

Simulatie van corrosie op complexe objecten

Panagiotis Korovessis

Promotor: Prof. dr. Philippe Bekaert

Begeleider: dr. Tom Mertens

Eindverhandeling voorgedragen tot het behalen van de graad van
Licentiaat in de Informatica afstudeervariant Multimedia

School voor Informatie Technologie
Transnationale Universiteit Limburg
Diepenbeek, 12 augustus 2005

Samenvatting

Fotorealistische beeldgeneratie kan zeer mooie beelden leveren, maar veel van deze beelden zien er toch nog onnatuurlijk uit omdat de voorwerpen in de scene te perfect overkomen. Om een natuurlijker realisme te bekomen is het dan ook belangrijk verweerde oppervlakken te kunnen modelleren en visualiseren.

Met de huidige modellen kan men stofaccumulatie, afschilfering, metaalverkleuring, mosbegroeiing en roest simuleren. Deze modellen genereren automatisch een textuur waarin de verwerkingseffecten tot uiting komen. Het is zeer moeilijk om alle soorten verwerkingen door eenzelfde techniek te realiseren. Vermits corrosie op zich een veel voorkomend fenomeen is, gaan wij ons beperken tot het modelleren van roestvorming.

Het bestaand algoritme om roest te simuleren is intuïtief en merkt zich op door zijn simpliciteit. Het gebruik van een corrosiefunctie die de penetratie van corrosie op een locatie evalueert in functie van de tijd levert een vrij goede fysische benadering op. Het algoritme kan ook uitgebreid worden om andere verwerkingen te simuleren. Zijn eenvoud uit zich doordat de simulatie zich afspeelt in het 2D vlak dat wij willen uitbreiden naar arbitraire oppervlakken in 3D.

Ons model werkt volgens het hetzelfde principe, maar met het verschil dat de simulatie nu doorgaat in de objectruimte. Roest gaat voorgesteld worden aan de hand van losse deeltjes die rechtstreeks op het oppervlak geplaatst worden. Op elke tijdstip wordt er een deeltje toegevoegd aan de rand van een corrosievlek. Onze simulatie maakt gebruik van dezelfde corrosiefunctie die gebaseerd is op experimentele data.

Voor het renderen maken we gebruik van een raytracing techniek en het resultaat wordt opgeslagen in een textuur. De keuze van de kleur wordt beslist aan de hand van een empirisch verkregen kleurentabel uit echte voorbeelden (foto's).

Onze nieuwe techniek geeft ons een alternatief om destructieve corrosie te simuleren en te renderen. Ons spreidingsproces zou ook gebruikt kunnen worden voor andere verwerkingen (bv. mos) waarvan het onderliggende patroon een vlek is. Er wordt ook automatisch een textuur gegenereerd omdat het moeilijk is om via manuele textuurcreatie dezelfde verwerkingseffecten te verkrijgen. Kortom, de corrosie, verkregen door onze benadering, draagt zijn steentje bij voor realistische beeldsynthese.

Voorwoord

Ik zou graag enkele mensen willen bedanken die een onmisbare steun zijn geweest tijdens het schrijven van deze thesis. Mijn promotor Prof. dr. Philippe Bekaert zonder wie deze thesis nooit tot stand zou zijn gekomen. Mijn begeleider dr. Tom Mertens die mij vele nuttige opmerkingen heeft gegeven en bij wie ik altijd terecht kon met al mijn vragen. Tot slot mijn familie die mij altijd tot steun is geweest en die ik wil bedanken voor het nalezen van deze thesis.

Inhoudsopgave

1	Inleiding	1
2	Corrosie in de natuur	3
2.1	Inleiding	3
2.2	Verschijningsvormen	5
2.3	De corrosievergelijking	7
3	Overzicht van verwerkingssimulaties	8
3.1	Simulatie van patina	8
3.1.1	Tendensdistributie	8
3.1.2	Structuur van meerdere lagen	14
3.1.3	L-systemen	18
3.2	Simulatie van destructieve corrosie	22
3.2.1	Cellulaire automaten	22
3.3	Conclusie	26
4	Simuleren van corrosie op complexe objecten	28
4.1	Representatie	28
4.2	Selectie van startpunten	28
4.3	Algoritme	29
4.4	Weergave	34
4.5	Besluit	36

5 Resultaten	37
6 Algemeen Besluit	41
6.1 Samenvatting	41
6.2 Toekomstig werk	42

Hoofdstuk 1

Inleiding

Het uitzicht van materialen rondom ons wordt grotendeels bepaald door verweringsprocessen. Er zijn verschillende vormen van verwerking zoals stofophoping, patina's, erosie, schimmel op organisch materiaal enz.

Alhoewel de technieken voor realistische beeldsynthese er de voorbije jaren dramatisch goed op vooruit zijn gegaan, blijft er toch de algemene kritiek bestaan dat de beelden er te onnatuurlijk uitzien omdat de voorwerpen in de scene te perfect zijn. Men doet dan ook een beroep op de vakkundigheid van artiesten om een natuurlijker realisme te verkrijgen. Maar deze benadering is moeilijk en tijdrovend. Vandaar dat er nood is aan modellering- en visualisatiewerktuigen om gemakkelijker oude en versleten objecten te creëren.

Tot dus ver zijn er in de computer graphics slechts enkele studies geweest omtrent dit onderwerp. Stofophoping werd onderzocht door Miller [Mil94]. Dorsey [DPH96] vertelt ons hoe de regen objecten bezoedelt. In [WNH97] introduceert Wong een methode om stofophoping, patina's en afgeschilferde oppervlaktes voor te stellen. Het verweren van steen werd bestudeerd in [DEJ⁺99]. Dorsey [DH96] stelt ook een model voor om patina's te renderen. [CS00] stelt een model voor voor de ontwikkeling van patina's op begraven objecten.

Er is voornamelijk veel onderzoek geweest over het modelleren en renderen van patina's. Een andere vorm van corrosie, namelijk destructieve corrosie, werd geïntroduceerd door Merillou [MDG01] en blinkt uit door zijn eenvoud. De spreiding van corrosie wordt gesimuleerd op een textuur door op elk tijdstip een willekeurige aangetaste pixel te kiezen en vervolgens willekeurig een niet gecorrodeerde buurpixel als gecorrodeerd te bestempelen. Het grote nadeel is dat er moeilijkheden optreden aan de textuurranden bij complexe objecten.

In deze thesis stellen we een nieuw model voor om destructieve corrosie te simuleren en te renderen. Ons systeem heeft zowel een invloed op de geometrie van een object als op zijn kleur en weerspiegeling. Onze benadering is een uitbreiding van het cellulair automaat algoritme van Merillou [MDG01]. In plaats van een textuur te gebruiken voor de beschadigingen en de veranderingen van het uiterlijk in op te slaan, maken we hiervoor gebruik van partikels die rechtstreeks gegenereerd worden op het oppervlak van een object. Door met losse deeltjes te werken, is er geen reguliere grid en connectiviteit vereist. Een soortgelijke aanpak werd ons voorgedaan door Desbenoit [DGA04] voor de groei van mos op objecten.

In het algemeen kunnen corrosiereacties verdeeld worden in twee hoofdcategorieën: metalen patina's en roestige lagen. Metalen zoals koper ontwikkelen een extreem beschermende laag, of patina, onder invloed van de atmosfeer. Aan de andere kant worden metalen zoals ijzer op een destructieve manier aangetast door corrosie. Grote hoeveelheden van materie kunnen verloren gaan in tegenstelling tot patinerende materialen. Er kunnen gaten verschijnen in het oppervlak.

Om de verloren hoeveelheid materie te berekenen en om een aanvaardbare simulatie te verkrijgen, maken we gebruik van een wiskundige vergelijking over verloop van de tijd. In ons model worden de startpunten van corrosie willekeurig over het oppervlak van een object gekozen. Ons spreidingsproces resulteert in een aantal uniforme corrosieplekken bestaande uit nauw aaneengesloten partikels. De spreiding van de partikels over het oppervlak gebeurt gelijkaardig als bij Merillou [MDG01]. We kiezen nu willekeurig een deeltje dat zich aan de rand van een corrosieplek bevindt en plaatsen er een nieuwe partikel rond.

Voor het renderen maken we gebruik van een raytracing techniek die een interpolatie inhoudt volgens [She68] en het resultaat bergen we op in een textuur. Voor de kleur werken we met een kleurentabel verkregen uit echte wereldfoto's.

Overzicht van de thesis

In het volgende hoofdstuk zullen we kort kijken hoe corrosie ontstaat en in welke vormen dit voorkomt in de natuur. Daarna gaan we wat dieper in op de wiskundige vergelijking die de hoeveelheid verloren stof beschrijft.

Het hoofdstuk daarna is het grootste deel van de thesis. Het geeft een overzicht van bestaande simulatietechnieken voor corrosie. De technieken worden opgesplitst op basis van verwerings-type: patina en destructieve corrosie.

Hoofdstuk 4 is het belangrijkste deel van de thesis. Hier leggen we de nadruk op de algoritmie en de weergave van onze simulatie.

In hoofdstuk 5 worden de resultaten van onze simulatie besproken.

In het laatste hoofdstuk tenslotte besluiten wij onze thesis.

Hoofdstuk 2

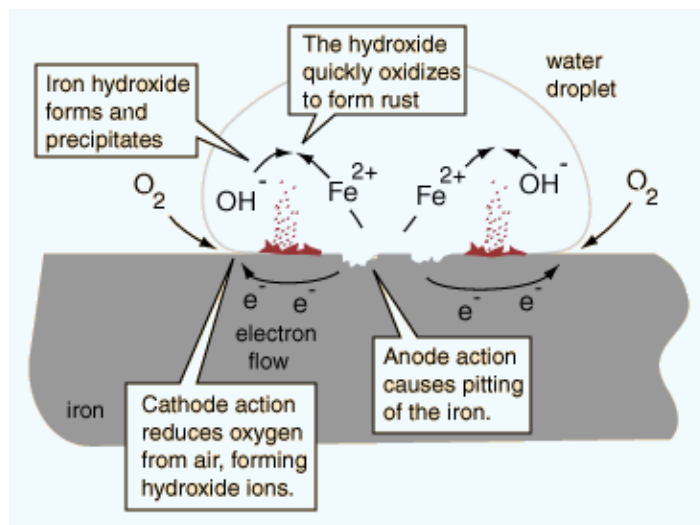
Corrosie in de natuur

In dit hoofdstuk zal het ontstaan van corrosie in de natuur worden besproken. Er zal een kort overzicht van de verschillende vormen van corrosie gegeven worden. De vergelijking die de hoeveelheid verloren materie uitdrukt over verloop van tijd, zal in dit hoofdstuk worden uitgelegd.

2.1 Inleiding

Corrosie kan gedefinieerd worden als: de ongewenste aantasting van een materiaal ten gevolge van chemische of elektrochemische reacties met componenten uit de omgeving. Deze definitie geldt voor elk materiaal: metalen, kunststoffen, rubber, glas, beton, enz. Zoals Dorsey [DH96] het beaamt, is corrosie moeilijk te modelleren omwille van een immense verscheidenheid aan reacties, condities en materialen. Wij gaan ons dan ook in deze thesis beperken tot de groep der metalen.

Bij corrosie van metalen hebben we het over een elektrochemische aantasting van het materiaal. Volgens [Eng] zijn hiervoor drie elementen nodig: een elektrolyt, zuurstof en een potentiaalverschil. Het elektrochemisch proces loopt onder gelijktijdige inwerking van de elektrolyt (vocht). Dit elektrolyt (bv. een dun laagje water) is aanwezig op een metalen oppervlak zodra een zekere vochtigheidsgraad is bereikt (bv. voor ijzer is dit ongeveer 60% in een onvervuilde atmosfeer). Hoe dan ook is deze graad sterk afhankelijk van milieuverontreinigende stoffen. Door opgeloste bestanddelen is dit water elektrisch geleidend. Op het metalen voorwerp komen plaatsen voor met een verschillend elektrisch oplossingspotentiaal. De verschillen in potentiaal ontstaan o.a. door verschillen in samenstelling van het metaal op het oppervlak, bv. door de aanwezigheid van insluitsels of door verschillen in samenstelling van de vloeistoffilm. Ten gevolge van deze potentiaalverschillen gaat een elektrische stroom lopen, waarbij bepaalde gedeelten van het metaal anodisch en andere gedeelten kathodisch worden (vgl. elektrolyse). Op anodische plaatsen gaat het metaal in oplossing en er ontstaan corrosieproducten of metaaloxiden (zie figuur 2.1).



Figuur 2.1: Corrosie van ijzer: Elektrochemisch proces waar een druppel water een elektrochemische cel vormt, waarin een elektrische stroom begint te vloeien, die het ijzer oxideert (Bron [Eng]).

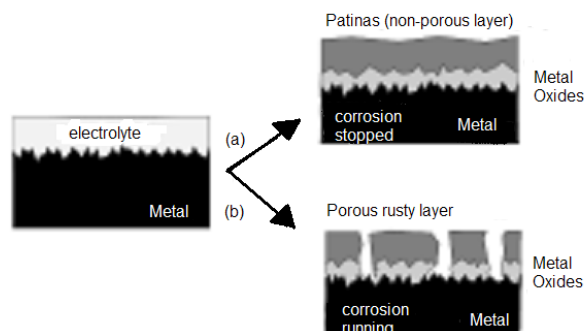
De chemische reacties produceren verschillende oxiden met verschillende chemische structuren. Deze oxiden hebben elk hun eigen kleuren (zie tabel 2.1) wat een verklaring vormt voor de gewoonlijk drukke en niet uniforme kleur van roest.

<i>formulatie</i>	<i>naam</i>	<i>kleur</i>
$\alpha - FeOOH$	goethiet	geel
$\gamma - FeOOH$	lepidocrociet	donker geel
Fe_3O_4	magnetiet	zwart
$\alpha - Fe_2O_3$	hematiet	rood
$\gamma - Fe_2O_3$	maghemiet	bruin

Tabel 2.1: Enkele onderdelen van roestige ijzer en hun kleuren (Bron [MDG01]).

Zoals we eerder hebben aangehaald, kan corrosie twee hoofdsenario's volgen zoals figuur 2.2 afbeeldt.

