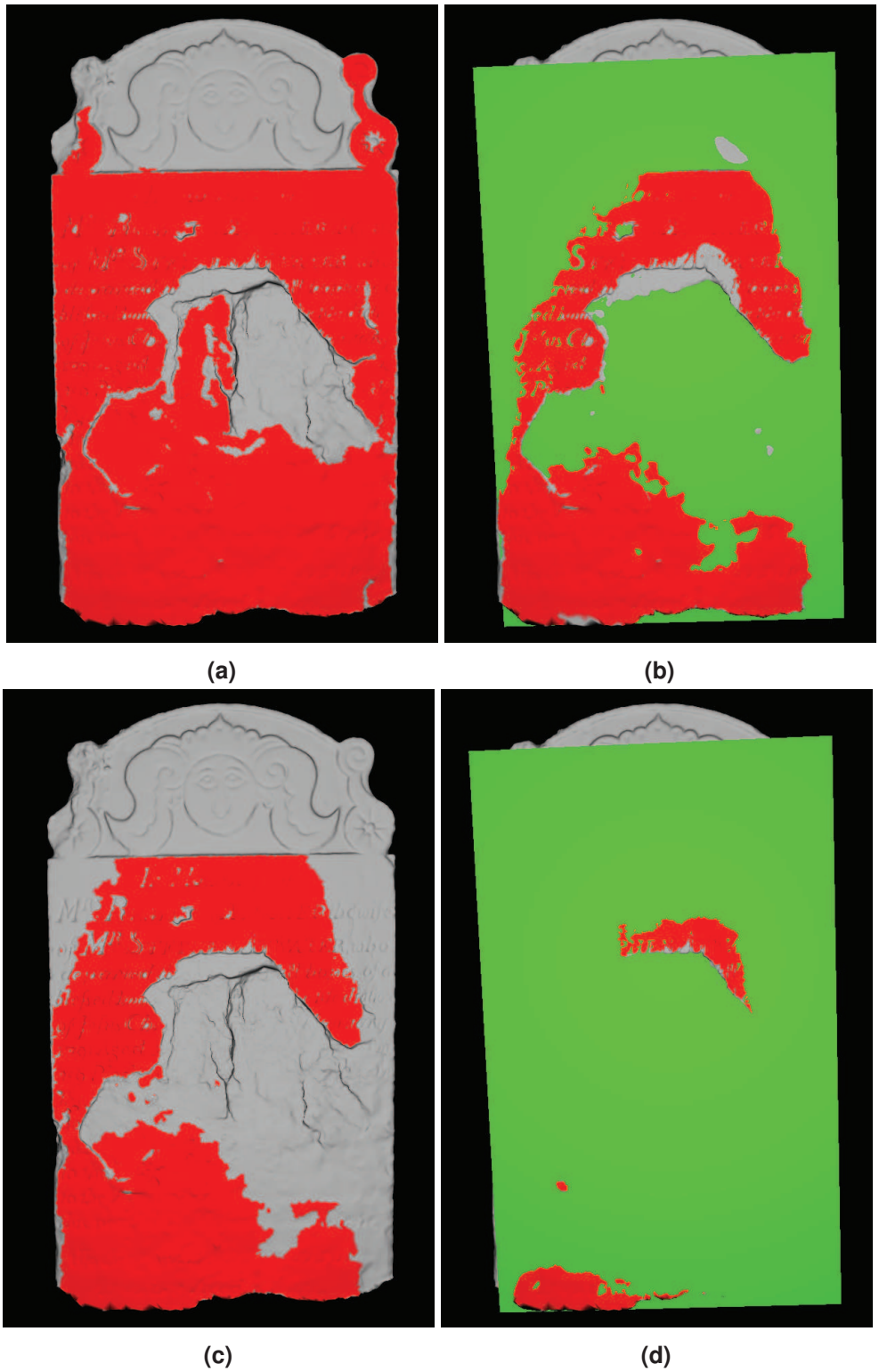


Figure V.6: *In order to detect the loss of material in the model, we first need to estimate the corresponding unweathered surface. This is done (a) by performing an initial segmentation starting from an unweathered point marked by the user. Then, we perform a RANSAC process to obtain the unweathered surface estimation (b). This process is iteratively refined removing outliers from segmentation and computing RANSAC again, until no outliers are segmented (c) and the definitive unweathered surface is found (d). We use red color to indicate the segmented faces and green color to show the estimated surface.*

(cc) BY-NC-ND

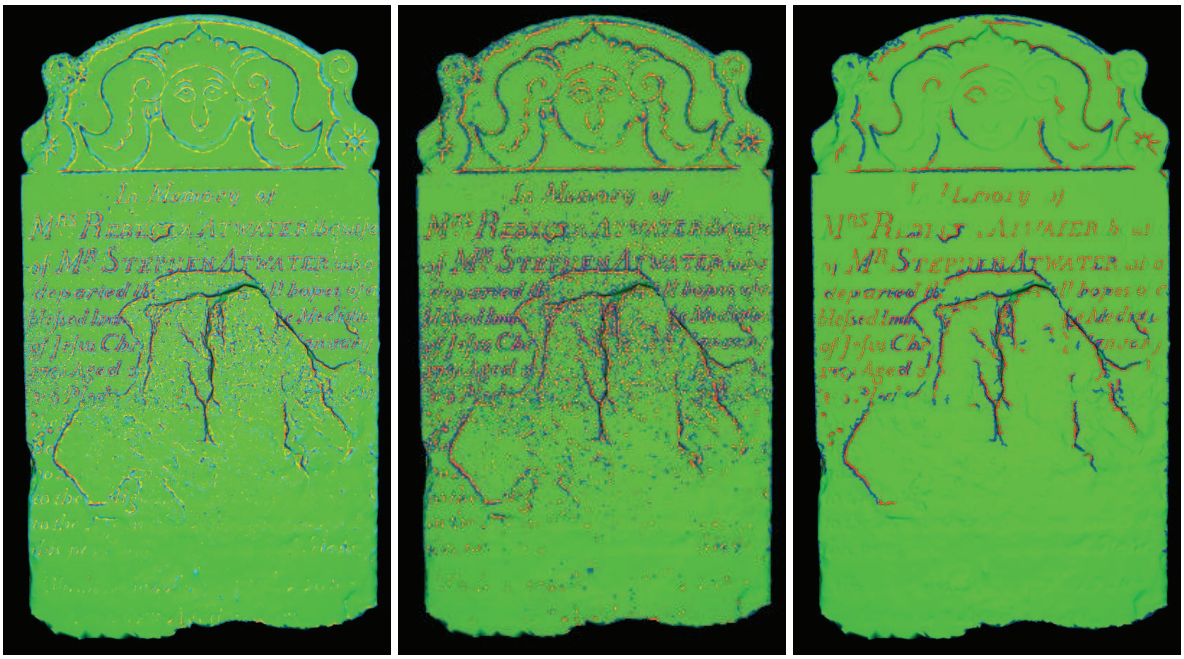


Figure V.7: *Scaling weathering effects are related to the geometrical features of the model. In order to detect them, we use the mean curvature (left) to identify the vertex where it is minimum and maximum (center) and we process it taking into account the highest and lowest values and filtering the small features (right).*

V.5.2 Detecting Scaling Weathering Effects

After tagging the features, we can use them to detect the borders of the scaling weathering effects. As we have said, one of the most typical signs of a scaling feature is a staircase effect in the margins of the lost area. Our idea is to use close features with an opposite tag to detect the typical steps of this staircase effect. Moreover, if we analyze the depth differences between the tagged features, we will measure the depth of the step and we can use it to classify the feature in the two subtypes of scaling processes.

In order to identify the vertices corresponding to these steps, we propose to go through the tagged vertices and check if there are opposite tagged vertices nearby. We check that by going through the neighbor vertices that are closer than a user defined distance $distance_{feat}$. We compute the distance between two vertices projecting them to the estimated original surface S_{base} and computing the geodesic distance between the projections on that surface. After that, we compute the difference of depth between their two vertices Δ_{depth}. If Δ_{depth} is greater or equal than 0.5cm we consider that the current vertex is part of a scaling weathering feature and, otherwise, we consider it as part of a flake. We use this value after in situ observations taking into account the general tendency of the layers separation. Our values apply to the sandstone we are examining which is used widespread in the chosen region (Connecticut, USA). But, it can be adapted to other types of stone used in other regions. Our classification is not exclusive, some vertices can be considered part of a scaling and of a flake. A reason for that could be the presence of flakes on the same wall of a scaling weathering step. Moreover, in our experiments we have observed that setting $distance_{feat} = 1$cm is enough to detect most of the features. This process is described more specifically in the following algorithm:

(cc) BY-NC-ND

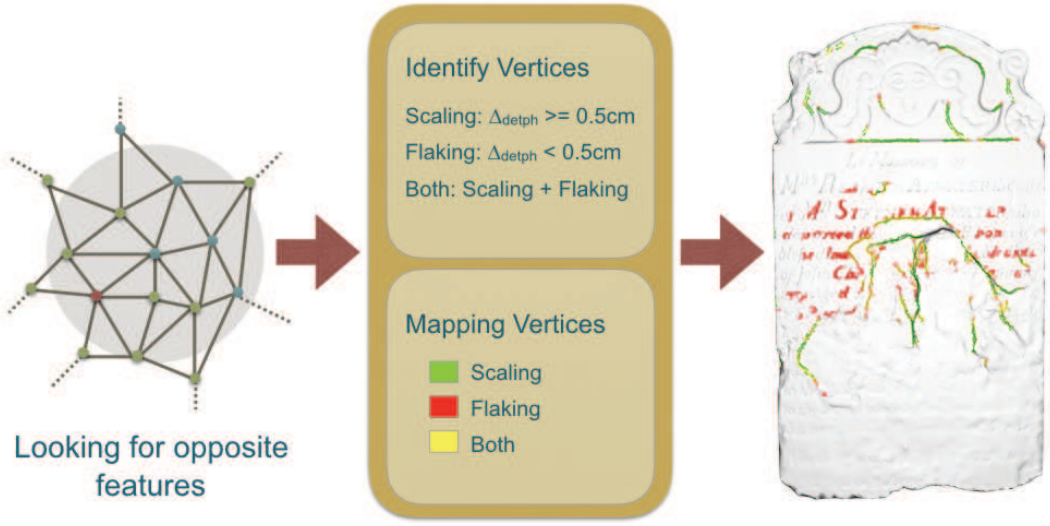


Figure V.8: *We detect the scaling features looking for opposite features in the near vertices after projecting them to the original surface (center). Then, we use the difference of distances to the original surface (center) as an indicator of scaling subtype of the surviving vertices (right).*

function CLASSIFYSCALINGVERTICES($vertices, S_{base}$)
  for all $v \in vertices$ **do**
    $v_{proj} \leftarrow S_{base}.projection(v)$
    $candidates \leftarrow v.neighbors()$
    for all $c \in candidates$ **do**
      $c_{proj} \leftarrow S_{base}.projection(c)$
      $dist_{v,c} \leftarrow S_{base}.geodesicDistance(v_{proj}, c_{proj})$
      if $dist_{v,c} \leq distance_{feat}$ **then**
        $candidates \leftarrow candidates \cup c.neighbors()$
        if $(v \in CV_{max} \wedge c \in CV_{min}) \vee$
         $(v \in CV_{min} \wedge c \in CV_{max})$ **then**
          $depth_v \leftarrow S_{base}.distance(v)$
          $depth_c \leftarrow S_{base}.distance(c)$
          $depth_{v,c} \leftarrow abs(depth_v - depth_c)$
          if $depth_{v,c} \geq \Delta_{depth}$ **then**
            $v.scaling \leftarrow true$
          else
            $v.flaking \leftarrow true$

Figure V.8 illustrates the process to detect the scaling features. In order to identify the sub-types of scaling effects present on the model, our technique analyzes the vertices previously detected as features related to the scaling effect shown in Figure V.7. We only consider the nearby vertices after being projected onto the estimated surface (left). The color code of the vertices is the following: red for the flakes, green for the scaling and yellow for the vertices that are part of weathering scales and flakes (center). It classifies them using the difference of depth with respect to the unweathered surface between the vertex and its neighbors. Vertices without

(cc) BY-NC-ND

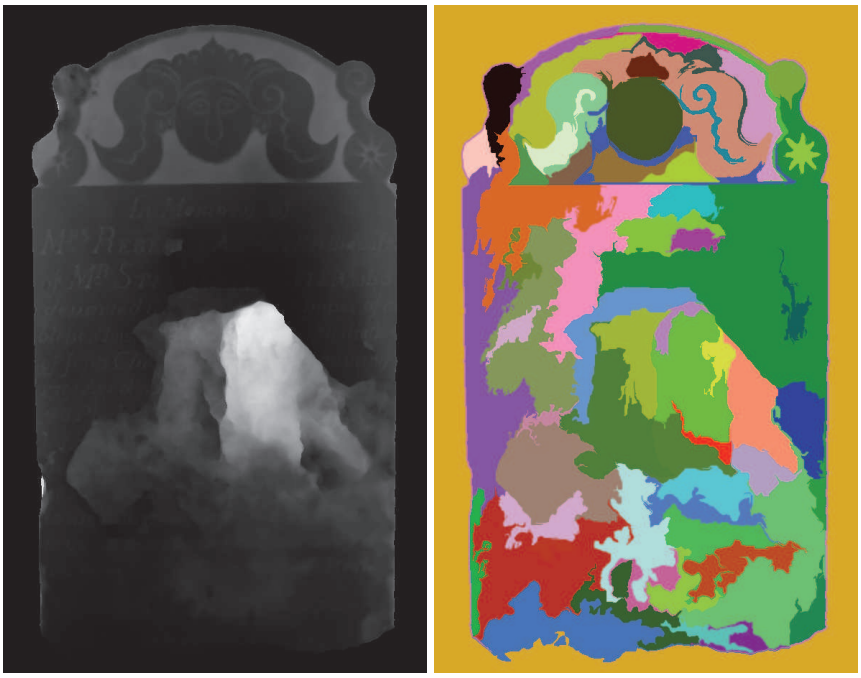


Figure V.9: *In order to detect regions affected by the scaling weathering effect, we first need to identify the different regions of the model. Our region segmentation uses a map of the distance between the model and the estimated unweathered surface (left) and computes the region segmentation in image-space (right).*

nearby opposite tagged vertices are discarded as part of scaling features and will not be taken into account in the following steps (right).

