omringd door informatica

Bennie Mols

onringd door informatica

Bennie Mols

Informatics is changing the way we work, play, do business, spend money, and communicate. The computer is also subtly changing the way we think, communicate, and view the world.

J.E. Savage, S. Magidson, A.M. Stein (The Mystical Machine, 1986)

Every significant technological innovation of the 21st century will require new software to make it happen.

Bill Gates (2007)

Informatica gaat net zomin over computers als astronomie over telescopen.

Edsger W. Dijkstra, 1930-2002
(Nederlandse informaticus en winnaar van de Turing Award in 1972)

Voorwoord

Het voor u liggende boek geeft een indruk van recent wetenschappelijk onderzoek op het gebied van de informatica. Aan de hand van interviews met onderzoekers geeft het een actueel en coherent beeld van interessante onderzoeksvragen en relevante toepassingen, en onderstreept het het belang van de informatica voor de hedendaagse maatschappij. Het boek is bestemd voor een breed publiek.

Alle geïnterviewde onderzoekers hebben meegewerkt in het BRICKS-project (*Basic Research in Informatics for Creating the Knowledge Society*). Dit project is mede gefinancierd met de overheidssubsidie BSIK (Besluit Subsidies Investeringen Kennisinfrastructuur), die wordt betaald uit aardgasbaten en is gericht op consortia van kennisinstellingen en bedrijven. Het algemene doel is het versterken van de kennisinfrastructuur in Nederland via wetenschappelijk onderzoek en het overdragen van resultaten uit het onderzoek naar het bedrijfsleven en de maatschappij. Op het gebied van de informatica heeft BRICKS gedurende de looptijd 2004-2009 daaraan een bijdrage geleverd.

BRICKS was een gezamenlijk initiatief van het Centrum Wiskunde & Informatica (CWI) en de Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO), gebied Exacte Wetenschappen. Naast deze twee partijen bestond het BRICKS-consortium uit de Universiteit Utrecht, de Universiteit Twente, de Technische Universiteit Delft, de Technische Universiteit Eindhoven, de Universiteit Leiden en de Radboud Universiteit Nijmegen. Het aantal betrokken onderzoekers bedroeg zo'n 150, waarvan 36 promovendi. Dit boek kan daarom slechts een selectie van de resultaten weergeven. Een volledige en meer technische beschrijving van het BRICKS-onderzoek is te vinden op de website www.bsik-bricks.nl.

Bij een groot en langdurig project als BRICKS zijn veel onderzoekers, instellingen en instanties betrokken. Onze dank gaat uit naar allen die hebben bijgedragen aan het succes van BRICKS. In het bijzonder noemen we de leden van de Adviesraad, het Monitorteam en de Commissie van Wijzen. Hun adviezen en suggesties zijn zeer welkom en effectief gebleken.

Jan Verwer BRICKS-projectleider
Peter Bosman BRICKS-projectmanager
Esther van Tienen interim communicatieadviseur

Centrum Wiskunde & Informatica (CWI)

Inhoudsopgave

Voorwoord 5

Inleiding
De wereld in je broekzak 9

Alledaagse informatica

Hoe werkt de Google zoekmachine? 23
Hoe werkt internetbankieren via iDEAL? 37
Hoe werkt Google Earth? 45
Hoe werkt intelligent cameratoezicht? 59
Hoe werkt televisie kijken via je mobiele telefoon? 73
Hoe werkt een simulatietraining voor de brandweer? 87
Hoe werkt Peer2Peer-bestandsuitwisseling? 113

Historisch overzicht
Mijlpalen van de informatica en haar toepassingen 122

Appendix
Gegevens over het BRICKS-project 137

Colofon 140

Interviews

Middenin de informatierevolutie	12	Jan van Leeuwen
De wereld gezien door de bril van data		
Magische compressieformule kijkt en vergelijkt	18	Paul Vitányi
Sneller data ophalen uit grote databestanden	26	Martin Kersten
Schatgraven in digitale databergen	32	Arno Siebes
Digitale veiligheid		
Digitaal touwtrekken tussen gemak en veiligheid	40	Wan Fokkink
Beelden interpreteren		
Nieuwe zoekmachine gidst gebruiker door beelduniversum	48	Michael Lew
Automatische borstkankerdetectie	54	Marina Velikova
Internet zonder files		
Betere postbezorging van datapakketjes	62	Jos Baeten
Nooit meer wachten op drukke websites	68	Wemke van der Weij
Planning & simulatie		
Gecombineerde gate- en busplanning	76	Guido Diepen
Vloeiend bewegen in een computergame	82	Mark Overmars
Stromingen rond schepen beter gesimuleerd	90	Jeroen Wackers
Snellere simulatie van bellen en druppels	96	Jok Tang
Betrouwbare software		
Faalkansen van softwaresystemen berekenen	102	Mariëlle Stoelinga
Logica legt systeemfouten bloot	108	Tim Willemse
Kwantumrekenen		
Rekenen aan de gedroomde kwantumcomputer	116	Harry Buhrman


Sony Ericsson
MENU

De wereld in je broekzak

In het jaar waarin ik naar de middelbare school ging – 1981 – introduceerde IBM de personal computer. Ik was twaalf jaar oud en had geen idee hoezeer de computer in de decennia daarna ieders leven zou binnendringen. Vrijwel niemand die daar toen een idee van had. In datzelfde jaar liet mijn wiskundeleraar me voor het eerst met die wonderbaarlijke machine kennismaken. We kregen computerles in BASIC, we programmeerden eenvoudige sommetjes en met het magische BASIC-commando: 10 PRINT "Hallo wereld" toverden we in groen fluorescerende lettertjes de begroeting 'Hallo wereld' op het beeldscherm.

Toch werd de computer nog vooral gepresenteerd als een verlengstuk van de wiskundeles. Niet zo gek misschien, want pas in hetzelfde jaar 1981 erkende de Nederlandse overheid voor het eerst de aparte studierichting informatica. Maar een jaar later al riep het Amerikaanse nieuwstijdschrift *Time* de personal computer tot 'Man of the Year 1982' uit. Het was de eerste keer dat niet een mens maar een machine werd gekozen.

Na mijn eerste kennismaking met de pc speelde het apparaat tijdens de rest van mijn middelbare-schooltijd geen enkele rol meer (of ik moet het hebben verdrongen). Sommige vriendjes speelden thuis computerspelletjes op een Atari of een Commodore en een enkeling waagde zich aan het programmeren. De enige computer die ik in die jaren bezat, was een fietscomputer die mijn raceprestaties tot mijn grote vreugde in keiharde getallen uiteen rafelde.

Toen ik zes jaar later met een natuurkundestudie aan de TU Eindhoven begon, bleek welke achterstand ik in de tussentijd op computergebied had opgelopen. Tijdens de eerste programmeerles in Pascal was ik de aan- en uitknop van de computer nog aan het zoeken terwijl sommige medestudenten al als volleerde programmeurs aan het werk sloegen. Ik worstelde me door de programmeerlessen heen en leerde hoezeer het

apparaat de wetenschap verrijkte. Je kon metingen automatiseren en sneller analyseren. Je kon moeilijke formules in een handomdraai door de computer laten uitrekenen. Je kon geheel nieuwe wetenschappelijke paden inslaan.

Tot dan toe was de computer voor mij nog niet meer dan een kruising van een uit de kluiten gewassen rekenmachine en een elektronische tekstverwerker – een eenzame, emotieloze relatie tussen mens en machine. Nog steeds had ik geen idee dat de computer niet alleen de wetenschap maar ook het alledaagse leven zou kunnen veranderen. Dat kwam voor mij pas in 1993, toen ik mijn eerste e-mail verstuurde.

Voor het eerst drong het tot me door welke communicatierevolutie de computer en het internet zouden kunnen ontketenen. De brieven die ik naar mijn vriendin in Polen stuurde werden binnen enkele jaren geheel vervangen door e-mails. Eindelijk kon je vrijwel meteen antwoord ontvangen – en veel goedkoper dan het internationaal bellen in die jaren. In de werkkamer kon je ineens per e-mail kletsen met collega's die tegenover je zaten – een geheel nieuwe dimensie aan het alledaagse communiceren. Een trend die zich binnen een decennium tot de zoveelste macht zou voortzetten in het chatten en in sociale netwerktoepassingen als Hyves en Facebook – digitaal voer voor sociologen en psychologen die de mens in een geheel nieuwe dimensie kunnen bestuderen.

In het nieuwe millennium bleek je ineens zelfs vanuit een dorpje in de hoge Andes e-mails te kunnen versturen en ontvangen, te kunnen bellen via internet (desgewenst met beeld erbij) en je eigen krant op internet te kunnen lezen. Achter de pc kon je je op willekeurig welke afgelegen plek in de wereld het vleugje heimwee ten minste voor even van je afschudden. En nu, anno 2009, brengen de iPhone en andere 'smart phones' alles draadloos bij elkaar: bellen, e-mailen, tv kijken, kranten lezen, muziek luisteren, surfen over het internet en zelfs een digitale wegwijzer naar de juiste straat en het dichtstbijzijnde pizzarestaurant. De hele wereld altijd en overal in je broekzak.

Deze persoonlijke ervaringen hebben vooral te maken met de nieuwe technologische snufjes die de informatica heeft voortgebracht. Dat is de buitenkant van het vakgebied. Maar er is ook een binnenkant: de informatica als de wetenschap die fundamentele vragen stelt over informatieverwerking.

Zonder binnenkant geen buitenkant. Het fundamentele informaticaonderzoek van nu legt de kiem voor de alledaagse toepassingen van later.

Vooral het schrijven van het historisch overzicht van de informatica (aan het einde van dit boek) confronteerde me met de gigantische maatschappelijke veranderingen die de informatica sinds de uitvinding van de computer ruim zestig jaar geleden heeft ontketend. Veel veranderingen begonnen met fundamentele wetenschap en waren totaal onvoorzien door de uitvinders van de computer. Het uitgebreide historisch overzicht neemt de lezer mee langs mijlpalen van de informatica en haar toepassingen.

In dit boek staat juist de binnenkant van de informatica centraal, de fundamentele wetenschap, maar zonder de buitenkant uit het oog te verliezen. Zestien hoofdartikelen proberen een breed publiek een gevoel te geven van de informatica als wetenschap. Het fundamentele onderzoek van nu kan aan de basis staan van toekomstige toepassingen. De kaders bij deze hoofdartikelen gaan een stuk dieper en zijn soms zelfs moeilijk. Maar moeilijke zaken overwinnen is noodzakelijk om tot grote hoogte te stijgen – of het nu gaat om het nemen van een rake vrije trap, het spelen van een ontroerende pianosonate of het bedenken van een efficiënt algoritme dat de kortste weg tussen Amsterdam en Zagreb berekent.

Daarnaast komen in zeven extra artikelen over alledaagse informaticatoepassingen de binnen- en buitenkant weer bij elkaar. Zoals informaticahoogleraar Jan van Leeuwen het in de ouverture van dit boek perfect verwoordt: "Meer dan enig ander vakgebied is de informatica een uniek samenspel tussen wetenschap en *engineering*, tussen wetenschappelijke ontdekkingen en het instrumentarium dat het vakgebied ontwikkelt."

Bennie Mols

Middenin de informatierevolutie

Waar de biologie de levende natuur bestudeert en de natuurkunde de levenloze natuur, daar bestudeert de informatica de wereld opgebouwd uit informatieprocessen. Of het nou gaat om administratieve, organisatorische, sociale, of zelfs natuurkundige of biologische processen, je kunt ze modelleren met informatieprocessen.

Zoals de stoommachine de industriële revolutie ontketende, zo hebben de computer en het internet de informatie-revolutie van eind twintigste en begin eenentwintigste eeuw ontketend. We chatten en e-mailen over de digitale snelweg, doen onze bankzaken achter de pc en vinden met zoekmachines onze weg door een alsmaar uitdijend informatie-universum. Internet is het verlengstuk van de mens geworden in zijn hele denken en doen.

De informatierevolutie staat nog maar aan het begin. Google wil alle op de wereld bestaande boeken digitaal beschikbaar maken; er komen digitale 'vinger-afdrukken' van je eigen erfelijke informatie en intelligentie wordt in alles om ons heen ingebouwd. Sommigen dromen al van een wereld waarin ieder mens en elk voorwerp – van voordeur tot auto en mobiele telefoon – altijd en overal verbonden is in een virtuele wereld, zodat iedereen

met alles kan communiceren. Deze technologische toepassingen zouden ondenkbaar zijn zonder de wetenschap van de informatica.

Maar waar gaat deze nog vrij jonge tak van wetenschap eigenlijk precies over? Volgens hoogleraar informatica Jan van Leeuwen van de Universiteit Utrecht werkt de informatica volgens een unieke, drievoudige methodologie. "Allereerst ontwikkelen informatici de theorie van informatieverwerkende systemen. Vanuit deze theorie ontwikkelen ze de concepten, modellen en technieken waarmee ze elk willekeurig gebied dat mensen interesseert effectief met de computer kunnen bestuderen. Vervolgens vertalen ze deze inzichten in werkende computerprogramma's. Elk model van een stukje wereld wordt dan een computerprogramma dat inzicht geeft in hoe dat stukje wereld werkt en ook hoe het bewerkt kan worden tot een product of systeem dat mensen kunnen gebruiken."

Informatiewetten

De eerste trap in de methodologie van de informatica is de wetenschappelijke theorie. Op dit niveau proberen informatici antwoorden te vinden op fundamentele vragen als: Wat is informatie? Wat is berekenbaar? Wat is intelligentie? Zijn er algemene informatiewetten? Zijn er algemene kaders om het effect van programma's te beschrijven en te meten? Kunnen we manieren vinden om problemen op te lossen die zelfs de snelste computers van nu nog niet aankunnen?

Waar de biologie de levende natuur bestudeert, de natuurkunde de levenloze natuur, daar bestudeert de informatica de wereld opgebouwd uit informatieprocessen. Of het nou gaat om administratieve, organisatorische, sociale of zelfs natuurkundige of biologische processen, je kunt ze modelleren met informatieprocessen. De informatica zoekt naar manieren om deze informatieprocessen te begrijpen en te beschrijven en de modellen te ontwikkelen waarmee je ze op computers kunt vormgeven en toetsen. Modellering is de tweede trap in de methodologie van de informatica.

Bij modellering spelen algoritmen een cruciale rol. Algoritmen geven de essentie weer van een stukje informatieverwerking en zijn tegelijk de rekenrecepten voor

Controlekamer CERN

de computer (zie kader p. 16). Sommige informatici zien het bedenken van algoritmen als de wetenschappelijke kern van de informatica. Maar voor Van Leeuwen is dat een te beperkte blik: "Neem bekende alledaagse toepassingen zoals eBay, Google, Amazon, of, dichter bij huis, de reisplanner van de NS. Ze bestaan dankzij hun onderliggende algoritmen. Maar het is te simplistisch om te denken dat ze alleen maar bestaan dankzij algoritmen. Zonder een gebruiksvriendelijk ontwerp en de creatieve softwaretechnologie voor het web zouden deze toepassingen nooit succesvol kunnen zijn."

Programma- en productontwerp

Voor Van Leeuwen is het ontwerpen van vernuftige, voor anderen bruikbare programma's en systemen uitdrukkelijk ook deel van de hele methodologie.
Een belangrijk aspect is de ontwikkeling van zowel de methoden als de technologie waarmee bedrijven en deskundigen snel de programma's kunnen ontwikkelen

Algoritmen

Algoritmen zijn de rekenrecepten voor computers. Ze zijn hét middel om oplossingen voor informatieverwerkingsproblemen efficiënt te berekenen. Zo zit in een TomTom een algoritme om op een wegenkaart de kortste weg van A naar B te vinden. Google gebruikt een algoritme om zoekresultaten te rangschikken op volgorde van relevantie. Biologen gebruiken algoritmen om genen, eiwitten of virussen met elkaar te vergelijken. Vervoersbedrijven stellen met behulp van algoritmen optimale dienstregelingen op. Waar een natuurwetenschapper een verschijnsel probeert te vangen in een formule, zal een informaticus altijd proberen een verschijnsel te vatten in een beschrijving die uitvoerbaar is op een computer. De natuurkundige wetten die luchtstromen en lichtwerking beschrijven, zijn bekend, maar dat betekent niet dat je dan ook weet hoe je ruisende bladeren aan een boom levensecht kunt nabootsen op een computer. Je zult creatief moeten nadenken over een rekenmethode – een algoritme – om bladeren en hun kleuren op een computer te simuleren. Het bedenken van efficiënte algoritmen, voor welke toepassing dan ook, is een van de grote uitdagingen voor informatici. Daarvoor is een combinatie van wetenschap en *engineering* nodig, van kennis en kunde. In al het informaticaonderzoek dat in dit boek aan bod komt, spelen algoritmen een centrale rol.

die aan hun eisen of die van hun klanten moeten voldoen. Ontwerp is daarom de derde trap in de informaticamethodologie. Deze trap leidt meteen weer tot nieuwe vragen voor de theorie: Hoe kunnen we complexe systemen simpel bouwen? Hoe kunnen we garanderen dat ze foutloos en veilig zijn? Hoe kunnen we ze onderhouden?

"Meer dan enig ander vakgebied", zegt Van Leeuwen, "is informatica een uniek samenspel tussen wetenschap en *engineering*, tussen wetenschappelijke ontdekkingen en het instrumentarium dat het vakgebied ontwikkelt. Computers en software worden in rap tempo slimmer dankzij de ontdekkingen in de informatica en dit stuwt het vakgebied weer voort naar nieuwe ontdekkingen. Zo heeft de ontwikkeling van het internet een geheel nieuwe dimensie geopend in het omgaan met informatie, die een informaticus van de jaren tachtig zich niet eens kon voorstellen, laat staan bestuderen."

De informatica als vakgebied ontstond in de jaren vijftig van de vorige eeuw. Vakgebieden als wiskunde, natuurkunde en sterrenkunde waren toen al vele eeuwen oud. Dat biedt voordelen. Waar de maatschappij het normaal vindt dat natuur- of sterrenkundigen ook theorieën ontwikkelen die alleen maar intellectuele vergezichten bieden zonder directe toepassing, daar wordt de informatica vooral afgerekend op de producten die ze de samenleving oplevert. Juist omdat de informatica zo'n jong vakgebied is, is het voor velen nog wennen dat de informatica ook een wetenschap is die hoogstaande abstracte kennis genereert.

Van Leeuwen: "Het sterke punt van de informatica is dat ze zoveel succesvolle toepassingen oplevert. De schaduwkant hiervan is dat de informatica als wetenschap te veel wordt afgerekend op alleen maar die toepassingen. De drietrapsraket theorie-model-ontwerp kan alleen maar succes boeken als we ook de ruimte geven aan de ontwikkeling van de fundamenten van de informatica, aan de eerste twee trappen van de raket. En dat is precies wat we in het BRICKS-programma hebben gedaan: het ontwikkelen van de fundamenten van de informatica, met in het achterhoofd de toepassingen ervan in de moderne informatiemaatschappij." ●

Magische compressieformule kijkt en vergelijkt

De compressieformule berekent net zo gemakkelijk de gelijkenis van virussen of diersoorten als die van muziekstukken of boeken.

