

## DE ROL VAN KLEUR BIJ HET ZOEKEN NAAR BEELDEN.

*Eén van de mooiste dingen van een computer is dat die niet moe wordt van zoeken. We kunnen al jaren op een eenvoudige manier zoeken naar bestanden die op de harde schijf van de eigen computer, of op het Internet beschikbaar zijn. Dat zoeken gebeurt dan met een zoekopdracht waarin is gespecificeerd waarnaar gezocht moet worden, bijvoorbeeld naar bestanden met een bepaalde naam, van een bepaalde grootte of datum, of waar een bepaald woord in voorkomt. Het wordt lastiger als er naar een bepaald beeld, plaatje of video-fragment gezocht moet worden. Want hoe weet een computer wat er in een beeld voorkomt?*

Het kan natuurlijk zo zijn dat de beelden een tag hebben, een korte omschrijving van wat er in het beeld staat. Die tag is dan door mensen aan het beeld toegevoegd en er kan dan gezocht worden naar beelden die deze tag hebben. Maar omdat een beeld meer zegt dan de spreekwoordelijke duizend woorden kan één enkele tag vrijwel nooit de inhoud van een beeld goed samenvatten, laat staan van een video. Als we zouden zoeken naar beelden met het label paard dan kan het resultaat een rennend zwart paard in de wei opleveren, of het paard van Pipi Langkous in de tuin van Villa Kakelbont, of een wit paard midden tussen een kudde bruine paarden. We zijn bij dit soort zoekopdrachten dus afhankelijk van de informatie die mensen zelf al hebben toegevoegd aan het beeldmateriaal. Helaas is het overgrote deel van het beschikbare beeldmateriaal, zowel op de eigen PC als op Internet, niet voorzien van tags, en dus zijn bestaande zoekmethoden zoals Google en Yahoo ontoereikend.

We kunnen een computer natuurlijk ook vragen om zèlf de inhoud van een beeld te analyseren, en op basis van die analyse aan te geven of er een match bestaat met datgene

waarnaar gezocht wordt. Dat is het werkerterrein van de computer vision. De software die deze taak moet uitvoeren moet dan in staat zijn om zelfstandig objecten te herkennen in het beeld. Net zoals een mens moet deze software eerst voorbeelden geleerd worden waar deze objecten inzitten. In Figuur 1 zijn drie voorbeelden gegeven van paarden. Het gaat in deze drie gevallen onmiskenbaar om paarden, maar in een andere hoedanigheid en context (rennend van voren gezien, wandelend van opzij, stilstaand tussen andere paarden). Wat de paarden verder onderscheidt is hun kleur. Kleur is onlosmakelijk verbonden met objecten en leent zich daarom goed als eigenschap waarop gezocht zou kunnen worden.

### KLEURCONSTANTIE

Helaas is het zo dat de kleur van een object afhankelijk is van een aantal zaken, zoals de intensiteit en spectrale samenstelling van de lichtbron(nen). Een zeer belangrijke eigenschap van het menselijke oog-hersen-systeem is *kleurconstantie*, de mogelijkheid om objectkleuren waar te nemen op een manier die redelijk onafhankelijk is van de kleur van de lichtbron. Gras is en blijft waargenomen als groen, zowel in de ochtend met wat blauwer licht als 's avonds met wat meer roder licht. Met andere woorden, we zijn (tot op zekere hoogte) invariant voor de kleur van de lichtbron. Een computerprogramma dat zelfstandig objecten van een bepaalde kleur moet gaan zoeken zal dus ook voorzien moeten worden van dezelfde kleurconstantie eigenschap. Als de software de kleur van de lichtbron zou kennen kan daarvoor gecorrigeerd worden, zodat de kleurbalans in het beeld hersteld kan worden. Het probleem is natuurlijk



Fig. 1: Drie voorbeelden van resultaten met de zoekopdracht "paard" gevonden met Google.