- (a): de corrosieproducten vormen een niet-poreuze laag op het oppervlak van het materiaal. In dit geval hebben zuurstof en water praktisch geen toegang meer tot het metaal. Dit betekent dat de corrosiereactie zeer traag gaat verlopen. Deze “beschermende” laag wordt een metalen patina genoemd [DH96].
- (b): de corrosieproducten zijn geconcentreerd in een poreuze laag (bv. roest). De poreusheid van zo een laag laat nog altijd toe dat water en zuurstof het onderliggende metaal bereiken. Vermits de corrosiereactie niet gestopt wordt, zal het metaal verder aangetast worden. We spreken dan ook van een “vernietigende” laag [MDG01].

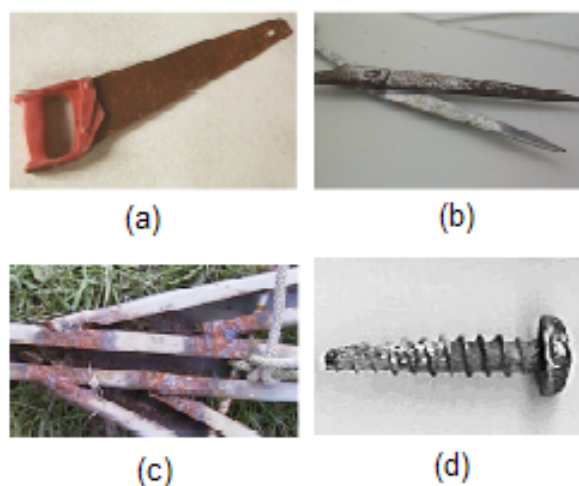


Figuur 2.2: Schema van het corrosie principe (a) patina's, (b) destructieve corrosie (Bron [MDG01]).

Vermits we het model van Merillou [MDG01] voor destructieve corrosie als basis voor ons nieuw systeem nemen, gaan wij ons dan ook beperken tot destructieve corrosie. Dat neemt niet weg dat het spreidingsproces niet gebruikt zou kunnen worden voor de vorming van patina's. Patina's hebben trouwens hetzelfde onderliggende patroon in de vorm van vlekken. In de volgende sectie gaan we enkele verschijningsvormen van destructieve corrosie bespreken die simuleerbaar zijn.

2.2 Verschijningsvormen

Een populaire classificatie van destructieve corrosievormen is die van Fontana [Fon86]. Van de acht klassen zijn er slechts vier met het oog gemakkelijk te onderscheiden. We gaan ons dan ook beperken tot deze gevallen.



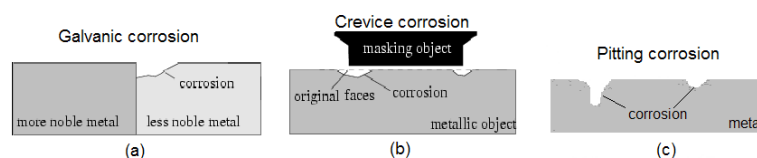
Figuur 2.3: Gecorrodeerde oppervlakken: (a) een zaag met uniforme corrosie (Bron [MDG01]), (b) een schaar met putcorrosie (Bron [MDG01]), (c) een aaneengelaste metalen frame met galvanische corrosie (Bron [MDG01]), (d) een bout met spleetcorrosie (Bron [Mac]).

Wanneer een metaal gelijkmatig over het gehele oppervlak wordt aangetast, spreekt men van *uniforme corrosie*. Deze doet een fraai uiterlijk zeer snel in kwaliteit teruglopen, maar voor de stevigheid van de aangetaste voorwerpen is deze vorm het minst schadelijk. Het is één van de belangrijkste vormen van corrosie. Zijn aantastingsnelheid is bijna constant over het gehele oppervlak.

Indien twee verschillende metalen met elkaar in contact komen en bedekt of ondergedompeld zijn in een elektrolyt, dan zal het meest edele metaal beschermd worden terwijl het minst edele metaal aangetast wordt. Men spreekt dan over *galvanische corrosie* of *contactcorrosie* (figuur 2.4(a)). De snelheidsfactor waarmee het minst edele metaal aangetast wordt is gelijk aan de verhouding tussen de “beschermende” en de “gecorrodeerde” onbeschutte oppervlakte.

Het probleem van *spleetcorrosie* (figuur 2.4(b)) doet zich voor bij zeer kleine spleten waarbij de verhouding tussen spleethoogte en spleetdiepte van maatgevend belang is op het corrosiegedrag. Water aan de buitenzijde van de spleet dringt door capillaire werking de spleet binnen. Diep in de spleet wordt zuurstof verbruikt door beginnende corrosie. Doordat de buitenzijde van de spleet wordt belucht, ontstaat een elektrolytische corrosiecel.

Putcorrosie (figuur 2.4(c)) is een plaatselijke putvormige aantasting van materialen, vaak ingeleid door plaatselijke beschadigingen aan het oppervlak of door de plaatselijke samenstelling van het medium (bv. aanwezigheid van chloriden). Meestal zijn de putten geheel gevuld met corrosieproduct. Putcorrosie wordt gekarakteriseerd door een putfactor die de verhouding uitdrukt tussen de diepste put en de gemiddelde verloren dikte (gemiddelde penetratie).



Figuur 2.4: (a) Contactcorrosie: minder edel metaal wordt aangetast, terwijl het sterker metaal ongemoeid blijft. (b) Spleetcorrosie: aantasting aan de buitenzijde van de spleet doordat daar meer zuurstof aankomt. (c) Putcorrosie: lokaal puntvormige aantasting van het metaal. (Bron [Fon86].)

De overige vormen: *interkristallijne corrosie*, *erosieve corrosie*, *spanningscorrosie* en *corrosievermoeïing* komen voor in meer specifieke gevallen (bv. in bepaalde legeringen of onder mechanische spanning). In realiteit is de waargenomen corrosie meestal een combinatie van deze verschillende vormen.

In de echte wereld is een metalen oppervlak immers nooit volledig puur. Zo kunnen bijvoorbeeld zeer dunne smeerlagen het corrosieproces vertragen. Ook kunnen er op het oppervlak onzuiverheden zitten waardoor er plaatselijke contactcorrosie optreedt. Microgeometrische details van het oppervlak zoals krassen zijn ook belangrijk. Een kras kan beschermende laagjes verwijderen waardoor de vorm van een kras lokale spleetcorrosie introduceert.

2.3 De corrosievergelijking

De kinetiek van corrosie wordt geëvalueerd aan de hand van corrosiesnelheid: de hoeveelheid aangetaste materiaal over de tijd per oppervlakte eenheid. Een generieke analytische schatting voor deze aantastingssnelheid blijft nagenoeg onmogelijk. Dit komt omdat er rekening moet gehouden worden met een zeer brede waaier van verschillende paramaters (exacte samenstelling van het materiaal, in welke staat bevindt het oppervlak zich, atmosferische condities, vervuiling, enz).

In [MDG01] heeft men uit experimentele data een wiskundige vergelijking kunnen afleiden. Vermits veel hedendaagse materialen gemaakt worden uit staal, heeft men de schattingen voor de corrosiesnelheid van staal gebruikt (ISO Norm 9223). Uiteraard kan men deze techniek ook gebruiken voor andere materialen. Tabel 2.2 toont de gemiddelde corrosiesnelheid van staal naar gelang simpele atmosferische toestanden.

<i>Atmosferische Conditie</i>	<i>eerste jaar</i>	<i>eerste 10 jaar</i>	<i>gemiddelde corrosiesnelheid</i>
zeer laag	< 10	< 5	< 0.8
laag	10-200	5-40	0.8-12
medium	200-400	40-100	12-50
hoog	400-650	100-250	50-150
zeer hoog	650-1500	250-750	150-700

Tabel 2.2: Gemiddelde corrosiesnelheid van staal ($g/m^2/jaar$) versus atmosferische condities als een functie van de tijd.

Met deze waardes kan de hoeveelheid verspilde materiaal geschat worden over een zekere tijd gegeven in jaren. Door deze hoeveelheid te berekenen voor verschillende tijden (bv. 1, 5, 10, 20 en 30 jaren), komen we tot de vergelijking die de aantastingssnelheid evalueert door het verloren gewicht te evalueren in $g/jaar$:

$$w(t) = k * t^n \quad (2.1)$$

w is het gecorrodeerde gewicht per oppervlakte eenheid, k en n zijn constanten die afhankelijk zijn van de atmosferische conditie en t is de tijd gegeven in jaren. De waarden van k en n , afgeleid van tabel 2.2, vinden we terug in tabel 2.3.

<i>Atmosferische Conditie</i>	<i>k</i>	<i>n</i>
zeer laag	5.5	0.565
laag	100.4	0.441
medium	274.6	0.516
hoog	477.4	0.638
zeer hoog	954.8	0.789

Tabel 2.3: Constanten bij de analytische functie om de corrosiesnelheid te evalueren.

Hoofdstuk 3

Overzicht van verwerkingssimulaties

In dit hoofdstuk zullen we verschillende technieken bespreken die kunnen gebruikt worden om corrosie te simuleren. Vermits corrosie volgens twee scenario's kan verlopen, splitsen we de simulatietechnieken op volgens hun verwerkingstype.

Deze technieken zijn simulaties die automatisch een textuur genereren, ze berusten op één of ander model. Men maakt een abstractie van de wereld en men probeert de spreiding van corrosie te modelleren. Er worden verschillende veronderstellingen gemaakt over het corrosieproces. Uitgaande van deze veronderstellingen maakt men een wiskundig model waarmee men probeert te voorspellen hoe corrosie een object aantast. Sommige modellen leggen restricties op verweringsprocessen die kunnen gesimuleerd worden, terwijl anderen meerdere verwerkingen aankunnen.

3.1 Simulatie van patina

3.1.1 Tendensdistributie

Hoewel de patronen van ontsieringen aan het oppervlak onregelmatig schijnen, kan men toch de geometrische afhankelijke aard van de onderliggende verdeling van de smet waarnemen.

Het modelleren van deze onderliggende distributie noemt men de *tendensverdeling* in [WNH97]. Het mogelijk voorkomen van een verweringspatroon aan een punt op het oppervlak wordt gemodelleerd als een scalaire hoeveelheid, *tendens* genaamd.

Dit geeft een erosiefunctie op het object die dan in een driedimensionale textuur kan omgezet worden. Met dit model kunnen o.a. stofaccumulatie, afschilfering en patina gesimuleerd worden.

Abstracte verweringsbronnen

De formatie van ontsieringen gebeurt onder invloed van verschillende soorten abstracte verweringsbronnen. Bijvoorbeeld, krassen komen vooral voor aan de handvat van een autodeur

ten gevolge van frequent menselijk contact. In dit geval kunnen we ons een abstracte krasbron inbeelden in de buurt van het handvat. Deze abstractie van verweringsbronnen is analoog aan het concept van lichtbronnen.

Beschouw de fluxdichtheid van een oppervlaktepunt P vanwege een specifieke verweringsbron S_i . Dit stelt het effect voor van een verweringsbron op dat oppervlaktepunt. De waarde van de fluxdichtheid is afhankelijk van de geometrische relatie tussen het oppervlaktepunt en de verweringsbron. Het neemt waarden aan in $[0,1]$. De waarde 0 betekent dat het oppervlaktepunt vrij is van onvolmaaktheden, terwijl de waarde 1 aangeeft dat het volledig verweerd is.

Zoals bij lichtbronnen, zijn er verschillende vormen van verweringsbronnen: ambient, point, directional, spotlight en slide projector. De fluxdichtheden van deze vijf vormen zijn gegeven in (3.1). F_{S_i} is de fluxdichtheid vanwege de i^{de} verweringsbron op positie S_i . Figuur 3.1 toont het visueel effect van deze vijf verweringsbronnen.

$$\begin{aligned}
(a) : F_{S_i} &= \text{constant}, \\
(b) : F_{S_i} &= \frac{1}{\rho_i^2} \cos^{\eta_c}(\theta_i) \text{ of } \frac{1}{\rho_i^2} \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right) \\
(c) : F_{S_i} &= \cos^{\eta_c}(\theta_i) \text{ of } \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right) \\
(d) : F_{S_i} &= \frac{s}{\rho_i^2} \cos^{\eta_c}(\theta_i) \text{ of } \frac{s}{\rho_i^2} \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right) \\
\text{where } s &= \text{smoothstep}(c_{out}, c_{in}, S_i P \cdot A), \\
(e) : F_{S_i} &= \frac{m}{\rho_i^2} \cos^{\eta_c}(\theta_i) \text{ of } \frac{m}{\rho_i^2} \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right) \\
\text{where } m &= \text{map}(S_i P).
\end{aligned} \tag{3.1}$$

Waar ρ_i de afstand is tussen P en S_i . θ_i is de hoek tussen de oppervlaknormaal in P en de vector PS_i , η_c controleert de cosinus functie, η_g is de Gaussiaanse standaardafwijking en A is de richting waarheen het spotlight naartoe schijnt. c_{in} is de cosinus van de halve hoek aan de punt van de binnenste spotlichtkegel, oppervlaktepunten binnen deze kegel zullen volledig belicht worden. Daarentegen is c_{out} de cosinus van de halve hoek aan de punt van de buitenste spotlichtkegel, oppervlaktepunten buiten deze kegel zullen niet beïnvloed worden door de bron. We merken op dat $c_{out} \leq c_{in}$. De functie $\text{smoothstep}(a, b, v)$ geeft 1 terug als $v \geq b$ en 0 als $v \leq a$, het interpoleert v tussen 1 en 0 als $a \leq v \leq b$. De functie $\text{map}()$ geeft de textuurwaarde die geïndexeerd is door de vector $S_i P$.