Zodra ergens in de wereld een nieuw virus opduikt, proberen biologen zo snel mogelijk om zijn erfelijke informatie te ontrafelen. Die erfelijke informatie kun je zien als een lang woord bestaande uit een combinatie van maar vier letters (de DNA-baseparen A, C, G en T). Vervolgens willen biologen weten met welke van de al bekende virussen het nieuwe virus het meest verwant is. Daarvoor kunnen ze zoeken naar specifieke virologische overeenkomsten. Maar ze kunnen dit ook uitvogelen door een computer het erfelijk materiaal van het ene virus letter voor letter te laten vergelijken met dat van andere virussen. Deze rekenmethode is echter traag en ongeschikt om lange stukken erfelijke informatie met elkaar te vergelijken. Bio-informatici hebben zogeheten *alignment*-technieken bedacht om die vergelijking sneller uit te voeren.

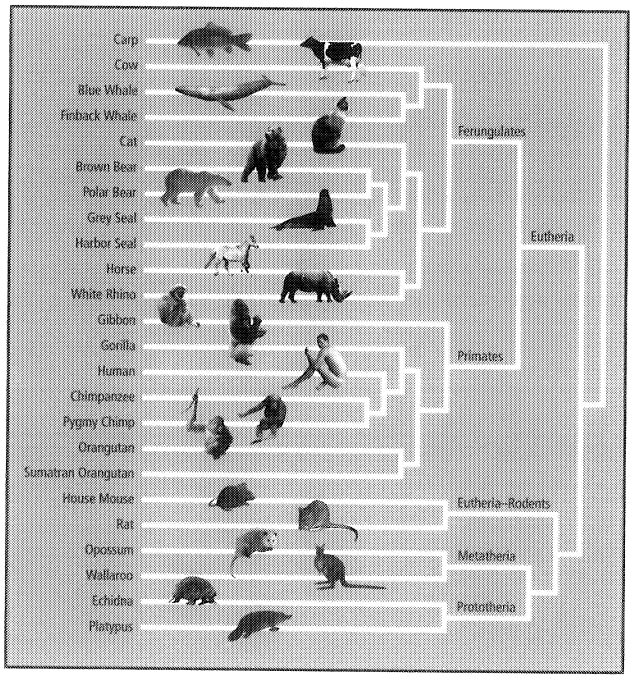
Theoretisch informaticus Paul Vitányi van het Centrum Wiskunde & Informatica (CWI) heeft met zijn collega's een alternatieve, zowel praktische als snelle manier ontwikkeld om te berekenen hoezeer virussen op elkaar lijken. De methode is zelfs zo algemeen dat ze alles kan vergelijken wat in een reeks enen en nullen kan worden uitgedrukt, of het nou virussen zijn, muziekstukken of boeken. 'Clusteren door compressie', noemt Vitányi de nieuwe methode. Het idee is voortgekomen uit het jarenlange onderzoek dat hij heeft gedaan naar de vraag hoe je de complexiteit van informatie zo precies mogelijk wiskundig beschrijft.

Een digitaal bestand bestaat uit een lange rij van enen en nullen. Tenzij het alleen maar ruis is, bevat het bestand altijd wel enige structuur, bijvoorbeeld een terugkerend patroon van tien nullen achter elkaar. Speciale compressieprogramma's herkennen zulke structuren en gebruiken die informatie om het bestand compacter op te slaan. Hoe beter de compressor, hoe meer structuur het programma ziet en hoe kleiner het gecomprimeerde bestand wordt.

Vitányi stelde een praktisch bruikbare formule op die een goede compressor gebruikt om te berekenen hoezeer twee digitale bestanden op elkaar lijken (zie kader). "De formule ziet er heel eenvoudig uit", vertelt Vitányi, "maar het heeft jaren gekost om hem af te leiden en te bewijzen dat hij de juiste eigenschappen heeft om de complexiteit van twee digitale bestanden te vergelijken. We hebben ook een kwaliteitsnorm opgesteld waaraan een goede compressor moet voldoen om hem voor het uitrekenen van de formule te mogen gebruiken."

Stamboom

Het eerste succes van de nieuwe methode bleek uit een experiment om diersoorten op grond van hun



Deze evolutionaire stamboom is geconstrueerd uit het mitochandriale DNA van de afgebeelde zoogdiersoorten

Magische compressieformule

Als je beeld, geluid of tekst codeert in enen en nullen, dan is de Kolmogorov-complexiteit de lengte van het kortste programma dat zo'n stuk informatie beschrijft. Stel dat je een bestand van honderdduizend nullen hebt. Dan hoeft het kortste programma alleen maar te zeggen: 'print honderdduizend nullen'. Maar als je honderdduizendmaal een muntstuk opgooit en achter elkaar noteert of je 'kop' (1) of 'munt' (0) krijgt, dan kun je het resultaat vrijwel nooit comprimeren. In de praktijk is de Kolmogorov-complexiteit onberekenbaar, waardoor deze maat niet gebruikt kan worden om de complexiteit van bestanden met elkaar te vergelijken. Vitányi heeft op grond van de formule voor de Kolmogorov-complexiteit echter een nieuwe formule afgeleid die in de praktijk wel berekenbaar is. Voor de genormaliseerde compressieafstand NCD tussen twee bestanden x en y geldt namelijk:

$$NCD(x,y) = \frac{C(xy) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}}$$

De genormaliseerde compressieafstand drukt in een getal tussen 0 en 1 uit hoezeer twee bestanden op elkaar lijken. Hoe dichter bij 0, hoe meer de bestanden op elkaar lijken. Hoe dichter bij 1, hoe meer de bestanden van elkaar verschillen. In de praktijk gebruik je een compressieprogramma C om de formule uit te rekenen. Compressieprogramma C comprimeert zowel alle afzonderlijke bestanden x en y tot C(x) en C(y), als ook alle combinaties van twee aan elkaar geplakte bestanden xy tot C(xy). Reken je voor een heleboel paren van bestanden de NCD uit, dan kun je vervolgens een boomstructuur maken waarin bestanden die meer op elkaar lijken dichter bij elkaar in de boomstructuur zitten. Deze compressieformule levert een resultaat op dat voor natuurlijke data waarschijnlijk nauwelijks afwijkt van de theoretische Kolmogorov-waarde, hoewel dat formeel niet te bewijzen is.

al bekende DNA te classificeren in een evolutionaire stamboom. Vitányi: "De stamboom die de compressiemethode zonder enige biologische voorkennis opstelde, bleek geheel identiek aan de stamboom die biologen, met al hun achtergrondkennis, hadden opgesteld."

Een volgende bevestiging van het praktisch nut van de methode bleek tijdens de uitbraak van het SARS-virus in 2003. Nadat de erfelijke informatie van het virus was ontrafeld, berekende het compressieprogramma in een mum van tijd dat SARS sterk verwant is aan het Corona-229-virus. "Dat was nog voordat biologen, met hun jarenlang opgebouwde kennis van virussen, dezelfde conclusie stelden", aldus Vitányi.

Daarna bleek clusteren door compressie ook geschikt om talen te rangschikken in taalbomen, muziekstukken te classificeren naar componist en boeken te clusteren naar auteur. En dat allemaal zonder dat het programma ook maar een flauw benul heeft van talen, muziek, literatuur of biologie.

Amerikaanse wetenschappers vergeleken de compressiemethode met de belangrijkste andere bekende dataminingtechnieken, die wel a priori kennis van een vakgebied gebruiken. Ze voerden de vergelijking uit op een groot aantal belangrijke databases. Vitányi: "Zij concluderen dat onze compressiemethode in het algemeen minstens even goed werkt en dat ze zelfs veel beter presteert als er ineens gekke afwijkingen in de gegevens zitten. Andere programma's zijn minder goed in het herkennen daarvan, omdat ze van tevoren gedefinieerde eigenschappen gebruiken, zoals een bepaald ritme in muziekstukken of een bepaalde lettervolgorde in het DNA van virussen. Onze methode zoekt niet naar eigenschappen die je van tevoren moet definiëren. En dat is vooral handig in gevallen waarbij je niet precies weet naar welke regelmaat je zoekt."

Google-afstand

Een van de jongste toepassingen van clusteren door compressie, is het gebruik van zoekmachine Google als woordenboek om computers automatisch woorden te laten begrijpen. Samen met zijn promovendus Rudi Cilibrasi ontwikkelde Vitányi een afstandsmaat voor de betekenis van woorden. Zo levert het Engelse woord 'horse' 168 miljoen Google-hits op; het woord 'rider' 68 miljoen. Zoeken op pagina's waarin beide woorden tegelijk voorkomen, geeft 6 miljoen treffers. Uit die getallen kan worden berekend hoezeer 'horse' en 'rider' qua betekenis met elkaar in verband staan. Door een web van meer en minder verwante woorden te creëren, kunnen computers zo automatisch woorden begrijpen.

Inmiddels wordt de compressiemethode al in bijna negenhonderd wetenschappelijke artikelen geciteerd en in uiteenlopende wetenschapsgebieden toegepast. "Voor een theoreticus zoals ik is het een enorme kick om te zien dat al een paar jaar na de theorie de toepassingen zo'n enorme vlucht nemen", besluit Vitányi. ●

Hoe werkt de Google zoekmachine?

Google, Bing en Yahoo zijn de drie meest gebruikte zoekmachines ter wereld, maar Google spant de kroon. In de meeste Europese landen ligt het marktaandeel van Google boven de negentig procent, in de VS tegen de tachtig procent. Dagelijks beantwoordt Google honderden miljoenen zoekopdrachten. In een oogwenk verschijnt een lange lijst van resultaten op je scherm. Hoe dat kan? In essentie door de brute kracht van heel veel computers en een rekenformule die op een slimme manier inschat welke antwoorden je zoekt.

Google – afgeleid van het woord 'googol': een 1 met honderd nullen – beschikt over een gigantisch gegevensbestand met

kopieën van webpagina's. Speciale software (*webspinnen* genaamd) zoekt geregeld naar zo veel mogelijk bestaande websites. De zoekmachine slaat vervolgens kopieën van de gevonden pagina's op, verspreid over honderdduizenden computers (het precieze aantal is geheim). Die kopieën vormen de database waarin de zoekmachine speurt. Zelfs Google ziet maar een deel van alle webpagina's. Precieze cijfers daarover zijn niet bekend, maar sommige experts denken dat het maar één procent is. Naar schatting bestaat Google's database uit tientallen miljarden webpagina's en dat aantal groeit voortdurend.

Pageranking

De crux van een goede zoekmachine zit in een slimme zoekstrategie, gebaseerd op drie principes. Allereerst telt mee hoe vaak een zoekwoord op een bepaalde pagina voorkomt. Dit deden alle zoekmachines vóór de introductie van Google ook al. Google was in 1998 echter de eerste die liet meewegen hoe vaak er naar de betreffende pagina wordt verwezen vanaf andere pagina's. Hoe meer andere webpagina's naar een site verwijzen, hoe belangrijker deze waarschijnlijk is. Dit tweede zoekprincipe bleek een gouden zet, die Google op grote voorsprong zette.

Met de oude zoekstrategie, die alleen het aantal gezochte woorden per pagina telde, zou het kunnen zijn dat je, als je 'Shell' intikt in de zoekmachine om de thuispagina te vinden, terechtkomt op de site van Greenpeace, omdat deze bijvoorbeeld in kritische stukken vele malen de naam van het bedrijf noemt. Door nu ook mee te wegen hoe vaak er naar een pagina wordt verwezen, is de kans veel groter dat je meteen terechtkomt bij de thuispagina van het bedrijf. Het derde principe is dat pagina's die langer bestaan ook een hogere waardering krijgen. De drie principes bij elkaar heeft Google verwerkt in het zoekalgoritme *PageRank*.

PageRank kent een waardering toe aan elke vondst en rangschikt ze naar belangrijkheid. Voortdurend wordt geprobeerd om het zoekalgoritme te verbeteren, maar de details van de toverformule zijn geheim. Anders kun je al te gemakkelijk je eigen site in de resultatenlijst kunstmatig naar boven stuwen.

Waarschijnlijkheden

Laten we nu iets meer ingaan op het algoritme PageRank. Wiskundig gezien is PageRank een waarschijnlijkheidsverdeling die de waarschijnlijkheid geeft dat iemand die willekeurig op weblinks klikt bij een bepaalde pagina terechtkomt. De waarschijnlijkheid is een getal tussen 0 en 1. Het algoritme berekent de PageRank van een pagina stapje voor stapje, waarbij elk stapje de uiteindelijke PageRank beter benadert.

Neem het eenvoudige voorbeeld van vier webpagina's: A, B, C en D. We beginnen met de veronderstelling dat de PageRank (PR) van alle vier de pagina's gelijk is: PR(A) = PR(B) = PR(C) = PR(D) = 0,25. We gaan nu een stap verder. Stel dat de pagina's B, C

en D alle drie alleen maar een link naar A bevatten. Dan is PR(A) = PR(B) + PR(C) + PR(D) = 0,75. We gaan nu nog een stap verder en veronderstellen dat B ook nog een link naar C en D heeft en dat D naar alle drie de andere pagina's verwijst. Dan is PR(A) = PR(B)/2 + PR(C)/1 + PR(D)/3.
Meer in het algemeen: de waarde van een link van pagina X naar A is de PageRank van X gedeeld door het aantal links vanuit X. De deling zorgt voor de normering. De PageRank van A is dan de som van alle genormeerde links die naar A verwijzen. Dit voorbeeld geeft de basis van de PageRank-berekening. In werkelijkheid neemt de berekening nog meer details mee.

De combinatie van zoekalgoritme en veilingalgoritme hebben van Google een succesvolle zoekmachine gemaakt.

Veilingalgoritme

Voor de gewone gebruiker is Google gratis, maar toch is de zoekmachine tegenwoordig *big business*. Al onze gratis zoekopdrachten vertellen namelijk iets over wie we zijn en wat we willen en dat is een vermogen waard. Wie bijvoorbeeld als zoekterm 'Tenerife' intikt, krijgt aan de rechterkant van de zoekresultaten meteen ook een lijst met gesponsorde links. Deze lijst wordt bepaald door een veilingalgoritme en dat is het algoritme waarmee Google haar geld verdient.
Adverteerders kunnen bieden om bij een bepaalde zoekopdracht vermeld te worden. Het veilingalgoritme evalueert vervolgens de biedingen op relevantie voor de zoekopdrachtgever, de kwaliteit van de pagina van de adverteerder en het voorbije klikgedrag op de advertentie. Daarna berekent het een ranking door de bieding van de adverteerder te vermenigvuldigen met de kwaliteitsscore. Ten slotte berekent het de prijs die de adverteerder daadwerkelijk moet betalen. Die prijs is gelijk aan de een-na-hoogste bieding maal de een-na-hoogste kwaliteitsscore gedeeld door de kwaliteitsscore van de adverteerder.
De combinatie van zoekalgoritme en veilingalgoritme hebben van Google zo'n succesvolle zoekmachine gemaakt dat het werkwoord 'googelen' synoniem is geworden voor zoeken op internet.

Sneller data ophalen uit grote databestanden

Hoe groter digitale databestanden worden, hoe moeilijker het wordt om erin te zoeken. Twee nieuwe informaticatrucs versnellen het zoekproces flink.

Hoe kun je in een grote hoeveelheid gegevens snel vinden wat je zoekt? Dat is de centrale vraag voor datamanagementsystemen. Of je nu online winkelt, online bankiert, of airmiles spaart: de manier waarop je dat doet, wordt in een database bewaard door de betrokken bedrijven. Die gebruiken je klikgegevens om hun bedrijfsvoering te verbeteren. Maar ook overheden beschikken over steeds grotere databestanden, die ze in toenemende mate online beschikbaar stellen.

De grootste bekende database in het publiek-private domein is die van veilingsite eBay, die anno 2009 maar liefst 2,5 petabyte aan data bevat. Eén petabyte (10^{15} bytes) komt overeen met driehonderdduizend volle dvd's. eBay gebruikt zijn database niet alleen voor de marketing, maar ook voor de opsporing van fraude, zoals prijsafspraken of wanbetalers.

Daarnaast creëren ook wetenschappelijke experimenten, zoals sterrenkundige waarnemingen en DNA-analyses, steeds grotere hoeveelheden data. Wereldwijd is de jaarlijkse omzet van databasetechnologie tussen de twintig en veertig miljard dollar en de omzet groeit jaarlijks omdat steeds meer gegevens digitaal beschikbaar komen.

De eerste databasetechnologie stamt uit de jaren zeventig en tachtig. Toen keek men tegen data aan als gegevens die corresponderen met eigenschappen van een persoon of een object; denk aan een bankrekeningnummer dat bij een bepaalde persoon hoort. Omdat er tegenwoordig veel meer gegevens beschikbaar zijn, kijkt men in de moderne datawereld heel anders tegen data aan, vertelt hoogleraar Informatica Martin Kersten van het Centrum Wiskunde & Informatica (CWI): "Het wordt steeds belangrijker om uit de verzamelde hoeveelheid gegevens nieuwe statistische inzichten over groepen af te leiden, in plaats van alleen maar informatie over een individu. Dat stelt nieuwe eisen aan een datamanagementsysteem."

Recycling

Kersten werkt al sinds 1993 met zijn collega's aan het voortdurend verbeteren van een databasearchi-

Het Savvis, Inc. data-centrum (NJ2) in Weehawken, New Jersey, USA

tectuur voor deze nieuwe wereld. Ze hebben het open source datamanagementsysteem MonetDB ontwikkeld, dat wereldwijd zowel in de wetenschappelijke wereld als daarbuiten wordt gebruikt. Zo gebruikt het Nederlands Forensisch Instituut het om efficiënt te zoeken in grote hoeveelheden geconfisqueerde harde schijven.

De afgelopen jaren hebben Kersten en zijn collega's MonetDB met twee nieuwe trucs uitgebreid, die allebei het zoeken in databestanden flink versnellen. Beide trucs zijn ontwikkeld en experimenteel getest in de proeftuin van een grote database met sterrenkundige gegevens: de *Sloan Digital Sky Server*. Deze bevat momenteel ruim drie terabyte (3×10^{12} bytes) aan gegevens over waargenomen sterren – een soort digitale hemelkaart. Per maand raadplegen sterrenkundigen van over de hele wereld deze database zo'n twee miljoen maal.

Net zoals een index achterin een boek je helpt om op trefwoord te zoeken in de volledige tekst van het boek, zo heeft de Sloan Digital Sky Server een catalogus die beschrijft hoe de databasestructuur in elkaar zit. Daarin staan zo'n zeventig tabellen en honderden routines beschreven in een van de standaard databasetalen: SQL (Structured Query Language). SQL lijkt op een programmeertaal, maar dan toegespitst op het beschrijven van verzamelingen. Wanneer een sterrenkundige op zoek is naar gegevens van een bepaalde ster, dan vertaalt het datamanagementsysteem de zoekopdracht eerst naar SQL, waarna het de zoekopdracht op de database kan uitvoeren.

"De eerste noviteit die wij hebben verzonnen", vertelt Kersten, "is gebaseerd op het idee van recycling. Wij bewaren de tussenresultaten van alle zoekopdrachten, terwijl ze voorheen altijd werden weggegooid. Bij elke nieuwe zoekopdracht kijkt het systeem eerst of delen van de zoekopdracht al eens eerder zijn uitgerekend. Is dat het geval, dan wordt het eerdere resultaat uit het geheugen opgehaald. Dit recyclen van kleine onderdelen van vragen uit het verleden gaat veel sneller dan het opnieuw beantwoorden van de zoekopdracht."