V.5.3 Segmentation of Weathered Regions

At this point, we are able to detect the vertices that belong to the typical steps of the scaling weathering processes and we are even able to classify them in subtypes. But, we are not yet able to map the regions of the surface affected by this weathering effect as it is traditionally done in Cultural Heritage. Mapping the regions is important because it gives us the information about the area affected and its spread. Moreover, this is the kind of data we need to understand and process in weathering simulations and/or texture transfer.

To solve that we propose segmenting the surface in regions by adapting the image-based segmentation approach of Felzenszwalb and Huttenlocher [FH04]. This technique is able to segment an image comparing the resemblance between the different pixels. The authors measure the evidence for a boundary between two regions using a graph-based representation of the image. The edges of the graph are assigned with a weight that measures the dissimilarity between neighbors. Then, they join regions of the image if a boundary is not found. A boundary appears when the minimum weight between the elements of the border of both regions is higher than the minimum internal weight between the elements of each region. This simple predicate results in a fast solution that preserves the features of the images. Although other segmentation techniques could be used, we have chosen this technique because it is fast and it allows us to take into account the depth (distance to S_{base}) as a segmentation condition. As mentioned before, the presence of scales parallel to the original surface is one of the properties of scaling

(cc) BY-NC-ND

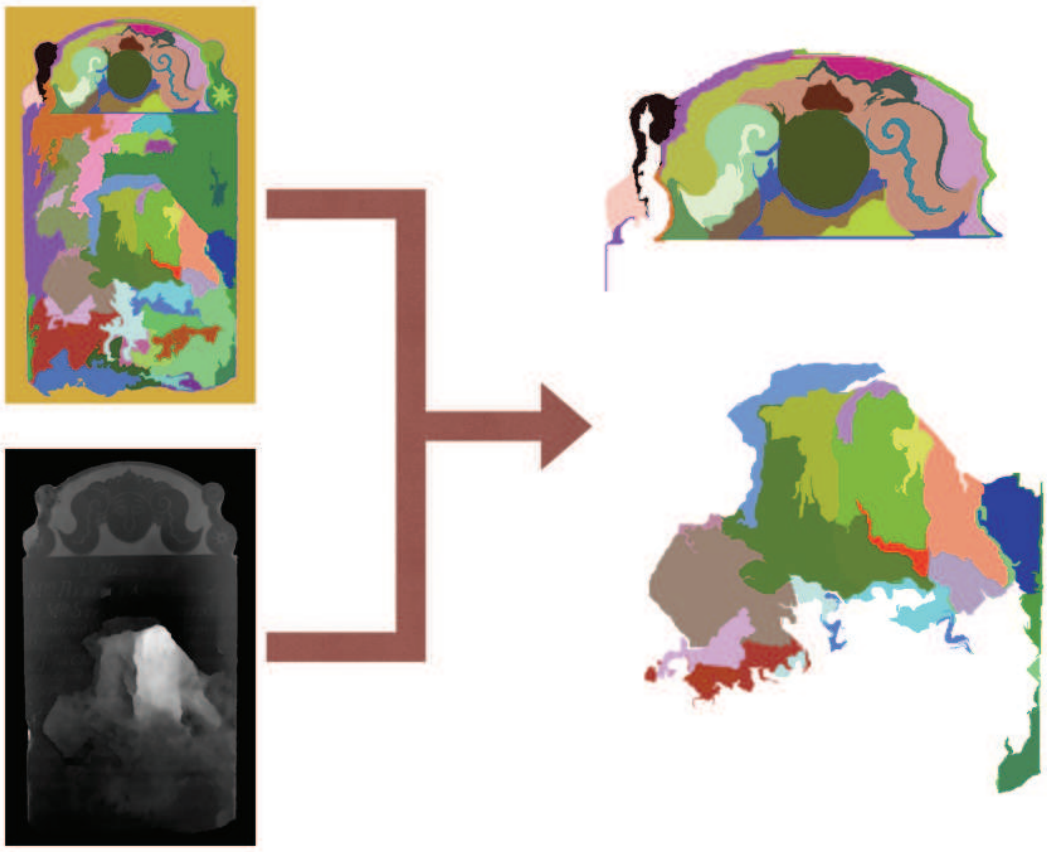


Figure V.10: *The segmented regions (top-left) need to be culled in order to obtain a map of weathered zones (right). We perform it by discarding the regions in which mean distance to the original surface, previoulsy computed in the depth map (bottom-left) is lower than* $dist_{min}$.

weathering effect.

In our case, we have a 3D mesh instead of a 2D image, and we are interested in segmenting regions based on depth rather than color. For this, we need to transform our 3D model to a 2D image that depicts the distance to the original mesh. Thus, we represent each surface point in terms of its distance to the original unweathered surface S_{base}, resulting in a depth map over this surface. Then, we generate a 2D image of this depth map using a direct parametrization, and we use this map as input to the segmentation algorithm.

Figure V.9, on the left, shows an example of the generated depth image used as input to the image segmentation algorithm and, on the right, the resulting segmentation. As can be observed, this depth-based segmentation follows the main characteristics of the model shape.

V.5.4 Mapping the Regions

At this point, we have all the information needed to map the scaling weathering effect onto the model. First, we need to process the segmentation of the surface to discard the segmented regions of the unweathered parts. The goal is to discard the segmented regions which belong to the unweathered zone. To achieve that, we perform a culling of the segmented regions taking into account again their distance to the estimated unweathered surface (S_{base}). Such distance is

(cc) BY-NC-ND

computed as the mean depth of all the points inside of each region (R). We define this culling process following this rule:

$$Culling(R) = \begin{cases} true, & \text{if } mean(S_{base}.distance(p_i)) < dist_{min} \\ false, & \text{otherwise} \end{cases} \quad \text{(V.6)}$$

where $dist_{min}$ is a user defined threshold. In our experiments we have observed that we achieve good results using $dist_{min} = 1$cm. However, this value can be adapted taking into account the formation properties of the studied stone. This culling process takes into account the segmented regions instead of evaluating each point individually, thus avoids considering as weathered region most of the carvings of the model (which fall on regions whose mean distance is under $dist_{min}$). Figure V.10 depicts this culling process and shows the detected weathered zones of the example.

After that, we propose tagging the segmented regions that have not been discarded during the culling taking into account the identification of the scaling features described in Section V.5.2. In this way, for each region we check if the vertices inside that region contain detected scaling and its sub-type. We define the tagging of each region (R) following this rule:

$$Tagging(R) = \begin{cases} Scaling, & \text{if } \forall v \in R : v \in \mathcal{C}_s \vee v \in \mathcal{C}_u \\ Flaking, & \text{if } \forall v \in R : v \in \mathcal{C}_f \vee v \in \mathcal{C}_u \\ Both, & \text{if } \exists v \in R : v \in \mathcal{C}_b \\ Unknown, & \text{otherwise} \end{cases} \quad \text{(V.7)}$$

where $\mathcal{C}_s$ is the set of vertices that are tagged only as scaling, $\mathcal{C}_f$ is the set of vertices tagged only as flaking, $\mathcal{C}_b$ is the set of vertices that are tagged as flaking and scaling, and $\mathcal{C}_u$ is the set of untagged vertices.

Figure V.11 illustrates the final classification (right) of the segmented regions (left) that have survived to the culling process depicted in Figure V.10. The regions where we only detect scaling are displayed in green, the regions where we detect flaking are in red, the regions with both effects are shown in yellow and, finally, the regions where we found detachment but they are not tagged with any type of scaling are colored blue.

V.6 Results and Discussion

V.6.1 Results

We have tested our approach on different models, that belong to ancient graves made of sandstone. They form part of Grove Street Cemetery, one of the earliest burial grounds placed at New Haven, Connecticut, USA. The sandstone of the graves is a kind of sandstone widely used in the region, so most of the user configurations used in this chapter could be used with other models of the area. All the studied graves are more than 100 years old and weathering processes are very present on them. We have reconstructed all of them with the same setup. We used a Nikon D7000 digital camera mounted on a tripod, with a fixed lens with a focal length of 18mm. We took pictures at maximum resolution (4928x3264). Table V.1 shows the characteristics of the models and the performance we obtain, for an unoptimized code, on a MacBook Pro 15" with an Intel Core i7 2.6GHz, 16GB memory and a NVIDIA GeForce GT 750M with 2GB memory. We have used large sets of pictures to get models with enough quality

(cc) BY-NC-ND

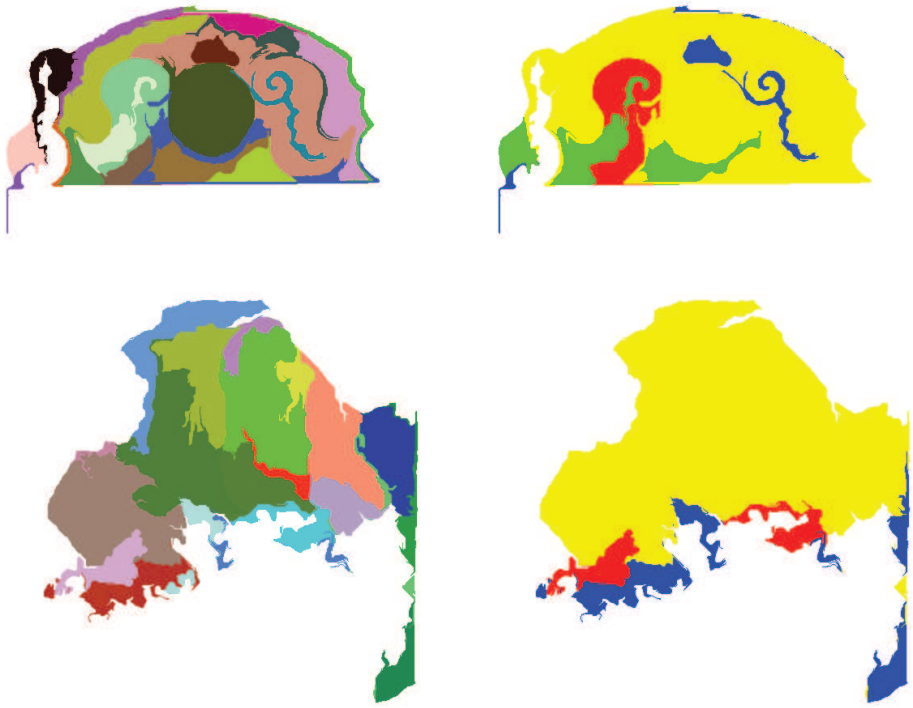


Figure V.11: *From the surviving regions of the previous culling process (left), we need to tag them according to the sub-type of scaling effect present on them (right) taking into account the scaling weathering features.*

following the guidelines provided by Autodesk ReMake. Despite the number of triangles, most of the processes are performed in seconds. The smoothing of the mesh takes some minutes, but is only computed once.

Figures V.12 to V.16 show the potential of our technique. They consist of a set of comparisons between the results of our technique and the mapping performed by an expert in Cultural Heritage. The left image of each figure shows the original reconstructed model with its original texture. In the middle, we show the model with the mapping of the detected weathering using our approach. The weathered regions detected and classified by our method (see Sections V.5.3 and V.5.4) are mapped on the corresponding model blending the scaling subtype color with the original texture to help the viewer to judge our mapping. Moreover, the weathering features used in the classification of each region (see Section V.5.2) are highlighted to provide extra information to the user. This helps to judge the results of our technique and to give additional feedback of the location of the weathering features and their relation with the weathered regions. Finally, the right image of each figure corresponds to the mapping performed by a stone preservator expert member of the Yale University Cultural Heritage Department who took several minutes analyzing and mapping the models. In middle and right images, we have used the same color code: green for scaling, red for flaking, yellow for the zones where there is scaling and flaking and blue for the detached regions where no scaling/flaking processes are detected. The width of these models are in the range between 57cm for Model depicted in Figure V.13 and 85cm for Model depicted in Figure V.12.