Aangezien het aantal kleuren en vormen oneindig is, zijn ook de combi



**Fig. 2: Het grey-world kleurconstantie algoritme aan het werk. Boven het originele beeld, in het midden het beeld onder geel licht dat gecorrigeerd moet worden door het algoritme, onder het kleur-gecorrigeerde beeld. Zie tekst voor verdere uitleg.**

dat de lichtbron niet zomaar beschikbaar is. De oplossing is dan dat de kleur van de lichtbron wordt geschat uit het beeld, waarna de correctie kan plaatsvinden. Dit gebeurt al volop in bijvoorbeeld fotocamera's. De schatting van de lichtbron uit het beeld kan op veel verschillende manieren gebeuren, waarbij er bepaalde aannamen worden gemaakt. Een van de aannamen is bijvoorbeeld dat de lichtbron een uniforme belichting op de scene geeft. Een bekende methode voor kleurcorrectie (maar zeker niet de enige) is gebaseerd op de zogenaamde *grey-world* aanname, waarin veronderstelt wordt dat de gemiddelde kleur van objecten in een scene grijs of neutraal is. Een beeld dat overwegend groen is zou onder deze aanname tot stand gekomen zijn door belichting met groen licht. Dus, een *grey-world* algoritme voor kleurcorrectie zal een bias in de gemiddelde kleur in een beeld interpreteren als het effect van een lichtbron. Het algoritme schat de sterkte van de kleur van de lichtbron, en past een correctie hiervoor toe. De beelden in Fig.2 illustreren dit proces. Merk op dat de kleurcorrectie niet perfect is (er zit nog een blauwe waas over het beeld), dit komt doordat in dit geval niet wordt voldaan aan de aanname dat het originele beeld (geheel links) gemiddeld neutraal is. De gemiddelde kleur in het originele beeld is namelijk een beetje geel-groen (CIELAB coördinaten  $a^*=-3.5$ ,  $b^*=6.9$ ) waardoor het effect van de gele lichtbron door het algoritme te sterk wordt ingeschat, en waardoor de correctie ook door "wit heen schiet" in de richting van de complementaire kleur.

## SCHADUWEN

Waar een computer vision systeem ook mee te maken krijgt is schaduwen. Mensen hebben geen enkel probleem om in een beeld schaduwen als zodanig te herkennen, maar hoe laat je een computerprogramma schaduwen herkennen? Waar het in feite op neerkomt is dat er een scheiding tussen licht en materie moet worden aangebracht. Overgangen tussen objecten zijn in de regel vrij scherp en gaan gepaard met een overgang in kleur. Een schaduwrand is meestal minder scherp en geeft minder kans op een kleurovergang. Als een banaan voor de helft in de schaduw ligt is aan beide kanten van de schaduwgrens sprake van een gele kleur, maar aan de kant van de schaduw is deze donkerder. Schaduwen hebben dus met name een effect op de helderheid, niet zozeer op de kleur (er bestaan overigens zeer krachtige demonstraties van gekleurde schaduwen, maar die laten we hier buiten beschouwing). In het voorbeeld van Figuur 3 zien we een paard in de schaduw grazen. We hebben snel door dat het grasveld één geheel vormt waar een schaduw overeen loopt. De kleur en het profiel van de schaduw-overgang is voor een algoritme voldoende om het te herkennen. Het aardige is dat er software is waarmee



**Fig. 3. Staat hier een paard te grazen op een donkerder stuk gras, of staat het in de schaduw? Links het originele beeld, rechts het resultaat van een schaduw-correctie algoritme.**

schaduwen ook verwijderd kunnen worden. In Figuur 3 is dit geïllustreerd. In het gecorrigeerde beeld aan de rechterkant is nog steeds wel iets van een schaduw te zien, maar lang niet zoveel meer als in het origineel. Merk op dat de schaduw onder de bomen op de achtergrond niet is weggewerkt.

#### TOEKOMSTBEELD

In het bovenstaande zijn twee voorbeelden gegeven waarbij problemen rondom kleur opgelost worden om zo de werkelijke kleur van een object beter te kunnen bepalen en te kunnen gebruiken als zoek eigenschap. De huidige generatie van zoekmethodes werken met name op basis van object-

vormen en vormherkenning. Bij toekomstige zoekmethoden zal kleur een meer prominente rol gaan spelen. Een en ander hangt samen met de toegenomen reken capaciteit van computers zodat een praktische responstijd gehaald wordt bij zoekacties.

Kleuronderzoek voor objectherkenning zal er toe leiden dat de computer in de toekomst zelf op Internet het plaatje kan vinden van dat ene bruine paard in een groene wei in de schaduw van de ondergaande zon.

Met dank aan Arjan Gijsenij voor de kleur-bewerkingen.

# ADVERTORIAL

HIER HAD UW ADVERTORIAL KUNNEN STAAN.

Heeft u interesse neem dan contact op met [info@kleurenvisie.nl](mailto:info@kleurenvisie.nl) o.v.v. advertorial.

**I** Mijn bewondering gaat uit naar de schok. Ik ontving de schokken, observeerde ze, vergeleek ze.