De onderzoekers hebben het recycling-idee geïmplementeerd in hun eigen MonetDB-systeem en vervolgens getest op de Sloan Digital Sky Server. Kersten: "Uit dit experiment is gebleken dat je zelfs op de optimaal ge-

Sorteren terwijl je zoekt

Het idee van 'cracking' is om niet alles vooraf te sorteren, maar telkens een subsortering te doen wanneer er een nieuwe zoekvraag komt. Deze truc valt het handigste uit te leggen aan de hand van het voorbeeld van een ongeordende stapel speelkaarten. Hierin mogen de datagebruikers zoeken met een zoekopdracht. Stel, de eerste gebruiker vraagt naar de kaart 'harten twee'. Dan moet het systeem in het algemeen kaart voor kaart doorlopen om de harten twee te vinden. Maar als het systeem toch al aan het zoeken is naar een harten twee, kan het ook wel meteen alle harten die het onderweg tegenkomt op een stapel met alleen harten leggen en alle niet-harten op een tweede stapel. Stel, dat de tweede gebruiker alle klaveren zoekt, dan weet het datamanagementsysteem dat het nu alleen nog maar hoeft te zoeken in de stapel met niet-harten. En tijdens het uitvoeren van deze zoekopdracht kan het bijvoorbeeld twee nieuwe stapeltjes maken van schoppen en klaveren. Elke nieuwe stapel vereenvoudigt de volgende zoekopdracht. Wanneer het nu gaat om realistische digitale data in plaats van om speelkaarten, dan schrijft het systeem de data bij elke nieuwe zoekopdracht in een nieuwe volgorde terug waardoor automatisch een steeds betere sortering ontstaat.

configureerde Sky Server-database een factor vier wint in zoeksnelheid."

Cracking

Een tweede techniek die Kersten en zijn collega's hebben verzonnen om het zoekproces te versnellen, gaat in tegen een van de centrale paradigma's in databasemanagement. Kersten: "Volgens dit paradigma staat of valt efficiënt zoeken met een goede zoekindex. Stel dat je een ongeordende stapel speelkaarten hebt. Een index legt alle speelkaarten nu in de juiste volgorde, gesorteerd op harten, ruiten, schoppen en klaveren en ook op de cijfers en plaatjes. Dat lijkt heel handig, maar het kost wel veel sorteertijd."

Kersten verzon de nieuwe *'cracking'-methode*, die niet meer alles vooraf precies indexeert, maar telkens

wanneer een nieuwe zoekopdracht wordt gegeven de data hersorteert. Zo wordt het bij elke volgende vraag gemakkelijker om een antwoord te vinden (zie kader). Het grote voordeel van cracking, is ook dat je beter gebruikmaakt van dat waarnaar mensen zoeken. Je hoeft dat niet meer vooraf vast te leggen.

"Toegepast op de Sloan Digital Sky Server als proeftuin blijkt de cracking-methode het zoeken met een factor tien tot twintig te versnellen", zegt Kersten. "De efficiëntie van een nieuwe datamanagementtechniek kun je niet alleen op papier bewijzen, maar moet je altijd op een echte database demonstreren. Wat dat betreft lijkt ons informaticawerk eerder op experimentele fysica dan op wiskunde." ●

Schatgraven in digitale databergen

Supermarkten, banken, ziekenhuizen en wetenschappelijke experimenten genereren steeds grotere digitale databergen. Informatici proberen daar met nieuwe algoritmen het verborgen goud uit te halen.

Elke twintig maanden verdubbelt de hoeveelheid digitale informatie in de wereld, zo is de schatting. Aan de ene kant gaat het om praktische gegevens, zoals verkoopgegevens van supermarkten, banktransacties, transportgegevens en medische gegevens. Aan de andere kant gaat het ook om uitkomsten van wetenschappelijke experimenten, zoals de ontrafeling van welk gen bij een bepaalde ziekte is betrokken, interacties tussen eiwitten in een lichaamscel, of sterrenkundige observaties. *Datamining* is de tak van sport die probeert om interessante patronen te vinden in een grote databrij. Een vorm van goud delven in digitale databergen.

Grote datahoeveelheden dwingen ons om op een andere manier dan vroeger te werk te gaan. "De klassieke manier van onderzoek doen", vertelt hoogleraar datamining Arno Siebes van de Universiteit Utrecht, "is dat je eerst een hypothese opstelt en daarna een experiment doet waarbij je data verzamelt. Ten slotte toets je of de hypothese klopt met het experiment. Maar bij grote databergen kan er goud in verborgen liggen zonder dat je er naar op zoek bent. Dan heeft het zin om ongericht – zonder een hypothese – te gaan schatgraven. Door nieuwe patronen in

Albert Heijn, Utrecht

een databerg te ontdekken, kun je namelijk op een interessante, onvermoede wetenschappelijke hypothese stuiten."

Boerenkool met rookworst

De belangrijkste commerciële toepassing van datamining ligt momenteel in de supermarktwereld. De kassa van een supermarkt slaat de gegevens van de afgerekende boodschappen op. Die kennis kan de supermarkt gebruiken om het koopgedrag van klanten te benutten, voor henzelf, maar ook voor de klant. Stel dat uit de kassagegevens blijkt dat mensen die boerenkool kopen ook vaak rookworst kopen. Dan weet de supermarkt dat wanneer ze boerenkool in de reclame doen, ze niet alleen moeten zorgen voor extra boerenkool, maar ook voor extra rookworst.

Siebes ontwikkelt samen met zijn collega's algoritmen om zinvolle informatie uit databergen te halen. Een succesvol algoritme dat ze in de afgelopen jaren hebben ontwikkeld, is het *Krimp-algoritme*. Siebes: "Krimp kun je voor veel toepassingen gebruiken, maar ik kan de methode het beste uitleggen in de context van de supermarkt. Stel, de supermarkt wil weten welke producten vaker samen worden gekocht dan een bepaalde drempelwaarde. Als je de drempel heel hoog legt, dan vind je alleen oninteressante

informatie. Bijvoorbeeld dat een plastic tas samen wordt gekocht met alle mogelijke producten, omdat mensen vaak vergeten hun eigen boodschappentas mee te nemen. Maar als je de drempel heel laag legt, dan vind je juist een explosie aan resultaten. Bijna alles wordt vaker gekocht dan die lage drempelwaarde. Toch kunnen daar juist de interessante combinaties bij zitten waar niemand nog aan heeft gedacht. Het kan bijvoorbeeld blijken dat mensen die een dure fles rode wijn kopen ook vaker tegelijk dure bonbons als toetje kopen."

Het Krimp-algoritme kan de grote hoeveelheid gevonden resultaten op een slimme manier terugbrengen tot een behapbare hoeveelheid. Dan wordt het ineens veel gemakkelijker om het verborgen goud in de data te herkennen. Het achterliggende idee van Krimp is gebaseerd op het principe van 'leren is comprimeren'.

Optimale codetabel berekenen

Het comprimeren van digitale bestanden is een krachtige praktische methode om de complexiteit van bestanden te meten. De compressiemethode die Arno Siebes in het Krimp-algoritme gebruikt om in databergen te zoeken, is gebaseerd op de berekenbare versie van de theoretische *Kolmogorov-complexiteit*, die door Paul Vitányi en Ming Li is ontwikkeld (zie kader op pagina 21). Om Krimp toe te passen, moeten de data in een codetabel staan, het model dat de databerg beschrijft. In een codetabel staan in de linkerkolom verzamelingen van items (bijvoorbeeld producten uit de supermarkt) en in de rechterkolom een code voor de betreffende verzameling. Voor Krimp maakt het niet uit wat de codes zijn, het gaat om hun lengte. Stel dat code c_1 staat voor de verzameling {bier, luiers} en c_2 voor {kaas}, dan is een mogelijke transactie van een klant c_1c_2. Een groot aantal codes wordt zo gecombineerd tot vele mogelijke transacties. Vervolgens kun je Krimp gebruiken om een optimale codetabel te vinden. Het algoritme begint met een geldige codetabel en een gesorteerde lijst van kandidaat-itemsets. Daarna voeg je kandidaat-itemsets een voor een toe aan de codetabel en telkens comprimeer je het geheel. Alleen als het geheel beter comprimeert, voeg je de itemset toe aan de codetabel, maar anders niet. Stap voor stap kun je zo de meest gecomprimeerde codetabel vinden. Dat is dan het model dat de kortste beschrijving geeft van de databerg.

Als je twee modellen hebt om een database te beschrijven, dan is het meest gecomprimeerde model het beste (zie kader p. 35).

Privacybescherming

Een belangrijke sociale toepassing van datamining gaat over privacybescherming. Hoe kan een instantie in grote datahoeveelheden zoeken zonder dat ze achter privacygevoelige informatie komen?

Stel dat je uit een grote hoeveelheid medische gegevens te weten komt dat er één roodharige man is die aids heeft en die in een dorpje van een paar honderd inwoners woont. De kans is dan groot dat die informatie maar op precies één persoon kan slaan. Wanneer deze informatie op straat komt te liggen, is de privacy geschonden. Stel nu, dat uit de gegevens ook volgt dat er een roodharige man in Amsterdam is die aids heeft, dan is de privacy waarschijnlijk niet geschonden. Amsterdam heeft zoveel inwoners dat de kans groot is dat er meerdere roodharige mannen wonen die aids hebben. En dan is de informatie uit de database niet tot een persoon te herleiden.

Siebes: "Wij hebben ons Krimp-algoritme gebruikt om uit een originele database, die de privacy niet garandeert, een nieuwe database te maken. Deze nieuwe database garandeert wel de privacy, terwijl je deze toch nog voor alle nuttige toepassingen kunt gebruiken. We hebben laten zien dat de twee databases statistisch als twee druppels water op elkaar lijken, behalve voor die gegevens die heel weinig voorkomen. Op dat punt wijken de originele en de gegenereerde database sterk van elkaar af. Maar dat is precies wat je uit privacyoverwegingen wilt."

Hoe groter de databases worden, hoe moeilijker het wordt om erin te schatgraven. En dus blijven informatici zoeken naar nog geavanceerdere algoritmen die nieuwe patronen ontdekken – patronen die inzichten onthullen waar nog niemand aan heeft gedacht. ◉

Hoe werkt online betalen met iDEAL?

We kenden al het internetbankieren en het webwinkelen, toen in 2005 het nieuwe online betalingssysteem iDEAL op het digitale betalingstoneel verscheen. Hiermee kun je een product bij een webwinkel kopen en direct via je eigen bank betalen. iDEAL is ontwikkeld door de drie Nederlandse banken Rabobank, ING/Postbank en ABN AMRO. Sinds 2006 is het eigendom van Currence, de exploitant van andere digitale betaalmiddelen zoals PIN en Chipknip. De meeste grote banken in Nederland zijn inmiddels aangesloten bij iDEAL.

Webwinkelen via iDEAL biedt twee voordelen. Ten eerste kun je als consument online een product direct afrekenen via de vertrouwde website van je eigen bank. En ten tweede kan de webwinkel direct zien of je betaling is voldaan en daarmee de bestelling sneller afhandelen dan wanneer je

alleen maar via je eigen bank zou betalen, zonder dat er tegelijkertijd contact is met de bank van de webwinkel.
Bij een iDEAL-transactie zijn vier partijen betrokken: de klant, de webwinkel, de bank van de klant en de bank van de webwinkel. Vanuit informaticaoogpunt kunnen we drie belangrijke aspecten in iDEAL onderscheiden. Ten eerste hoeft de klant niet een apart stuk software te downloaden, maar kan hij gewoon zijn standaardbrowser gebruiken. iDEAL maakt in deze browsertoepassing veel gebruik van http-*redirecting*. Hierbij bevat een webpagina een instructie om automatisch naar een andere webpagina te gaan. Ten tweede moeten de vier partijen onderling op een veilige manier informatie kunnen uitwisselen. iDEAL-berichten zijn geschreven in XML, een van de standaardtalen voor webtoepassingen en belangrijke berichten worden elektronisch ondertekend. Ten derde moet de betaling ook nog eens voldoende snel gebeuren.

iDEAL in vogelvlucht

Stel nu dat je een product bij een webwinkel wilt kopen en aangeeft dat je wilt betalen met iDEAL. Nadat je hebt geselecteerd welke bank je wilt gebruiken, gaat er een betaalverzoek van de webwinkel naar de bank van de webwinkel. Dit betaalverzoek bevat het orderkenmerk, het bedrag en de terugkeer-URL van de webwinkel (het specifieke internetlabel dat verwijst naar de webwinkel). Het betaalverzoek wordt elektronisch ondertekend.

De stap van het betalen via je bank naar de bevestiging dat de webwinkel het product gaat leveren, duurt minder dan twee seconden.

Vervolgens vindt er een interbancair afhandelingsverzoek plaats tussen de bank van de webwinkel en jouw bank. Als alles goed gaat, geeft de bank van de webwinkel betaaltoestemming aan de webwinkel. In het bericht van de bank aan de webwinkel staat het orderkenmerk van de webwinkel, een transactie-ID van de bank van de webwinkel en ook een URL van de elektronische betalingstoepassing van jouw bank. Ook dit bericht wordt weer elektronisch ondertekend. iDEAL is zo ontworpen dat het in principe minder dan twee seconden duurt vanaf het moment dat je hebt gekozen voor het betalen via iDEAL tot en met deze toestemming.

Vervolgens stuurt de webwinkel je via http-*redirecting* automatisch door naar de gebruikelijke webtoepassing voor elektronisch bankieren bij je eigen bank. Je betaalt dan via de jou vertrouwde authenticatiemethode. Als je de betaling hebt afgerond, stuurt je eigen bank je vervolgens weer door naar de webwinkel. De webwinkel vraagt bij zijn bank of de betaling al is voldaan, een verzoek dat ook weer vergezeld gaat van een elektronische handtekening. Als het goed is, bevestigt de bank van de webwinkel dat de betaling is voldaan. Ook deze boodschap is weer elektronisch ondertekend. Na ontvangst van het betaalbericht ontvang je van de webwinkel de bevestiging dat je het product zult ontvangen.

De stap van het betalen via je bank naar de bevestiging dat de webwinkel het product gaat leveren, duurt ook weer minder dan twee seconden.

Elektronische handtekening

We hebben gezien dat voor het waarborgen van een veilige informatie-uitwisseling steeds een elektronische handtekening wordt gebruikt. Deze handtekening waarborgt de authenticiteit (het bericht is echt afkomstig van de vermelde afzender), de integriteit (het bericht is onderweg niet veranderd) en de onweerlegbaarheid (de verzender kan niet ontkennen dat hij het bericht heeft verzonden).

Het algoritme dat voor deze elektronische handtekening wordt gebruikt, is gebaseerd op zogeheten *RSA-cryptografie*. RSA gebruikt een publieke sleutel voor het versleutelen en een geheime sleutel voor het ontcijferen. Iedereen kan de publieke sleutel gebruiken om gecodeerde berichten te verzenden, maar alleen wie in het bezit is van de geheime sleutel kan het bericht ook ontcijferen. De veiligheid van RSA-cryptografie is gebaseerd op het feit dat het in de praktijk zeer onwaarschijnlijk is om twee priemgetallen p en q te achterhalen als $p \times q$ bekend is en p en q groot genoeg zijn, hoeveel huidige computerkracht je ook inzet. Hiervoor moeten p en q minstens uit 512 bits bestaan, maar tegenwoordig worden al vaak getallen van 1024 bits gebruikt om de kans op het kraken van de sleutel nog kleiner te maken.

Met dank aan Eric Verheul, hoogleraar informatieveiligheid aan de Radboud Universiteit Nijmegen en manager van de afdeling security & technology bij PricewaterhouseCoopers.

Digitaal touwtrekken tussen gemak en veiligheid

De elektronische stemmachine, de ov-chipkaart en het elektronische patiëntendossier werden overhaast ingevoerd. Veiligheid kreeg te weinig aandacht, wat tot grote problemen leidde. Met nieuwe informaticatechnieken kan de digitale veiligheid veel beter.

In het moderne leven identificeren we ons met loginnamen, toegangswoorden, pincodes, creditcardcijfers en bijbehorende veiligheidsnummers. Maar als deze in handen van criminelen vallen, kunnen zij eenvoudig misbruik maken van onze identiteit. Kenmerken van ons lichaam zoals gezicht, vingerafdruk, iris of stem, zijn veel moeilijker na te bootsen. Zelfs de manier waarop we bewegen, schrijven of op een toetsenbord tikken, verschilt van persoon tot persoon en kan dienen ter identificatie.

Wan Fokkink, hoogleraar aan de Vrije Universiteit in Amsterdam, leidde het BRICKS-onderzoek naar digitale veiligheid. Hij vertelt dat informatici van het Centrum Wiskunde & Informatica (CWI) en van de Radboud Universiteit Nijmegen hebben onderzocht op welke manier biometrische kenmerken het veiligst kunnen worden gebruikt voor identificatie. "Zij doen twee belangrijke aanbevelingen", zegt Fokkink. "Ten eerste concluderen ze dat het te kwetsbaar is om alleen maar één enkel biometrisch kenmerk aan te brengen in het paspoort." Mede op basis van dit advies

bevat het Nederlandse paspoort vanaf juni 2009 naast een biometrische scan van het gezicht ook een scan van twee vingerafdrukken.

Daarnaast concluderen ze ook dat er nu nog te veel alleen vanuit de gebruiker van de biometrische informatie wordt gedacht, de partij die de identiteit controleert. Informatie kan echter ook via deze zogenaamd betrouwbare partij naar buiten lekken. Fokkink: "Als dat gebeurt, dan kan een willekeurige vreemdeling zich toch voordoen alsof hij jou is. We hebben in het project een raamwerk ontworpen om ervoor te zorgen dat die biometrische gegevens niet zomaar naar buiten kunnen lekken. Centraal hierbij staat dat de afnemer van de biometrische informatie, bijvoorbeeld een centrale databank van de overheid, zich ook moet identificeren om toegang te krijgen tot die informatie."

Vertrouwen in plaats van wantrouwen

Naast biometrie als manier om je als persoon te identificeren, is het beveiligen van de toegang tot digitale informatie een ander belangrijk en actueel thema. Denk bijvoorbeeld aan het beveiligen van de toegang tot medische gegevens in ziekenhuizen. Als je in het ene ziekenhuis een röntgenfoto hebt laten maken en daarna naar een ander ziekenhuis gaat, dan wil je eigenlijk dat dezelfde röntgenfoto door het nieuwe ziekenhuis kan worden gebruikt.

Traditioneel is de beveiliging van dergelijke gegevens gebaseerd op wantrouwen. We vertrouwen iemand niet, tenzij hij het goede *password* intikt of het juiste pasje voor een scanner houdt. Soms is dat een handig systeem, maar niet altijd. Vooral als het gaat om uitwisseling van informatie in verschillende met elkaar verbonden organisaties blijkt dit in de praktijk juist belemmerend te werken. In zo'n heterogene omgeving, met een bonte verzameling software- en hardwareplatformen en verschillende lees- en schrijfpermissies op de diverse bestanden, blijkt een restrictief systeem moeilijk te implementeren.