(cc) BY-NC-ND

Model	#Pictures	#Triangles	Smoothing (s)	Unweathered surface (s)	Scaling features (s)	Map regions (s)
1	188	201K	181.50	0.13	2.12	4.89
2	165	193K	237.87	0.28	2.75	4.91
3	166	427K	612.42	23.97	5.73	4.94
4	123	173K	157.52	0.88	2.03	4.73
5	214	167K	168.68	4.36	1.97	4.83

Table V.1: *Performance of our technique for different models. For each model, we show the number of pictures used in the reconstruction, the number of triangles of the reconstructed 3D model and the performance in seconds of each step of our technique: smoothing the model (Section V.3.2), estimating the unweathered surface (Section V.4), detecting the scaling weathering features (Sections V.5.1 and V.5.2) and mapping the weathered regions (Sections V.5.3 and V.5.4).*

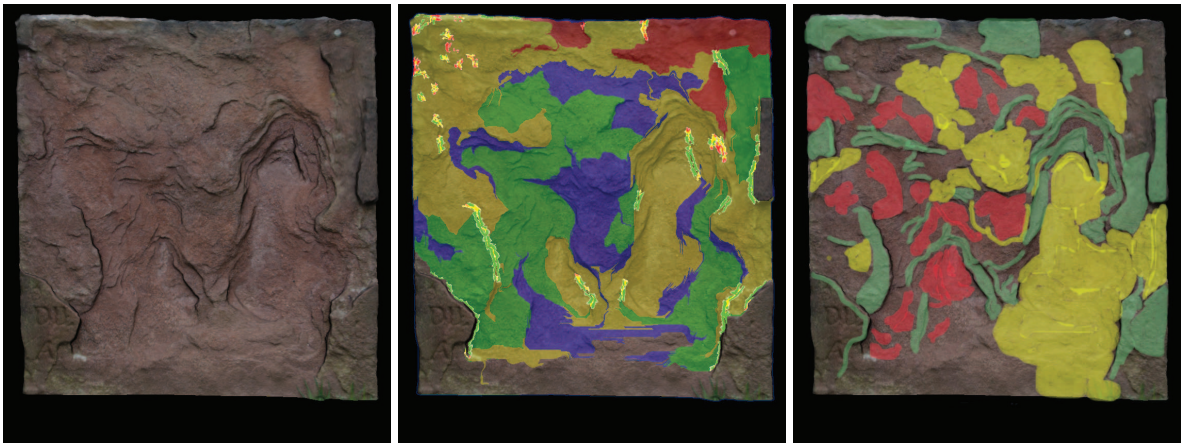


Figure V.12: *Model 1. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment.*

(cc) BY-NC-ND

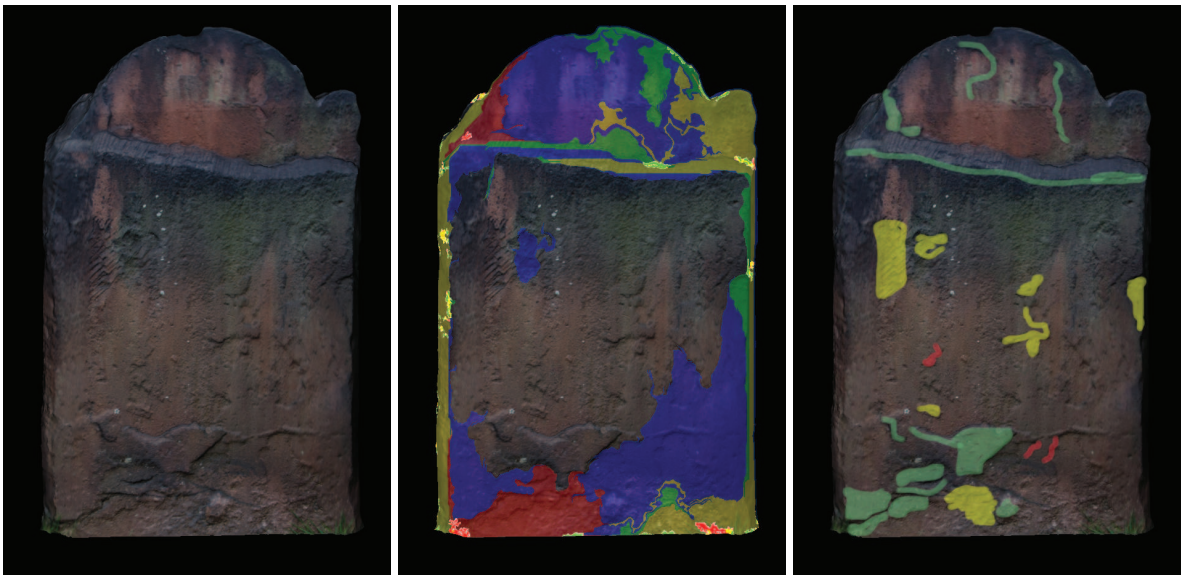


Figure V.13: *Model 2. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment.*

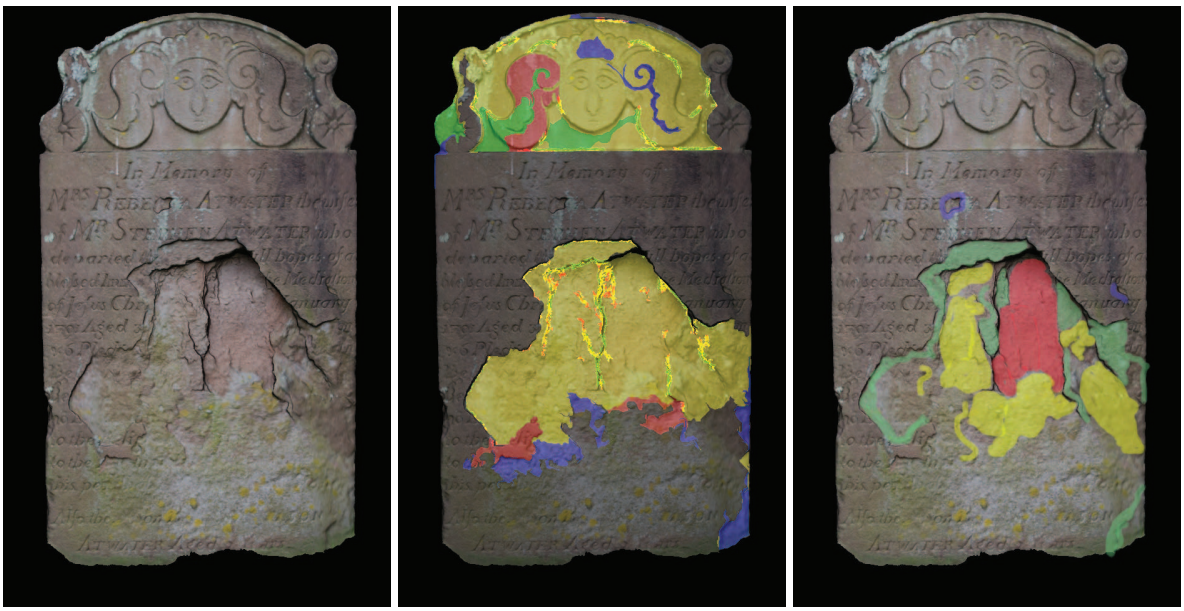


Figure V.14: *Model 3. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment.*

(cc) BY-NC-ND

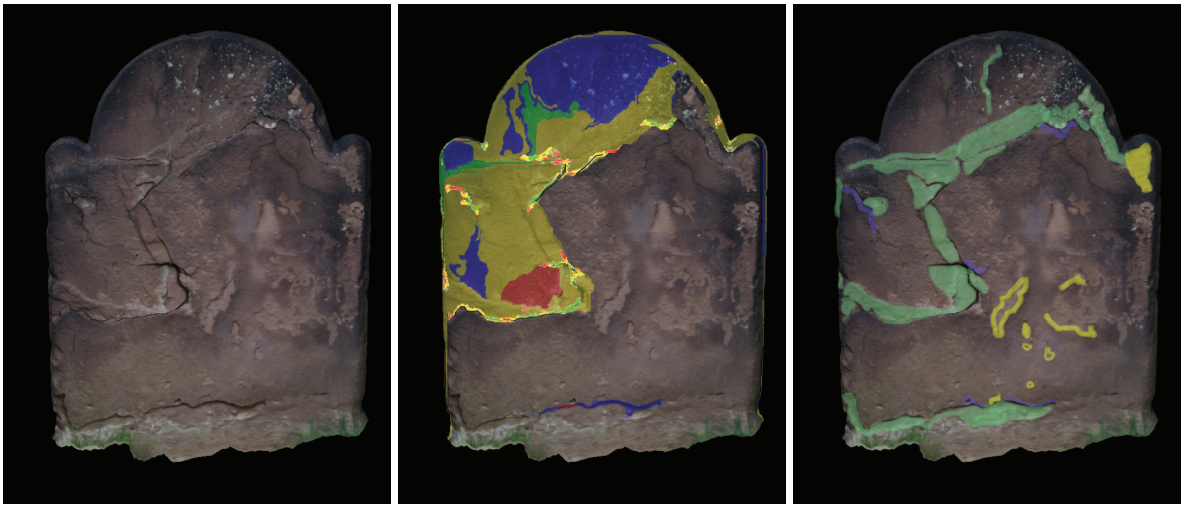


Figure V.15: *Model 4. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment.*

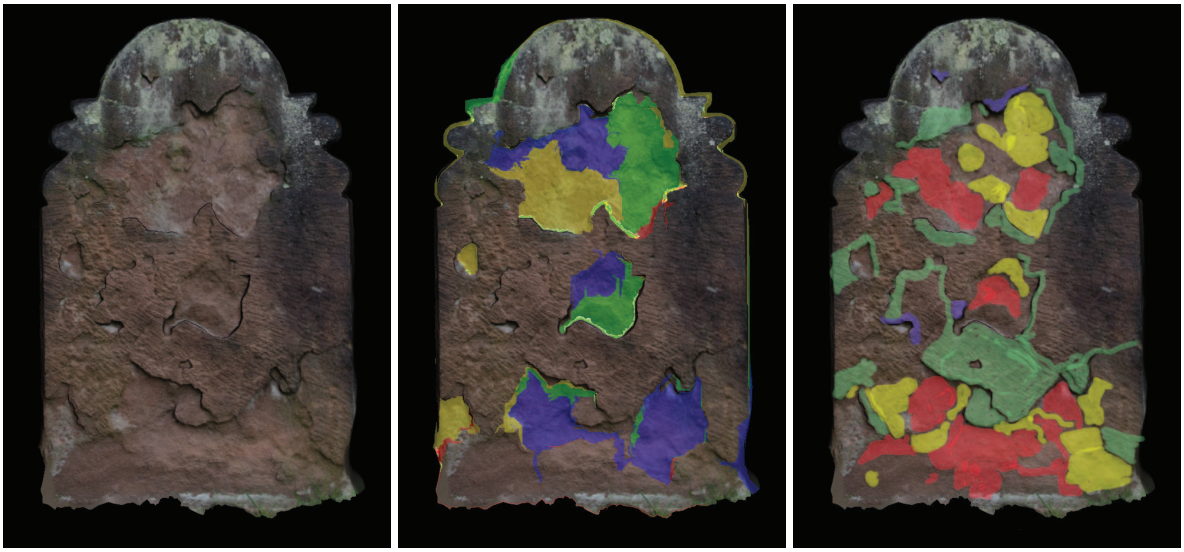


Figure V.16: *Model 5. Result of our method (middle) for a reconstructed model (left) compared to a mapping of an expert in Cultural Heritage (right). The color code of the weathering regions describes the corresponding sub-type of scaling weathering effect. It is red for flaking, green for scaling, yellow for both and blue for detachments which we are not able to classify in a specific sub-type. We also display the identified scaling features to help in the assessment.*

(cc) BY-NC-ND

V.6.2 Discussion

We think our approach represents a first step forward in the mapping of weathering effects. It represents a new tool that is able to reduce the ambiguity and subjectivity of current mapping procedures. Moreover, our approach is suitable for all kind of users. Once the user has taken the pictures and reconstructed the model, our technique only requires a single click on an unweathered region and the adjustment of some intuitive thresholds to perform an automatic analysis and segmentation. Furthermore, our proposed pipeline is independent of the reconstruction method and it can be used with LIDAR data, for instance. Although we have tested our method only in grave markers with flat shapes, it can be extended to all the forms supported by RANSAC: planes, cylinders, spheres, cones and tori. Fortunately, as a result of being man-made, many ancient monuments and buildings can be decomposed in these shapes. In that case, the normals similarity strategy used in Section V.4 should be adapted appropriately. Besides that, we would need to define a 2D parametrization of the new shapes to obtain a distance image whose computation should be adapted to store the lowest distance between the model and the estimated shape taking into account the normal at those points. This parametrization is necessary for the segmentation of the weathered regions.