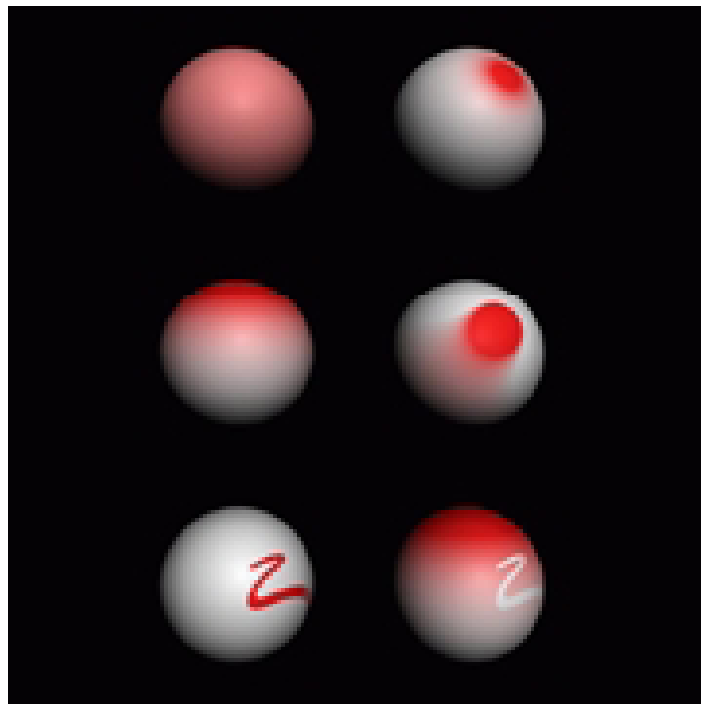
Alle vergelijkingen in 3.1 behalve de ambiënte bron, beschrijven een cosinus en een Gaussiaanse verdeling. Het verschil tussen beide wordt enkel zichtbaar als de hoek $\geq \frac{\pi}{2}$. De cosinus functie geeft dan 0 terug, terwijl de Gaussiaanse functie nog steeds waarden die niet gelijk zijn aan 0, teruggeeft. Dit betekent dat de cosinus functie een plotse visuele verandering met zich meebrengt. Hoe dan ook is de cosinus functie efficiënter te berekenen dan de Gaussiaanse functie.

Als er meerdere verweringsbronnen aanwezig zijn, dan wordt de totale fluxdichtheid F_S van

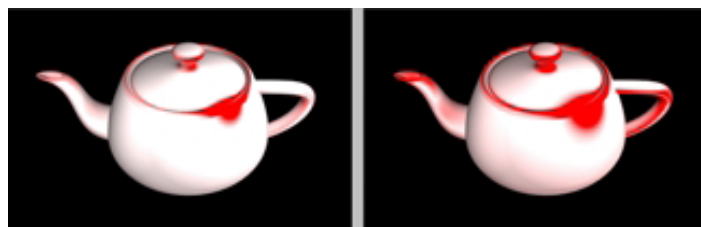
een oppervlaktepunt gegeven door:

$$F_S = \sum_{i=1}^m \gamma_i F_{S_i}, \quad (3.2)$$

Waar m het totaal aantal verweringsbronnen is, F_{S_i} is een van de functies in (3.1), γ_i is een vergrotingsfactor $\in \mathfrak{R}$. γ_i is positief als S_i een positieve bron is, anders negatief. Door de vergrotingsfactor te verhogen, kan de groei van de onderliggende tendensverdeling gesimuleerd worden (figuur 3.2).



Figuur 3.1: Verschillende bronnen. (a)linksboven, ambient. (b)rechtsboven, point. (c)links in het midden, directional, (d)rechts in het midden, spotlight. (e)linksonder, slide projector. (f)rechtsonder negatieve slide projector en positief directional. Bron [WNH97].



Figuur 3.2: Van links naar rechts, de vergrotingsfactoren van de verweringsbronnen zijn verhoogd. Bron [WNH97].

Geometrische factoren

Het gebruik van abstracte verweringsbronnen is echter niet voldoende om de tendensverdeling in detail te modelleren. Dit is te wijten aan het niet triviale patroon van de verdeling. Daar de verdeling afhankelijk is van de geometrie, kunnen we de geometrische informatie gebruiken om de verdeling automatisch meer in detail te genereren. De blootstelling en de kromming van het oppervlak zijn zulke belangrijke geometrische factoren. De uiteindelijke tendens T in een oppervlaktepunt P , is de totale fluxdichtheid F_S verstoord door een functie α die alle geometrische factoren G_j in rekening brengt.

$$\begin{aligned} T' &= F_S * \alpha(G_1, G_2, G_3, \dots), \\ T &= \begin{cases} 0 & \text{als } T' < 0, \\ T' & \text{als } 0 \leq T' \leq 1, \\ 1 & \text{als } T' > 1 \end{cases} \end{aligned} \quad (3.3)$$

Waar G_j is de j^{de} geometrische factor. De verstoringsfunctie $\alpha()$ is een functie van alle geometrische factoren en geeft een positief reële waarde terug. Het geeft 1 terug als er geen verstoring is.

Oppervlakte blootstelling en toegankelijkheid

De blootstelling van het oppervlak in een oppervlaktepunt P is een maat voor de blootstelling aan lucht. Beschouw een straal R_i vanuit P die een object intersecteert op een afstand d_i (Figuur 3.3(a)). Als d_i vergroot (het obstakel beweegt weg van P), dan verkleint het effect van dat obstakel op de oppervlakteblootstelling G_ξ , het is meer blootgesteld. We definiëren een monotoon stijgende weegfunctie ω voor d_i die waarden teruggeeft tussen $[0,1]$: $\omega(d_i) = 0$ als $d_i = 0$ en $\omega(d_i) \rightarrow 1$ als $d_i \rightarrow +\infty$ (geen intersectie). De volgende functie voldoet aan deze eisen en is eenvoudig te berekenen:

$$\omega(d_i) = \frac{d_i}{d_h + d_i} \text{ met } d_h > 0 \quad (3.4)$$

De constante d_h is de afstand waarbij het object half blootgesteld is en die, grof geschat, de gemiddelde ruimte is tussen P en andere obstakels die de blootstelling zou reduceren tot 0.5. Het is een indirecte parameter om te definiëren hoe ver ver is.

De blootstelling G_ξ wordt vastgesteld met behulp van raycasting. We schieten een eindig aantal teststralen af, gelijk verdeeld over de bovenste hemisfeer van P , om de blootstelling te detecteren (figuur 3.3(b)). Dezelfde stralenverdeling kan hergebruikt worden voor andere oppervlaktepunten op het object. De blootstelling van het oppervlak wordt benaderd door het gemiddelde van de monsters:

$$G_\xi \approx \frac{1}{n_0} \sum_{i=1}^{n_0} \omega(d_i) \quad (3.5)$$

Waar n_0 is het eindig aantal afgeschoten stralen, d_i is de afstand tussen P en het intersectiepunt en $\omega(d_i)$ is de weegfunctie in 3.4. Hoe meer stralen er afgeschoten worden, hoe preciezer de benadering. Het aliasing probleem, eigen aan raycasting, kan geminimaliseerd worden met supersampling.



Figuur 3.3: (a): Ray R_i intersekteert met een object. (b): n gelijk verdeelde stralen vanuit het punt P . Bron [WNH97].

De geometrische functie α wordt gedefinieerd als een lineaire functie:

$$\alpha = 1 + r_0 G_\xi \quad (3.6)$$

Waar $r_0 \in \mathfrak{R}$ een schaalfactor is voor de blootstelling van het oppervlak en G_ξ is de blootstelling van het oppervlak gelegen in $[0,1]$.

Voor iedere vertex van een polygon wordt de blootstelling berekend en opgeslagen bij die vertex. De blootstelling voor een punt in een polygon kan lineair geïnterpoleerd worden volgens de waardes bij zijn vertices.

Oppervlakte kromming

Verf heeft meer kans om af te schilferen op een uitpuilend oppervlak dan op een plat oppervlak. Deze afschilfering van verf breidt zich dan vervolgens uit naar de omliggende omgeving. Dit is omdat het uitstekende gedeelte van het oppervlak meer kans heeft om beïnvloed te worden door uitwendige krachten. De blootstelling van het oppervlak alleen kan dit niet in rekening brengen, beschouw hiervoor een convex oppervlak en een plat oppervlak. Beiden zijn volledig blootgesteld, maar de verf op het gebogen oppervlak heeft een grotere kans om af te schilferen.

De kromming G_κ in een vertex P van een polygoon wordt benaderd zoals in [Tur92]. Laat PQ_1, PQ_2, \dots en PQ_n de edges voorstellen van een object en laat N de vertexnormaal van P zijn, dan is ϕ_i de hoek tussen PQ_i en N . De kromming met edge PQ_i wordt gegeven door:

$$G_{\kappa_i} = \frac{|PQ_i|}{2} * \tan(\phi_i) \quad (3.7)$$

Vervolgens is de kromming G_κ bij een vertex P :

$$G_\kappa = \frac{G_{\kappa_{min}} + G_{\kappa_{max}}}{2} \quad (3.8)$$

Waar $G_{\kappa_{min}}$ het minimum is van alle G_{κ_i} en $G_{\kappa_{max}}$ het maximum is van alle G_{κ_i} .

Wanneer de kromming berekend is, kan het net zoals bij de blootstelling opgeslagen worden voor later gebruik. Nu kunnen we deze kromming gebruiken in de berekening van de geometrische functie α , deze functie wordt uitgebreid als volgt:

$$\alpha = 1 + r_0 G_\xi + r_1 G_\kappa \quad (3.9)$$

Waar $r_0, r_1 \in \Re$ een schaalfactor, $G_\xi \in [0, 1]$ is de blootstelling van het oppervlak en $G_\kappa \in \Re$ is de kromming van het oppervlak.

Textuur generatie

Eens de tendens T berekend is, kan er een textuur gegenereerd worden volgens de tendens verdeling. Meer ontsieringen zouden moeten voorkomen op plaatsen met hogere tendens. Om textuurdistortie te voorkomen, wordt er gebruik gemaakt van solid texturing [Pea85]. De 3D textuur wordt bekomen met behulp van een wiskundige vergelijking $f : \Re^3 \rightarrow \Re$ die gegeven een positie, een scalaire waarde teruggeeft. Een populair model is de Perlin's noise functie [Per85]. De finale tendens T'' in een punt P wordt gegeven door:

$$T'' = T * noise(P) \quad (3.10)$$

Waar T de tendens uit 3.3 is, P is de positie en de functie $noise()$ is Perlin's noise functie die een waarde in $[0,1]$ teruggeeft.

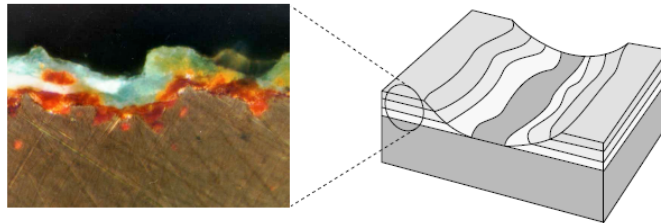
Besluit

Deze techniek laat de gebruiker toe om intuïtief de totale tendensdistributie te controleren door het plaatsen van verschillende verweringsbronnen rond het object. De groei wordt gesimuleerd door de invloed van een verweringsbron te vergroten met een scalaire parameter, waardoor de groei zich bij iedere tijdstap over alle richtingen gelijktijdig uitbreidt. Door de tendens voor te stellen als een vectoriële grootheid zou men de groeirichting van de smet kunnen bepalen.

Als het aantal verweringsbronnen stijgt, wordt de berekening van de tendens in elk oppervlaktepunt steeds zwaarder. Vooral de berekening van de geometrische blootstelling via raycasting lijkt hieronder. Dit kan men optimalizeren door Miller [Mil94] zijn benadering voor oppervlakteblootstelling te gebruiken. Hierbij is de blootstelling gelijk aan de straal van de grootste bol in een oppervlaktepunt zonder dat die bol een ander object raakt.

3.1.2 Structuur van meerdere lagen

In [DH96] wordt er een model voorgesteld om metalische patina's te simuleren gebaseerd op een structuur van meerdere lagen. Zoals we in hoofdstuk 2 hebben gezien, is de ontwikkeling van een patina een proces dat resulteert in een complex systeem van lagen. De veranderingen van het oppervlak die deze lagen creëren, gebeuren aan de hand van enkele specifieke operatoren en worden gecontroleerd door welbepaalde fysische spreidingstechnieken.



Figuur 3.4: Een oppervlak van een patina als een stapel lagen. (a) Microscopische foto van een patina op een koperoppervlak die de lagen toont. (b) Abstractie van de lagenstructuur. Bron [DH96].

Lagen en materialen

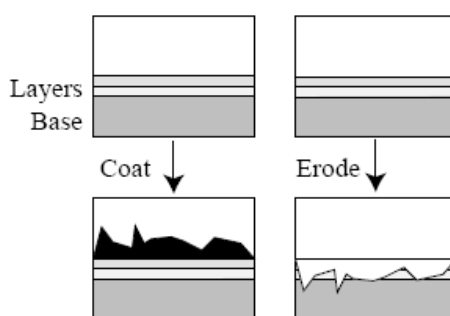
In deze representatie bestaat een oppervlak uit een stapel van n lagen (Figuur3.4). De 0^{de} laag komt overeen met het basismateriaal van oneindige dikte. De totale dikte van een stapel lagen wordt verondersteld relatief klein te zijn t.o.v. de oppervlakte die de laag bedekt. Elke individuele laag heeft ook een dikte, die op sommige plaatsen 0 kan zijn. Elke laag bestaat uit homogeen materiaal. Enkel de dikte van het materiaal varieert in functie van plaats.

Operatoren

De ontwikkeling van patina wordt geïmplementeerd als een serie van operatoren op de lagen. Door de sequentie van de operatoren te controleren, kunnen er verschillende effecten gerealiseerd worden. De volgende operatoren zijn beschikbaar:

- **coat materiaal dikte diktemap.** De *coat* operator voegt een nieuwe materiaallaag toe aan een oppervlak. De nieuwe laag heeft een maximale dikte die gemoduleerd wordt door de diktemap (zie figuur 3.5).
- **erode dikte diktemap.** De *erode* operator verwijdert materiaal van een gelaagd oppervlak. De diepte van de erosie kan gecontroleerd worden door een diktemap. De erosie operator gaat doorheen de stapel van lagen en vermindert de dikte van elke laag tot de gewenste hoeveelheid materiaal is verwijderd (zie figuur 3.5).