"Een nieuwe kijk op digitale veiligheid", legt Fokkink uit, "gaat niet uit van wantrouwen, maar juist van vertrouwen. Het idee is dat je een systeem ontwerpt waarbij de toegangspoort weliswaar open staat, maar waarbij het systeem automatisch nagaat of mensen zich aan de vooraf afgesproken toegangsregels hebben gehouden. Een

Stemcomputer Nedap

controle achteraf dus. De toegangsregels specificeren bijvoorbeeld wie wel en niet toegang hebben, maar ook wie gegevens alleen mag inzien maar niet mag veranderen en wie de gegevens zowel mag inzien als veranderen."

Deze aanpak maakt het leven veel gemakkelijker en komt overeen met de manier waarop mensen in het dagelijks leven met elkaar omgaan. Mensen zijn verantwoordelijk voor hun eigen acties. Als blijkt dat ze de regels toch hebben overtreden dan volgen sancties. Fokkink: "Informatici van de Universiteit Twente hebben binnen ons veiligheidsproject een formeel bewijssysteem ontworpen dat achteraf vaststelt wie de regels wel of niet hebben overtreden. Dit zou een nieuw ontwerp kunnen zijn voor het elektronische patiëntendossier van de toekomst."

Lichtvaardig

"Terwijl digitale veiligheid en identificatie een steeds belangrijkere rol spelen in de maatschappij, is de politiek er tot nu toe veel te lichtvaardig mee opgesprongen", meent Fokkink. "Daarom is het misgegaan met de invoering van de stemmachine, de ov-chipkaart en het elektronisch patiëntendossier. Het geloof in computers is erg groot. Als de politiek niet inhoudelijk stuurt op het thema veiligheid, dan nemen de bedrijven die de stem-

Elektronisch stemmen: verifieerbaarheid bijt privacy

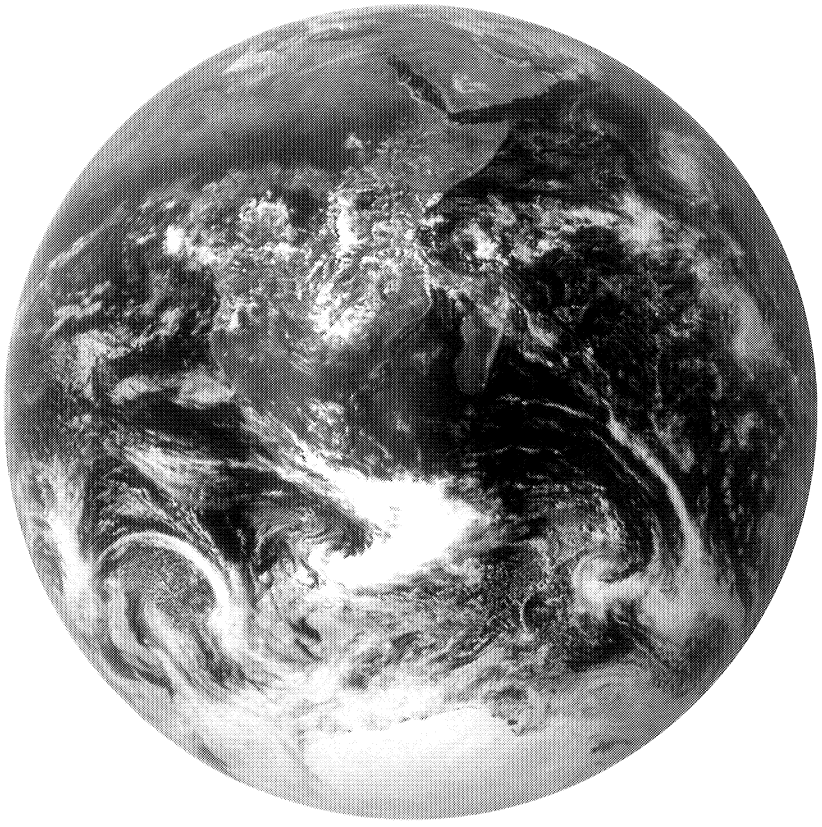
Bij het stemmen met potlood en papier is de stem anoniem en worden de stembiljetten met de hand geteld. Het aantal getelde stemmen zou door onzorgvuldig of opzettelijk verkeerd tellen kunnen afwijken van het echte aantal uitgebrachte stemmen. Maar de privacy van de stemmer is in principe gegarandeerd. Hoe zit dat bij elektronisch stemmen? Onderzoekers van de Technische Universiteit Eindhoven hebben een speciale logica ontwikkeld waarmee de privacy van een stemmer formeel kan worden geanalyseerd, bij een gegeven programmering van de elektronische stemmachine. Ze hebben op basis hiervan laten zien dat verificatie van stemaantallen en privacy van stemmers elkaar bijten. De reden voor dit conflict is dat de verificatie van een elektronische stemming in principe vereist dat elke uitgebrachte stem wordt gekoppeld aan de persoon die deze stem heeft uitgebracht – een systeem dat *verified vote* heet. Maar dan is de stem niet meer anoniem. Een kwaadwillende partij kan daardoor bijvoorbeeld nagaan op wie jij hebt gestemd. Dus kan een stemmer worden gechanteerd of hij kan zijn stem verkopen. Een grote uitdaging voor de toekomst wordt om een digitaal stemprotocol te ontwerpen dat zowel het aantal stemmen gegarandeerd juist telt, als de privacy van de stemmer voldoende waarborgt.

machine of de ov-chipkaart moeten ontwikkelen, het thema ook niet serieus, want ze worden er toch niet op afgerekend. Binnen BRICKS hebben we laten zien dat het veiliger kan, maar dan moet er eerst wel uitvoerig onderzoek plaatsvinden."

Fokkink vergelijkt het met de aanleg van de Afsluitdijk in de jaren dertig. Toen werd eerst jarenlang onderzoek gedaan naar alle mogelijke gevolgen van de afsluiting van de toenmalige Zuiderzee, onder leiding van Nobelprijswinnaar in de natuurkunde Hendrik Antoon Lorentz. Fokkink: "Het lijkt erop dat politici tegenwoordig snel resultaat willen laten zien, zonder grondig vooronderzoek. Net zo belangrijk als het wetenschappelijke werk, is het daarom om politici ervan bewust te maken dat digitale veiligheid niet zomaar komt aanwaaien." ◉

Hoe werkt Google Earth?

Tik 'New York' in op Google Earth en binnen enkele seconden vlieg je naar een luchtfoto van de *Big Apple* waarop je kunt in- en uitzoomen. Je kunt een wegenkaart over de luchtfoto leggen, je kunt bezienswaardigheden aanklikken en zelfs een realistische indruk krijgen van hoe de gebouwen er in drie dimensies uitzien. Google Earth heeft een geheel nieuwe dimensie toegevoegd aan de ouderwetse wereldatlas.
De kracht van Google Earth zit allereerst in de enorme hoeveelheid hoogkwalitatieve satelliet- en luchtfoto's die het bedrijf heeft gekocht. De tweede voorwaarde voor



haar succes is de ontwikkeling van een gebruikersvriendelijke interface. En ten derde heeft de relatief open architectuur de dienst een flinke steun in de rug gegeven. Zo kunnen andere partijen Google Earth voor hun eigen doelen gebruiken (zoals bijvoorbeeld het NOS-journaal doet) en er zelfs functionaliteiten aan toevoegen.

Ruimtelijke data

Hoewel de details van Google Earth bedrijfsgeheim zijn, valt er voldoende te zeggen over de belangrijkste informaticacomponenten van deze geografische dienst. Vooral het datamanagement en de snelheid van het ophalen van ruimtelijke gegevens zijn cruciaal.

Decennialang was databasemanagement gericht op alfanumerieke gegevens: het sorteren van getallen, woorden en letters. Maar satellietfoto's en luchtfoto's kun je niet op die manier sorteren. Hoe dan wel? Globaal gesproken op dezelfde manier als het in een atlas opzoeken waar je New York kunt vinden. Je legt een soort ruitjespatroon over de ruimtelijke data, houdt bij welke plaats in welk hokje ligt en maakt daarmee een index van waar je welke plek kunt vinden.

Google Earth gebruikt het 'ruitjespatroon' zoals gedefinieerd door lengte- en breedtegraden op de aardbol. Dan ontstaat echter het probleem dat er in het ene gebied een heleboel hokjes leeg zijn, zoals hokjes in de Atlantische Oceaan die alleen uit water 'bestaan', en dat er in een ander gebied hokjes liggen die juist veel relevante informatie bevatten, bijvoorbeeld een hokje dat over New York heen ligt. Dat maakt het zoeken inefficiënt. Informatici hebben diverse datastructuren en bijbehorende algoritmen verzonnen om dit probleem efficiënter op te lossen, onder andere de *Quadtree* en *R-tree* datastructuren.

Quadtree stelt bij elk hokje de vraag of het verder moet worden opgesplitst (in vier gelijke kleinere hokjes) of niet. Het opsplitsen gaat net zo lang door totdat het hokje niet meer te veel objecten bevat voor de opslag. De data worden zo uitgewerkt als een zich vertakkende zoekboom. Het nadeel is dat sommige takken van de boom meteen stoppen met groeien en dat andere takken heel lang blijven doorgroeien. Dat leidt tot een ongebalanceerde zoekboom, wat het efficiënt doorzoeken van de data belemmert.

R-tree is een alternatief algoritme dat wel zorgt voor een gebalanceerde zoekboom. R-tree gaat niet uit van de ruimte, maar van de objecten in deze ruimte, zoals een stad, een gebouw of een rivier. Die objecten worden geometrisch benaderd door bijvoorbeeld een punt, een lijn of een vlak. De truc van R-tree is dat objecten die geografisch dicht bij elkaar liggen in de zoekboom ook dicht bij elkaar terechtkomen.

Streamen

Het indexeren van de geografische data behoort tot de preprocessing. Vervolgens is de vraag hoe je zo snel kunt inzoomen op het beeld van New York nadat je de zoekterm hebt ingetikt. Als je de hele wereld

met een resolutie van een meter zou willen downloaden, dan ben je zelfs met een 10-megabit-per-seconde internetverbinding 69 jaar bezig. Dat schiet dus niet op. Google Earth haalt dan ook maar een heel klein stukje van al die informatie op. Hoe verder weg van het aardoppervlak je gezichtspunt, hoe lager de resolutie van dat stukje hoeft te zijn. Pas wanneer je gaat inzoomen wordt het stukje opgedeeld in nieuwe stukjes waarvan de gedetailleerde informatie wordt opgehaald.

Eigenlijk krijg je nog voordat het inzoomen begint al een heel grofkorrelige blik op New York te zien. Dit is niets meer dan een opgeblazen versie van de grotere kaart die al op je scherm stond, bijvoorbeeld een blik op het gehele Noord-Amerikaanse continent. Terwijl dit beeld op je scherm verschijnt, is het programma al bezig om nieuwe informatie te verzamelen en te versturen. Voor dit echte inzoomen wordt een aangepaste vorm van *streaming* gebruikt. Je computer ontvangt steeds meer details en gaat daarmee het grofkorrelige beeld van New York snel invullen. Zo herken je eerst de globale plattegrond van de stad en een paar seconden later verschijnen steeds meer details van wegen en gebouwen.

Naast het datamanagement en de aangepaste vorm van streaming bevat Google Earth nog veel meer informaticatoepassingen, zoals het omgaan met geografische projecties en het toevoegen van extra informatielagen op de satelliet en luchtbeelden (wegenkaarten, foto's, driedimensionale gebouwen, tekstuele informatie).

Je legt een soort ruitjespatroon over de ruimtelijke data, houdt bij welke plaats in welk hokje ligt en maakt daarmee een index van waar je welke plek kunt vinden.

Met dank aan Peter van Oosterom, hoogleraar GIS-technology aan de Technische Universiteit Delft

Nieuwe zoekmachine gidst gebruiker door beelduniversum

Woorden schieten vrijwel altijd tekort om beelden te beschrijven. Een nieuwe zoektechniek interpreteert de beeldinhoud en leidt de zoekopdrachtgever sneller naar een beter zoekresultaat in een grote hoeveelheid beelden.

Zoeken in grote hoeveelheden teksten is tegenwoordig gemakkelijk. Zoeken in grote hoeveelheden beelden staat daarentegen pas in de kinderschoenen. Wie nu via Google een foto zoekt van een eskimo die een iglo aan het bouwen is, vindt wel foto's, illustraties en cartoons van iglo's, maar niet van een iglo in aanbouw door een eskimo. Dat komt omdat Google nu alleen zoekt op grond van de teksten die bij de plaatjes staan en de namen die iemand aan de plaatjes heeft gegeven. Maar omdat een plaatje vaak meer zegt dan duizend woorden, is elke poging om een beeld in tekst te beschrijven gemankeerd. Daardoor blijft ook het zoeken in beeldcollecties via zoektermen nog behelpen.

De grote uitdaging is een zoekmachine te maken die zelf ziet wat er op een beeld staat, zoals mensen dat ook doen. Dat is precies wat Michael Lew van de Universiteit Leiden probeert, geïnspireerd door technieken


Inupiat Eskimo igloo
468 x 304 - 36 kB - jpg
alaska-in-pictures.com
Coloring Worksheet:
500 x 339 - 9 kB - gif
teach-nology.com
Building a real igloo
400 x 348 - 62 kB - jpg
jaspernationalpark.com
igloo-inside_large
640 x 480 - 37 kB - jpg
lifewithoutbuildings.net
igloo
640 x 512 - 51 kB - jpg
troop361.com
If you're looking for an
382 x 270 - 12 kB - jpg
alaska.net
Igloo.jpg
700 x 475 - 73 kB - jpg
commons.wikimedia.org
The Igloo
640 x 480 - 62 kB - jpg
richdeneault.com
Northern Lights Igloo
468 x 313 - 22 kB - jpg
travel-destination-pictu
Abandoned Igloo
1024 x 768 - 118 kB - jpg
bazos.com
an igloo hotel offers
500 x 332 - 78 kB - jpg
luxuo.com
Igloo Logo.jpg (38836
565 x 499 - 65 kB - gif
www3.sympatico.ca
Igloo
618 x 355 - 17 kB - jpg
maidsmusings.wordpress.com
Work has started on the
449 x 337 - 38 kB - jpg
emagine-travel.co.uk
igloo under northern
533 x 809 - 42 kB - jpg
trekearth.com


Two of the greatest words in
400 x 268 - 26 kB - jpg
qslsblogcast.blogspot.com
Morsecode voor eskimo's.
468 x 304 - 36 kB - jpg
absurd.nl
F-onduen in een Iglo met
512 x 349 - 117 kB
gva.be
A young Inuit girl
265 x 400 - 24 kB - jpg
bobmckenow.blogspot.com
van de Groenlandse
469 x 307 - 12 kB - jpg
urbanbeguide2.wordpress.com
Een Iglo
290 x 194 - 9 kB - jpg
proto.thinkquest.nl
Eco-Iglo antwoord op
120 x 120 - 4 kB - jpg
landvanenergie.nl
stay in an Inuit Igloo
451 x 300 - 127 kB - jpg
lawandmore.co.uk
Inuit Igloo
350 x 233 - 28 kB - jpg
nvernen.ca
sleeps in an Igloo and
349 x 233 - 55 kB - jpg
apps.exploratorium.edu
Any Eskimo's using Skype ?
980 x 735 - 299 kB
weblown.typepad.com
Eskimo's maken geen wak
800 x 501 - 101 kB
trompetoeuf.wordpress.com
Bouwkundig is de Iglo
531 x 392 - 17 kB - jpg
wolventand.eu
Eskimo en Iglo
640 x 480 - 63 kB
leenhuys.nl
Igloo betekent
240 x 160 - 9 kB - jpeg
bam3v1.googlepages.com


Inuit building igloo
450 x 315 - 52 kB - jpg
touchnote.com
An Inuit building an
550 x 265 - 37 kB - jpg
britannica.com
Inuit building igloo,
505 x 337 - 25 kB
gettyimages.com
Inuit building igloo
233 x 350 - 20 kB
photos.com
Inuit building igloo,
640 x 428 - 34 kB
gettyimages.com
Inuit building an
164 x 219 - 7 kB - jpg
worldtour.www8.50megs.com
An Inuit men can
800 x 463 - 144 kB - jpg
allthingsquebec
The building of an
400 x 260 - 26 kB - jpg
tslr.net
Caption: Eskimo build
478 x 350 - 46 kB - jpg
topfoto.co.uk
Eskimo building igloo
233 x 350 - 24 kB
photos.com
several Inuit
400 x 265 - 11 kB - jpg
windows.ucar.edu
Hunters could build an
394 x 300 - 71 kB - jpg
historyforkids.org
Eskimo Building an
634 x 480 - 65 kB
pro.corbis.com
Building Igloo
225 x 176 - 11 kB - jpg
takeoffleh.com
Top: Eskimo Igloo
381 x 517 - 43 kB - jpg
custance.org

uit de kunstmatige intelligentie. "Als het om zoeken in beelden gaat, vindt Google alleen het topje van de ijsberg", zegt Lew. "Omdat er zoveel beelden online staan, vindt de zoekmachine altijd wel iets, maar zeker als je iets specifieks zoekt, is de kans dat je het vindt veel te klein."

Ultieme bibliothecaris

Alle zoektechnieken die informatici tot nu toe hebben verzonnen om met een enkele zoekopdracht precies de gezochte beelden te vinden, hebben gefaald. Daarom hebben ze sinds een jaar of vijf het roer omgegooid. Voor de nieuwe zoekstrategie in beelden, staat de ultieme bibliothecaris model.

Lew: "De gebruiker tikt bijvoorbeeld in dat hij beelden van een iglo zoekt. De zoekmachine schotelt de gebruiker vervolgens een collectie van die beelden voor en vraagt de gebruiker verder te specificeren wat hij zoekt. De gebruiker kijkt naar de eerste zoekresultaten en selecteert het type beeld dat hij zoekt, bijvoorbeeld alleen iglo's die in aanbouw zijn. Via een dialoog van vragen en antwoorden leidt de zoekmachine de gebruiker door het beeldduniversum, tot de gebruiker heeft gevonden wat hij zoekt: een eskimo die een iglo bouwt. Precies zoals een goede bibliothecaris je in een bibliotheek helpt met zoeken. De kunst is om dat in zo min mogelijk vraag- en antwoordstappen voor elkaar te krijgen."

Informatici die zich met het zoeken in beeldcollecties bezighouden, zijn naarstig op zoek naar algoritmen die voor ultieme bibliothecaris kunnen spelen. Dat blijkt veel lastiger dan gedacht. Lew en zijn collega's hebben daarom het nieuwe concept van de 'kunstmatige verbeelding' verzonnen. Mensen kunnen zo goed in beelden zoeken, omdat ze in hun hoofd gemakkelijk associëren. Zo kan de mens bijvoorbeeld een paard en een hoorn denkbeeldig samenvoegen tot de niet-bestaande eenhoorn.