In Figures V.12 to V.16, we can see how our technique is very good at detecting the weathering regions and isolating them from the unweathered surface, even if it contains inscriptions or superficial carvings (see Figure V.14). It can detect weathered regions of multiple degrees and the region segmentation is very useful to divide them in different entities (see Figure V.12, central hole in Figure V.14, upper part of Figure V.15 and main holes of Figure V.14). Moreover, in many cases, the presented technique is able to classify these weathered regions into the different scaling sub-types thanks to the feature analysis (see Figure V.12, central hole of Figure V.14 and upper holes of Figure V.16). There are cases where the classification differs considerably between our technique and the expert mapping (see Figures V.13 and V.15). This is caused by differences between the mapping strategy and feature classification between the expert and our technique. These differences vary from an expert to the other due to the subjectivity in the judgement. Finally, the highlighting of the detected scaling features during visualization provides an interesting tool for user inspection to uncover subtle features that would be very costly to produce by an expert.

Our method also has some limitations. It can be misled by deep carvings and sculpture reliefs, and it can map them as weathered regions of the surface (see top zone of Figure V.14 and margins in Figure V.13). Although some techniques to try to detect these regions can be applied to identify and reduce this problem, we think that these cases are simple enough to detect for a user and they can be removed with some simple interaction. Another limitation of our technique is that the mapping procedure can skip weathered regions if they are small and not deep enough (see unweathered surfaces in Figures V.13 and V.15). This is caused by how the image-space depth segmentation computes borders and it is more noticeable if the model has large and deep weathered zones. In that case, avoiding small detached zones is a problem of low incidence if we consider the whole model. Finally, our technique is dependent on the initial reconstruction. If its surface is not able to represent the features of the surface, our technique will not be able to detect the scaling features (see the small features of Figure V.12 that are only considered in the texture and not in the geometry). In addition, as our technique consists of several independent steps, there is the possibility that the error of each step is accumulated in the final result. This is a common problem of step-based solutions and it requires a study of the uncertainty of each

(cc) BY-NC-ND

step which is out of the scope of this initial work. But, this problem emphasizes the need of using high quality reconstructed models to reduce the error from the beginning. For these reasons, our technique performs better with large sets of pictures with high resolution as it is recommended in photogrammetry guidelines.

V.7 Conclusions

In this chapter, we have introduced a semi-automatic technique to detect weathering effects on stone monuments. From an easy to perform 3D reconstruction of a model, we are able to analyze its geometry and to identify and map the features and characteristics (depth, spread, ...) of scaling weathering effects. Moreover, we are able to detect two sub-types of scaling and to represent them on the mapping. Our technique has the advantage of requiring little interaction from the user to isolate well weathered regions from unweathered ones, a fact that reduces the disparity in mappings. Although our method has some limitations in terms of identifying scaling subtypes and detecting some carvings as weathering effects, current manual mapping also has its own limitations. The most important is the subjectivity of its analysis, which limits the comparison between different mappings due to the possible disparity of criteria between different observers. By contrast, the presented method is a first big step towards the reduction of ambiguity and subjectivity of manual maps. We think that this technique can open a new line to improve the mapping and tracing of weathering effects in Cultural Heritage conservation projects. Moreover, we think this technique can set up a basis to improve existing weathering simulation techniques for realistic image synthesis or for developing new ones. It can also be very useful to detect characteristics of real weathering effects and to transfer them to objects to either synthesize the effects themselves or additional details not obtained with simulations.

(cc) BY-NC-ND

Chapter VI. Conclusions and Future Work

VI.1 Conclusions

In general, we can describe weathering effects as the result of the incidence of several weathering/environmental factors on a surface followed by decay processes that can be described with different models. In this thesis we have analyzed and proposed techniques that focus on the different parts of a weathering pipeline in Computer Graphics: factors, models and effects.

First, in Chapter II we have reviewed the previous work concerning the simulation of weathering effects. We have divided the presented techniques into two main groups depending on whether they are single-effect centered or, conversely, they try to simulate more than one effect at the same time. After that, we have also reviewed techniques that estimate some of the main weathering factors, in particular solar radiation and wind. Finally, we have surveyed the analysis and classification of weathering effects in real objects from two points of view. On one hand, we have presented studies focused in the preservation of Cultural Heritage and, on the other hand, we have introduced techniques to analyze weathering in the field of Computer Graphics. In this review, we have observed a need for adequate techniques to estimate weathering factors in order to reliably use them in weathering simulations. Moreover, we have noticed a significant drawback in current weathering techniques, which is the lack of models that take into account the interaction of factors and processes from a global point of view. Finally, we have realized that current weathering effects are disconnected from real processes and there is an important opportunity for improvement by linking real and virtual weathering simulations.

Based on these findings, in Chapter III we have presented a new technique to estimate solar exposure on urban models. Our method is based on a screen-space approach that is able to achieve interactive frame rates allowing to handle dynamic modifications of the scene. More specifically, we estimate the solar exposure at each point by sampling a precomputed solar radiation map inside the sky visible regions computed thanks to a two-scale approach. Our technique surpasses existing methods proving to be fast and reliable. This improvement makes possible to use a realistic estimation of this factor in any weathering simulation that was not possible before. It also has advantages in other fields such as architecture, urban planning and engineering design, as we have shown before.

In Chapter IV we have introduced a new technique to simulate the changes of appearance of urban environments due to pollution deposition, which is a very common issue in modern

(cc) BY-NC-ND

cities. Our work is based on a new conception of weathering models: rather than focus on the local simulation of the weathering effect, we expand the decay simulation to a global point of view taking into account the different factors that affect a city and their interrelations. On one hand, we compute the pollution distribution along the city using a simulation of pollution particles spreading and interacting with surfaces thanks to a new and fast physically-based 2D method to simulate wind. On the other hand, we estimate the solar exposure factor using the technique described in Chapter III and we compute rain and wind accessibility using an adapted version of the screen-space sky visibility computation used by this estimation. At run-time, we are able to evaluate and combine these factors using a global model that simulates the three main steps of this pollution process on surfaces: pollution deposition, binding by chemical reactions and pollution stains washing. Our approach results in an interactive method that modifies the appearance of buildings showing the deposition of pollution and providing realistic images. Moreover, our model can be easily adapted to simulate other weathering factors.

Finally, in Chapter V we have presented a new method to analyze weathering effects in real objects. More specifically, our method identifies and maps scaling weathering effects on flat stone monuments. This approach consists of capturing real weathered models using photogrammetry and pre-processing them to reduce the acquisition noise and computing its original unweathered surface. Then, on the one hand, we analyze the geometry of the models identifying those features belonging to scaling effects. On the other hand, we segment the weathering zones of the surface using an image-space segmentation method based on the distance from each region to the unweathered surface. Finally, we map the weathered regions on the model taking into account the detected scaling features making it suitable to identify two subtypes of this weathering effect. The goal of this chapter is to be able to understand better weathering processes in order to improve weathering simulations using real life data. It also offers the ability to recognize weathering effects that will help during their transfer to other virtual models. Moreover, this technique is very useful in other fields like Cultural Heritage preservation projects.

Rather than performing very specific work, this thesis has tried to improve the general simulation of weathering effects in Computer Graphics by dealing with the different parts that influence these processes. We think that this results in a generalized study that opens different research lines to continue the understanding of decay processes from different points of view. We think that attacking this problem across several fronts is the key point to assure a successful improvement of the existing weathering simulations in a near future. However, this improvement will not only bring about being able to reproduce more realistic material appearance in Computer Graphics, it is also an opportunity to improve the understanding of real materials decay that can have implications in other areas such as Cultural Heritage preservation, urban planning and engineering works.

VI.2 Future Work

In the context of this thesis, there are still some open problems and possible research directions. In the following subsections, we will discuss them divided into each of the steps that this thesis deals with.

(cc) BY-NC-ND

VI.2.1 Solar Radiation Estimation

Future work is not only oriented towards solving the drawbacks discussed in the specific chapter, but also towards additional aspects such as indirect reflections caused by the ground and neighboring objects; neither of which have been considered in our current method. Also, our two-scale screen-space visibility estimation has room for improvement. On one hand, a combination of visible regions from the two scales would produce a more accurate sky sampling. On the other hand, we could reduce errors introduced in the computation of the opening angles of the large-scale step by refining them using viewer information or an adapted version of depth-peeling based on the height map of the scene. Moreover, currently the samplings of the exposure map are regularly distributed along the visible regions. Taking into account the variation of solar radiation distribution on the map, it would be interesting to guide the sampling using importance sampling. Last but not least, we plan to improve the sky model's accuracy by introducing climatological data during the computation of the map to be able to reproduce partially cloudy skies and by using real captured data when available. Our technique can be easily adapted to use real data in the exposure maps, particularly if it is captured saving its directional component. Nowadays, this is very common for the direct component but still not so usual for diffuse components.

VI.2.2 Urban Pollution Weathering Model

In this part of the thesis we have proposed a very complete framework. But, although our results are visually plausible, we do not face the washing and deposition processes due to surface flows. As future work, we would like to incorporate such water flows in our model in order to achieve flow stains under building features. A starting point could be adapting existing particle-based approaches [DPH96][Bos+11] to screen-space, following the image-based solution proposed by Endo et al. [End+10]. Furthermore, we believe that screen-space approaches such as the one proposed in this chapter will open the door to simulate other weathering processes not directly related to pollution, particularly for big scenarios such as urban environments. We would like to extend our work to simulate the growth of biological agents like lichens, moss and vegetation. Moreover, we would like to adapt our method to simulate chemical attacks that also depend on global factors like rust, corrosion and patina. Finally, we would like to generalize even more our model to be able to simulate several weathering effects at the same time as well as their interactions.

VI.2.3 Stone Weathering Analysis

Analyzing weathering on stones is still a work in progress. In the near future, we would like to extend our work to make it handle all the shapes detected by RANSAC. This will open up the opportunity to apply our technique to a broader range of monuments and buildings. Moreover, we would like to incorporate the detection of other weathering characteristics like volume loss, area of the loss and spread of it, in order to be able to compute the damage indices and the medical analogies described in the literature. Finally, we would like to improve the current tagging of the regions using an statistical analysis and deriving a confidence map rather than a piecewise classification. In the mid-term, we would like to generate a model that takes into account the characteristics analyzed in the captured data to be able to simulate and transfer weathering effects in virtual models inspired by real captured effects. A good point to start would

(cc) BY-NC-ND

be procedural modeling techniques, which might be useful to take into account the characteristics and structure of the model. Finally, we would like to adapt the aforementioned techniques to screen-space to be able to reproduce them interactively and to include those in our existing global model.

(cc) BY-NC-ND

Bibliography

[AAP12] Oscar Argudo, Carlos Andujar, and Gustavo Patow. “Interactive rendering of urban models with global illumination”. In: 2012 (cited on page 21).

[Aut] *Autodesk Insight*. https://insight360.autodesk.com. Accessed: 2017-09-15 (cited on page 22).

[BS09] Louis Bavoil and Miguel Sainz. “Multi-layer Dual-resolution Screen-space Ambient Occlusion”. In: *SIGGRAPH 2009: Talks*. SIGGRAPH ’09. New Orleans, Louisiana: ACM, 2009, 45:1–45:1. ISBN: 978-1-60558-834-6 (cited on page 37).

[BSD08] Louis Bavoil, Miguel Sainz, and Rouslan Dimitrov. “Image-space Horizon-based Ambient Occlusion”. In: *ACM SIGGRAPH 2008 Talks*. SIGGRAPH ’08. Los Angeles, California: ACM, 2008, 22:1–22:1. ISBN: 978-1-60558-343-3. DOI: 10.1145/1401032.1401061. URL: http://doi.acm.org/10.1145/1401032.1401061 (cited on pages 37, 63).

[BB90] Welton Becket and Norman I. Badler. “Imperfection for realistic image synthesis”. In: *The Journal of Visualization and Computer Animation* 1.1 (1990), pages 26–32. ISSN: 1099-1778. DOI: 10.1002/vis.4340010108. URL: http://dx.doi.org/10.1002/vis.4340010108 (cited on page 16).

[BKCO16] Rachele Bellini, Yanir Kleiman, and Daniel Cohen-Or. “Time-varying Weathering in Texture Space”. In: *ACM Trans. Graph.* 35.4 (July 2016), 141:1–141:11. ISSN: 0730-0301. DOI: 10.1145/2897824.2925891. URL: http://doi.acm.org/10.1145/2897824.2925891 (cited on page 19).

[BA05] Bedrich Benes and X Arriaga. “Table mountains by virtual erosion”. In: *Proceedings of the First Eurographics conference on Natural Phenomena*. Eurographics Association. 2005, pages 033–039 (cited on page 10).

[Ben+06] Bedrich Benes, Vaclav Tesinsky, Jan Hornys, and Sanjiv K. Bhatia. “Hydraulic erosion”. In: *Computer Animation and Virtual Worlds* 17.2 (2006), pages 99–108. ISSN: 1546-427X. DOI: 10.1002/cav.77. URL: http://dx.doi.org/10.1002/cav.77 (cited on page 10).