- **fill materiaal hoogte hoogtemap** De *fill* operator legt materiaal neer tot een gegeven hoogte boven het basismateriaal. Dit is ongeveer gelijkaardig met het vullen van alle valleien met water tot het water een gegeven hoogte bereikt. Dit is een simpele manier om de afzetting van materiaal te simuleren in spleten en barsten.
- **polish hoogte hoogtemap.** De *polish* operator verwijdert materiaal totdat er een gegeven hoogte boven het basismateriaal bereikt is. Het resultaat is een glad effect over een oppervlak. Een variatie van de operator verwijdert materiaal tot een gegeven oppervlakte van het oppervlak is blootgelegd. Deze versie kwam tot stand na het observeren van profielen en statistieken van ruwe oppervlaktes nadat deze waren zuiver gemaakt.
- **offset radius.** De *offset* operator past een materiaal toe op een oppervlak door eerst een dikke laag toe te voegen. Daarna wordt het gedeelte dat toegankelijk is voor een bol met een gegeven radius, verwijderd. De offset oppervlak wordt berekend volgens de techniek in [Mil94].



Figuur 3.5: De coat en erode operatoren (bron [DH96]). Terwijl de coat operator een extra laag toevoegt, verwijdert de erode operator materie.

Aan de hand van *scripts* wordt er een gelaagd oppervlak aangemaakt uit een verzameling materialen en diktemappen. Elk materiaal en diktemap stelt de huidige aard voor van het gelaagd oppervlak. In een latere sectie zal er een voorbeeld gegeven worden van zo'n *script*.

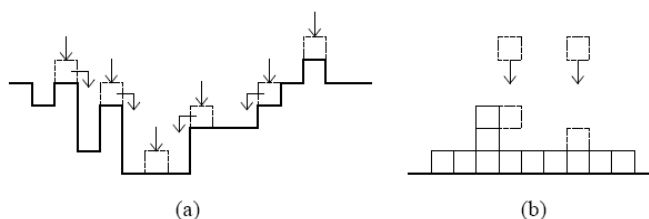
Textuur/diktemappen

De beschreven operators zijn in staat om een meerdere lagen patina te modelleren, maar deze laagstructuur is volledig uniform en mist de variaties en rijkdom van natuurlijke details. Om deze reden stelt men nu enkele fysische aanvaardbare methodes voor om deze operatoren te moduleren over het oppervlak. Er zijn twee representaties voor ruimtelijke variërende diktemappen:

1. **Rechthoekige textuurmaps:** de dikte wordt gecontroleerd door een standaard 2D textuurmap.
2. **Driehoeken als textuurmaps:** de diktes worden voorgesteld als waardes die horen bij de vertices van een driehoek.

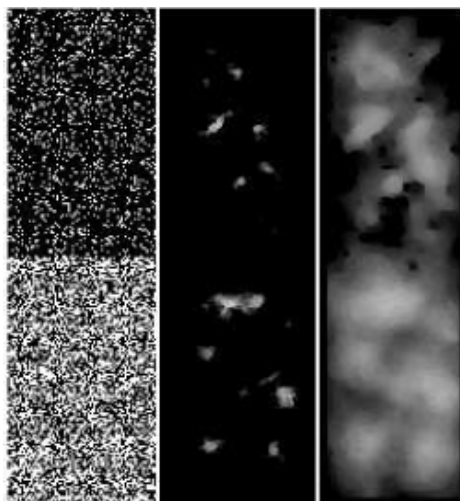
Om variaties van dikte over tijd te simuleren gebruikt men groeimodellen voor fractalen over een oppervlak [BS95]. Deze groeimodellen zijn ontwikkeld door natuurkundigen en simuleren de fysische groei van fractalen op oppervlakken. Enkel de modellen die te maken hebben met het afzetten van materiaal en met de laterale groei van vlekken op oppervlakken, worden beschouwd. Deze modellen worden ook voorzien van een groeisnelheid zoals lineair, parabolisch en logaritmisch. De volgende modellen worden voorzien:

- **Steady thickening (ST)**. Dit model creëert een eenvoudig, relatief uniform patroon dat in dikte toeneemt in verloop van tijd. Het oppervlak wordt gelijkmatig bemonsterd met een klein aantal punten. Een initiële dikte wordt toegekend aan elk punt en de dikte voor tussenliggende punten wordt geïnterpoleerd. In loop van de tijd wordt de dikte aan ieder sample punt verhoogd door een door de gebruiker opgegeven groeisnelheid. Een kleine hoeveelheid van ruis wordt toegevoegd aan het patroon om zijn verschijning natuurlijk te houden.
- **Random deposition (RD)**. Vanuit een willekeurig gekozen positie over het oppervlak valt een partikel verticaal totdat het de top van het oppervlak onder zich bereikt waar het vervolgens aan blijft plakken. Dit model wordt geïmplementeerd door willekeurig een positie i te kiezen en door zijn hoogte $h(i, t)$ met 1 te verhogen, waarbij t de tijdstap is in $h(i, t)$. Een variant op dit model (zie figuur 3.6) laat het afgezette particle toe te bewegen langs het oppervlak over een eindige afstand. Het vestigt zich op de positie met de laagste hoogte. Het originele model verschaft ons een ruwer oppervlak dan de variant die voor een gladder patroon zorgt.
- **Ballistic deposition (BD)**. Dit model laat ook een partikel los vanuit een willekeurige positie boven een oppervlak. Het partikel valt verticaal totdat het het oppervlak raakt en waar het bijgevolg aan blijft plakken (zie figuur 3.6). De hoogte van de interface in dat punt i wordt verhoogd tot $\max[h(\text{buurpunten}, t), h(i, t) + 1]$. De groei wordt gedefinieerd door een simpele functie die de gemiddelde hoogte van het oppervlak berekent. Als het aantal partikels die aankomen op een positie constant is, dan verhoogt de gemiddelde hoogte lineair over tijd. Het verschil tussen RD en BD is dat bij RD de dikte in ieder punt op het oppervlak onafhankelijk groeit, omdat er geen mechanisme is dat correlatie kan genereren over het RD interface. Door het feit dat de partikels bij BD aan de edge van nabijgelegen punten kunnen blijven plakken, leidt dit tot laterale groei.



Figuur 3.6: (a) Random deposition: de gebogen pijlen duiden op de diffusie van de variant. (b) Ballistic deposition. Bron [DH96] en [BS95].

- **Directed percolation depinning (DPD)**. Startend met een verzameling van initiële vlekken op een oppervlak, ontwikkelen we hiermee een interface dat in alle richtingen groeit volgens een eenvoudige kansfunctie in 2D en de dikte verhoogt. We beginnen met een simpele 2D grid en markeren een hoeveelheid cellen als geblokt en de rest als ongeblokt. Over de tijd gezien, breiden de vlekken zich uit naar ongeblokte cellen omdat deze een zekere vochtigheidsgraad hebben die de groei stimuleert (zie figuur 3.7).



Figuur 3.7: links: een grid van geblokte en ongeblokte cellen. Onderaan zijn meer ongeblokte cellen dan bovenaan., midden: een vroege stage van DPD, rechts: een late stage van DPD. Bron [DH96].

Scripts

De ontwikkeling van de lagenstructuur gebeurt door gebruik te maken van *scripts*. Deze *scripts* beschrijven de volgorde van de operaties en specificeren de groeimodellen. Ze genereren zo de diktemappen in functie van de tijd. Deze groeimodellen kunnen gecontroleerd worden door de omgeving en de geometrie van het oppervlak. Uiteindelijk is de laagstructuur de output die voorgesteld wordt als een serie van materiaaleigenschappen en textuurmappen die doorgegeven worden aan het renderingsysteem. Hieronder volgt een voorbeeld van zo'n *script*. De groeimodellen worden geïdentificeerd door hun naam, groeisnelheid, stapgrootte en totaal aantal stappen. De operatoren krijgen de nodige parameters mee zoals wordt aangeduid in bovenstaande sectie 3.1.2.

```
new copper;
coat cuprite 1.2 texture(DPD linear 5 40);
coat patina 3.0 texture(BD linear 10 20)
erode 0.5 texture(BD linear 5 20);
render maps;
```

Besluit

De voorstelling van patina door meerdere lagen komt overeen met de structuur van patina's in de natuur. De groei wordt gecontroleerd door operatoren die gebaseerd zijn op fysische groeimodellen. De gebruiker heeft een volledige controle over de groei en het uitzicht van het materiaal door het invoeren van *scripts*. Terwijl zulke *scripts* de artiest wel controle bieden om bepaalde effecten te realiseren, zou het beter zijn om ze via een interactieve wijze te kunnen gebruiken.

Het groeiproces wordt uitgevoerd in textuurmapping die overeenkomen met de verschillende lagen. Dit houdt in dat er moeilijkheden optreden aan de textuurranden bij het mapping van de textuur op een complex object.

3.1.3 L-systemen

Een Lindenmayer systeem (L-systeem) [Lin68] is een parallel herschrijfsysteem. Het bestaat uit een axioma en één of meerdere productieregels. De productieregels beschrijven door welk nieuw woord ieder symbool in het axioma vervangen moet worden. Door herhaaldelijk de symbolen in het axioma te vervangen zoals aangegeven door de productieregels, bekomt men een (meestal lang en ingewikkeld) woord.

Het toepassen van de herschrijfgeregels wordt vaak gevisualiseerd door gebruik te maken van de schildpadinterpretatie [PL90]. De opzet hiervan is om een woord, bekomen door toepassing van een L-systeem, van links naar rechts te scannen. De tegengekomen symbolen worden vervolgens geïnterpreteerd om als het ware een schildpad voort te bewegen in een 3D-ruimte. Deze bewegingen resulteren in een verandering van toestand.

L-systemen zijn een formalisme dat toelaat de vorm van planten en bomen zeer compact en volgens de regels van de biologie te beschrijven. Prusinkiewicz [PJM94] kwam op de proppen met omgevingsgevoelige L-systemen om de interactie tussen de plant en zijn omgeving te simuleren. Door gebruik te maken van een eenvoudige functie op de positie van de schildpad kunnen we de groei van planten limiteren. De open L-systemen van Měch en Prusinkiewicz [MP96] zijn een verdere uitbreiding van de omgevingsgevoelige L-systemen, waarbij communicatie tussen de plant en de omgeving in twee richtingen mogelijk is. De omgeving kan allerlei informatie doorgeven aan de plant, maar de plant kan ook informatie verschaffen aan de omgeving, die hierop eventueel kan reageren.

We hebben in hoofdstuk 2 gezien dat de transport van de metaalionen en elektronen een belangrijke rol spelen bij de vorming van corrosie, in [CS00] simuleert men dit transport met L-systemen. Metalische patina bestaat uit verschillende lagen en iedere laag bestaat uit verschillende chemische structuren. We beschouwen een homogeen patina model voor koper. Elke laag van patina's wordt verondersteld homogeen te zijn, waardoor er enkel twee lagen zijn: de groene patina en de adherente patina.

Erosiefunctie

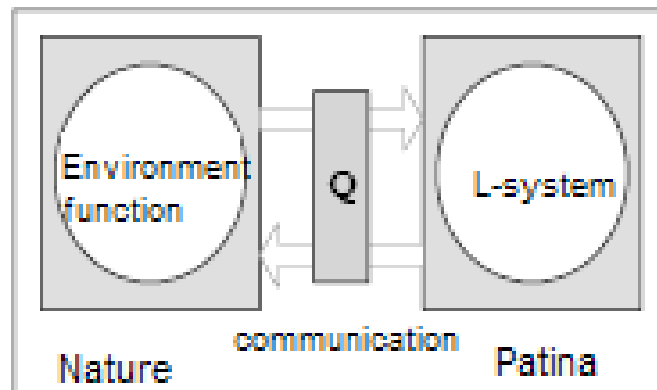
Voordat we de invloeden van de natuur kunnen modelleren op de ontwikkeling van patina, moeten we eerst een tendensverdeling zoals in het vorig hoofdstuk 3.1.1 definiëren om aan te geven waar het object het sterkst zal eroderen. We gebruiken hier geen verweringsbronnen waardoor de tendens T in een oppervlaktepunt als volgt wordt gedefinieerd:

$$T = \alpha * C + \beta * A + \gamma * G + \delta * S \quad (3.11)$$

Waar C de kromming van het oppervlak voorstelt, A de blootstelling van het oppervlak, G is het tropisme en S is de karakteristiek van de omgeving. α , β , γ en δ zijn constante gewichten. $G = N \cdot g$ is een maat voor de hoeveelheid water dat een oppervlak kan vasthouden met N de normaal van het oppervlak en g de zwaartekracht.

Patina model

Het framework van open L-systemen wordt gebruikt om de relatie te beschrijven tussen de groei van patina's en de nabije omgeving. Het framework in figuur 3.8 bestaat uit twee modules: de patina module en de natuur module. Een communicatie interface handelt als een brug tussen beide modules. Een gereserveerd symbool Q duidt de communicaties aan.



Figuur 3.8: Conceptueel framework van de patinatie.

De patina module stuurt een boodschap naar de natuur module voordat het L-systeem produceert. Deze boodschap bevat de volgende informatie: de positie van de schildpad, het type patina en de levensduur van de schildpad. De natuur module verwerkt alle informatie verkregen door de patina module en verzamelt de overeenkomstige tendenswaarden van de schildpad zijn burens volgens zijn type patina en zijn nabije omgeving. De volgende toestand van de schildpad wordt geëvalueerd en er wordt een record bijgehouden in de module. De resultaten worden dan teruggestuurd naar de patina module. De volgende toestand van de schildpad wordt er geëvalueerd en hier wordt er ook een record bijgehouden. De volgende afleidingsstap start volgens de productieregels van het L-systeem nadat alle informatie van de natuur module is ontvangen.

Bij aanvang van de patinatie worden er willekeurig een aantal vertices gekozen. Volgens het L-systeem gaat de patina ontwikkeling vanuit deze vertices verder en volgt het de edges die verbonden zijn met deze vertices. De groeirichting van de patina op de huidige positie, aangevraagd door de patina module, wordt beslist door de natuur module. De natuur module begint de tendensen van de buurvertices van de huidige vertex te verzamelen nadat het bericht van de patina module is ontvangen. De buurvertices die al patina's hebben, worden buiten beschouwing gelaten. Onder de burens worden er vertices geproduceerd als hun tendens waarde hoger is dan een treshold en hoger is dan de tendens van de huidige vertex. De natuur module beslist of de groei moet worden stopgezet of als er moet overgegaan worden naar een andere type van patina als er geen gepaste vertex gevonden wordt om te groeien.