Een computer kan kunstmatige verbeelding krijgen door hem bestaande beelden te laten combineren tot nieuwe, kunstmatige beelden. Zo'n kunstmatig beeld is niet het uiteindelijke zoekresultaat, maar een plaatje dat bedoeld is om de gebruiker te helpen in de verfijning van zijn zoekopdracht, net zoals een bibliothecaris je een boek kan tonen en kan vragen: 'Zoekt u misschien iets in deze

Kunstmatige verbeelding

Om te zoeken in een grote verzameling beelden, gebruiken informatici algoritmen die elk plaatje in een reeks kenmerken vertalen. Die bevatten in ieder geval de drie dominante beeldkarakteristieken: kleur, vorm en textuur. Deze kenmerken worden wiskundig voorgesteld als een vector in een vectorruimte. Een realistische vector stelt al snel tussen honderd en duizend kenmerken voor. Elk plaatje kun je dan voorstellen als een punt in een honderd- of duizenddimensionale vectorruimte.
Wanneer je een database van bijvoorbeeld tienduizend plaatjes in deze honderddimensionale ruimte gaat beschrijven, dan wordt de ruimte maar heel sporadisch gevuld: gemiddeld slechts honderd plaatjes per dimensie. Dat levert een groot probleem op bij het zoeken. Stel, je geeft een zoekopdracht. Het algoritme vertaalt de zoekopdracht naar een specifiek punt in de vectorruimte. Omdat de vectorruimte zo leeg is, is de kans echter groot dat er geen corresponderend plaatje bij het gezochte punt hoort.
Via het idee van de kunstmatige verbeelding, kan de computer zelf een plaatje creëren dat wel bij dat punt hoort. Een van de manieren die Michael Lew heeft gebruikt, is om een verzameling van honderd miljoen foto's van het internet te downloaden en die toe te voegen aan de broncollectie. De eerst vrij lege vectorruimte, alleen bevolkt door foto's uit de broncollectie, wordt zo kunstmatig opgevuld. Het algoritme voor de kunstmatige verbeelding neemt vervolgens enkele plaatjes die dicht in de buurt liggen van het gezochte punt en combineert deze tot een kunstmatig beeld. Dit kunstmatige beeld kan aan de gebruiker worden voorgelegd en verfijnt de oorspronkelijke zoekopdracht.

richting?' Welke plaatjes het algoritme combineert en hoe dat rekenkundig gebeurt, hangt af van de gekozen techniek (zie kader).

Google te snel af

Lew heeft het concept van de kunstmatige verbeelding in de praktijk getest: "Daaruit blijkt dat we het aantal stappen om tot een goed zoekresultaat te komen met minstens de helft reduceren ten opzichte van concurrerende algoritmen uit de wetenschappelijke literatuur."
De resultaten zijn zo veelbelovend dat de onderzoekers nu de laatste hand leggen aan een voor iedereen toegankelijke beeldzoekmachine, met het rekenhart in Leiden.

"Als dat ons lukt", zegt Lew, "dan hebben we de Nederlandse informatica op een mooie manier op de wereldkaart gezet. Wij zijn al blij als we een werkend prototype kunnen presenteren. Dan hebben we laten zien dat onze algoritmen goed werken. Daarna is het aan commerciële partijen als Google of Microsoft om dit type zoekmachine verder te optimaliseren. Hij moet in een zo groot mogelijke database kunnen zoeken en door een grote hoeveelheid mensen tegelijk en snel gebruikt kunnen worden."

Uiteindelijk moet de zoekmachine ook in collecties van tekeningen, schilderijen en medische beelden kunnen zoeken. Ook zou het handig zijn als de invoer van de zoekopdracht kan variëren van tekst, tot tekeningen en foto's. Zo zou de politie bijvoorbeeld een tekening van een verdachte als zoekopdracht aan een computer kunnen geven. De computer kan dan in een fotodatabase van eerder opgepakte criminelen gaan zoeken of er een foto is die sterk op de tekening lijkt. ●

Automatische borstkankerdetectie

Bij borstkankerscreening missen radiologen nu ongeveer een kwart van de kankergevallen. Slimme software moet helpen om dit aantal terug te brengen.

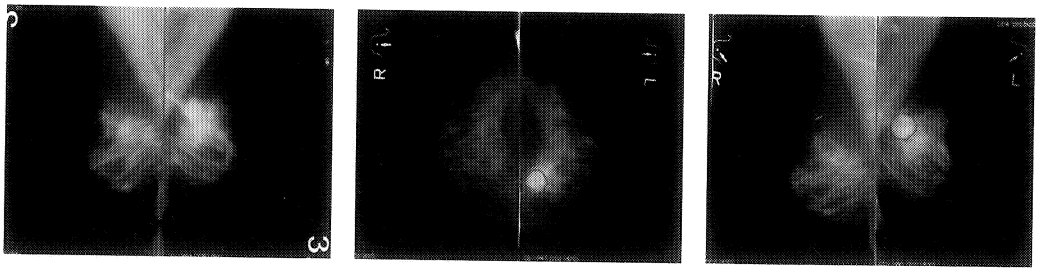
De zangeressen Ellen ten Damme en Kylie Minogue zijn twee bekende jonge vrouwen waarbij borstkanker werd ontdekt. Een op de negen vrouwen ontwikkelt in haar leven borstkanker. Hoe ouder, hoe groter de kans. Gelukkig hoeft borstkanker niet dodelijk te zijn, als het maar vroeg genoeg wordt gedetecteerd. Om die reden ontvangen in Nederland jaarlijks een miljoen vrouwen tussen de 50 en 75 een oproep voor een borstkankerscreening.

Steeds meer screenings gebeuren digitaal en vanaf 2010 moeten alle screenings digitaal gaan gebeuren. Maar digitaal of niet, de radiologen die de beelden afspeuren op de aanwezigheid van borstkanker missen gemiddeld ongeveer een kwart, ondanks hun grote ervaring.

Getrainde computer

Postdoc Marina Velikova werkt bij het Medisch Centrum van de Radboud Universiteit Nijmegen aan de ontwikkeling van slimme software die de radioloog moet helpen bij de vroege detectie van borstkanker. "De computer zal de radioloog niet gaan vervangen, maar hem wel helpen bij de interpretatie van wat hij ziet", zegt Velikova. "Het grote voordeel van de computer, is dat je hem kunt

trainen met veel meer voorbeelden van borstkanker dan een radioloog ooit in zijn leven te zien krijgt. Die training willen we gebruiken om de vroege detectie van borstkanker te verbeteren."



Mammogrammen van een linker- en rechterborst in bovenaanzicht en zijaanzicht. In het rood omcirkelde gebied zit mogelijk een tumor.

Borstkankerscreening gebeurt met een röntgenfoto van een borst, een mammogram. Verschillende weefsels absorberen de röntgenstraling in verschillende mate, zodat de foto een verzameling witte, grijze en zwarte structuren toont. Als borstkanker er op alle mammogrammen altijd hetzelfde zou uitzien, zou de detectie gemakkelijk zijn, maar dat is nu juist niet het geval. Een tweede complicatie is dat de structuur van de borst varieert van vrouw tot vrouw. Een derde moeilijkheid is dat die structuur ook met het klimmen van de jaren verandert.

Om de detectie van borstkanker te verbeteren, worden vaak twee mammogrammen van elke borst genomen: eentje in bovenaanzicht en eentje in zijaanzicht. Elk digitaal mammogram bestaat uit zo'n tienduizend pixels, allemaal met een eigen grijswaarde. Om de computer te leren borstkanker in een vroeg stadium te herkennen, wordt een hele serie aan eigenschappen berekend van gebiedjes waarvan biopsie al heeft aangetoond dat er kanker zit: onder andere de vorm, de grijswaarde, de grootte, het contrast met de omgeving en de architectuur van de draadachtige structuren die een gebiedje kunnen omgeven. Vervolgens kun je de computer leren welke gebiedjes hij als verdacht moet omcirkelen.

Expertkennis

Nu ziet het zijaanzicht van een borst er anders uit dan het bovenaanzicht. Je weet niet meteen welk gebiedje

in het ene aanzicht overeenkomt met welk gebiedje in het andere aanzicht. De radioloog is getraind om beide aanzichten te combineren en één enkel oordeel te vellen. Maar de computer kan dat in beginsel niet. Hij heeft alleen maar weet van pixels met een bepaalde grijswaarde. Toch heeft Velikova samen met haar collega's de computer geleerd om net als de radioloog uit beide aanzichten één conclusie te trekken: Zit er kanker in of niet? Velikova: "Omdat ik kansmodellen gebruik, is de computer zelfs nog specifieker. Hij vertelt wat de kans is dat een vrouw borstkanker heeft. Verder geeft de computer ook aan waarom hij tot zijn conclusie komt. Dan kan de radioloog vervolgens op grond van zijn eigen ervaring kijken of hij zich daar in kan vinden."

Lerend netwerk van kankerkansen

Hoe kun je de computer leren om één diagnose te stellen op grond van twee verschillende aanzichten van dezelfde borst? Dat kan door eerst in kaart te brengen wat de verdachte gebiedjes in elk aanzicht zijn. Vervolgens bepaal je wat de kans is dat een verdacht gebiedje ook echt kanker bevat. En ten slotte combineer je de kansen van beide aanzichten als het ware tot een enkele kans, die vertelt of een vrouw borstkanker heeft.

Elk verdacht gebiedje in het bovenaanzicht wordt met een pijl verbonden aan elk verdacht gebiedje in het zijaanzicht. Bij elke pijl komt een kans te staan die aangeeft dat een van beide gebiedjes kanker bevat. Maar waarom zou je gebiedje A in het ene aanzicht verbinden met een ander gebiedje B in het andere aanzicht? Die hebben toch niets met elkaar te maken? Dat is een slimmigheid om de mogelijkheid mee te nemen dat kanker in het ene gebiedje zich wel degelijk kan verraden door een afwijkende structuur van een ander gebiedje in dezelfde borst. Zo ontstaat een pijlenmodel van verdachte gebiedjes en kansen op kanker: een zogeheten *Bayesiaans netwerkmodel* gebaseerd op Bayesiaanse statistiek. Dit type statistiek is een systematische manier om te berekenen hoe de kans op een bepaalde gebeurtenis verandert wanneer nieuwe informatie aan het licht komt.

De kansen die bij de pijlen horen, worden berekend door het Bayesiaanse netwerk te trainen met een dataset van duizenden mammogrammen. Velikova trok een heel jaar uit om van de radiologen te leren hoe je mammogrammen interpreteert. Die expertkennis vertaalde ze vervolgens in een computermodel en dat model traint ze via een dataset van mammogrammen met en zonder kanker.

Samen met radiologen test Velikova hoe het systeem werkt en welke specifieke expertkennis van de radiologen de computer ook moet meenemen. Een voorbeeld van die expertkennis is dat de plek van de kanker vaak omringd is door een groeiende sterachtige structuur.

Vrouwen boven de vijftig worden elke twee jaar voor een screening opgeroepen. Als volgende stap wil Velikova de computer dan ook leren om vandaag gemaakte mammogrammen automatisch te vergelijken met jaren eerder gemaakte mammogrammen. Geen sinecure, want de structuur van de borst verandert met de tijd.

Het zal nog wel enkele jaren duren eer de methode in de praktijk van borstkankerscreening wordt gebruikt. Nu nog beoordelen altijd twee radiologen onafhankelijk van elkaar een mammogram. Het zou al een hele winst zijn als het computeroordeel net zo goed wordt als het oordeel van de radioloog. In dat geval kan de computer een van de twee radiologen vervangen. "En dat niveau beginnen we nu te benaderen", besluit Velikova. "Met de computeranalyse als goede *second opinion* kunnen we in de toekomst levens redden." ◉

Hoe werkt intelligent cameratoezicht?

Cameratoezicht in de openbare ruimte is in de afgelopen jaren steeds meer opgerukt: in winkels, op straat, in parkeergarages en in voetbalstadions. Met deze toename is ook de behoefte aan intelligent cameratoezicht toegenomen, waarbij niet een mens maar een computer de beelden automatisch interpreteert. Nu is het menselijke vermogen om beelden te interpreteren in miljoenen jaren geëvolueerd tot een uiterst snel en efficiënt waarnemingssysteem. Maar voor een computer is het interpreteren van een beeld een van de moeilijkste uitdagingen die er zijn – nog moeilijker dan het analyseren van schrift, geluid en spraak. In essentie komt dat doordat hetzelfde voorwerp er voor een computer onder een andere hoek en bij een andere belichting heel anders uitziet. Nog moeilijker wordt het als voorwerpen ook nog in het beeld bewegen.

Gezichtsherkenning

Stel dat een intelligent camerasysteem een mensenmenigte in de gaten houdt en naar personen moet zoeken die in een database met verdachten voorkomen. Dan moet de camera eerst gezichten onderscheiden van de omgeving en erop inzoomen. Dat is een taak die de computer nog vrij gemakkelijk uitvoert op basis van het herkennen van een ovale vorm met twee ogen en een neus. Maar daarna wordt het moeilijker: hoe komt de gezichtsherkenningssoftware erachter of het gezicht in de bestaande database voorkomt?

Traditionele tweedimensionale gezichtsherkenning gaat ervan uit dat het gezicht recht in de camera kijkt en dat de belichting nauwelijks afwijkt van die van de foto in de database. Tegenwoordig wordt er hard gewerkt aan driedimensionale gezichtsherkenning waarbij het gezicht niet

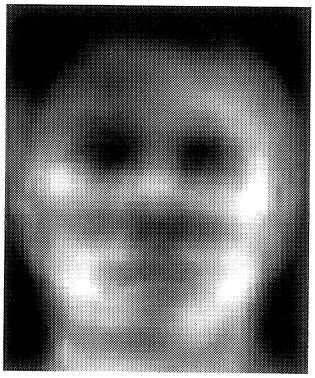
Je kunt elk gezicht uitdrukken in een combinatie van basisvormen die in alle menselijke gezichten voorkomen.

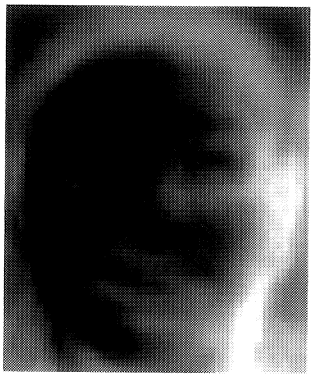
langer recht in de camera hoeft te kijken. Zowel bij de twee- als bij de driedimensionale gezichtsherkenning moet het gezicht worden uitgedrukt in getalsmatige kenmerken, zoals de afstand tussen de ogen, de breedte van de neus, de diepte van de oogbollen, de vorm van de jukbeenderen en de lengte van de kaaklijn – allemaal kenmerken die bij volwassenen in de tijd nauwelijks veranderen. Deze getallen worden samengevoegd in een unieke numerieke code: de digitale gezichtsafdruk. Haarstijlen, baarden, snorren en brillen tellen trouwens niet mee, omdat die in de tijd nogal veranderen.

Eigenfaces

Voor het genereren van de digitale gezichtsafdruk bestaan talloze algoritmen. Een van de meest toegepaste is gebaseerd op het idee dat je elk gezicht kunt uitdrukken in een combinatie van basisvormen die in alle menselijke gezichten voorkomen (*eigenfaces*). Net zoals je elk punt in de driedimensionale ruimte kunt beschrijven in termen van drie eigenvectoren (in x-, y- en z-richting), kun je ook proberen om gezichten uit te drukken in termen van een optelsom van eigenfaces. Bij de ene persoon is de ene basisvorm sterker aanwezig dan de andere, vandaar dat de som van eigenfaces een gewogen som moet zijn, die verschillende gewichten toekent aan verschillende basisvormen.

De eigenfaces kun je construeren uit een grote verzameling gedigitaliseerde gezichten. Omdat de afstand tussen de ogen onderling en die tussen de horizontale ooglijn en de mond van persoon tot persoon verschillen, worden alle gedigitaliseerde gezichten eerst als het ware in dezelfde mal geperst waarin deze afstanden kunstmatig gelijk worden getrokken. Dan vallen de ogen en monden in alle foto's over elkaar heen. Vervolgens kun je met de wiskundige techniek van de *principal component analyse* de eigenfaces uit de database bepalen. Praktische toepassingen werken meestal met honderd tot tweehonderd eigenfaces. Vervolgens kun





je elke digitale foto van een gezicht uitdrukken in een gewogen combinatie van eigenfaces.

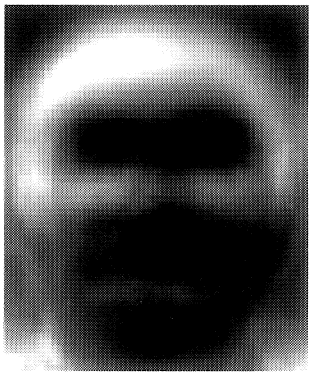
Identificatie

Hoewel deze methode snel is, werkt hij niet goed onder verschillende aangezichten en verschillende belichtingen. Sinds de eerste toepassing van eigenfaces voor gezichtsherkenning in 1991 zijn daarom talloze verbeterde versies ontwikkeld. Maar aan de basis van veel commercieel verkrijgbare gezichtsherkenningssoftware staat vaak nog steeds de eigenface-methode.

Wil je nu weten of een willekeurig persoon in een database voorkomt (identificatie), dan moet je in een score uitdrukken in welke mate het gezicht van die persoon overeenkomt met foto's in de database. Wil je weten of iemand de persoon is voor wie hij zich uitgeeft (verificatie), dan hoef je alleen maar een een-op-eenvergelijking te maken tussen de huidige gezichtsopname en die uit een database. De softwaregebruiker bepaalt zelf hoe hoog de score moet zijn om te gelden als identificatie of verificatie.

Hoe goed presteert deze gezichtsherkenningssoftware? Volgens het Amerikaanse National Institute of Standards and Technology identificeert de beste software 0,1 procent van de gezichten ten onrechte, terwijl de software tussen 1,0 en 2,5 procent van de gezichten juist niet herkent terwijl ze wel in de database voorkomen. Maar deze percentages gelden alleen voor niet bewegende gezichten, die recht in de camera kijken, goed belicht zijn en een neutrale gezichtsuitdrukking hebben. Volautomatische *realtime* gezichtsherkenning van bewegende personen in de publieke ruimte is daar nog ver van vandaan. Vooralsnog dient intelligent cameratoezicht daarom vooral als een handige aanvulling op cameratoezicht door mensen. Het grote voordeel is dat de software niet in aandacht verslapt en de mens wel.