[Bli78] James F. Blinn. “Simulation of Wrinkled Surfaces”. In: *SIGGRAPH Comput. Graph.* 12.3 (Aug. 1978), pages 286–292. ISSN: 0097-8930. DOI: 10.1145/965139.507101. URL: http://doi.acm.org/10.1145/965139.507101 (cited on page 14).

(cc) BY-NC-ND

[Bli82] James F Blinn. “Light Reflection Functions for Simulation of Clouds and Dusty Surfaces”. In: *Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’82. New York, NY, USA: ACM, 1982, pages 21–29. ISBN: 0-89791-076-1 (cited on page 8).

[BJH12] B. Blocken, W.D. Janssen, and T. van Hooff. “CFD simulation for pedestrian wind comfort and wind safety in urban areas: General decision framework and case study for the Eindhoven University campus”. In: *Environmental Modelling & Software* 30.Supplement C (2012), pages 15 –34. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2011.11.009. URL: http://www.sciencedirect.com/science/article/pii/S1364815211002647 (cited on page 23).

[Blo14] Bert Blocken. “50 years of Computational Wind Engineering: Past, present and future”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 129.Supplement C (2014), pages 69 –102. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2014.03.008. URL: http://www.sciencedirect.com/science/article/pii/S016761051400052X (cited on page 23).

[Blo15] Bert Blocken. “Computational Fluid Dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations”. In: *Building and Environment* 91.Supplement C (2015). Fifty Year Anniversary for Building and Environment, pages 219 –245. ISSN: 0360-1323. DOI: https://doi.org/10.1016/j.buildenv.2015.02.015. URL: http://www.sciencedirect.com/science/article/pii/S0360132315000724 (cited on page 23).

[BCS07] Bert Blocken, Jan Carmeliet, and Ted Stathopoulos. “CFD evaluation of wind speed conditions in passages between parallel buildings - effect of wall-function roughness modifications for the atmospheric boundary layer flow”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 95.9 (2007), pages 941 –962. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2007.01.013. URL: http://www.sciencedirect.com/science/article/pii/S0167610507000268 (cited on page 23).

[Boi+00] L Boissieux, G Kiss, N Magnenat-Thalmann, and Karla P. “Simulation of skin aging and wrinkles with cosmetics insight”. In: *Computer animation and simulation 2000*. Springer, 2000, pages 15–27 (cited on page 14).

[Bos+04] Carles Bosch, Xavier Pueyo, Stéphane Mérillou, and Djamchid Ghazanfarpour. “A Physically-Based Model for Rendering Realistic Scratches”. In: *Computer Graphics Forum* 23.3 (2004), pages 361–370. ISSN: 1467-8659 (cited on page 8).

[Bos+11] Carles Bosch, Pierre-Yves Laffont, Holly Rushmeier, Julie Dorsey, and George Drettakis. “Image-guided Weathering: A New Approach Applied to Flow Phenomena”. In: *ACM Trans. Graph.* 30.3 (2011), 20:1–20:13. ISSN: 0730-0301 (cited on pages 11, 55, 86, 113).

[Bru17] E. Bruneton. “A Qualitative and Quantitative Evaluation of 8 Clear Sky Models”. In: *IEEE Transactions on Visualization and Computer Graphics* PP.99 (2017), pages 1–1. ISSN: 1077-2626. DOI: 10.1109/TVCG.2016.2622272 (cited on page 21).

(cc) BY-NC-ND

[BN08] Eric Bruneton and Fabrice Neyret. “Precomputed Atmospheric Scattering”. In: *Computer Graphics Forum* 27.4 (2008), pages 1079–1086. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2008.01245.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2008.01245.x (cited on page 20).

[CBV15] Riccardo Cacciotti, Miroslav Blasko, and Jaroslav Valach. “A diagnostic ontological model for damages to historical constructions”. In: *Journal of Cultural Heritage* 16.1 (2015), pages 40 –48. ISSN: 1296-2074. DOI: https://doi.org/10.1016/j.culher.2014.02.002. URL: http://www.sciencedirect.com/science/article/pii/S1296207414000259 (cited on page 26).

[CS00] Yao-Xun Chang and Zen-Chung Shih. “Physically-Based Patination for Underground Objects”. In: *Computer Graphics Forum* 19.3 (2000), pages 109–117. ISSN: 1467-8659. DOI: 10.1111/1467-8659.00403. URL: http://dx.doi.org/10.1111/1467-8659.00403 (cited on page 12).

[CS03] Yao-Xun Chang and Zen-Chung Shih. “The synthesis of rust in seawater”. In: *The Visual Computer* 19.1 (2003), pages 50–66. ISSN: 1432-2315. DOI: 10.1007/s00371-002-0172-0. URL: https://doi.org/10.1007/s00371-002-0172-0 (cited on page 12).

[Che+05] Yanyun Chen, Lin Xia, Tien-Tsin Wong, Xin Tong, Hujun Bao, Baining Guo, and Heung-Yeung Shum. “Visual Simulation of Weathering by γ-ton Tracing”. In: *ACM Trans. Graph.* 24.3 (July 2005), pages 1127–1133. ISSN: 0730-0301. DOI: 10.1145/1073204.1073321. URL: http://doi.acm.org/10.1145/1073204.1073321 (cited on pages 15, 16, 55).

[Che+11] Yin Chen, Zhi-Quan Cheng, Jun Li, Ralph R. Martin, and Yan-Zhen Wang. “Relief extraction and editing”. In: *Computer-Aided Design* 43.12 (2011), pages 1674–1682 (cited on pages 26, 27).

[CP10] Olivier Clément and Eric Paquette. “Adaptable aging factory for multiple objects and colorations”. In: *Computers & Graphics* 34.4 (2010). Procedural Methods in Computer Graphics Illustrative Visualization, pages 460 –467. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2010.05.009. URL: http://www.sciencedirect.com/science/article/pii/S0097849310000762 (cited on page 16).

[CK07] Mark Colbert and Jaroslav Křivánek. “GPU-Based Importance Sampling”. In: *GPU Gems 3*. Edited by Hubert Nguyen. Addison-Wesley, 2007, pages 459–479 (cited on page 52).

[Cor+16] Guillaume Cordonnier, Jean Braun, Marie-Paule Cani, Bedrich Benes, Éric Galin, Adrien Peytavie, and Éric Guérin. “Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion”. In: *Computer Graphics Forum* 35.2 (2016), pages 165–175. ISSN: 1467-8659. DOI: 10.1111/cgf.12820. URL: http://dx.doi.org/10.1111/cgf.12820 (cited on page 10).

[Cor+17] Guillaume Cordonnier, Éric Galin, James Gain, Bedrich Benes, Éric Guérin, Adrien Peytavie, and Marie-Paule Cani. “Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation”. In: *ACM Transactions on Graphics* 36.4 (2017). The paper was presented at Siggraph 2017. DOI: 10.1145/3072959.3073667. URL: https://hal.archives-ouvertes.fr/hal-01518967 (cited on pages 10, 11).

(cc) BY-NC-ND

[Cre+14] Benoit Crespin, Richard Bezin, Xavier Skapin, Olivier Terraz, and Philippe Meseure. "Generalized maps for erosion and sedimentation simulation". In: *Computers & Graphics* 45 (2014), pages 1 –16. ISSN: 0097-8493. DOI: http://dx.doi.org/10.1016/j.cag.2014.07.001. URL: http://www.sciencedirect.com/science/article/pii/S0097849314000624 (cited on page 10).

[Cut+02] Barbara Cutler, Julie Dorsey, Leonard McMillan, Matthias Müller, and Robert Jagnow. "A Procedural Approach to Authoring Solid Models". In: *ACM Trans. Graph.* 21.3 (July 2002), pages 302–311. ISSN: 0730-0301. DOI: 10.1145/566654.566581. URL: http://doi.acm.org/10.1145/566654.566581 (cited on page 16).

[DR15] José Delgado Rodrigues. "Defining, mapping and assessing deterioration patterns in stone conservation projects". In: *Journal of Cultural Heritage* 16.3 (2015), pages 267–275 (cited on page 25).

[DGA04] Brett Desbenoit, Eric Galin, and Samir Akkouche. "Simulating and modeling lichen growth". In: *Computer Graphics Forum* 23.3 (2004), pages 341–350. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2004.00765.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2004.00765.x (cited on pages 14, 70).

[DGA05] Brett Desbenoit, Eric Galin, and Samir Akkouche. "Modeling cracks and fractures". In: *The Visual Computer* 21.8-10 (2005), pages 717–726. ISSN: 0178-2789 (cited on page 9).

[Dob+94] Yoshinori Dobashi, Kazufumi Kaneda, Takanobu Nakashima, Hideo Yamashita, Tomoyuki Nishita, and Kastumi Tadamura. "Skylight for Interior Lighting Design". In: *Computer Graphics Forum* 13.3 (1994), pages 85–96. ISSN: 1467-8659. DOI: 10.1111/1467-8659.1330085. URL: http://dx.doi.org/10.1111/1467-8659.1330085 (cited on page 21).

[Dob+00] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. "A Simple, Efficient Method for Realistic Animation of Clouds". In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pages 19–28. ISBN: 1-58113-208-5. DOI: 10.1145/344779.344795. URL: http://dx.doi.org/10.1145/344779.344795 (cited on page 21).

[Doe05] Martin Doerr. "The CIDOC CRM, an Ontological Approach to Schema Heterogeneity". In: *Semantic Interoperability and Integration*. Edited by Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold. Dagstuhl Seminar Proceedings 04391. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005. URL: http://drops.dagstuhl.de/opus/volltexte/2005/35 (cited on page 25).

[DRM12] T. Dogan, C. Reinhart, and P Michalatos. "Urban daylight simulation: Calculating the daylit area of urban designs." In: *Proceedings of SimBuild*. 2012, pages 613–620 (cited on page 22).

[Don+15] Zhao Dong, Bruce Walter, Steve Marschner, and Donald P. Greenberg. "Predicting Appearance from Measured Microgeometry of Metal Surfaces". In: *ACM Trans. Graph.* 35.1 (Dec. 2015), 9:1–9:13. ISSN: 0730-0301. DOI: 10.1145/2815618. URL: http://doi.acm.org/10.1145/2815618 (cited on page 9).

(cc) BY-NC-ND

[DH96] Julie Dorsey and Pat Hanrahan. “Modeling and Rendering of Metallic Patinas”. In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’96. New York, NY, USA: ACM, 1996, pages 387–396. ISBN: 0-89791-746-4 (cited on page 12).

[DPH96] Julie Dorsey, Hans Køhling Pedersen, and Pat Hanrahan. “Flow and changes in appearance”. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH ’96* (1996), pages 411–420. ISSN: 00978930 (cited on pages 11, 113).

[DRS08] Julie Dorsey, Holly Rushmeier, and Franois Sillion. *Digital Modeling of Material Appearance*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008. ISBN: 9780080556710, 9780122211812 (cited on pages 7, 55).

[Dor+99] Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans Køhling Pedersen. “Modeling and Rendering of Weathered Stone”. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pages 225–234. ISBN: 0-201-48560-5 (cited on pages 12, 13).

[End+10] Yuki Endo, Yoshihiro Kanamori, Jun Mitani, and Yukio Fukui. “An interactive design system for water flow stains on outdoor images”. In: *International Symposium on Smart Graphics*. 2010, pages 160–171 (cited on page 113).

[FLW13] D. Fan, S. Liu, and Y. Wei. “Fruit Ring Rot Simulation Based on Reaction-Diffusion Model”. In: *2013 International Conference on Virtual Reality and Visualization*. 2013, pages 199–205. DOI: 10.1109/ICVRV.2013.38 (cited on page 14).

[FH04] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision* 59.2 (2004), pages 167–181 (cited on page 102).

[FB12] Eduardo Fernández and Gonzalo Besuievsky. “Inverse lighting design for interior buildings integrating natural and artificial sources”. In: *Computers & Graphics* 36.8 (2012). Graphics Interaction Virtual Environments and Applications 2012, pages 1096 –1108. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2012.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0097849312001550 (cited on page 21).

[FHB00] B. Fitzner, K. Heinrichs, and D. La Bouchardiere. “Damage index for stone monuments”. In: *Protection and Conservation of the Cultural Heritage in the Mediterranean Cities: Proceedings of the 5th International Symposium*. Sevilla, Spain, 2000, pages 315–326 (cited on pages 24, 25).

[FHK95] B. Fitzner, K. Heinrichs, and R. Kownatzki. “Weathering forms - classification and mapping”. In: *Denkmalpflege und Naturwissenschaft, Natur- steinkonservierung*. Berlin, Germany, 1995, pages 41–88 (cited on page 24).

[FHK97] B. Fitzner, K. Heinrichs, and R. Kownatzki. “Weathering forms at natural stone monuments - classification, mapping and evaluation”. In: 3.2 (1997), pages 105–124 (cited on page 24).