De productieregels van het patina model zijn als volgt geformuleerd:

```
#define L1 80 / * initial activity of patina type 1 * /
#define L2 2 / * initial activity of patina type 2 * /
axiom : Q(M,0)A(V1,L1,1)Q(M,0)A(V2,L1,1)...Q(M,0)A(Vk,L1,1)
p1 : Q(M,n) < A(p,l,t) : l == 0 → R(p,t)
p2 : Q(M,n) < A(p,l,t) : n == 0 & t == 1 → R(p,t)Q(M,0)A(p,L2,t+1)
p3 : Q(M,n) < A(p,l,t) : n == -1 → R(p,t)
p4 : Q(M,n) < A(p,l,t) : * → R(p,t)Q(M,0)A(M[0],l-1,t)
      Q(M,0)A(M[1],l-2,t)...Q(M,0)A(M[n-1],l-n,t)
```

Hier is M een array van kandidaten voor groeiende patina die teruggeven is door de natuur module, en is n het aantal kandidaten. De natuur module geeft $n = -1$ terug wanneer het besluit om de groei te stoppen. De parameter l geeft de levensduur aan van apex A, een apex stelt een cel voor waarin metaalionen kunnen bewegen over het oppervlak. t geeft het type patina aan. De levensduur van een apex vermindert geleidelijk aan tot 0, wat wilt zeggen dat de apex een natuurlijke dood sterft. Het communicatiesymbool $Q(M, n)$ beschrijft zowel een verzoek aan de natuur module als het ontvangen resultaat van de query. Het symbool $R(p, t)$ geeft aan dat er op positie p een patina aanwezig is van type t .

De gesimuleerde ontwikkeling begint met een initieel woord V_1, V_2, \dots, V_k dat willekeurig gekozen wordt met een initiële levensduur $L1$, dit startwoord is het axioma in het L-systeem. De afleiding begint wanneer alle queries zijn afgehandeld. Eerst controleert de patina module of de apex al dan niet leeft volgens de contextgevoelige productie p_1 . De module stopt met groeien en vervangt het paar, bestaande uit de apex $A(p, l, t)$ en het linker contextsymbool $Q(M, n)$, door een patinaknoop $R(p, t)$ als de levensduur gelijk is aan 0. De natuur module beslist de volgende toestand, gebaseerd op de toestanden van de huidige vertex en zijn burens, indien geen gepaste vertex gevonden is om dezelfde patina te doen groeien. Een patina wordt geïntroduceerd in de tweede laag volgens productie p_2 als de vertex niet bedekt is met een patina uit de tweede laag. Zoniet genereert productie p_3 een patinaknoop en beëindigt het het groeiproces als alle burens bedekt zijn met patina's. De primaire groei van patina's wordt beschreven door productieregel p_4 . Een apex A met een zoekopdracht Q creëert een patinaknoop R en een aantal nieuwe apices A 's voorafgegaan door de queryknopen Q 's in elke iteratie nadat het resultaat van de query ontvangen is van de natuur module. De nieuwe apex met een hogere tendens heeft een langere levensduur dan diegene met een lagere tendens.

De patina in een punt P wordt afgeleid uit de patina's van de vertices van de polygoon waarin P zich bevindt. Als de vertices verschillende patina's bevatten, dan beslist een Perlin's noise

functie [Per85] de patina in P . In het ander geval toont P dezelfde patina als de polygoon waarin het zich bevindt.

Besluit

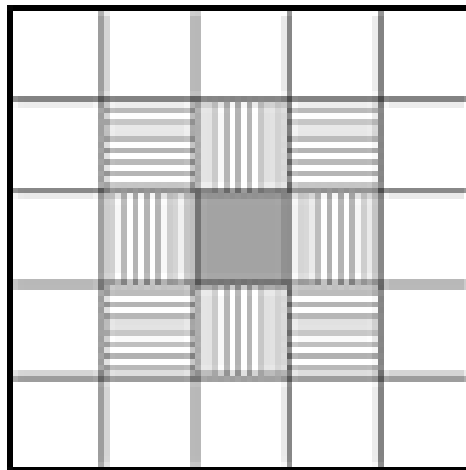
Deze techniek berust op een erosiefunctie die als invoer dient voor het open L-systeem die de groei van de patina imiteert. De artiest kan enkel de erosiefunctie beïnvloeden door de natuurlijke omgeving waarin het object zich bevindt, te specificeren.

Dankzij het feit dat het groeiproces doorgaat op het oppervlak zelf, zullen er geen problemen optreden aan de textuurranden. Maar doordat het groeiproces van vertex naar vertex verloopt volgens hun gemeenschappelijke edge, wordt er enkel aan de vertices patina toegekend. De tusseliggende punten krijgen een geïnterpoleerde waarde van de vertices van de driehoek waarin ze liggen. Dit ondermijnt toch het realisme van het verweringsproces dat ieder oppervlaktepunt kan aantasten.

3.2 Simulatie van destructieve corrosie

3.2.1 Cellulaire automaten

Cellulaire automaten bestaan uit een verzameling cellen, die gepositioneerd zijn in een grid (zie figuur 3.9). Elke cel heeft een eigen toestand en deze toestand is een element van een eindige verzameling mogelijke toestanden die een cel kan aannemen. Op de ticks van een gezamenlijke klok wordt de toestand van al de cellen opnieuw berekend. Hoe de toestand van een cel verandert bij een tick van de klok wordt beschreven door bepaalde regels. Deze regels zijn functies die afhankelijk van de toestand van deze cel en de toestanden van de burens van deze cel, een nieuwe toestand voor de huidige cel berekenen. Welke de burens van een cel zijn moet op voorhand worden gedefinieerd. Zo is het voor sommige toepassingen nuttig om enkel de cellen die verticaal zijn gearceerd in figuur 3.9 als burens van de grijze cel te beschouwen terwijl in andere toepassingen zowel de horizontaal als de verticaal gearceerde cellen bij de burens moeten gerekend worden.



Figuur 3.9: Burens van een cel.

Merillou [MDG01] gebruikt een cellulaire automaat in de vorm van een textuurmap waarin de cellen overeenkomen met de texels van de textuur. Hun simulatie van destructieve corrosie gebeurt in deze textuurmap, die ze de corrosiemap noemen.

Startpunten van corrosie

Ten eerste wordt het aantal startpunten n_0 gekozen. Vervolgens moet er beslist worden waar de startpunten het best geplaatst worden op het object. Dit komt overeen met de kans dat een polygoon een startpunt bevat. Elke polygoon P_i heeft een kans p_i die gelijk is aan:

$$p_i = \text{Area}(P_i) / \sum_i \text{Area}(P_i) \quad (3.12)$$

Dit betekent dat alle polygoonen een kans hebben verhoudingsgewijs met hun oppervlakte. Dit is in tegenstelling tot wat we in hoofdstuk 2 gezien hebben, namelijk dat corrosie van

interne en externe condities afhangt. De kansen p_i worden aangepast volgens deze condities, als volgt:

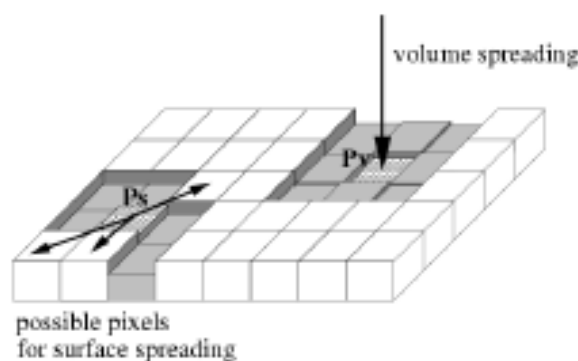
$$p_i = p_{i,initial} * k_{str,i} * k_{gr,i} * k_{ext} \quad (3.13)$$

Waar de coëfficiënt $k_{str,i} > 1$ duidt op microscopische imperfecties (bv. een krets) op het oppervlak. Deze coëfficiënt wordt gebruikt om de kans voor elke defecte polygoon te verhogen. Coëfficiënt $k_{gr,i} < 1$ wordt gebruikt om de kans te verlagen voor elke polygoon die bedekt is met een vette laag. k_{ext} duidt op externe invloeden zoals de aanwezigheid van een andere metalen legering, hierbij wordt de kans op een minst nobele polygoon verhoogd, terwijl de kans op de meer nobele verlaagd wordt (contactcorrosie).

Nu kunnen de startpunten willekeurig gekozen worden. Eerst wordt er een polygoon willekeurig gekozen volgens zijn kans p_i . Vervolgens wordt er willekeurig een punt binnen deze polygoon gekozen. De coëfficiënten worden door de gebruiker gekozen en laten hem toe om de startpunten van corrosie te beïnvloeden.

Spreiding van corrosie

Het corrosieproces wordt gesimuleerd op een corrosiemap. Er wordt verondersteld dat de corrosiemap een grootte van $1m^2$ en een voorgedefinieerde dikte (bv. in mm) heeft. Deze dikte komt overeen met de dikte van het materiaal van het object. Daarna wordt de corrosiemap gemapt op het object. Elke pixel van de corrosiemap bevat de volgende informatie: een kleur, een toestand en een hoogte. Die hoogte komt bij aanvang overeen met de dikte van de map. De startpunten die we net berekend hebben worden via een inverse mapping getransformeerd naar de corrosiemap, met deze startpunten kan het spreidingsproces gebeuren op de map. De n_0 startpunten worden gemarkeerd als gecorrodeerd.



Figuur 3.10: Het spreidingsproces (Bron[MDG01]): rond P_s wordt er een nieuwe pixel gecorrodeerd en duidt het spreidingsproces in de richting van de pijlen aan, terwijl P_v al een gecorrodeerde pixel is waar het spreidingsproces doorgaat in het volume.

De eerste stap van het spreidingsproces bestaat eruit om een lijst l_{cp} van al gecorrodeerde pixels te bouwen. In deze lijst kiezen we willekeurig een pixel P_s waarvan er minstens 1 van

de acht buren niet gecorrodeerd zijn en we kiezen er ook eentje, P_v , willekeurig uit waarvan alle buren aangetast zijn (zie figuur 3.10).

Het spreidingsprincipe wordt toegepast op P_s door willekeurig 1 van zijn niet aangetaste buren te kiezen. Zijn toestand wordt aangepast als gecorrodeerd, zijn kleur wordt gekozen uit een kleurentabel en zijn hoogte wordt verminderd met δ . Wat P_v betreft als die bestaat, gaat het proces onmiddellijk zijn hoogte verminderen met δ . Merk op dat voor elke pixel, δ afhankelijk is van de lokale hoeveelheid verloren hoogte. Hieruit volgt dat de δ voor P_s niet noodzakelijk hetzelfde is als de δ voor P_v . Als de hoogte van een pixel kleiner is dan 0, dan verschijnt er een gat in het oppervlak en telt de betreffende pixel niet meer mee. De nieuwe aangetaste pixels worden tenslotte toegevoegd aan de lijst l_{cp} .

Het berekenen van δ

Indien een volledig oppervlak uniform aangetast werd door corrosie, dan zou het gemakkelijk zijn om de verloren dikte tijdens iedere spreidingsstap te berekenen door gebruik te maken van functie 2.1 in hoofdstuk 2. Men zou dan dezelfde δ voor iedere pixel kunnen gebruiken. Maar corrosie beïnvloedt niet alle pixels van de map op hetzelfde moment door het tijdsafhankelijke spreidingstechniek met willekeurige startpunten. Dus moet er voor iedere pixel een specifieke δ berekend worden. Er wordt dan ook voor elke pixel een lokale corrosietijd t_{lokaal} berekend. Uitgaande van de verloren dikte van een pixel, dit is *materiaaldikte – pixelhoogte*, kunnen we het equivalente verloren gewicht w_{eq} als volgt berekenen:

$$w_{eq} = \text{verloren pixeldikte} * S_p * \rho_{ijzer} \quad (3.14)$$

Waar $S_p = (100\text{cm}/\text{Mapresolutie})^2$ de pixeloppervlakte is en $\rho_{ijzer} = 7.8\text{g}/\text{cm}^3$ de dichtheid van ijzer is. Als we functie 2.1 oplossen in t dan krijgen we:

$$t_{lokaal} = e^{(1/n)*\ln(w_{eq}/k)} \quad (3.15)$$

Het gemiddelde verloren gewicht op t_{lokaal} is:

$$\Delta w = w(t_{lokaal+1}) - w(t_{lokaal}) \quad (3.16)$$

Waar w de corrosiefunctie 2.1 is. De gemiddelde verloren hoogte voor deze pixel wordt gegeven door:

$$h_l = \Delta w / (S_p * \rho_{ijzer}) \quad (3.17)$$

In het specifiek geval van putcorrosie wordt δ verhoogd met een putfactor die de ratio aangeeft tussen de maximale diepte van een put en de gemiddelde verloren dikte.

Besluit

Met deze methode kan de gebruiker het aantal startpunten van corrosie aanduiden. Ook kan de artiest bepaalde eigenschappen meegeven aan het object die de corrosie op een positieve of negatieve manier beïnvloeden.

Het cellulaire automaat wordt gebruikt om het gedrag van corrosie te imiteren. Het spreidingsproces gebeurt zowel over de textuur, als in de diepte. Hierdoor wordt ook de erosie tegelijkertijd gesimuleerd. Maar dankzij het feit dat het groeiproces zich afspeelt in de textuurruimte, krijgen we problemen aan de textuurranden bij een mapping naar complexe objecten.

3.3 Conclusie

Wij hebben verschillende methodes gezien waarmee corrosie gesimuleerd kan worden. We kunnen opmerken dat al deze technieken streven naar eenzelfde doel, namelijk het automatisch genereren van een textuur. De reden hiervoor is dat manuele textuurcreatie veel inspanning van de artiest vergt. Het merendeel van de simulaties is gewijd aan de ontwikkeling van patina's.