Eigenfaces: enkele basisvormen die in menselijke gezichten voorkomen

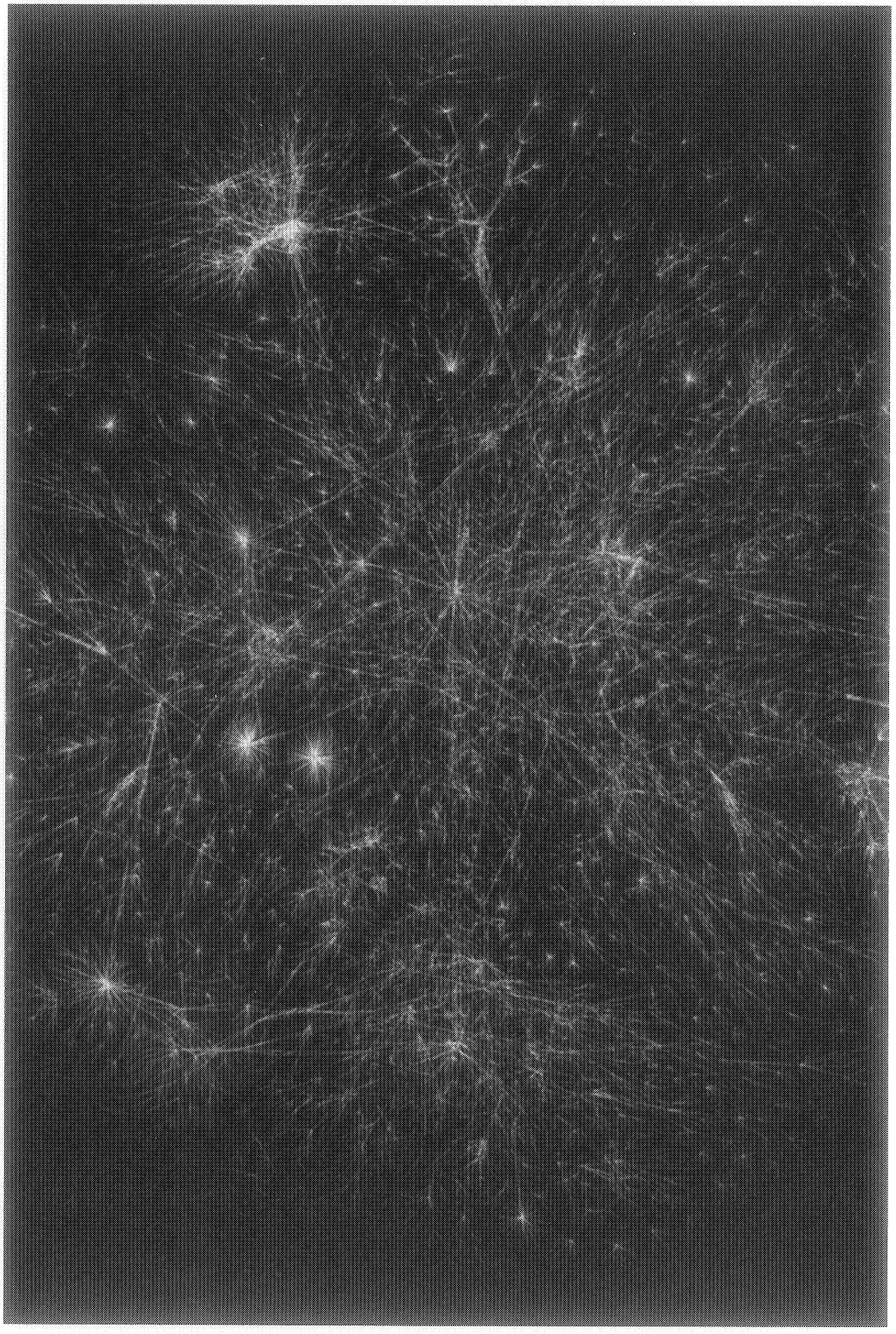


Betere postbezorging van datapakketjes

Een nieuwe wiskundige analyse van het internetprotocol dat datapakketjes rondstuurt, moet ervoor zorgen dat ons internetverkeer vloeiender verloopt.

Of je nu een e-mail verstuurt, rondsurft op het internet, of een film downloadt, je wilt dat de digitale bestanden die je opvraagt zo goed mogelijk van de digitale snelweg op je computer aankomen. Het protocol dat daarvoor automatisch zorgt, heet het *transmission control protocol* (TCP). TCP zit ingebouwd in elke computer en elk tussenstation waarlangs het internetverkeer loopt. Het is als het ware de ruggengraat van internet. De internationale non-profitorganisatie ICANN (Internet Corporation for Assigned Names and Numbers) heeft deze standaard voor internet afgesproken en beoordeelt ook voorstellen voor verbetering.

Het protocol zorgt ervoor dat alle gegevens die worden verstuurd, worden opgeknipt in pakketjes. Elk pakketje krijgt een etiket waarop staat waar het vandaan komt en waar het naartoe moet. Vervolgens worden de pakketjes als in een estafette van het ene naar het andere tussenstation overgedragen tot ze uiteindelijk bij de ontvanger aankomen. Pakketjes van hetzelfde bestand kunnen ook via verschillende wegen uiteindelijk bij de ontvanger aankomen. Helaas kunnen er onderweg pakketjes



kwijtraken of beschadigd raken. Dat gebeurt vooral op internetknooppunten waar het heel druk is.

YouTube

TCP is ontworpen om te zorgen dat de gebruiker ondanks mogelijke problemen onderweg toch de juiste informatie ontvangt. Daarvoor gebruikt het protocol een denkbeeldig raam waarin het bekijkt of het vaste aantal pakketjes dat het door het raam kan zien, wel of niet goed zijn ontvangen. En dat gebeurt op elk tussenstation. Stel dat tussenstation D door zijn raam bijvoorbeeld de pakketjes 11 tot en met 16 bekijkt. En stel dat het constateert dat 11 tot en met 14 goed zijn ontvangen, dat er iets mis is met 15 en dat 16 helemaal niet is ontvangen. Dan meldt tussenstation D aan het vorige tussenstation C dat het nog een keer de pakketjes 15 en 16 moet versturen en dat het tegelijk de nieuwe pakketjes 20 tot en met 23 kan meesturen. Zo gaat het doorsturen van pakketjes verder totdat alle pakketjes goed bij de ontvanger zijn aangekomen.

Nu zijn de mogelijkheden van het internet in de loop van de tijd veranderd. Toen mensen massaal YouTube-filmpjes gingen bekijken, werd het bijvoorbeeld belangrijker dat je elk filmpje kunt blijven kijken zonder dat het blijft hangen, dan dat precies elk datapakketje ook echt aankomt. Liever dat een datapakketje kwijtraakt, dan dat je filmpje steeds blijft hangen. Maar bij het verzenden van data is het belangrijker dat alle data aankomen dan dat de datastroom tijdelijk blijft hangen.

"In de loop van de tijd", vertelt professor Jos Baeten van de Technische Universiteit Eindhoven, "is het oorspronkelijke TCP-protocol daarom steeds meer vervuild met nieuwe eisen. Het is steeds ingewikkelder geworden. Zeker met de opkomst van het bekijken van filmpjes merk je af en toe dat TCP begint te kraken in zijn voegen. Wij willen TCP met nieuwe wiskundige hulpmiddelen analyseren, om te onderzoeken hoe het protocol kan worden verbeterd."

Die wiskundige hulpmiddelen komen uit de procesalgebra, een wiskundige taal waarmee je kunt redeneren over wat er gebeurt in een willekeurig systeem met aan elkaar gekoppelde processoren of computers. De algebra beschrijft hoe toestanden in zo'n systeem veranderen. Een typische opdracht in de procesalgebra luidt bijvoorbeeld: "Stuur over kanaal C een 0 en ga dan verder met x en doe

Gedeeltelijke kaart van de internetverbindingen op 15 januari 2005 (p. 64)

Stochastische procesalgebra

Bestaande stochastische procesalgebra's nemen aan dat de kansverdeling van een bepaalde gebeurtenis verloopt volgens een negatieve e-macht. Bovendien nemen ze aan dat je weet wanneer twee gebeurtenissen tegelijk optreden. Markovski en Baeten ontwikkelden een algemenere stochastische procesalgebra waarin de kansverdeling niet langer een negatieve e-macht hoeft te zijn en waarin je niet van tevoren weet wanneer twee processen tegelijk optreden. Hiermee analyseerden ze bijvoorbeeld hoe de bezetting van datakanaal K afhangt van de onbetrouwbaarheid van K en van de onbetrouwbaarheid van het terugmeldkanaal L. Langs kanaal K komen de data binnen en langs kanaal L wordt teruggemeld aan het vorige tussenstation welke datapakketjes wel of niet goed zijn aangekomen. Hoe vaak het pakketje kwijtraakt, wordt aangegeven door een stochastische grootheid. Het resultaat van de analyse geven de onderzoekers weer in een driedimensionale grafiek, die eruitziet als een berglandschap voor de bezettingsgraad. De top in het berglandschap leert je vervolgens welke betrouwbaarheden voor de communicatiekanalen je moet realiseren voor een optimale bezetting van communicatiekanaal K.

dat in parallel met het ontvangen over C van een o en ga dan verder met y."

Datarace

"Procesalgebra bestaat sinds eind jaren zeventig", vertelt Baeten. "Wat wij hebben gedaan is een uitbreiding ervan met stochastische variabelen. Dat zijn variabelen die op elk tijdstip een gebeurtenis met een bepaalde kans laten gebeuren, bijvoorbeeld het moment waarop je in een wachtrij aan de beurt komt. Een berucht probleem ontstaat wanneer een protocol met meerdere stochastische variabelen tegelijk te maken heeft. Er is een bepaalde kans dat A gebeurt en tegelijk een bepaalde kans dat B gebeurt. Vaak is een systeem ontworpen met het idee dat A altijd voor B gebeurt, maar toch bestaat er een kans dat B voor A komt. Nu kan het in de praktijk 99 keer goed gaan, maar de honderdste keer kan het ineens fout gaan. Het is onvoorspelbaar wanneer dat gebeurt. Dat noemen we de race-

conditie, omdat het gaat om een soort race tussen twee of meer dingen die tegelijk kunnen gebeuren."

Promovendus Jasen Markovski heeft samen met Baeten een stochastische procesalgebra ontwikkeld waarmee ze kunnen analyseren hoe een protocol de mist in kan gaan bij een race-conditie (zie kader). Met deze nieuwe procesalgebra kunnen ze wiskundig bewijzen of een protocol wel of niet een race-conditie bevat. Daarnaast kunnen ze ook kwantitatief evalueren hoe goed een protocol werkt. "Helaas zijn we uiteindelijk niet toegekomen aan het toepassen van onze algebra op TCP", vertelt Baeten. "Maar we hebben het succesvol gedemonstreerd bij een eenvoudiger protocol, dat een soort uitgeklede versie is van TCP. In een volgend project gaan we het toepassen op TCP. We hopen dat we daar voorstellen voor verbetering kunnen uithalen. TCP moet echt beter kunnen dan de huidige versie." ◉

Nooit meer wachten op drukke websites

Google.com, Amazon.com, nu.nl, nos.nl – alle drukbezochte websites kunnen met nieuwe wiskundige strategieën de wachtrijen voor hun websites flink verkorten.

Stel, er vindt een terroristische aanslag plaats. Het nieuws verspreidt zich als een lopend vuurtje. Mensen gaan ineens massaal nieuwswebsites checken. Hongerig naar beeld en geluid, klikken ze op video- en audiofragmenten. Binnen de kortste keren raken de servers van de nieuwssites overbelast. Niemand krijgt meer de gevraagde informatie. Dit is het gevolg van de naïeve manier waarop de wachtende internetter nu wordt bediend.

Het kan veel beter, zo heeft promovenda Wemke van der Weij van het Centrum Wiskunde & Informatica (CWI) in haar proefschrift laten zien. Zij vond wiskundige strategieën die wachtrijen voor drukbezette webservers met meer dan de helft kunnen verkorten. Pure wiskunde, waarvan de toepassing volledig informatica is. Van der Weij promoveerde begin 2009 aan de Vrije Universiteit Amsterdam.

Laadbalkje

Een willekeurige webpagina bestaat vaak uit zowel tekst en plaatjes als audio- en videofragmenten. Om al dat materiaal op je scherm te krijgen, moet een processor ze uit diverse databases ophalen van de websiteaanbieder. Dat ophaalproces verloopt meestal over meerdere webser-

vers. Terwijl de gevraagde informatie wordt opgehaald, staar je als gebruiker van een drukke website ongeduldig naar een zandloper, een laadbalkje of een soort klokje. Hoe lang je moet wachten, wordt bepaald door hoe slim de informatie wordt opgehaald en op je scherm wordt aangeboden.

De tegenwoordige strategie is meestal nogal naïef. Ofwel iedereen wordt tegelijk bediend, ofwel de bediening gebeurt een voor een. Als iedereen tegelijk wordt bediend, zoals bij nieuwssites, dan krijgen alle bezoekers een klein deel van de capaciteit. Maar dat gaat mis wanneer veel mensen tegelijk dezelfde website bezoeken. Als mensen een voor een worden bediend, dan heb je pech als je iemand voor je hebt die een lange video wil downloaden, terwijl je zelf alleen de tekst van een nieuwsbericht wilt lezen. Dezelfde pech als wanneer voor je in de supermarktrij iemand met een gevulde winkelwagen staat, terwijl je zelf alleen maar drie producten wilt afrekenen.

De supermarkt kan de wachtrijen verkorten door bijvoorbeeld een rij te maken voor alleen mensen met een winkelwagen en een rij voor mensen met alleen een mandje. Iets soortgelijks kan ook bij wachten op websites. Iemand die een film downloadt, houdt zelf al rekening met een flinke wachttijd. Hij kan best iets langer wachten en daarmee iemand die alleen maar koppen van artikelen op de site leest laten voorgaan. Toch wordt daar in de praktijk nog geen rekening mee gehouden.

"Tussen het een voor een bedienen van klanten en het allemaal tegelijk bedienen, zitten nog veel andere smaken", zegt Van der Weij. "Je kunt bijvoorbeeld een bepaalde capaciteit reserveren voor moeilijke klanten. Of je kunt trouwe klanten voorrang geven."

Slager helpt bakker

Het is dit soort strategieën die Van der Weij wiskundig heeft geanalyseerd. Pionierswerk, want hoe alomtegenwoordig webdiensten ook zijn, er was nog nauwelijks onderzocht hoe de wachtrijen van webservers zo kort mogelijk kunnen worden. Dat komt omdat wachtrijen van webservers anders zijn dan klassieke wachtrijen, zoals die in de supermarkt. Bij de klassieke wachtrijen is het aantal klanten dat in de ene rij wordt geholpen onafhankelijk

van het aantal klanten dat in andere rijen wordt geholpen. Bij webservers werkt het anders. De totale bedieningscapaciteit wordt hier dynamisch verdeeld over de verschillende wachtrijen. "Het is alsof de slager in een grote supermarkt de bakker gaat helpen als deze het ineens heel druk heeft", legt Van der Weij uit.

De promovenda leidde voor een algemeen wachtrijprobleem met gedeelde capaciteit een wiskundige formule af die tot de kortste wachttijden leidt. Hoeveel korter de wachttijd wordt, hangt sterk af van de specifieke toepassing. Er blijkt vooral veel te winnen wanneer de bezoekers een uiteenlopend beroep doen op de bedieningscapaciteit: de ene bezoeker wil veel informatie, de ander gemiddeld en weer een ander wil weinig. "Voor een eenvoudig voorbeeld hebben we in een testomgeving laten zien dat de wachttijd gehalveerd wordt", aldus Van der Weij. "Maar in werkelijkheid verwacht ik dat de wachttijd nog korter wordt, omdat er in ingewikkeldere gevallen meer te winnen valt."

In principe kunnen haar resultaten voor alle webtoepassingen worden gebruikt. "Maar", voegt Van der Weij eraan toe, "vooral websites die soms plat gaan door

Kansverdelingen van klanten

De wiskundige technieken die Van der Weij heeft gebruikt om wachtrijen voor webservers te minimaliseren, zijn gebaseerd op statistiek. Aan de basis staat het inzicht in hoe klanten de website gebruiken. Klanten komen volgens een bepaalde kansverdeling op een bepaald moment naar de website. Een tweede kansverdeling geeft aan wat de kans is dat een klant een bepaalde bedieningstijd vraagt. Is hij bijvoorbeeld een korte bezoeker die alleen de koppen van de nieuwsartikelen scant, of is hij iemand die uitgebreide videofragmenten gaat bekijken? Om de wachttijd zo kort mogelijk te houden, moet de eigenaar van de website beschikken over dit soort kansverdelingen, die over een lange tijd in de praktijk zijn bepaald.

Als deze twee kansverdelingen bekend zijn, kun je in een simulatie testen hoe goed een bepaalde wachtrijstrategie werkt. Eerst trek je een kans dat er op een bepaald moment een bepaald aantal klanten aankomt. Vervolgens trek je voor elke klant een kans op een bepaalde bedieningstijd. Deze gegevens stop je volgens de optimale strategie in een denkbeeldige webserver en dan kun je in de simulatie uitrekenen hoeveel korter de wachttijden worden, vergeleken met bestaande strategieën. Zo liet Van der Weij zien dat de strategie waarvan ze wiskundig al had bewezen dat het de beste is, ook in de praktijk veel beter werkt dan bestaande strategieën.

een te grote toestroom van bezoekers en websites vanwaar je films kunt downloaden, kunnen de wachttijden flink verkorten." Of bedrijven als Google en Amazon al een exemplaar van haar proefschrift hebben aangevraagd en de resultaten al gaan toepassen? "Dat weet ik niet, maar ik weet wel zeker dat als drukbezochte websites mijn verbeterstrategieën oppikken, ze hun servercapaciteiten beter kunnen benutten en minder servers nodig hebben." ◉

Hoe werkt televisie kijken via je mobiele telefoon?

Een YouTube-filmpje of een journaaluitzending op je mobiele telefoon bekijken is al vrij gewoon. Dat is mogelijk via zogeheten *streaming*-technologie. De filmbeelden worden niet eerst gedownload, maar direct afgespeeld zonder dat ze worden bewaard op je telefoon. Als jij aangeeft dat je een bepaald filmpje wilt bekijken, worden de beelden eerst via het internet verspreid naar het mobiele telefoonnetwerk of een Wi-Fi-verbinding. Van daaruit komen de beelden op je telefoon terecht.

Een stap verder en technisch gezien veel moeilijker is televisie kijken via je mobiele telefoon. Dat gebeurt niet met streamingtechnologie (waarbij de beelden maar naar één persoon worden gestuurd), maar met *broadcasting*-technologie (waarbij de beelden naar veel mensen tegelijk worden gestuurd). Een satelliet stuurt de signalen van alle aangesloten tv-kanalen naar de aarde. Zendmasten van mobiele telefoonaanbieders sturen vervolgens continu tv-beelden van deze kanalen de lucht in en op een geschikte mobiele telefoon kies je dan naar welke zender je wilt kijken. Je mobiele telefoon gebruikt dezelfde ingebouwde radio-ontvanger die hij ook gebruikt voor telefoon- en dataverkeer om tv-beelden te ontvangen, alleen in een andere frequentieband.

Compressie cruciaal

De eerste grote uitdaging bij televisie kijken op de mobiele telefoon is de enorme hoeveelheid informatie die in tv-beelden zit. Om televisie te kijken moet je telefoon dertig beelden per seconde kunnen ontvangen. Elk beeld is ruwweg één megabyte groot. Dat betekent dat je in een seconde

Om televisie te kijken moet je telefoon dertig beelden per seconde kunnen ontvangen. Zonder beeldcompressie is dat onmogelijk.

dertig megabyte aan data zou moeten versturen en ontvangen. En dan gaat het nog maar over één enkele zender, terwijl de aanbieders vele zenders tegelijk versturen. Zonder beeldcompressie is dat onmogelijk.

Omdat een beeld vaak uit stukken bestaat waarin de pixels sterk op elkaar lijken, kun je de informatie van deze stukken comprimeren door te zeggen dat al deze pixels dezelfde kleurwaarde hebben. Dat gebeurt met ruimtelijke compressie. Daarnaast wordt ook nog tijdscompressie toegepast. Bij de dertig beelden per seconde zitten vaak opeenvolgende beelden die nauwelijks van elkaar verschillen. Van zo'n reeks wordt dan alleen één enkel beeld verstuurd. De derde vorm van compressie gebruikt het voorspellen van een beweging die aan de gang is. Stel, dat er een bal van links naar rechts beweegt, dan weet je dat in het volgende beeld de bal niet ineens stilstaat, maar nog steeds beweegt al is het minder snel.