(cc) BY-NC-ND

[FHB03] Bernd Fitzner, Kurt Heinrichs, and Dennis La Bouchardiere. “Weathering damage on Pharaonic sandstone monuments in Luxor-Egypt”. In: *Building and Environment* 38.9-10 (2003). Building Stone Decay: Observations, Experiments and Modeling, pages 1089–1103 (cited on page 24).

[FF01] Nick Foster and Ronald Fedkiw. “Practical Animation of Liquids”. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’01. New York, NY, USA: ACM, 2001, pages 23–30. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383261. URL: http://doi.acm.org/10.1145/383259.383261 (cited on pages 22, 23).

[FM97] Nick Foster and Dimitris Metaxas. “Modeling the Motion of a Hot, Turbulent Gas”. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pages 181–188. ISBN: 0-89791-896-7. DOI: 10.1145/258734.258838. URL: http://dx.doi.org/10.1145/258734.258838 (cited on page 22).

[Fre+15] S. Freitas, C. Catita, P. Redweik, and M.C. Brito. “Modelling solar potential in the urban environment: State-of-the-art review”. In: *Renewable and Sustainable Energy Reviews* 41 (2015), pages 915–931 (cited on page 22).

[FVG15] Dhana Frerichs, Andrew Vidler, and Christos Gatzidis. “A Survey on Object Deformation and Decomposition in Computer Graphics”. In: *Comput. Graph.* 52.C (Nov. 2015), pages 18–32. ISSN: 0097-8493. DOI: 10.1016/j.cag.2015.06.004. URL: http://dx.doi.org/10.1016/j.cag.2015.06.004 (cited on page 7).

[FVG16] Dhana Frerichs, Andrew Vidler, and Christos Gatzidis. “Biologically inspired simulation of livor mortis”. In: *The Visual Computer* (2016). ISSN: 1432-2315. DOI: 10.1007/s00371-016-1291-3. URL: https://doi.org/10.1007/s00371-016-1291-3 (cited on page 14).

[Glo+12] L. Glondu, L. Muguercia, M. Marchal, C. Bosch, H. Rushmeier, G. Dumont, and G. Drettakis. “Example-Based Fractured Appearance”. In: *Computer Graphics Forum* 31.4 (2012), pages 1547–1556. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2012.03151.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2012.03151.x (cited on page 9).

[GC01a] Stephane Gobron and Norishige Chiba. “Crack pattern simulation based on 3D surface cellular automata”. In: *The Visual Computer* 17.5 (2001), pages 287–309. ISSN: 1432-2315. DOI: 10.1007/s003710100099. URL: https://doi.org/10.1007/s003710100099 (cited on pages 9, 10).

[GC01b] Stéphane Gobron and Norishige Chiba. “Simulation of Peeling Using 3D-Surface Cellular Automata”. In: *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*. PG ’01. Washington, DC, USA: IEEE Computer Society, 2001, pages 338–. ISBN: 0-7695-1227-5. URL: http://dl.acm.org/citation.cfm?id=882473.883422 (cited on page 10).

[Gol+06] Aleksey Golovinskiy, Wojciech Matusik, Hanspeter Pfister, Szymon Rusinkiewicz, and Thomas Funkhouser. “A Statistical Model for Synthesis of Detailed Facial Geometry”. In: *ACM Trans. Graph.* 25.3 (July 2006), pages 1025–1034. ISSN: 0730-0301. DOI: 10.1145/1141911.1141988. URL: http://doi.acm.org/10.1145/1141911.1141988 (cited on page 14).

(cc) BY-NC-ND

[Gos+10] Prashant Goswami, Philipp Schlegel, Barbara Solenthaler, and Renato Pajarola. “Interactive SPH Simulation and Rendering on the GPU”. In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’10. Madrid, Spain: Eurographics Association, 2010, pages 55–64. URL: http://dl.acm.org/citation.cfm?id=1921427.1921437 (cited on page 23).

[Gou+11] P. Gousseau, B. Blocken, T. Stathopoulos, and G.J.F. van Heijst. “CFD simulation of near-field pollutant dispersion on a high-resolution grid: A case study by LES and RANS for a building group in downtown Montreal”. In: *Atmospheric Environment* 45.2 (2011), pages 428 –438. ISSN: 1352-2310. DOI: https://doi.org/10.1016/j.atmosenv.2010.09.065. URL: http://www.sciencedirect.com/science/article/pii/S1352231010008514 (cited on page 23).

[Gu+06] Jinwei Gu, Chien-I Tu, Ravi Ramamoorthi, Peter Belhumeur, Wojciech Matusik, and Shree Nayar. “Time-varying Surface Appearance: Acquisition, Modeling and Rendering”. In: *ACM Trans. Graph.* 25.3 (July 2006), pages 762–771. ISSN: 0730-0301. DOI: 10.1145/1141911.1141952. URL: http://doi.acm.org/10.1145/1141911.1141952 (cited on page 17).

[Gu+07] Jinwei Gu, Ravi Ramamoorthi, Peter Belhumeur, and Shree Nayar. “Dirty Glass: Rendering Contamination on Transparent Surfaces”. In: *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. EGSR’07. Grenoble, France: Eurographics Association, 2007, pages 159–170. ISBN: 978-3-905673-52-4. DOI: 10.2312/EGWR/EGSR07/159-170. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/159-170 (cited on page 11).

[GRG12] Tobias Gunther, Kai Rohmer, and Thorsten Grosch. “GPU-accelerated Interactive Material Aging”. In: *Vision, Modeling and Visualization*. Edited by Michael Goesele, Thorsten Grosch, Holger Theisel, Klaus Toennies, and Bernhard Preim. The Eurographics Association, 2012. ISBN: 978-3-905673-95-1. DOI: 10.2312/PE/VMV/VMV12/063-070 (cited on page 16).

[GP14] Jie Guo and Jin-Gui Pan. “Real-time simulating and rendering of layered dust”. In: *The Visual Computer* 30.6 (2014), pages 797–807. ISSN: 1432-2315. DOI: 10.1007/s00371-014-0967-9. URL: https://doi.org/10.1007/s00371-014-0967-9 (cited on page 8).

[Ḧ17] Torsten Hädrich, Bedrich Benes, Oliver Deussen, and Sören Pirk. “Interactive Modeling and Authoring of Climbing Plants”. In: *Computer Graphics Forum* 36.2 (2017), pages 49–61. ISSN: 1467-8659. DOI: 10.1111/cgf.13106. URL: http://dx.doi.org/10.1111/cgf.13106 (cited on pages 14, 15).

[HL97] X He and L-S Luo. “Lattice Boltzmann model for the incompressible Navier-Stokes equation”. In: *Journal of Statistical Physics* 88.3-4 (1997), pages 927–944 (cited on page 59).

[HB95] David J. Heeger and James R. Bergen. “Pyramid-based Texture Analysis/Synthesis”. In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’95. New York, NY, USA: ACM, 1995, pages 229–238. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218446. URL: http://doi.acm.org/10.1145/218380.218446 (cited on page 18).

(cc) BY-NC-ND

[HTK98] Koichi Hirota, Yasuyuki Tanoue, and Toyohisa Kaneko. “Generation of crack patterns with a physical model”. In: *The Visual Computer* 14.3 (1998), pages 126–137 (cited on page 9).

[HB10] T. Van Hooff and B. Blocken. “Coupled urban wind flow and indoor natural ventilation modelling on a high-resolution grid: A case study for the Amsterdam ArenA stadium”. In: *Environmental Modelling & Software* 25.1 (2010), pages 51 –65. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2009.07.008. URL: http://www.sciencedirect.com/science/article/pii/S1364815209001790 (cited on pages 23, 24).

[HW12] Lukas Hosek and Alexander Wilkie. “An Analytic Model for Full Spectral Sky-dome Radiance”. In: *ACM Trans. Graph.* 31.4 (July 2012), 95:1–95:9. ISSN: 0730-0301. DOI: 10.1145/2185520.2185591. URL: http://doi.acm.org/10.1145/2185520.2185591 (cited on pages 20, 21, 33).

[HW13] Lukas Hosek and Alexander Wilkie. “Adding a Solar-Radiance Function to the Hošek-Wilkie Skylight Model”. In: *IEEE Computer Graphics and Applications* 33.3 (May 2013), pages 44–52 (cited on pages 30, 32, 33, 35, 80).

[HW95] Siu-chi Hsu and Tien-tsin Wong. “Simulating Dust Accumulation”. In: *IEEE Comput. Graph. Appl.* 15.1 (1995), pages 18–22. ISSN: 0272-1716 (cited on page 8).

[IO06] Hayley N. Iben and James F. O’Brien. “Generating Surface Crack Patterns”. In: *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’06. Vienna, Austria: Eurographics Association, 2006, pages 177–185. ISBN: 3-905673-34-7. URL: http://dl.acm.org/citation.cfm?id=1218064.1218088 (cited on page 9).

[ICO08] ICOMOS. *Monuments and Sites XV - Illustrated glossary on deterioration patterns*. ICOMOS International Scientific Committee for Stone, 2008 (cited on pages 25, 90).

[IG+15] Jose A. Iglesias-Guitian, Carlos Aliaga, Adrian Jarabo, and Diego Gutierrez. “A Biophysically-Based Model of the Optical Properties of Skin Aging”. In: *Computer Graphics Forum* 34.2 (2015), pages 45–55. ISSN: 1467-8659. DOI: 10.1111/cgf.12540. URL: http://dx.doi.org/10.1111/cgf.12540 (cited on pages 14, 15).

[Iiz+16] Satoshi Iizuka, Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. “Single Image Weathering via Exemplar Propagation”. In: *Computer Graphics Forum* 35.2 (2016), pages 501–509. ISSN: 1467-8659. DOI: 10.1111/cgf.12850. URL: http://dx.doi.org/10.1111/cgf.12850 (cited on page 19).

[Ito+03] T. Ito, T. Fujimoto, K. Muraoka, and N. Chiba. “Modeling rocky scenery taking into account joints”. In: *Proceedings Computer Graphics International 2003*. 2003, pages 244–247. DOI: 10.1109/CGI.2003.1214475 (cited on page 10).

[JBC07] Hans Janssen, Bert Blocken, and Jan Carmeliet. “Conservative modelling of the moisture and heat transfer in building components under atmospheric excitation”. In: *International Journal of Heat and Mass Transfer* 50.5 (2007), pages 1128 – 1140. ISSN: 0017-9310. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2006.06.048. URL: http://www.sciencedirect.com/science/article/pii/S0017931006005011 (cited on page 23).

(cc) BY-NC-ND

[JPK13] SoHyeon Jeong, Si-Hyung Park, and Chang-Hun Kim. “Simulation of Morphology Changes in Drying Leaves”. In: *Computer Graphics Forum* 32.1 (2013), pages 204–215. ISSN: 1467-8659. DOI: 10.1111/cgf.12009. URL: http://dx.doi.org/10.1111/cgf.12009 (cited on page 14).

[JW84] Glenn T. Johnson and Ian D. Watson. “The Determination of View-Factors in Urban Canyons”. In: *Journal of Climate and Applied Meteorology* 23.2 (1984), pages 329–335 (cited on page 35).

[Jon+10] M. D. Jones, M. Farley, J. Butler, and M. Beardall. “Directable Weathering of Concave Rock Using Curvature Estimation”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.1 (2010), pages 81–94. ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.39 (cited on pages 12, 13).

[Kaj86] James T. Kajiya. “The Rendering Equation”. In: *SIGGRAPH Comput. Graph.* 20.4 (Aug. 1986), pages 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902. URL: http://doi.acm.org/10.1145/15886.15902 (cited on page 32).

[KRB11] Joseph T. Kider, Samantha Raja, and Norman I. Badler. “Fruit Senescence and Decay Simulation”. In: *Computer Graphics Forum* 30.2 (2011), pages 257–266. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2011.01857.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2011.01857.x (cited on pages 14, 15).

[Kid+14] Joseph T. Kider Jr., Daniel Knowlton, Jeremy Newlin, Yining Karl Li, and Donald P. Greenberg. “A Framework for the Experimental Comparison of Solar and Skydome Illumination”. In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 180:1–180:12. ISSN: 0730-0301. DOI: 10.1145/2661229.2661259. URL: http://doi.acm.org/10.1145/2661229.2661259 (cited on pages 21, 32).

[Kim+13] Bradley W. Kimmel, Gladimir V. G. Baranoski, T. F. Chen, Daniel Yim, and Erik Miranda. “Spectral Appearance Changes Induced by Light Exposure”. In: *ACM Trans. Graph.* 32.1 (Feb. 2013), 10:1–10:13. ISSN: 0730-0301. DOI: 10.1145/2421636.2421646. URL: http://doi.acm.org/10.1145/2421636.2421646 (cited on pages 12, 13).

[KST08] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. “Demarcating Curves for Shape Illustration”. In: *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)* 27.5 (Dec. 2008), 157:1–157:9. ISSN: 0730-0301 (cited on pages 26, 27).