We kunnen systemen en technieken beoordelen volgens hun fysische correctheid. Fysisch gebaseerde modellen volgen de wetten van de natuurkunde in hun simulatie. De tegenpool van deze modellen zijn empirische modellen, die hun simulatie baseren op observaties in de natuur. Zo zijn L-systemen en cellulaire automaten ontwikkeld om het gedrag van bepaalde fenomenen te imiteren. L-systemen worden in het algemeen gebruikt om het gedrag van planten te imiteren en cellulaire automaten worden bijvoorbeeld gebruikt voor vloeistofsimulaties.

Fysische modellen brengen zowel voor- als nadelen met zich mee. Het grootste voordeel is de realistische factor van het verweringsproces, maar het grootste nadeel is dat de simulatie veel complexer is. Ook laat het fysisch model weinig controle over aan de gebruiker.

Van al de besproken technieken is de techniek met de verweringsbronnen verre van fysisch correct. Er wordt totaal geen rekening gehouden met natuurlijke factoren die de vertering aansporen op bepaalde regio's van het object. De gebruiker is hier in volledige controle. Daartegenover sluit de techniek met de operatoren het meest aan om een fysisch model genoemd te worden. De voorstelling en de groei zijn fysisch correct weergegeven, maar de simulatie houdt geen rekening met de omgeving van het object.

De techniek gebaseerd op L-systemen en de techniek die werkt met verweringsbronnen berekenen een onderliggende verdeling van corrosie via een erosiefunctie. Door de invloed van een verweringsbron te vergroten, simuleert men de groei, terwijl de andere techniek die onderliggende verdeling gebruikt als input voor het L-systeem. Bij deze laatste gebruikt men geen door de gebruiker geplaatste verweringsbronnen, maar natuurlijke factoren zoals hoeveel water een oppervlak kan ophouden, wat als voordeel heeft dat de simulatie realistisch getrouwer is.

De techniek met L-systemen beschouwt een patina ook als een lagenstructuur, maar verschilt in het groeiproces. In de techniek met de lagenstructuur gebruikt men 2D fysische groeimodellen, terwijl L-systemen op observatie gebaseerd zijn. De meerdere lagenstructuur die opgebouwd wordt via operatoren, maakt gebruik van textuurmapping om de groei te simuleren. Hierdoor krijgt men moeilijkheden aan de textuurranden bij het mapping. De simulatie via L-systemen speelt zich af in de objectruimte, waardoor er geen distorties optreden bij het mapping.

Van al deze technieken laat het cellulaire automaat zich toch opmerken door zijn simpliciteit in vergelijking met de anderen. Het vergt geen zware precalculatie zoals het geval is bij de technieken die werken met een tendensverdeling. Met het spreidingsproces kan men heel gemakkelijk smetten creëren en niet alleen voor corrosie maar ook voor andere verweringszaken zoals schimmel en mos. Het nadeel is dat het spreiden niet fysisch gebaseerd is, maar dat wilt niet zeggen dat men hiermee geen realistische resultaten kan verkrijgen. Het grootste

probleem aan het cellulaire automaat is dat we een textuurmap nodig hebben. Dit houdt in dat we zeker moeilijkheden aan de textuurranden gaan verkrijgen bij textuurmapping.

Ons doel is om de techniek met het cellulaire automaat te transformeren van textuuruimte naar objectruimte. Hierdoor krijgen we geen textuurdistorties. Het reguliere karakter verbreken we door gebruik te maken van losse partikels. De losse partikels komen overeen met de cellen van het cellulair algoritme en worden rechtstreeks op het oppervlak van een object geplaatst. Door met losse partikels te werken, is er geen reguliere grid en connectiviteit vereist. Onze grootste uitdaging is om enige regelmaat in te bouwen in het irreguliere karakter van losse partikels.

Hoofdstuk 4

Simuleren van corrosie op complexe objecten

In dit hoofdstuk zal onze simulatie van destructieve corrosie uit de doeken gedaan worden. Onze techniek is gebaseerd op het cellulaire automaat algoritme [MDG01] in sectie 3.2.1. Onze doelstelling is om het spreidingsproces dat werkt met een cellulaire automaat te verplaatsen van textuurruimte naar objectruimte.

We beschrijven eerst de voorstelling van corrosie, daarna lichten we de automatische selectie van corrosiestartpunten toe. Vervolgens beschrijven we het simulatieproces van corrosie dat gebaseerd is op het cellulaire automaat algoritme van Merillou [MDG01]. Bij dit proces brengen wij ook de corrosiefunctie uit hoofdstuk 2 in rekening om de hoeveelheid verloren materie over de tijd uit te drukken. Ook de visualisatie van ons systeem wordt hier beschreven, we gebruiken hiervoor een simpele raytracer met een interpolatietechniek.

4.1 Representatie

De corrosie zelf stellen we voor als een verzameling van losse partikels. De partikels worden rechtstreeks op de mesh geplaatst waardoor iedere partikel geometrisch gedefinieerd wordt door zijn centrum, straal en een index van de driehoek waarop het zich bevindt. Elke partikel bevat ook een RGBA kleur, een toestand en een diepte. De toestand van een partikel kan drie waardes aannemen, namelijk midden, rand en gat. De diepte wordt aangepast door het spreidingsalgoritme en geeft aan hoe fel het partikel is gepenetreerd onder invloed van de corrosiereactie. De kleur is afhankelijk van de diepte en wordt gekozen uit een kleurentabel.

4.2 Selectie van startpunten

Het aantal startpunten n_0 wordt gekozen door de gebruiker, maar de locatie van deze startpunten wordt automatisch berekend. De kans p_i dat een polygoon P_i een startpunt zal bezitten wordt gegeven door [MDG01]:

$$p_i = \text{Area}(P_i) / \sum_i \text{Area}(P_i) \quad (4.1)$$

Dit betekent dat alle polygonen een kans hebben om gekozen te worden volgens hun oppervlakte. In hoofdstuk 2 hebben we gezien dat deze kans volgens interne en externe condities kan verhoogd of verlaagd worden. Dit wordt gerealiseerd in [MDG01] door de kans p_i te vermenigvuldigen met enkele gebruikersspecifieke coëfficiënten.

Wij gaan ons beperken tot uniforme corrosie, of met andere woorden, al de coëfficiënten, uit sectie 3.2.1, hebben de waarde 1. Vermits deze schaalfactoren door de gebruiker zelf gespecificeerd worden, kan onze kansverdeling gemakkelijk gelijkaardig uitgebreid worden zoals in sectie 3.2.1.

Eens we een willekeurige polygoon P_i hebben, kunnen we willekeurig een punt binnen P_i kiezen.

Alhoewel het niet fysisch correct lijkt dat de startpunten op deze wijze gekozen worden, is het dat toch volgens Mullins [MST97]: de lokaties van corrosie reacties zijn willekeurige functies in functie van tijd en positie zodat hun effect op het totale oppervlak kan beschouwd worden als een klassieke “white noise” bron.

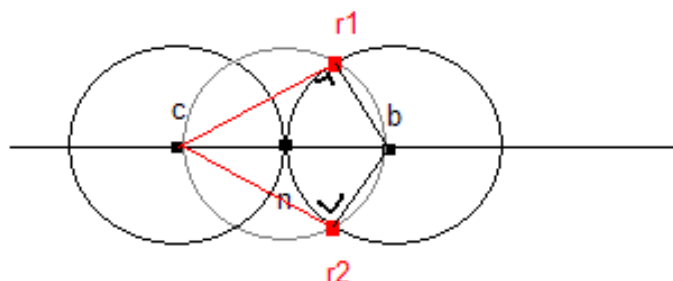
4.3 Algoritme

Vermits we gaan werken met losse partikels, is er geen reguliere grid en geen connectiviteit meer. Hierdoor moeten we toch een zekere regelmaat inbouwen om de burens te bepalen van een partikel en om nieuwe burens te plaatsen die mooi aansluiten op de andere partikels. De partikels worden ruimtelijk opgedeeld volgens een octree-structuur [Sut], waardoor we de burens van een partikel binnen een straal efficiënt kunnen opzoeken.

Bij het plaatsen van een nieuwe partikel moeten we ervoor zorgen dat die mooi aansluit met de andere partikels. We kunnen dit ofwel iteratief (met stappen doorschuiven totdat hij iets raakt), ofwel exact doen. Om het exact te doen, werken we met een lijst van burens, die gesorteerd is volgens hoek, en op basis daarvan kan de positie van het nieuwe partikel bepaald worden.

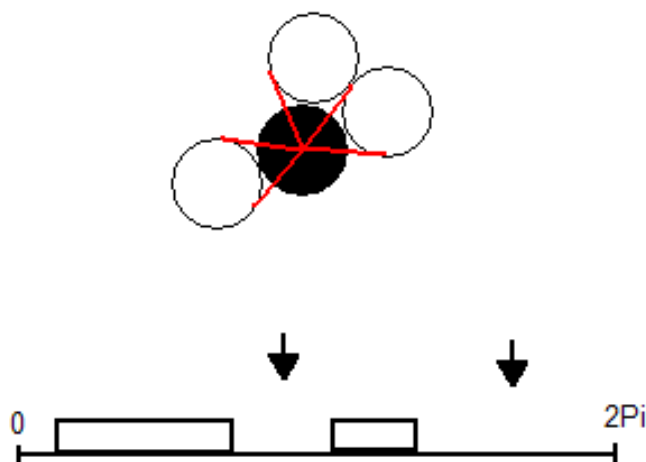
Rond iedere partikel p is er een domein $[0, 2\Pi]$ waarin nieuwe burens geplaatst kunnen worden. We berekenen eerst de lijst met burens en vervolgens berekenen we voor iedere buur b_i de hoek h_i die hij beslaat in het $[0, 2\Pi]$ domein. De hoek die een buur b_i beslaat wordt bepaald door een beginhoek en een eindhoek die berekend worden door de twee raaklijnen aan buur b_i vanuit het centrale partikel p te construeren [KHL].

Om de raaklijnen te bepalen vanuit het middelpunt c van partikel p aan het buurpartikel b_i met centrum b , zoeken we een punt r op de cirkel b_i zodat de rechte cr loodrecht staat op de rechte br . We zoeken eerst het midden n van het lijnstuk cb , en beschouwen een cirkel door n met als straal de afstand tussen cn . Bijgevolg zijn de snijpunten van deze cirkel met de buurpartikel de twee raakpunten r_1 en r_2 (zie figuur 4.1). Nu kunnen we de begin- en eindhoek, cr_1 en cr_2 , van de hoek h_i berekenen die de buur b_i beslaat.



Figuur 4.1: Constructie van raaklijnen. Vanuit het partikel met centrum c worden de raaklijnen geconstrueerd met buurpartikel b_i . De grijze cirkel is de cirkel die ons de raakpunten r_1 en r_2 geeft.

Eens wij alle hoeken h_i berekend en gesorteerd hebben, zoeken we met een zoekopdracht waar er nog voldoende plaats is om een nieuwe partikel te plaatsen (vgl. geheugenbeheer) (Zie figuur 4.2). Het domein rond een partikel wordt geïmplementeerd als een circulaire lijst en de zoekopdracht vergelijkt de eindhoek van een hoek h_i met de beginhoek van een hoek h_{i+1} .

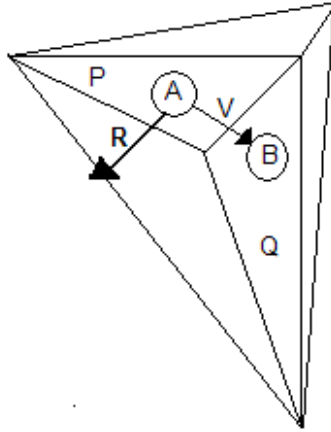


Figuur 4.2: Het algoritme om een nieuwe partikel te plaatsen rond het zwarte centrale partikel. De rode raaklijnen bepalen de hoeken h_i van de burens b_i . In het domein $[0, 2\Pi]$ komen de blokjes overeen met de h_i 's en de pijlen duiden op mogelijke plaatsen waar een nieuw partikel kan toegevoegd worden.

Vermits de partikels op een 2D oppervlak liggen, gebeurt de berekening van de positie van het nieuwe partikel in het draagvlak van het centrale partikel p of met andere woorden in een 2D lokaal assenstelsel. We nemen de centrale partikel p als oorsprong en de normaal van het vlak waarop p ligt, als Z -as. Vervolgens nemen we een loodrechte vector op deze Z -as als X -as en als Y -as nemen we het kruisproduct van de X -as en Z -as. Als een buur in het vlak van de centrale partikel ligt of met andere woorden als het op dezelfde polygoon ligt, dan wordt hij getransformeerd naar het lokaal stelsel. Als een buur niet op dezelfde polygoon ligt als de centrale partikel en dus niet in hetzelfde vlak als de centrale partikel ligt, dan wordt

hij eerst gerooteerd naar het vlak van p en daarna getransformeerd naar het lokaal stelsel.

Het roteren van de buren b_i naar het draagvlak van de centrale partikel p , gebeurt door hun te roteren rond partikel p (zie figuur 4.3). Om te kunnen roteren, hebben we een rotatiehoek en een rotatie-as nodig. Allereerst stellen we een lokaal assenstelsel op dat gedefinieerd is door de normaal van het vlak waarop p ligt als Z-as, het kruisproduct van de Z-as en de verschilvector $b_i - p$ als Y-as en een vector loodrecht daarop als X-as. Vervolgens wordt de verschilvector $b_i - p$ geprojecteerd in de lokale basis en met de resulterende vector verkrijgen we de rotatiehoek. De rotatie-as is de Y-as van het lokaal systeem.