Alle tv-beelden worden met deze drie methoden gecomprimeerd en verstuurd. Vervolgens pakt een speciale decompressiechip op je mobiele telefoon de gecomprimeerde beelden razendsnel uit.

De tweede grote uitdaging ligt in het zo zui-

nig mogelijk gebruiken van de batterij van je mobiele telefoon. Als de radio-ontvanger continu aanstaat om beelden te ontvangen, loopt de batterij razendsnel leeg. Daarom gebruikt je telefoon een truc waarbij de radio-ontvanger slechts zo'n tien procent van de tijd aanstaat en beelden ontvangt (*time division multiplexing*). De zendmast zendt voortdurend alle beschikbare tv-kanalen uit. Stel dat je naar honderd kanalen kunt kijken, dan is afgesproken dat het eerste blok in een reeks van honderd informatieblokken voor zender 1 is, het tweede blok voor zender 2 enzovoort. Als jij voor zender 50 kiest, dan weet je telefoon dat hij alleen maar blok nummer 50 hoeft te ontvangen. Omdat je telefoon weet wanneer blok 50 binnenkomt, schakelt de radio-ontvanger alleen aan om blok 50 te ontvangen en uit wanneer de andere blokken zouden binnenkomen.

Foutencorrectie

Een derde uitdaging is om voor een goede foutencorrectie te zorgen. Juist mobiele telefoongebruikers bevinden zich in omgevingen die het signaal verstoren, bijvoorbeeld in een gebouw dat vol staat met muren en andere obstakels, of in een bus die net een tunnel in rijdt. Hiervoor wordt *forward error correction* gebruikt. Daarbij stuurt de verzender extra data mee met het oorspronkelijke signaal. Als er onderweg geen fouten zouden ontstaan, zijn deze data overbodig. Maar omdat er in de praktijk altijd fouten ontstaan, gebruikt je mobiele telefoon deze extra data om deze fouten te corrigeren.

Sinds 2004 is in Europa de standaard *Digital Video Broadcasting-Handheld* (DVB-H) afgesproken voor de techniek die televisie kijken op je mobiele telefoon mogelijk maakt. DVB-H is een zusje van DVB-T (met de T van *terrestial*), de open standaard die wordt gebruikt voor de uitzending van digitale televisie via zendmasten. Naast het slim omgaan met datacompressie, batterijduur en foutencorrectie, is een ander belangrijk element van DVB-H het efficiënt gebruiken van de beschikbare bandbreedte. Het radiosignaal wordt opgedeeld in een heleboel kleinere subsignalen die dan tegelijk over verschillende frequenties naar de ontvanger worden gestuurd (*orthogonal frequency-division multiplexing*).

Met dank aan Dick Bulterman, hoofd van de afdeling Distributed Multimedia Languages and Infrastructures *van het Centrum Wiskunde & Informatica (CWI) in Amsterdam en hoogleraar aan de Vrije Universiteit van Amsterdam*

Gecombineerde gate- en busplanning

Nieuw algoritme lost Schipholplannings-probleem veel sneller op dan voor mogelijk werd gehouden.

Dagelijks landen op luchthaven Schiphol gemiddeld zo'n zeshonderd vliegtuigen. Het grootste deel daarvan taxiet na de landing naar een van de vaste gates, waar de passagiers door een slurfachtige luchtbrug het vliegtuig verlaten. Zo'n dertig procent van de gelande vliegtuigen gaat niet naar een vaste gate, maar parkeert op een *remote stand* op het vliegveld. Meestal zijn dat kleinere vliegtuigen. Van daaruit worden passagiers met een bus vervoerd naar de terminal.

Elke dag maakt een computerprogramma een planning welk geland vliegtuig de volgende dag aan welke gate wordt toegekend. Deze gatetoekenning wordt bepaald door een reeks van eisen. Zo mag een vliegtuig dat van buiten de EU komt niet terechtkomen aan een gate die alleen bestemd is voor EU-landen – dit vanwege de paspoortcontrole. Verder worden vliegtuigen in acht groottes onderscheiden, van klein tot supergroot en de grootte van het vliegtuig moet passen bij wat een luchtbrug van een bepaalde gate aan kan. Om mogelijke vertragingen te kunnen opvangen, moet er ook altijd minstens twintig minuten zitten tussen het vertrek van het ene vliegtuig van een gate en de aankomst van het volgende vliegtuig bij dezelfde gate.

Een tweede planningsprobleem dat de luchthavenplanners moeten oplossen, is hoeveel bussen beschikbaar moeten zijn en welke bus op welk moment naar welke *remote stand* moet rijden om passagiers op te halen of weg te brengen. Dat is het bustoekenningsprobleem.

Robuust

De computerplanning van de dag ervoor kan op de dag zelf nog handmatig worden aangepast, maar de uitdaging is om de computerplanning zo robuust mogelijk te maken. De belangrijkste eis waaraan beide planningen daarom moeten voldoen, is dat een kleine wijziging in de aankomst- of vertrektijd van een vlucht zo min mogelijk mag leiden tot een herplanning op de dag zelf.

Momenteel wordt eerst het gatetoekenningsprobleem opgelost en de uitkomsten daarvan worden gebruikt voor de oplossing van het bustoekenningsprobleem. "Deze aanpak levert in het algemeen echter niet een optimale oplossing", zegt Guido Diepen, die promoveerde op onderzoek naar een betere oplossing voor beide planningsproblemen. "Het kan zijn dat een optimale oplossing van het gatetoekenningsprobleem alleen maar een slechte oplossing van het bustoekenningsprobleem oplevert. De oplossing van het tweede probleem zou dan flink kunnen verbeteren als je het eerste probleem iets minder dan optimaal oplost. De gecombineerde planning van beide problemen tegelijk zou daardoor veel beter kunnen worden."

Typerend voor deze planningsproblemen is dat het aantal mogelijke planningen weliswaar eindig is, maar zo gigantisch groot dat het voor geen enkele bestaande computer mogelijk is om ze allemaal door te rekenen op zoek naar de beste planning. Een aantal trucs is dan nodig om toch een optimale oplossing te vinden. Diepen ontwikkelde eerst een rekenmethode om beide problemen afzonderlijk op te lossen, gebaseerd op geheeltallig lineair programmeren (zie kader p. 80) en daarna een manier om het gecombineerde probleem op te lossen.

Groeperen

"Een van de trucs die ik heb gebruikt", vertelt Diepen, "is om niet individuele vliegtuigen aan individuele

Schiphol, D-Pier

gates toe te kennen, maar groepen van vliegtuigen aan groepen van gates. Een groep vliegtuigen wordt een gateplan genoemd en bestaat uit een serie vliegtuigen die aan dezelfde gate staan. Gates groepeer je dan bijvoorbeeld op basis van de mogelijkheid om vluchten van binnen of buiten de EU af te handelen en op basis van de toegestane vliegtuiggrootte. Het probleem wordt dan om die groepen vliegtuigen te vinden waarvoor geldt dat elk vliegtuig precies eenmaal in een groep zit en dat de tijd tussen alle vliegtuigen aan dezelfde gate zo groot mogelijk is. Deze laatste eis zorgt ervoor dat de planning

Geheeltallig lineair programmeren

Lineair Programmeren (LP) is een methode voor het oplossen van optimaliseringsproblemen waarin zowel de te optimaliseren functie als de randvoorwaarden lineair zijn. De beslissingsvariabelen mogen alle mogelijke waarden aannemen. Maar in het geval van de gate- en bustoekenningsproblemen op Schiphol mogen de beslissingsvariabelen alleen maar de waarde 0 of 1 aannemen: een 1 bij toekenning en een 0 bij niet-toekenning. In dit geval moet een LP-variant worden gebruikt die *Geheeltallig Lineair Programmeren* (ILP) heet. Zowel het gatetoekenningsprobleem als het bustoekenningsprobleem formuleerde Diepen als een geheeltallig lineair programmeringsprobleem. Helaas zijn ILP-problemen veel moeilijker op te lossen dan LP-problemen. Diepen loste daarom eerst met een techniek die kolomgeneratie heet de veel gemakkelijkere LP-variant op. Hierbij doe je alsof de beslissingsvariabelen van de gate- en bustoekenningsproblemen wel continu zijn. Maar dan kan het gebeuren dat een vliegtuig voor zestig procent wel wordt toegekend aan een gate en voor veertig procent niet. Omdat dat in de praktijk natuurlijk niet kan, wordt nu afzonderlijk doorgerekend wat er gebeurt wanneer je het vliegtuig wel toekent en wat als je het vliegtuig niet toekent. Dat leidt al snel tot een gigantische beslissingsboom om uit te zoeken welk vliegtuig aan welke gate wordt toegekend. Alhoewel je zou verwachten dat het toevoegen van nog meer potentiële gateplannen deze beslissingsboom groter en lastiger maakt, bleek juist dat deze toevoeging het probleem gemakkelijker oplosbaar maakt.

zo min mogelijk wordt verstoord door veranderingen op de dag zelf."

Een mogelijke oplossing is bijvoorbeeld dat de vliegtuigen 3, 5 en 8 worden toegekend aan een groep van gates waar alleen vliegtuigen uit EU-landen mogen aankomen. Welk plan aan welke gate wordt toegekend kan de gateplanner handmatig oplossen. Diepen: "Als het gateplan maar eenmaal is uitgerekend, is dit een gemakkelijk op te lossen probleem."

Vervolgens koppelde Diepen de oplossingsmethoden voor het gatetoekenningsprobleem en het bustoekenningsprobleem aan elkaar en loste hij het gecombineerde planningsprobleem op. Dat was nog niet eerder gedaan. Voor het testen van deze nieuwe methode ge-

bruikte Diepen de geplande vliegtuiglandingen voor zes typische Schipholdagen, waarvan drie in het hoogseizoen en drie in het laagseizoen. Verder gebruikte hij de busgegevens voor dertig typische dagen. Elk van de zes dagen rekende Diepen met elk van de dertig busdagen door.

"Deze tests laten zien dat we het gecombineerde probleem op een gewone pc in twintig tot zestig minuten kunnen oplossen", zegt Diepen. "Dat is veel sneller dan voor mogelijk werd gehouden. Bovendien leveren onze oplossingen langere tijden tussen de vliegtuigen op. Daardoor zijn de gate- en de busplanning minder gevoelig voor vertragingen op de dag zelf." ◉

Vloeiend bewegen in een computergame

Een nieuw rekenmodel, gebaseerd op de psychologie van menselijk ontwijkgedrag, laat karakters vloeiender bewegen door een virtuele wereld.

Zowel in Nederland als in de rest van de wereld groeit de game-industrie fors. PricewaterhouseCoopers verwacht tot 2013 een jaarlijkse omzetgroei van 7,4 procent. Dat zou betekenen dat er in 2013 wereldwijd in de branche 73,5 miljard dollar wordt omgezet. Het grootste deel daarvan komt voor rekening van computergames voor de entertainmentindustrie. Een kleiner, maar snel groeiend aandeel komt voor rekening van 'serious gaming': virtuele omgevingen voor educatie en training, zoals vluchtsimulaties, virtuele brandweertrainingen en virtuele rondleidingen in bouwkundige ontwerpen.

Zowel bij de entertainment- als bij de serieuze games speelt het navigeren door de virtuele wereld een centrale rol. Karakters moeten zo vloeiend mogelijk rondbewegen en botsingen met obstakels en andere karakters vermijden. "Zelfs in geavanceerde huidige games", zegt hoogleraar informatica Mark Overmars van de Universiteit Utrecht, "komt het regelmatig voor dat een karakter tegen een obstakel botst en er maar tegenaan blijft botsen. Hij vindt geen weg eromheen, zelfs als die wel bestaat. Dat laat zien hoe moeilijk het is om een natuurlijk pad te plannen in een computergame."

Denkbeeldige gang

Ondanks dat er al veel onderzoek naar padplanning is verricht, is het probleem nog steeds niet realistisch

Een vloeiend pad in een virtuele stad met de bijbehorende corridor

genoeg opgelost, zoals de soms lachwekkende botsingen in games laten zien. Het padplanningsprobleem is dan ook gecompliceerder dan op het eerste gezicht lijkt. Game-ontwerpers moeten zowel met stilstaande als bewegende voorwerpen rekening houden. Verder moeten de paden er ook nog vloeiend uitzien. En wanneer je als het ware met een camera naar het karakter kijkt (zoals bij een *third-person-game* in plaats van een *first-person-game*), moet niet alleen het pad van het karakter er natuurlijk uitzien maar ook de camerabeweging.

Een belangrijke praktische randvoorwaarde voor het oplossen van het padplanningsprobleem is dat de processor die het probleem moet oplossen niet meer dan tien procent mag worden belast. De rest van de rekenkracht is hard nodig om de andere facetten van de game te berekenen. Ook moet de oplossing vrijwel in *realtime* worden berekend.

Overmars en zijn collega's hebben een nieuwe, snelle oplossingsmethode voor het padplanningsprobleem bedacht, die aansluit bij de manier waarop mensen in het dagelijks leven bewegen. "Het klinkt paradoxaal", zegt Overmars, "maar als je een pad berekent, wil je eigenlijk helemaal niet dat je een exact pad berekent. Als mens zoek je juist naar een globaal pad met een bepaalde ruimte om het pad heen waarin je vrij mag bewegen. Geïnspireerd

hierop berekenen wij eerst een indicatieve route – een soort ideale lijn – en daarna een gang er omheen. Globaal blijft een karakter de indicatieve route volgen, maar hij mag ervan afwijken, zolang hij maar binnen de gang blijft. Dit biedt de vrijheid om obstakels en andere karakters op het pad te ontwijken."

De routes die karakters kunnen bewandelen zijn gegeven door het spelontwerp. Dat maakt het mogelijk om de indicatieve routes en bijbehorende gangen te berekenen voordat een speler met de game begint. Wanneer de game begint, is de volgende vraag hoe een karakter dan precies beweegt door de denkbeeldige gangen langs de indicatieve route. Om het echte pad te berekenen, gebruikt Overmars een krachtenmodel. "Het hoofdkarakter stellen we als een cirkel voor die zich ergens in de gang beweegt. Langs de indicatieve route beweegt zich nu een punt dat

Nieuw krachtenmodel

Om karakters natuurlijker te laten bewegen in een game gebruikt Overmars een krachtenmodel dat lijkt op de manier waarop mensen in het dagelijks leven botsingen vermijden. Stel, je loopt rond in een stad. Dan kijk je af en toe om je heen om in te schatten met wie je wanneer in botsing zou kunnen komen. Vervolgens pas je je looprichting en -snelheid aan om toekomstige botsingen te vermijden. Dit idee kun je vertalen naar de gamewereld. Wanneer twee karakters op botsingkoers bewegen, pas je het pad van beide karakters aan. Dat gebeurt echter niet langer meer door een directe afstoting tussen de twee karakters die pas begint op het moment wanneer ze dicht genoeg bij elkaar zijn. Overmars laat al veel eerder de banen afwijken om een botsing te vermijden. De afwijking wordt bepaald door een soort virtuele, nieuwe kracht. Zowel de richting als de grootte van de nieuwe afstotende kracht hangt niet langer af van de afstand tussen de karakters, maar van de relatieve positie ten opzicht van het toekomstige botsingspunt. Op elk moment wordt de beweging van een karakter bepaald door de optelsom van de kracht die hem langs zijn huidige baan voortdrijft en de extra kracht die ervoor zorgt dat de toekomstige botsing wordt vermeden. Via de basisprincipes van de meetkunde en de vectorrekening wordt dit model geïmplementeerd in het computerprogramma van de game.

een aantrekkende kracht uitoefent op het hoofdkarakter. De wanden van de gang oefenen juist een afstotende kracht op het karakter uit. Ook alle obstakels en andere karakters oefenen een afstotende kracht uit op het hoofdkarakter."

Psychologische afstoting

Traditioneel gebruiken gamebouwers een krachtenmodel waarin karakters elkaar pas beginnen af te stoten wanneer ze dicht genoeg bij elkaar komen. Pas op het laatste moment ontwijken karakters elkaar. Dat leidt tot onnatuurlijke, schokkerige bewegingen. Overmars en zijn collega's hebben een krachtenmodel gemaakt dat tot veel vloeiendere bewegingen leidt. In dit model passen karakters eerder hun pad aan om een toekomstige botsing te vermijden (zie kader p. 85). Een interessante uitkomst van het model is dat het realistisch emergent gedrag laat zien dat traditionele modellen niet lieten zien. Zonder dat het bewust in het model is gestopt, genereert het nieuwe model namelijk groepsgedrag dat ook in de alledaagse werkelijkheid optreedt, zoals de vorming van rijen en groepen.

Voor het opstellen van het nieuwe krachtenmodel heeft Overmars in zijn speciaal opgezette Utrechtse bewegingslaboratorium bestudeerd hoe proefpersonen rondlopen en botsingen vermijden. Deze multidisciplinaire aanpak is een noviteit in de gamewereld. "We willen kwantitatief onderzoeken op welk moment mensen hun pad beginnen aan te passen. Die gegevens gebruiken we om ons krachtenmodel fijn af te stellen. In werkelijkheid hangen de gegevens ook van de situatie af. Ben je aan het winkelen, reis je naar je werk of loop je als een toerist in een stad rond? En daarnaast speelt emotie en persoonlijkheid een belangrijke rol. Wat dit betreft is padplanning deels ook gebaseerd op sociale psychologie."

Samen met twee Nederlandse gamebedrijven werkt Overmars inmiddels aan de implementatie van zijn indicatieve-route-model in de nieuwe generatie computergames. ●

Hoe werkt een simulatietraining voor de brandweer?

Brandweertrainingen in de praktijk zijn tijdrovend en duur. Brandweerkorpsen oefenen daarom maar een beperkt aantal keren en een beperkt aantal scenario's in het echt. Om die nadelen te omzeilen, ontwikkelde het Rotterdamse bedrijf VSTEP de virtuele brandweertraining *RescueSim*, die sinds 2006 commercieel verkrijgbaar is. RescueSim bereidt bevelvoerders en officieren van brandweerkorpsen voor op een groot aantal brandscenario's. De cursist ziet hoe zich in een realistische omgeving een brandscenario voltrekt. Hij schat de situatie in en bepaalt op welke manier het incident aangepakt wordt en het vuur zo snel en goed mogelijk geblust wordt.

Veel van de onderliggende informaticatechnologie van deze simulatietraining komt rechtstreeks uit de gamewereld, zoals de realistische graphics, het plannen van paden die de karakters kunnen bewandelen en het in realtime laten verlopen van de actie. Toch verschilt de simulatietraining op een aantal punten essentieel van entertainmentgames. Bij games draait alles om het plezier van de speler. Daarvoor worden vaak fantasieaspecten verwerkt in het spel en worden sommige effecten versterkt en andere juist verzwakt. Maar in een simulatietraining moeten zowel de fysieke omgeving als de gebeurtenissen de werkelijkheid zo dicht mogelijk benaderen, terwijl de rekentijd niet uit de hand mag lopen. In beide aspecten ligt een grote uitdaging voor de programmeurs.