[Kou+14] Anestis Koutsoudis, Blaz Vidmar, George Ioannakis, Fotis Arnaoutoglou, George Pavlidis, and Christodoulos Chamzas. “Multi-image 3D reconstruction data evaluation”. In: *Journal of Cultural Heritage* 15.1 (2014), pages 73–79 (cited on page 25).

[KM31] Paul Kubelka and Franz Munk. “Ein Beitrag zur Optik der Farbanstriche”. In: *Zeitschrift für technische Physik* 12 (1931), pages 593–601 (cited on pages 8, 12, 13).

[Law+17] K. Lawonn, E. Trostmann, B. Preim, and K. Hildebrandt. “Visualization and Extraction of Carvings for Heritage Conservation”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pages 801–810 (cited on pages 26, 27).

[LP02] Anita T. Layton and Michiel van de Panne. “A numerically efficient and stable algorithm for animating water waves”. In: *The Visual Computer* 18.1 (2002), pages 41–53. ISSN: 1432-2315. DOI: 10.1007/s003710100131. URL: https://doi.org/10.1007/s003710100131 (cited on page 22).

(cc) BY-NC-ND

[LWK03] Wei Li, Xiaoming Wei, and Arie Kaufman. “Implementing lattice Boltzmann computation on graphics hardware”. In: *The Visual Computer* 19.7 (2003), pages 444–456. ISSN: 1432-2315. DOI: 10.1007/s00371-003-0210-6. URL: https://doi.org/10.1007/s00371-003-0210-6 (cited on page 23).

[Lia+14] Jianming Liang, Jianhua Gong, Wenhang Li, and Abdoul Nasser Ibrahim. “A visualization-oriented 3D method for efficient computation of urban solar radiation based on 3D–2D surface mapping”. In: *International Journal of Geographical Information Science* 28.4 (2014), pages 780–798 (cited on pages 21, 22).

[Liu+12] Youquan Liu, Yanyun Chen, Wen Wu, Nelson Max, and Enhua Wu. “Physically based object withering simulation”. In: *Computer Animation and Virtual Worlds* 23.3-4 (2012), pages 395–406. ISSN: 1546-427X. DOI: 10.1002/cav.1459. URL: http://dx.doi.org/10.1002/cav.1459 (cited on page 14).

[Liu+05] Y.Q. Liu, H.B. Zhu, X.H. Liu, and E.H. Wu. “Real-time simulation of physically based on-surface flow”. In: *The Visual Computer* 21.8 (2005), pages 727–734. ISSN: 1432-2315. DOI: 10.1007/s00371-005-0314-2. URL: https://doi.org/10.1007/s00371-005-0314-2 (cited on page 10).

[Lu+07] Jianye Lu, Athinodoros S. Georghiades, Andreas Glaser, Hongzhi Wu, Li-Yi Wei, Baining Guo, Julie Dorsey, and Holly Rushmeier. “Context-aware Textures”. In: *ACM Trans. Graph.* 26.1 (Jan. 2007). ISSN: 0730-0301. DOI: 10.1145/1189762.1189765. URL: http://doi.acm.org/10.1145/1189762.1189765 (cited on pages 17, 18, 89).

[McG+11] Morgan McGuire, Brian Osman, Michael Bukowski, and Padraic Hennessy. “The Alchemy Screen-space Ambient Obscurance Algorithm”. In: *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. HPG ’11. Vancouver, British Columbia, Canada: ACM, 2011, pages 25–32. ISBN: 978-1-4503-0896-0 (cited on page 37).

[MFS09] Àlex Méndez-Feliu and Mateu Sbert. “From obscurances to ambient occlusion: A survey”. English. In: *The Visual Computer* 25.2 (2009), pages 181–196. ISSN: 0178-2789 (cited on pages 35, 37).

[Mer+12] N. Merillou, S. Merillou, E. Galin, and D. Ghazanfarpour. “Simulating How Salt Decay Ages Buildings”. In: *IEEE Computer Graphics and Applications* 32.2 (2012), pages 44–54. ISSN: 0272-1716. DOI: 10.1109/MCG.2011.107 (cited on pages 12, 13).

[Mer+10] Nicolas Merillou, Stephane Merillou, Djamchid Ghazanfarpour, Jean-Michel Dischler, and Eric Galin. “Simulating Atmospheric Pollution Weathering on Buildings”. en. In: *WSCG*. 2010 (cited on pages 12, 13, 65, 67, 68).

[MG08] S. Merillou and D. Ghazanfarpour. “A survey of aging and weathering phenomena in computer graphics”. In: *Computers & Graphics* 32.2 (2008), pages 159 –174. ISSN: 0097-8493. DOI: http://dx.doi.org/10.1016/j.cag.2008.01.003. URL: http://www.sciencedirect.com/science/article/pii/S0097849308000058 (cited on pages 7, 55).

(cc) BY-NC-ND

[MDG01a] Stephane Merillou, Jean-Michel Dischler, and Djamchid Ghazanfarpour. “Corrosion: Simulating and Rendering”. In: *Proceedings of Graphics Interface 2001*. GI ’01. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2001, pages 167–174. ISBN: 0-9688808-0-0 (cited on pages 8, 12).

[MDG01b] Stéphane Mérillou, Jean-Michel Dischler, and Djamchid Ghazanfarpour. “Surface scratches: measuring, modeling and rendering.” In: *The Visual Computer* 17.1 (2001), pages 30–45 (cited on page 8).

[Mer+10] L. Merino, E. Antaluca, B. Akinoglu, and B. Beckers. “Solar Energy Inputs Estimation for Urban Scales Applications”. In: *8th International Conference on System Simulation in Buildings*. Liege, 2010 (cited on pages 22, 31).

[Mer+06] Tom Mertens, Jan Kautz, Jiawen Chen, Philippe Bekaert, and Frédo Durand. “Texture Transfer Using Geometry Correlation”. In: *Proceedings of the 17th Eurographics Conference on Rendering Techniques*. EGSR ’06. Nicosia, Cyprus: Eurographics Association, 2006, pages 273–284. ISBN: 3-905673-35-5. DOI: 10.2312/EGWR/EGSR06/273-284. URL: http://dx.doi.org/10.2312/EGWR/EGSR06/273-284 (cited on pages 17, 18, 89).

[Mes+17] Tommy Messaoudi, Philippe Véron, Gilles Halin, and Livio De Luca. “An ontological model for the reality-based 3D annotation of heritage building conservation state”. In: *Journal of Cultural Heritage* (2017). ISSN: 1296-2074. DOI: https://doi.org/10.1016/j.culher.2017.05.017. URL: http://www.sciencedirect.com/science/article/pii/S1296207417304508 (cited on page 26).

[Mil94] Gavin Miller. “Efficient Algorithms for Local and Global Accessibility Shading”. In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’94. New York, NY, USA: ACM, 1994, pages 319–326. ISBN: 0-89791-667-0. DOI: 10.1145/192161.192244. URL: http://doi.acm.org/10.1145/192161.192244 (cited on page 11).

[Mit07] Martin Mittring. “Finding Next Gen: CryEngine 2”. In: *ACM SIGGRAPH 2007 Courses*. SIGGRAPH ’07. San Diego, California: ACM, 2007, pages 97–121. ISBN: 978-1-4503-1823-5 (cited on page 37).

[Moh12] A. A. Mohamad. *Lattice Boltzmann Method. Fundamentals and Engineering Applications with Computer Codes*. 1st edition. London: Springer-Verlag, 2012, page 178 (cited on page 58).

[Moh16] S Mariraj Mohan. “An overview of particulate dry deposition: measuring methods, deposition velocity and controlling factors”. In: *International Journal of Environmental Science and Technology* 13 (2016), pages 387–402 (cited on page 62).

[MBP14] Lien Muguercia, Carles Bosch, and Gustavo Patow. “Fracture modeling in computer graphics”. In: *Computers & Graphics* 45 (2014), pages 86–100. ISSN: 00978493 (cited on page 9).

[MCG03] Matthias Müller, David Charypar, and Markus Gross. “Particle-based Fluid Simulation for Interactive Applications”. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’03. San Diego, California: Eurographics Association, 2003, pages 154–159. ISBN: 1-58113-659-5. URL: http://dl.acm.org/citation.cfm?id=846276.846298 (cited on page 22).

(cc) BY-NC-ND

[MKM89] F. K. Musgrave, C. E. Kolb, and R. S. Mace. “The Synthesis and Rendering of Eroded Fractal Terrains”. In: *SIGGRAPH Comput. Graph.* 23.3 (July 1989), pages 41–50. ISSN: 0097-8930. DOI: 10.1145/74334.74337. URL: http://doi.acm.org/10.1145/74334.74337 (cited on page 10).

[Nag98] Kenji Nagashima. “Computer generation of eroded valley and mountain terrains”. In: *The Visual Computer* 13.9 (1998), pages 456–464. ISSN: 1432-2315. DOI: 10.1007/s003710050117. URL: https://doi.org/10.1007/s003710050117 (cited on page 10).

[NDL16] R. Nespeca and L. De Luca. “Analysis, thematic maps and data mining from point cloud to ontology for software development”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Volume XLI-B5. XXIII ISPRS Congress. Prague, Czech Republic, 2016, pages 347–354 (cited on page 26).

[Nis+96] T. Nishita, Y. Dobashi, K. Kaneda, and H. Yamashita. “Display method of the sky color taking into account multiple scattering.” In: *Pacific Graphics*. Volume 96. 1996, pages 117–132 (cited on page 20).

[NMK08] A. M. Noorian, I. Moradi, and G.A. Kamali. “Evaluation of 12 models to estimate hourly diffuse irradiation on inclined surfaces”. In: *Renewable Energy* 33 (2008), pages 1406–1412 (cited on page 32).

[OBS02] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. “Mesh Smoothing by Adaptive and Anisotropic Gaussian Filter Applied to Mesh Normals”. In: *vision modeling and visualization*. 2002 (cited on page 92).

[PPD01] Eric Paquette, Pierre Poulin, and George Drettakis. “Surface Aging by Impacts”. In: *Proceedings of Graphics Interface 2001*. GI ’01. Ottawa, Ontario, Canada: Canadian Information Processing Society, 2001, pages 175–182. ISBN: 0-9688808-0-0. URL: http://dl.acm.org/citation.cfm?id=780986.781008 (cited on page 9).

[PPD02] Eric Paquette, Pierre Poulin, and George Drettakis. “The simulation of paint cracking and peeling”. In: *Proceedings of Graphics Interface*. Canadian Human-Computer Communications Society. 2002, page 10 (cited on page 10).

[PSM93] R. Perez, R. Seals, and J. Michalsky. “All-weather model for sky luminance distribution—Preliminary configuration and validation”. In: *Solar Energy* 50.3 (1993), pages 235–245. ISSN: 0038-092X (cited on pages 20, 33).

[Pet+15] F. Peteler, E. Gattet, P. Bromblet, O. Guillon, J. M. Vallet, and L. De Luca. “Analyzing the evolution of deterioration patterns: A first step of an image-based approach for comparing multitemporal data sets”. In: *2015 Digital Heritage*. Volume 2. 2015, pages 113–116 (cited on page 26).

[Pfa+14] Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, and James F. O’Brien. “Adaptive Tearing and Cracking of Thin Sheets”. In: *ACM Trans. Graph.* 33.4 (July 2014), 110:1–110:9. ISSN: 0730-0301. DOI: 10.1145/2601097.2601132. URL: http://doi.acm.org/10.1145/2601097.2601132 (cited on pages 9, 10).

[Pin+16] Ruggero Pintus, Kazim Pal, Ying Yang, Tim Weyrich, Enrico Gobbetti, and Holly Rushmeier. “A Survey of Geometric Analysis in Cultural Heritage”. In: *Computer Graphics Forum* 35.1 (2016), pages 4–31 (cited on page 26).

(cc) BY-NC-ND

[PRD98] A. Pio, C. Ramos, and M. M. Duarte. “Atmospheric Aerosol and Soiling of External Surfaces in an Urban Environment”. In: *Atmospheric Environment* 32.11 (1998), pages 1979–1989 (cited on page 71).

[PSS99] A. J. Preetham, Peter Shirley, and Brian Smits. “A Practical Analytic Model for Daylight”. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pages 91–100. ISBN: 0-201-48560-5. DOI: 10.1145/311535.311545. URL: http://dx.doi.org/10.1145/311535.311545 (cited on pages 20, 32, 33).

[RB12] R. Ramponi and B. Blocken. “CFD simulation of cross-ventilation for a generic isolated building: Impact of computational parameters”. In: *Building and Environment* 53.Supplement C (2012), pages 34 –48. ISSN: 0360-1323. DOI: https://doi.org/10.1016/j.buildenv.2012.01.004. URL: http://www.sciencedirect.com/science/article/pii/S0360132312000133 (cited on page 23).

[RGB16] Boris Raymond, Gaël Guennebaud, and Pascal Barla. “Multi-scale Rendering of Scratched Materials Using a Structured SV-BRDF Model”. In: *ACM Trans. Graph.* 35.4 (July 2016), 57:1–57:11. ISSN: 0730-0301. DOI: 10.1145/2897824.2925945. URL: http://doi.acm.org/10.1145/2897824.2925945 (cited on pages 8, 9).