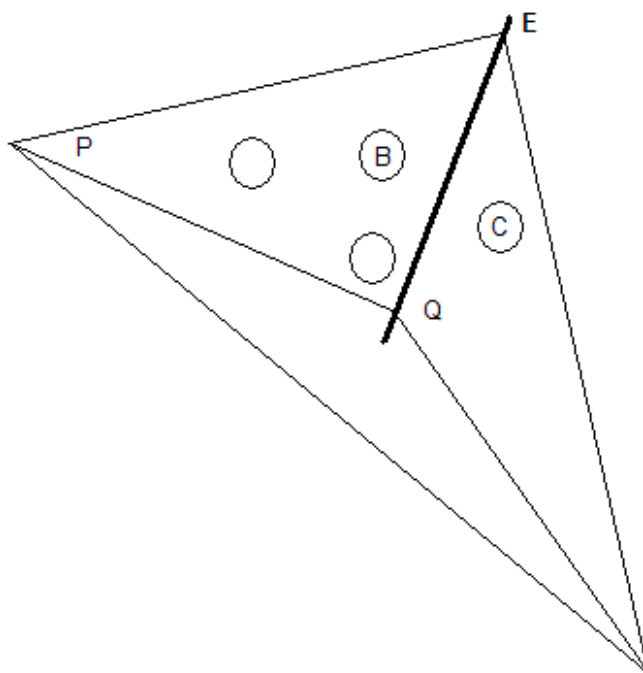
Figuur 4.3: Partikel B is een buur van partikel A, maar ligt niet in dezelfde polygoon als A. B wordt gerooteerd rond rotatie-as R die loodrecht staat op de verschilvector $V = B - A$. De rotatiehoek verkrijgen we via V na projectie in de lokale basis van A.

Als er een nieuw partikel wordt toegevoegd, ligt deze niet noodzakelijk in dezelfde polygoon als de centrale partikel (zie figuur 4.4). Dit lossen we op met een algoritme voorgesteld door Turk [Tur91]. Als het nieuwe partikel p_{nieuw} niet in de polygoon A van de centrale partikel p ligt, dan is de kans groot dat p_{nieuw} helemaal niet op het object ligt. We zoeken dan de edge van A waar p_{nieuw} het dichtst bij ligt volgens [Lan] en zoeken vervolgens de polygoon B die deze edge deelt met A. p_{nieuw} wordt nu gerooteerd rond deze gezamenlijke edge zodat het in het vlak van B ligt. Als p_{nieuw} niet in B ligt, wordt de iteratie verdergezet tot p_{nieuw} in een polygoon ligt. In sommige gevallen kan het zijn dat een mesh polygoonen bevat die niet aan iedere edge een buurpolygoon hebben. In dat geval wordt het nieuwe partikel p_{nieuw} niet beschouwd.

Het algoritme van Merillou verspreidt zich over de textuur als een gecorrodeerde pixel aan de rand van een corrosieplek ligt en gaat in de diepte bij een pixel die midden in een corrosiesmet gelegen is. Om een partikel p te bestempelen als rand of midden, beschouwen we een dichtheidsfunctie die gegeven is als volgt:

$$d = \frac{n}{A} \tag{4.2}$$

Waar n overeenkomt met het aantal buren van het partikel in kwestie p en A overeenkomt met de oppervlakte waarin de buren zijn gevonden. Aan de hand van een treshold, wordt er



Figuur 4.4: Partikel C is een nieuwe partikel die niet in polygoon P ligt. C ligt bijgevolg ook niet op het oppervlak. C ligt het dichtst bij edge E waarrond hij geroteerd wordt over een hoek die gegeven wordt door de normalen van P en buurpolygoon Q die E deelt met P . Het resultaat is dat C komt te liggen op polygoon Q .

bepaald of het partikel aan de rand of in het midden ligt.

Nu kunnen we overeenkomstig met het algoritme van Merillou [MDG01] een lijst l_r opbouwen van partikels die aan de rand liggen en een lijst l_m met partikels die in het midden liggen. Uit de lijst l_r kiezen we willekeurig een partikel p_r en plaatsen er een nieuwe partikel rond p_r die nauw aansluit bij de rest. Zijn diepte wordt vermeerderd met δ en zijn overeenkomstige kleur wordt opgezocht in een kleurentabel. Ook uit de lijst l_m wordt er willekeurig een partikel p_m gekozen en de diepte van deze partikel wordt rechtstreeks met een δ vermeerderd. Vervolgens wordt ook zijn kleur aangepast volgens zijn nieuwe diepte. Als de diepte van een partikel groter is dan de materiaaldikte, dan verschijnt er een gat in het oppervlak, het betreffend partikel telt dan niet meer mee.

Berekening van δ

De hoeveelheid verloren materie wordt op dezelfde manier berekend als in sectie 3.2.1, maar met de uitzondering dat nu het oppervlak van de partikels beschouwd wordt. Vermits corrosie niet het hele oppervlak aantast op hetzelfde moment, kan hier ook niet dezelfde δ voor iedere partikel gebruikt worden door de tijdsafhankelijke spreidingstechniek met willekeurige startpunten. Dus moet er voor iedere partikel een specifieke δ berekend worden. Er wordt dan ook voor elke partikel een lokale corrosietijd t_{lokaal} berekend. Uitgaande van de verloren dikte van een partikel d_p , kunnen we het equivalente verloren gewicht w_{eq} als volgt berekenen:

$$w_{eq} = d_p * S_p * \rho_{ijzer} \quad (4.3)$$

Waar S_p het oppervlak van het partikel is en $\rho_{ijzer} = 7.8g/cm^3$ de dichtheid van ijzer is. Als we de functie 2.1 oplossen in t dan krijgen we:

$$t_{lokaal} = e^{(1/n)*\ln(w_{eq}/k)} \quad (4.4)$$

Het gemiddelde verloren gewicht op t_{lokaal} is:

$$\Delta w = w(t_{lokaal+1}) - w(t_{lokaal}) \quad (4.5)$$

Waar w de corrosiefunctie 2.1 is. De gemiddelde penetratie wordt gegeven door:

$$h_l = \Delta w / (S_p * \rho_{ijzer}) \quad (4.6)$$

In het geval van putcorrosie kan de gebruiker δ verhogen met een putfactor zoals in sectie 3.2.1.

De kleurentabel

Zoals we eerder vermeld hebben is de kleur van een partikel afhankelijk van zijn diepte. De kleur wordt opgezocht in een kleurentabel die verkregen wordt door samples te nemen uit echte foto's van corrosie.

De kleurentabel wordt gecreëerd als een 1D textuur, namelijk 1 pixel hoog en c pixels breed, waarbij c overeenkomt met het aantal kleuren. De tabel gaat van initiële diepte links naar maximale diepte rechts (het einde van de tabel) en de sleutel van de tabel wordt gegeven door:

$$sleutel = p_d / max_d \quad (4.7)$$

Waar p_d de diepte van het partikel in kwestie is en max_d de maximale diepte, voorgesteld door de textuur. De resulterende $0 \dots 1$ waardes kunnen gebruikt worden als textuurcoördinaat.

Samengevat

Samengevat kunnen we het algoritme in pseudocode als volgt beschrijven:

Lokaliseer het aantal startpunten

Herhaal k keer

Bouw de lijsten l_r en l_m
Kies willekeurig een partikel uit l_r
 Voeg een nieuw partikel p_{nieuw} toe
 Pas diepte en kleur aan
Kies willekeurig een partikel uit l_m
 Pas diepte en kleur aan

4.4 Weergave

Om onze simpele raytracer te versnellen gebruiken we een uniforme grid als acceleratiestructuur, samen met het algoritme voor de doortocht van stralen [AW87].

Wij gebruiken een methode voor het interpoleren tussen ongestructureerde punten [She68], zoals wij hier hebben in de vorm van partikels. Zodra een straal het oppervlak raakt, zoeken we alle partikels die gelegen zijn in een afstand d van het intersectiepunt p . De zoekradius d wordt gegeven door:

$$d = r * \gamma \quad (4.8)$$

Waar r de straal van een partikel is en γ een schaalfactor. Als we alle partikels binnen een afstand d van intersectie p gevonden hebben, bekomen we de resulterende kleur k door als volgt te interpoleren:

$$k = \frac{\sum_i k_i * w_i}{\sum_i w_i} \quad (4.9)$$

Waarbij k_i de kleur is van de i^{de} buur en w_i het gewicht is dat als volgt berekend wordt:

$$w_i = \frac{1}{\|x_i - p\|^\alpha} \quad (4.10)$$

Waar x_i het centrum is van de i^{de} buurpartikel en α is een schaalfactor.

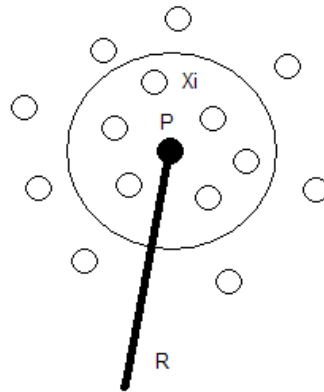
Om cirkelachtige artefacten te reduceren in figuur 4.6, kunnen we ook een Gaussiaanse weging gebruiken:

$$w_i = \exp(-(x_i \cdot p)/s^2) \quad (4.11)$$

Waarbij s een verzachtingsfactor is die evenredig is met de zoekradius d , $x_i \cdot p$ is het inproduct van het centrum x_i van de i^{de} buur en het intersectiepunt p .

Als er geen partikels gevonden worden in een afstand d , dan wordt de eigenschap van het basis-materiaal gebruikt, anders berekenen we de kleur volgens formule 4.9. De resulterende kleur wordt opgeslagen in een textuuratlas gegenereerd door het programma *Graphite* [Gra03].

Voor het creëren van gaten, gaan we pixelgewijs de som maken van de dieptes van de partikels in die pixel. Indien die som groter is dan de materiaaldikte, dan wordt die pixel als gat gemarkeerd, of met andere woorden alpha wordt op nul gezet.



Figuur 4.5: Interpolatiemethode. Al de witte partikels met centrum x_i binnen de zoekcirkel met centrum het intersectiepunt p (zwart), worden betrokken bij de interpolatie, gegeven door de formule 4.9, om de kleur in p te bepalen. De anderen worden niet beschouwd.



Figuur 4.6: Interpolatiemethode. (a) Circulaire artefacten door de weegfactor in 4.10 te gebruiken. (b) Reduceren van de artefacten met een Gaussiaanse weging volgens formule 4.11.

4.5 Besluit

Wij hebben hier onze methode voorgesteld om corrosie te simuleren op complexe objecten. Het cellulair automaat algoritme van Merillou [MDG01] is uitgebreid van 2D naar 3D door de cellen van het cellulaire automaat te vervangen door losse partikels. In het irreguliere karakter van deze benadering hebben wij een zekere regelmaat ingebouwd.

Wij zoeken de buren op via een octree en de nieuwe partikels worden zodanig toegevoegd dat ze nauw aansluiten aan de bestaande partikels. Dit gebeurt door de partikels in kwestie te transformeren naar een 2D lokaal assenstelsel. Het terugroteren van een buur in het vlak van een centraal partikel, zouden we preciezer kunnen doen door een geodesisch pad te zoeken tussen het centraal partikel en de buur. Vervolgens, volgens dit pad, de buur telkens te roteren naar de opeenvolgende polygonen volgens Turk [Tur91] totdat we in het vlak van de centrale partikel belanden. Maar omwille van de complexiteit en de daaraan verbonden tragere uitvoeringssnelheid hebben we dit achterwege gelaten.

Om het geheel te renderen maken we gebruik van een interpolatietechniek voor ongestructureerde punten (partikels). We elimineren de circulaire artefacten aan de hand van een Gaussiaanse weging.

Hoofdstuk 5

Resultaten

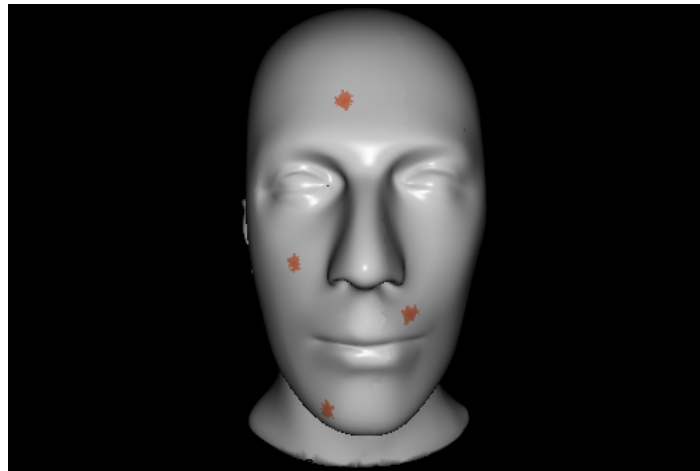
In dit hoofdstuk worden de uiteindelijke resultaten getoond aan de hand van enkele screenshots. De beelden werden gemaakt op een laptop voorzien van een 1.5GHz centrino processor, 512 MB geheugen en met een NVIDIA GeForce FX Go5650 grafische kaart.

De objecten werden allemaal voorzien van een materiaaldikte van 12mm en de grootte van een partikel wordt gegeven door $Oppervlakte_{object}/10000cm^2$. De uniforme grid die gebruikt werd door onze eenvoudige raytracer was telkens 50x50x50 groot.

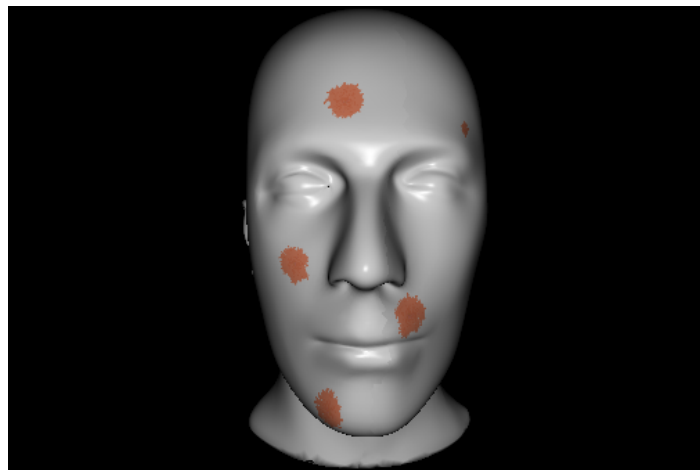
In tabel 5.1 staan de twee hoofdbewerkingen, namelijk de simulatie en visualisatie van corrosie, samen met de tijd die nodig was om die bewerking uit te voeren tijdens een toename van het aantal partikels in de scene bestaande uit 23402 polygonen, die te zien is in figuur 5.1.