[Reu+13] Jeroen De Reu et al. “Towards a three-dimensional cost-effective registration of the archaeological heritage”. In: *Journal of Archaeological Science* 40.2 (2013), pages 1108–1121 (cited on page 25).

[Ril+04] Kirk Riley, David S. Ebert, Martin Kraus, Jerry Tessendorf, and Charles Hansen. “Efficient Rendering of Atmospheric Phenomena”. In: *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*. EGSR’04. Norrköping, Sweden: Eurographics Association, 2004, pages 375–386. ISBN: 3-905673-12-6. DOI: 10.2312/EGWR/EGSR04/375-386. URL: http://dx.doi.org/10.2312/EGWR/EGSR04/375-386 (cited on page 21).

[RGS09] Tobias Ritschel, Thorsten Grosch, and Hans-Peter Seidel. “Approximating Dynamic Global Illumination in Image Space”. In: *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*. I3D ’09. Boston, Massachusetts: ACM, 2009, pages 75–82. ISBN: 978-1-60558-429-4 (cited on pages 37, 40, 52).

[RCOL09] Amir Rosenberger, Daniel Cohen-Or, and Dani Lischinski. “Layered Shape Synthesis: Automatic Generation of Control Maps for Non-stationary Textures”. In: *ACM Trans. Graph.* 28.5 (Dec. 2009), 107:1–107:9. ISSN: 0730-0301. DOI: 10.1145/1618452.1618453. URL: http://doi.acm.org/10.1145/1618452.1618453 (cited on page 18).

[Sat+17] P. Satilmis, T. Bashford-Rogers, A. Chalmers, and K. Debattista. “A Machine-Learning-Driven Sky Model”. In: *IEEE Computer Graphics and Applications* 37.1 (2017), pages 80–91. ISSN: 0272-1716. DOI: 10.1109/MCG.2016.67 (cited on page 21).

[SKS13] Henry Schäfer, Benjamin Keinert, and Marc Stamminger. “Real-time Local Displacement Using Dynamic GPU Memory Management”. In: *Proceedings of the 5th High-Performance Graphics Conference*. HPG ’13. Anaheim, California: ACM,

(cc) BY-NC-ND

2013, pages 63–72. ISBN: 978-1-4503-2135-8. DOI: 10.1145/2492045.2492052. URL: http://doi.acm.org/10.1145/2492045.2492052 (cited on page 9).

[SWK07] R. Schnabel, R. Wahl, and R. Klein. “Efficient RANSAC for Point-Cloud Shape Detection”. In: *Computer Graphics Forum* 26.2 (2007), pages 214–226 (cited on page 96).

[SMG05] S. Shahidi, S. Merillou, and D. Ghazanfarpour. “Phenomenological Simulation of Efflorescence in Brick Constructions”. In: *Eurographics Workshop on Natural Phenomena*. Edited by Pierre Poulin and Eric Galin. The Eurographics Association, 2005. ISBN: 3-905673-29-0. DOI: 10.2312/NPH/NPH05/017-023 (cited on page 13).

[SA07] Perumaal Shanmugam and Okan Arikan. “Hardware Accelerated Ambient Occlusion Techniques on GPUs”. In: *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*. I3D ’07. Seattle, Washington: ACM, 2007, pages 73–80. ISBN: 978-1-59593-628-8 (cited on page 37).

[SP07] B. J. Smith and R. Prikryl. “Diagnosing decay: the value of medical analogy in understanding the weathering of building stones”. In: *Geological Society, London, Special Publications* 271 (2007), pages 1–8 (cited on pages 24, 90).

[SF95] Jos Stam and Eugene Fiume. “Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes”. In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’95. New York, NY, USA: ACM, 1995, pages 129–136. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218430. URL: http://doi.acm.org/10.1145/218380.218430 (cited on page 22).

[Sta+08] Ondrej Stava, Bedrich Benes, Matthew Brisbin, and Jaroslav Krivanek. “Interactive Terrain Modeling Using Hydraulic Erosion”. In: *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’08. Dublin, Ireland: Eurographics Association, 2008, pages 201–210. ISBN: 978-3-905674-10-1. URL: http://dl.acm.org/citation.cfm?id=1632592.1632622 (cited on page 10).

[Ste+14] Chiara Stefani, Xavier Brunetaud, Sarah Janvier-Badosa, K’evin Beck, Livio De Luca, and Muzahim Al-Mukhtar. “Developing a toolkit for mapping and displaying stone alteration on a web-based documentation platform”. In: *Journal of Cultural Heritage* 15.1 (2014), pages 1–9 (cited on page 25).

[Sun+05] Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. “A Practical Analytic Single Scattering Model for Real Time Rendering”. In: *ACM Trans. Graph.* 24.3 (July 2005), pages 1040–1049. ISSN: 0730-0301. DOI: 10.1145/1073204.1073309. URL: http://doi.acm.org/10.1145/1073204.1073309 (cited on page 21).

[Sun+07a] Bo Sun, Kalyan Sunkavalli, Ravi Ramamoorthi, Peter N. Belhumeur, and Shree K. Nayar. “Time-Varying BRDFs”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.3 (May 2007), pages 595–609. ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.1013. URL: http://dx.doi.org/10.1109/TVCG.2007.1013 (cited on page 8).

[Sun+07b] X. Sun, P. Rosin, R. Martin, and F. Langbein. “Fast and Effective Feature-Preserving Mesh Denoising”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.5 (2007), pages 925–938 (cited on page 95).

(cc) BY-NC-ND

[Sus03] Mark Sussman. “A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles”. In: *Journal of Computational Physics* 187.1 (2003), pages 110 –136. ISSN: 0021-9991. DOI: https://doi.org/10.1016/S0021-9991(03)00087-1. URL: http://www.sciencedirect.com/science/article/pii/S0021999103000871 (cited on page 22).

[Wat] *The Effects of Air Pollution on Cultural Heritage*. Springer US, 2009 (cited on pages 61, 68).

[Töl08] Jonas Tölke. “Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA”. In: *Computing and Visualization in Science* 13.1 (2008), page 29. ISSN: 1433-0369. DOI: 10.1007/s00791-008-0120-2. URL: https://doi.org/10.1007/s00791-008-0120-2 (cited on page 23).

[Tur01] Greg Turk. “Texture Synthesis on Surfaces”. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’01. New York, NY, USA: ACM, 2001, pages 347–354. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383297. URL: http://doi.acm.org/10.1145/383259.383297 (cited on page 18).

[VPL06] Gilles Valette, Stephanie Prevost, and Laurent Lucas. “A Generalized Cracks Simulation on 3D-Meshes”. In: *Eurographics Workshop on Natural Phenomena*. Edited by Norishige Chiba and Eric Galin. The Eurographics Association, 2006. ISBN: 3-905673-38-X (cited on page 9).

[Wal+05] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. “Lightcuts: A Scalable Approach to Illumination”. In: *ACM Trans. Graph.* 24.3 (July 2005), pages 1098–1107. ISSN: 0730-0301. DOI: 10.1145/1073204.1073318. URL: http://doi.acm.org/10.1145/1073204.1073318 (cited on page 21).

[Wan+06] Jiaping Wang, Xin Tong, Stephen Lin, Minghao Pan, Chao Wang, Hujun Bao, Baining Guo, and Heung-Yeung Shum. “Appearance Manifolds for Modeling Time-variant Appearance of Materials”. In: *ACM Trans. Graph.* 25.3 (July 2006), pages 754–761. ISSN: 0730-0301. DOI: 10.1145/1141911.1141951. URL: http://doi.acm.org/10.1145/1141911.1141951 (cited on pages 17, 18, 89).

[War94] G. J. Ward. “The RADIANCE lighting simulation and rendering system.” In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. Edited by ACM. 1994 (cited on page 22).

[War+03] P.A. Warke, J.M. Curran, A.V. Turkington, and B.J. Smith. “Condition assessment for building stone conservation: a staging system approach”. In: *Building and Environment* 38.9-10 (2003). Building Stone Decay: Observations, Experiments and Modeling, pages 1113–1123 (cited on pages 24, 90).

[Wer+17] Sebastian Werner, Zdravko Velinov, Wenzel Jakob, and Matthias Hullin. “Scratch Iridescence: Wave-Optical Rendering of Diffractive Surface Structure”. In: (2017). arXiv: 1705.06086 (cited on pages 8, 9).

(cc) BY-NC-ND

[Wic+10] Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O'Brien. "Dynamic Local Remeshing for Elastoplastic Simulation". In: *ACM Trans. Graph.* 29.4 (July 2010), 49:1–49:11. ISSN: 0730-0301. DOI: 10.1145/1778765.1778786. URL: http://doi.acm.org/10.1145/1778765.1778786 (cited on page 9).

[Woj+07] Chris Wojtan, Mark Carlson, Peter J. Mucha, and Greg Turk. "Animating Corrosion and Erosion". In: *Proceedings of the Third Eurographics Conference on Natural Phenomena*. NPH'07. Prague, Czech Republic: Eurographics Association, 2007, pages 15–22. ISBN: 978-3-905673-49-4. DOI: 10.2312/NPH/NPH07/015-022. URL: http://dx.doi.org/10.2312/NPH/NPH07/015-022 (cited on page 12).

[WNH97] Tien-Tsin Wong, Wai-Yin Ng, and Pheng-Ann Heng. "A Geometry Dependent Texture Generation Framework for Simulating Surface Imperfections". In: *Proceedings of the Eurographics Workshop on Rendering Techniques '97*. London, UK, UK: Springer-Verlag, 1997, pages 139–150. ISBN: 3-211-83001-4. URL: http://dl.acm.org/citation.cfm?id=647651.731969 (cited on page 16).

[Wu+99] Yin Wu, Prem Kalra, Laurent Moccozet, and Nadia Magnenat-Thalmann. "Simulating wrinkles and skin aging". In: *The Visual Computer* 15.4 (1999), pages 183–198. ISSN: 1432-2315. DOI: 10.1007/s003710050171. URL: https://doi.org/10.1007/s003710050171 (cited on page 14).

[Xia+11] Lin Xia, Zilong Dong, Guofeng Zhang, Duanjin Chen, and Hujun Bao. "Interactive Weathering of Depth-inferred Videos". In: *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*. VRCAI '11. Hong Kong, China: ACM, 2011, pages 117–124. ISBN: 978-1-4503-1060-4. DOI: 10.1145/2087756.2087771. URL: http://doi.acm.org/10.1145/2087756.2087771 (cited on page 16).

[XDR11] Su Xue, Julie Dorsey, and Holly Rushmeier. "Stone Weathering in a Photograph". In: *Computer Graphics Forum* 30.4 (2011), pages 1189–1196. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2011.01977.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2011.01977.x (cited on pages 12, 13).

[Xue+08] Su Xue, Jiaping Wang, Xin Tong, Qionghai Dai, and Baining Guo. "Image-based Material Weathering". In: *Computer Graphics Forum* 27.2 (2008), pages 617–626. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2008.01159.x. URL: http://dx.doi.org/10.1111/j.1467-8659.2008.01159.x (cited on pages 18, 19).

[Yan+16] Ling-Qi Yan, Miloš Hašan, Steve Marschner, and Ravi Ramamoorthi. "Position-normal Distributions for Efficient Rendering of Specular Microstructure". In: *ACM Trans. Graph.* 35.4 (July 2016), 56:1–56:9. ISSN: 0730-0301. DOI: 10.1145/2897824.2925915. URL: http://doi.acm.org/10.1145/2897824.2925915 (cited on page 9).

[YL+92] YL Wu YL, CI Davidson, DA Dolske, and SI Sherwood. "Dry deposition of atmospheric contaminant: the relative importance of aerodynamic, boundary layer and surface resistance". In: *Aerosol Science and Technology* 16 (1992), pages 65–81 (cited on page 61).

[ZTS09] Rony Zatzarinni, Ayellet Tal, and Ariel Shamir. "Relief Analysis and Extraction". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 28.5 (Dec. 2009), 136:1–136:9. ISSN: 0730-0301 (cited on pages 26, 27).

(cc) BY-NC-ND

[Zho+17] Yang Zhou, Huajie Shi, Dani Lischinski, Minglun Gong, Johannes Kopf, and Hui Huang. “Analysis and Controlled Synthesis of Inhomogeneous Textures”. In: *Computer Graphics Forum* 36.2 (2017), pages 199–212. ISSN: 1467-8659. DOI: 10.1111/cgf.13119. URL: http://dx.doi.org/10.1111/cgf.13119 (cited on page 19).

[ZIK98] Sergey Zhukov, Andrei Iones, and Grigorij Kronin. “An Ambient Light Illumination Model.” In: *Rendering Techniques*. Edited by George Drettakis and Nelson L. Max. Springer, 1998, pages 45–56 (cited on page 37).

[ZH97] Qisu Zou and Xiaoyi He. “On pressure and velocity boundary conditions for the lattice Boltzmann BGK model”. In: *Physics of fluids* 9.6 (1997), pages 1591–1598 (cited on page 60).

(cc) BY-NC-ND