<i>corrosietijd in jaren</i>	<i>aantal partikels</i>	<i>benodigde tijd voor simulatie in sec</i>	<i>benodigde tijd voor visualisatie in sec</i>
na 1 jaar	673	39.395s	20.710s
na 5 jaren	3167	185.385s	21.329s
na 10 jaren	6142	359.531s	22.602s

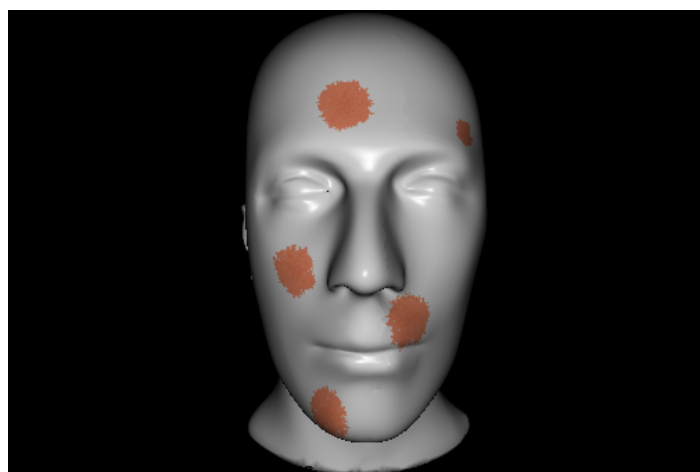
Tabel 5.1: Tijd die nodig was voor de spreiding en weergave van corrosie.



(a) 1 jaar/673 partikels

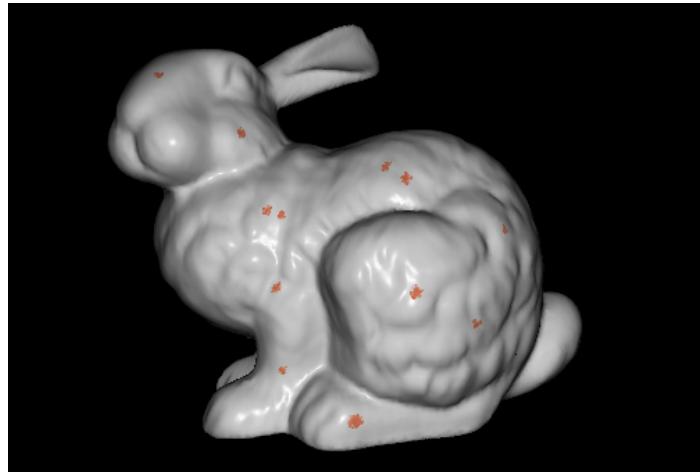


(b) 5 jaar/3167 partikels

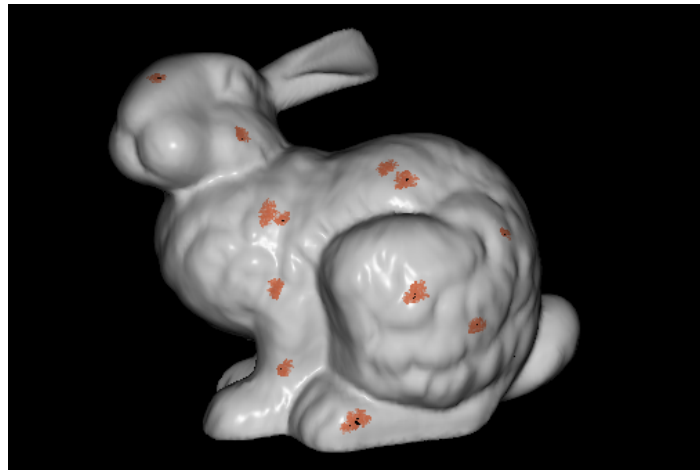


(c) 10 jaar/6142 partikels

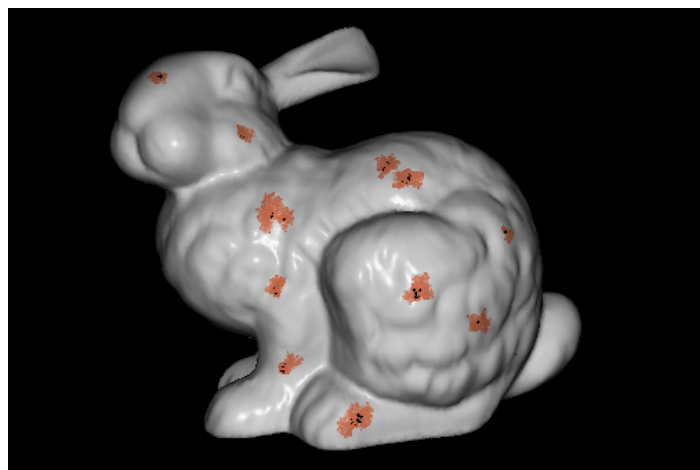
Figuur 5.1: Spreiding van corrosie met medium corrosiviteit en met 9 startpunten, na respectievelijk 1, 5 en 10 jaar op een object bestaande uit 23402 polygonen.



(a) 1 jaar/639 partikels

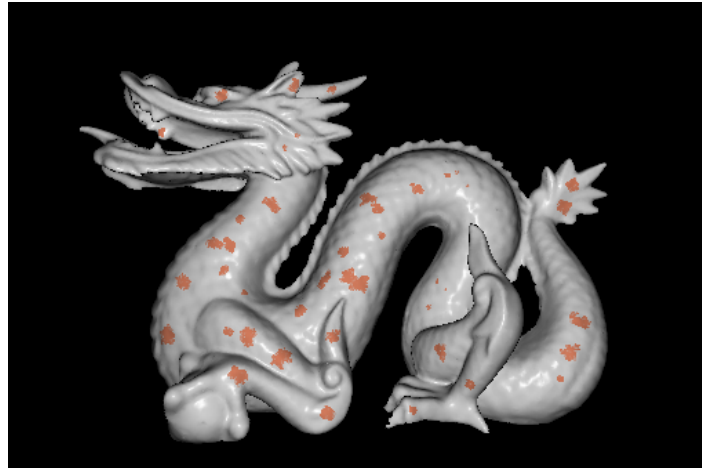


(b) 5 jaar/3027 partikels



(c) 10 jaar/6092 partikels

Figuur 5.2: Spreiding van corrosie met hoge corrosiviteit en met 20 startpunten na respectievelijk 1, 5 en 10 jaar op een object bestaande uit 69451 polygonen. Na vijf jaren verschijnen de eerste gaten in het oppervlak. De simulatietijd komt overeen met tabel 5.1, terwijl de tijd voor het renderen groter is, namelijk rond de 35 seconden.



(a) 18 jaar/12108 partikels



(b) 15 jaar/11424 partikels

Figuur 5.3: Twee objecten aangetast door uniforme corrosie met 30 startpunten in een lage atmosferische omgeving. De draak bestaat uit 202520 polygonen en de simulatie duurde 10 minuten. De boedha bestaat uit 1087716 polygonen en de simulatie gebeurde in 9,5 minuten. Het renderen gebeurde in het geval van de draak in 54 seconden en bij de boedha in 8 minuten.

Hoofdstuk 6

Algemeen Besluit

6.1 Samenvatting

Nieuwe ontwikkelingen zijn meestal gebaseerd op oudere werken die hun nut al uitvoerig bewezen hebben. Onderzoekers trachten de nadelen van deze oudere methodes te overkomen door ze te combineren met andere methodes of door nieuwe ideeën te introduceren. Regelmatig worden hierbij technieken uit andere domeinen van de computer graphics gebruikt, al dan niet in een aangepaste vorm. Bij het uitwerken van een nieuwe methode is het belangrijk als basis de juiste technieken te selecteren en die verder te verbeteren of aan te passen aan de eigen noden.

De literatuurstudie van deze thesis geeft een algemeen overzicht van de meest recente systemen en technieken die reeds succesvol werden toegepast bij het modelleren en simuleren van corrosie. Elk van deze systemen en technieken heeft zowel een aantal voordelen als nadelen. We kunnen systemen en technieken echter niet zomaar gaan beoordelen als zijnde goed of slecht. Soms zijn onderzoekers bereid nauwkeurigheid op te offeren voor snelheid of eenvoud, terwijl we in andere gevallen juist het omgekeerde scenario aantreffen. Dit is natuurlijk erg afhankelijk van het doel van het onderzoek. Wanneer we het hebben over de kwaliteiten van een systeem of een techniek, moeten we bijgevolg niet denken in termen van goed of slecht, maar in termen van geschikt of niet geschikt.

Empirisch gebaseerde systemen zijn bijvoorbeeld zeer geschikt in situaties waar het belangrijk is makkelijk en snel visueel attractieve modellen te bekomen. Wanneer we echter veel belang hechten aan de correctheid van de structuren, zijn deze systemen veel minder geschikt en gaat de voorkeur uit naar de meer complexe fysisch gebaseerde systemen. Tot op heden zijn er geen volledige fysische systemen voor corrosie, want deze type verweringsvorm is nog niet volledig begrepen.

Uit onze implementatie blijkt dat het gebruik van losse partikels geen slechte keuze is voor de simulatie van corrosie. Wij hebben succesvol de techniek van Merillou [MDG01] in het 2D vlak uitgebreid naar arbitraire oppervlakken in 3D. Hierdoor kunnen complexe objecten getextureerd worden zonder distorties. Onze techniek voldoet aan de verwachtingen: de simulatie is snel genoeg en de resultaten zijn goed. Enkel de weergave kan in de toekomst nog sneller en realistischer gemaakt worden.

6.2 Toekomstig werk

Wanneer we een korte blik in de toekomst werpen, lijkt onze techniek ook bruikbaar voor andere verwerkingstypes die eenzelfde onderliggende smet hebben als corrosie. Wij denken hierbij vooral aan mosgroei, metaalverkleuring en beschimmelings van organisch materiaal.

De simulatie van patina is in feite de tegenpool van destructieve corrosie. In plaats van ons spreidingsalgoritme in de diepte te laten werken, kunnen we nu in de hoogte gaan uitbreiden, waardoor het effect van een beschermende laag (patina) gesimuleerd wordt. Afhankelijk van een patinafunctie zouden we de toegenomen materie kunnen berekenen en afhankelijk daarvan een gepaste kleur uit een kleurentabel kiezen.

Om het realisme te verhogen kunnen we in onze visualisatie het BRDF-model van Cook en Torrance [CT82] gebruiken voor de niet gecorrodeerde regio's, terwijl we voor de aangetaste regio's het post BRDF-model van Merillou [MDG00] kunnen gebruiken om de poreusheid tot uiting te laten komen.

Het meeste onderzoek naar verweringsprocessen is vrij recent en boekte de laatste jaren een enorme vooruitgang. Dit onderwerp geniet nog steeds veel aandacht en de technieken zullen in de nabije toekomst zeker uitgebreid worden.

Bibliografie

- [AW87] John Amanatides and Andrew Woo. A fast voxel traversal algorithm for ray tracing. *Eurographics Association*, pages 3–9, 1987.
- [BS95] Albert-Laszls Barabasi and Harry Eugene Stanley. *Fractal Concepts in Surface Growth*. Cambridge University Press, Cambridge, UK, 1995.
- [CS00] Yao-Xun Chang and Zen-Chung Shih. Physically-based patination for underground objects. *Computer Graphics Forum*, 19(3), 2000.
- [CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, 1982.
- [DEJ⁺99] Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans Pedersen. Modeling and rendering of weathered stone. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 225–234. ACM Press/Addison-Wesley Publishing Co., 1999.
- [DGA04] Brett Desbenoit, Eric Galin, and Samir Akkouche. Simulating and modeling lichen growth. *Computer Graphics Forum*, 23(3):341–350, 2004.
- [DH96] Julie Dorsey and Pat Hanrahan. Modeling and rendering of metallic patinas. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 387–396. ACM Press, 1996.
- [DPH96] Julie Dorsey, Hans Pedersen, and Pat Hanrahan. Flow and changes in appearance. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 411–420. ACM Press, 1996.
- [Eng] Gordon England. The corrosion process. <http://www.gordonengland.co.uk/corrosion.htm>.
- [Fon86] MG Fontana. Corrosion engineering. <http://www.corrosion-club.com/forms.htm>, 1986.
- [Gra03] Graphite, 2003. <http://www.loria.fr/levy/Graphite/index.html>.
- [KHL] departement Rega KHLeuven. Raaklijnen aan cirkels. <http://wiskunde.rega.be/meetkunde/cirkel/raaklijn.php>.
- [Lan] Jeff Lander. Crashing into the new year: Collision detection. <http://www.gamasutra.com>.

- [Lin68] A. Lindenmayer. Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology*, 18:280–315, 1968.
- [Mac] Louis MacDowell. Crevice corrosion. <http://corrosion.ksc.nasa.gov/html/crevcor.htm>.
- [MDG00] Stéphane Mrillou, Jean-Michel Dischler, and Djamchid Ghazanfarpour. A brdf postprocess to integrate porosity on rendered surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):306–318, 2000.
- [MDG01] Stéphane Merillou, Jean-Michel Dischler, and Djamchid Ghazanfarpour. Corrosion: simulating and rendering. In *GRIN'01: No description on Graphics interface 2001*, pages 167–174. Canadian Information Processing Society, 2001.
- [Mil94] Gavin Miller. Efficient algorithms for local and global accessibility shading. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 319–326. ACM Press, 1994.
- [MP96] Měch and Przemyslaw Prusinkiewicz. Visual models of plants interacting with their environment. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 397–410. ACM Press, 1996.
- [MST97] WM Mullin, EJ Shumaker, and GJ Tyler. Stochastic kinetics of corrosion and fractal surface evolution. *Journal of Corrosion Science and Engineering*, 1(7), 1997.
- [Pea85] Darwyn R. Peachey. Solid texturing of complex surfaces. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press, 1985.
- [Per85] Ken Perlin. An image synthesizer. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 287–296. ACM Press, 1985.
- [PJM94] Przemyslaw Prusinkiewicz, Mark James, and Měch. Synthetic topiary. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM Press, 1994.
- [PL90] P. Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.
- [She68] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, New York, NY, USA, 1968. ACM Press.
- [Sut] Jaap Suter. Introduction to octrees. <http://www.flipcode.com>.
- [Tur91] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1991. ACM Press.
- [Tur92] Greg Turk. Re-tiling polygonal surfaces. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 55–64. ACM Press, 1992.

- [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*, pages 139–150. Springer-Verlag, 1997.