Belangrijk is ook dat het scenario niet alleen qua beeld, maar ook qua geluid zo realistisch mogelijk is. In de praktijk be-

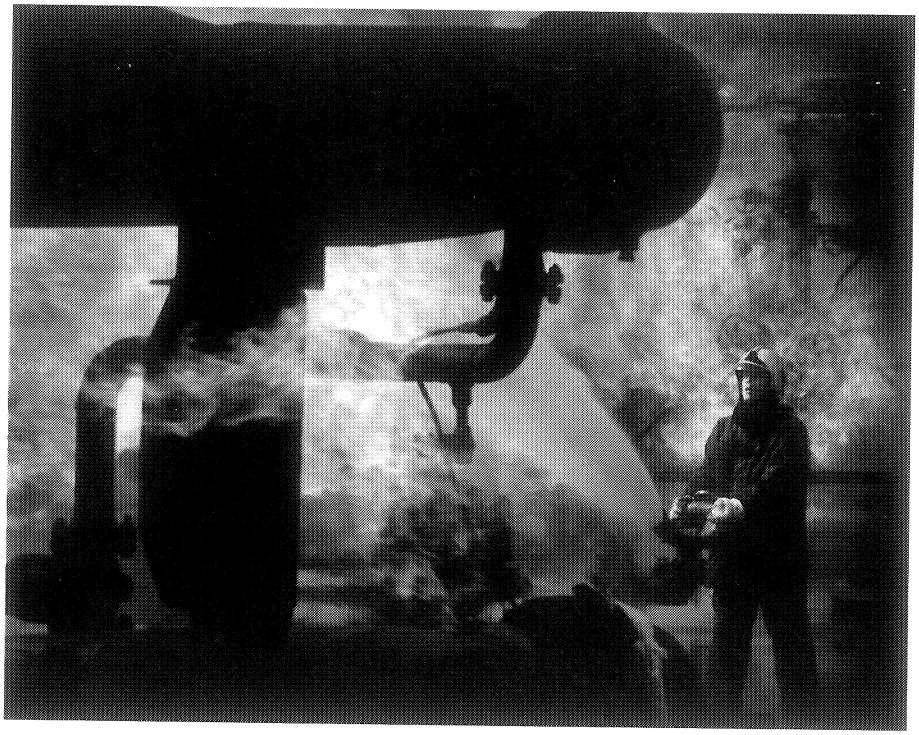
> De instructeur kan in de simulatie het weer veranderen, de plaats van voertuigen, het overige verkeer, welke ladingen de voertuigen hebben en zelfs hoe snel een olieplas zich uitbreidt.

oordelen brandweermannen een incident immers deels ook op wat ze horen. Om de brandscenario's zo realistisch mogelijk te simuleren, werken de programmeurs nauw samen met brandweerexperts.

Drag-and-drop

Twee bedrijven die hun brandweerkorps tegenwoordig virtueel trainen met Rescue-Sim, zijn chemieconcern DSM in Geleen en het Havenbedrijf Rotterdam. Voor beide bedrijven hebben de ontwikkelaars van VSTEP de specifieke bedrijfsomgeving op basis van plattegronden en foto's zo realistisch mogelijk nagebouwd in een virtuele omgeving. Daarna is het zaak om alle mogelijke paden die mensen en voertuigen kunnen afleggen te definiëren in het simulatieprogramma.

Een tweede belangrijk verschil tussen een simulatietraining en een entertainmentgame is dat brandweerkorpsen hun bevelvoerders niet zuiver en alleen tegen de computer willen laten trainen, maar samen met een instructeur – in de praktijk iemand die zelf veel ervaring heeft als bevelvoerder. De instructeur kan vooraf kiezen uit voorgeprogrammeerde scenario's, maar hij kan ook zelf een scenario verzinnen, creëren en laten uitvoeren.



RescueSim stelt de instructeur in staat om een groot aantal parameters te variëren, zoals het weer, de windrichting en windkracht, de plaats van voertuigen, het overige verkeer, welke ladingen de voertuigen hebben en zelfs hoe snel een olieplas zich uitbreidt. Via een *drag-and-drop*-systeem kan hij uit een reeks van voorgeselecteerde objecten kiezen welk object hij waar in het scenario wil plaatsen.

RescueSim is zo opgezet dat de cursist en de instructeur allebei naar een eigen scherm kijken. De cursist kijkt naar een groot scherm waarop zich een voor hem onbekend brandscenario voltrekt. Via een draadloze joypad kan hij zelf als bevelvoerder door het scenario bewegen. Terwijl de cursist virtueel in het scenario van RescueSim beweegt, geeft hij al pratend opdrachten aan zijn brandweermannen. In werkelijkheid zouden de brandweermannen die instructies via een koptelefoon in hun helm horen. In de simulatie gebeurt dat via de instructeur, die als het ware de rollen van alle andere brandweermannen speelt. Stel dat de cursist brandweerman 1 de opdracht geeft om naar plek B te gaan, dan zegt hij deze opdracht hardop en de instructeur, die meestal met zijn laptop achter de cursist zit, voert de opdracht uit via de muis of het toetsenbord.

Intelligente karakters

De instructeur kan het scenario vanaf de grond bekijken, maar hij kan de situatie ook in vogelvlucht overzien door op zijn scherm als het ware uit te zoomen. Hij speelt een actieve rol in de simulatie doordat hij het scenario op elk moment kan beïnvloeden. Hij kan bijvoorbeeld plotseling de wind van richting laten veranderen, een brand op een nieuwe plek laten ontstaan of opeens een slachtoffer laten vallen. Tegelijkertijd beoordeelt de instructeur ook hoe goed de cursist presteert. Hij kan tussentijds ingrijpen en de situatie nog een keer opnieuw laten spelen, maar hij kan ook achteraf evalueren hoe goed de cursist een bepaald brandscenario heeft opgelost. RescueSim houdt zelf ook een logboek van de gebeurtenissen bij.

Een van de informatica-uitdagingen voor de toekomst is om de huidige *non-playing-characters* in de simulatie kunstmatige intelligentie te geven, zodat ze zelf een actieve rol kunnen spelen, in plaats van volledig te worden aangestuurd door de instructeur, al dan niet in opdracht van de cursist. Een tweede uitdaging is om van de simulatie een multiplayersimulatie te maken, waaraan meerdere brandweermannen tegelijk kunnen deelnemen. In samenwerking met de informaticaopleiding van de Universiteit Utrecht doet VSTEP in beide richtingen langetermijnonderzoek.

Met dank aan Pjotr van Schothorst, technisch directeur van VSTEP

Stromingen rond schepen beter gesimuleerd

Een scheepsontwerp is gebaseerd op zowel fysieke schaalmodellen als numerieke modellen. Een nieuw numeriek algoritme geeft betere en snellere resultaten voor specifieke scheepsstromingen.

Voordat een scheepsbouwer aan de slag gaat, wil hij weten hoe het schip op zee gaat presteren. Gedraagt het zich stabiel in alle mogelijke golven? Wat is de scheepsweerstand? Functioneert de schroef wel efficiënt genoeg in het zog achter het schip? Kan het schip snel genoeg manoeuvreren?

Het maken van fysieke scheepsmodellen en het testen ervan in een scala van mogelijke stromingen, is echter duur. In de praktijk gebruiken scheepsbouwers daarom een numeriek model gebaseerd op wiskundige stromingsvergelijkingen om een eerste scheepsontwerp te maken. Op grond hiervan maken ze een fysiek scheepsmodel, dat ze in een laboratorium testen om te zien hoe goed het in een echte waterstroming werkt. Aan de hand van deze testen kunnen ze het numerieke model finetunen. Met het verbeterde computermodel wordt vervolgens het definitieve schip ontworpen.

Deze aanpak combineert het beste van twee werelden. Het numerieke model is beter dan het fysieke schaalmodel in staat de effecten van turbulentie in de stroming te voorspellen. Dat komt omdat het schaalmodel in een laboratoriumstroming nooit dezelfde turbulente

Jeroen Wackers

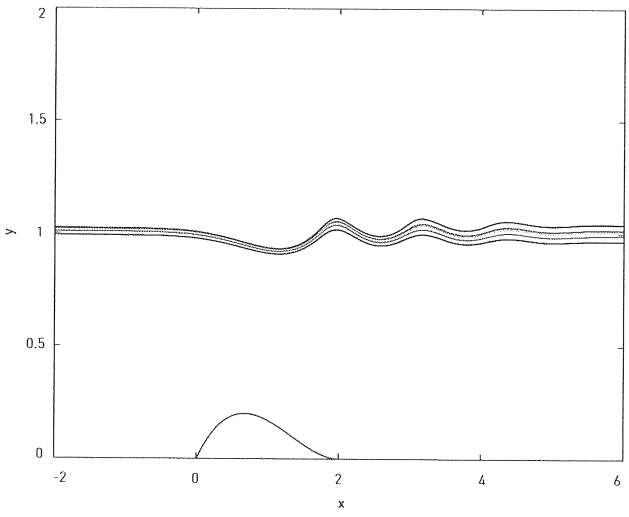
karakteristieken kan nabootsen als die van de stromingen op zee. Aan de andere kant kan het schaalmodel in de praktijk bestudeerd worden zonder dat er vereenvoudigende modelaannames nodig zijn.

De centrale moeilijkheid bij het simuleren van stromingen rond schepen is de aanwezigheid van het grensvlak tussen water en lucht. Een stromingsberekening rond een schip is een stuk moeilijker dan die rond een vliegtuig, vertelt lucht- en ruimtevaartingenieur Jeroen Wackers: "Bij de stroming rond een vliegtuig ligt de geometrie van het stromingsgebied exact vast door de vaste vorm van het vliegtuig. Bij de stroming rond een schip niet, omdat het oppervlak tussen water en lucht door de stroming verandert."

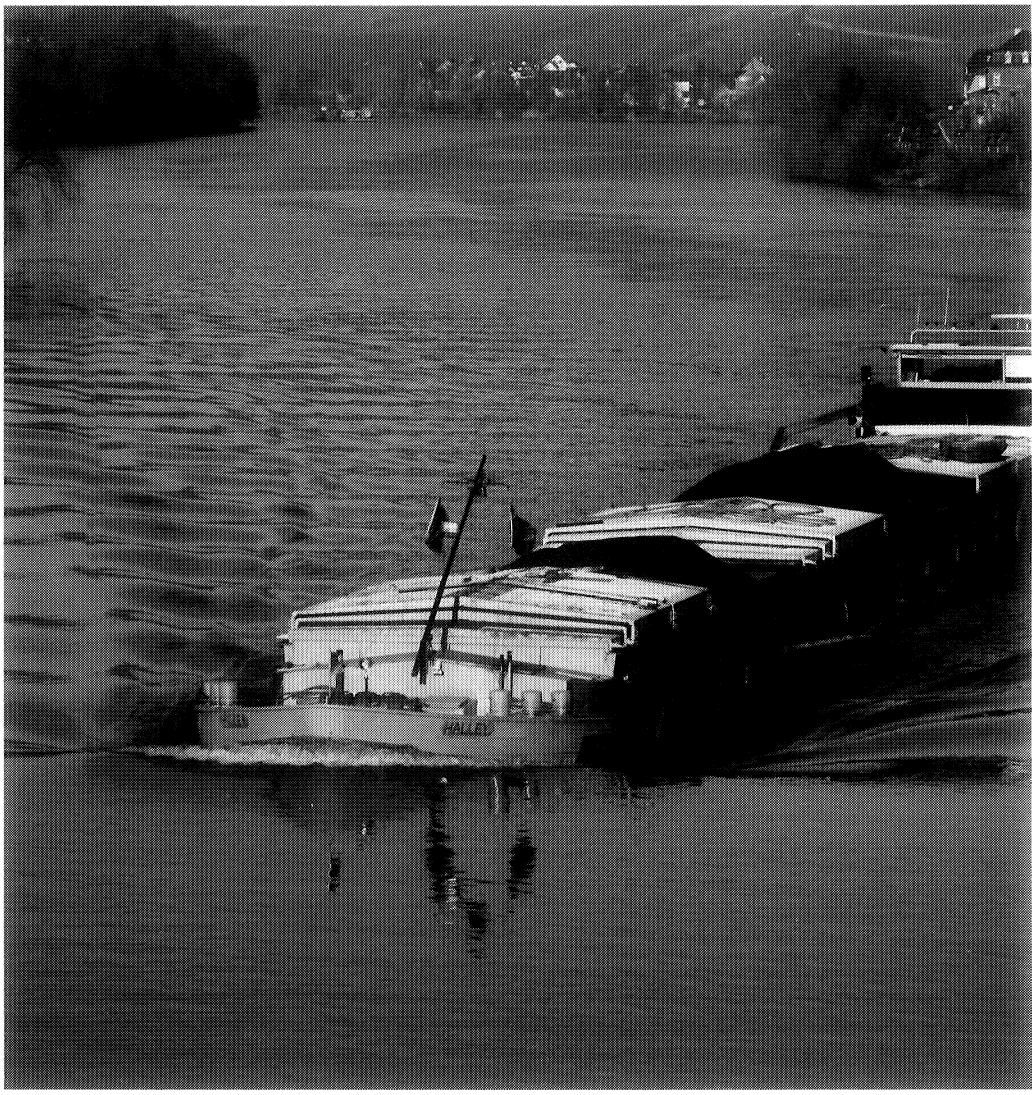
Wackers deed zijn promotieonderzoek naar numerieke computersimulatietechnieken die scheepsstromingen beter kunnen simuleren. Hij verrichtte zijn promotieonderzoek aan het Centrum Wiskunde & Informatica (CWI) in Amsterdam en promoveerde aan de Technische Universiteit Delft.

Mengsel

Veel numerieke stromingsmodellen doen alsof er geen lucht boven het water zit en lossen alleen het gedrag van het water op. Deze methode werkt weliswaar snel, maar is beperkt tot vrij simpele scheepsvormen.



Golven die worden veroorzaakt door een obstakel op de bodem van een tweedimensionaal kanaal (numerieke simulatie)



Geavanceerdere modellen doen alsof er helemaal geen grensvlak tussen lucht en water is, maar nemen daarvoor in de plaats een mengsel van water en lucht aan. Deze techniek levert in de praktijk betrouwbaardere resultaten. Diep genoeg in het water heeft het mengsel precies de eigenschappen van water. Hoog genoeg boven het echte grensvlak tussen water en lucht heeft het mengsel alle eigenschappen van lucht. Daartussenin varieert de dichtheid van het mengsel tussen beide uitersten in.

De stromingsvergelijkingen worden voor dit water-luchtmengsel opgelost. Hoewel het mengselmodel goed werkt en flexibel is in de aanname van de scheepsvorm, werkt de numerieke oplossingstechniek relatief langzaam. "Iedereen dacht dat er geen snelle oplossingstechniek bestond", zegt Wackers. "Voor mij was de uitdaging om toch een snelle numerieke oplossingstechniek te vinden."

In plaats van het simuleren van een echt schip, richtte Wackers zich op een klassiek stromingsprobleem met een eenvoudige geometrie, dat sterk verwant is aan de stroming rond een schip: een tweedimensionale en stationaire kanaalstroming met een drempel op de bodem van het kanaal. Dit is een eenvoudig model voor stromingen met golven aan het oppervlak, zoals bij een schip ook altijd het geval is. Dat de kanaalstroming stationair is, wil zeggen dat de stroming niet in de tijd verandert, zoals het geval is wanneer een schip rechtdoor vaart op een rustige zee.

Voor het oplossen van deze tweedimensionale kanaalstroming met golven op het wateroppervlak, combi-

Hybride oplostechniek

Een numeriek model legt een rekenrooster aan in het gebied van de stroming die gesimuleerd moet worden. De stroming wordt vervolgens opgelost op de roosterpunten. De bestaande multiroostertechniek gebruikt een fijn rooster om numerieke fouten met een hoogfrequent karakter weg te poetsen en een grof rooster om de numerieke fouten met een laagfrequent karakter te elimineren. Deze methode werkt alleen goed als de fysieke eigenschappen van de stroming op het fijne rooster vergelijkbaar zijn met die op het grove rooster. Dat is echter niet het geval in een mengselmodel dat een golvende stroming over een bodem beschrijft. Waar bijvoorbeeld in het grove rooster in een roosterhokje nog water zit, kan in een klein roosterhokje van het fijne rooster al alleen maar lucht zitten. Wackers loste dit probleem voor de multiroostertechniek op door de niet-lineaire vergelijkingen van het fijne rooster te lineariseren en te kopiëren naar het grove rooster. Op het fijne rooster loste hij de niet-lineaire vergelijkingen direct op en op het grove rooster loste hij de gelineariseerde vergelijkingen op. Deze nieuwe, hybride methode pakte goed uit en leverde een tientallen malen snellere numerieke oplossing.

neerde Wackers twee al bestaande numerieke technieken met elkaar. Aan de ene kant het mengselmodel (*surface capturing*) en aan de andere kant een *multiroostermethode*. Wackers paste allebei de technieken aan om ze met elkaar te combineren en zo vond hij de snelle numerieke oplossing waarop hij had gehoopt (zie kader). Dit numerieke model lost de onderzochte tweedimensionale kanaalstroming tientallen malen sneller op dan bestaande technieken.
In 2007 won Wackers voor zijn onderzoek de prijs voor het beste proefschrift van de *European Community on Computational Methods in Applied Sciences*.

Racewagen

Ondanks dat hij een snelle oplossing vond, verwacht Wackers niet dat zijn methode rechtstreeks gekopieerd kan worden naar een realistische scheepssimulatie: "Ik heb als het ware een racewagen gebouwd die supersnel is op een circuit. Maar een racewagen is niet automatisch geschikt voor het rijden op de gewone snelweg of in de stad."

Wackers ziet zijn werk als een ontdekkingstocht in het onbekende landschap van numerieke simulatietechnieken. "Men dacht dat je een mengselmodel niet snel numeriek kunt oplossen. Ik heb laten zien dat het voor speciale gevallen wel snel kan. Het punt is alleen dat die speciale gevallen niet voldoende zijn voor de simulatie van alle scheepsstromingen. Ik denk dat de beste praktijkmethode ergens tussen de huidige modellen en mijn model voor een geïdealiseerde stromingsgeometrie zal uitkomen. Verder denk ik dat uit mijn werk ideeën rollen die je in tal van numerieke simulaties kunt toepassen, zonder dat dit specifiek scheepsstromingen moeten zijn." ◉

Snellere simulatie van bellen en druppels

Geavanceerde numerieke methode lost stromingsproblemen van opstijgende bellen en vallende druppels veel sneller op.

Bij het oppompen van ruwe olie worden onder aan de olieproductieleiding vaak luchtbellen toegevoegd. Zo stuwt de olie gemakkelijker omhoog. Dit is een voorbeeld van belletjes in een stromende vloeistof. Een voorbeeld van het omgekeerde – druppeltjes vloeistof omringd door lucht – treedt op in een inktjetprinter, die druppeltjes vloeibare inkt op het papier spuit.

Beide zijn voorbeelden van *tweefasestromingen*, waarbij de vloeistoffase en de gasfase naast elkaar in een stroming voorkomen. Voor realistische toepassingen is het onmogelijk om met pen en papier de wiskundige stromingsvergelijkingen op te lossen en zo de stroming te begrijpen en te controleren. Ook is het niet altijd mogelijk om realistische experimenten uit te voeren. Daarom worden tweefasestromingen vaak gesimuleerd op de computer.

Wiskundige Jok Tang deed aan de Technische Universiteit Delft promotieonderzoek naar numerieke methoden om stromingen met druppels of bellen op een computer snel en betrouwbaar te simuleren. Hij vertelt met welk probleem zulke simulaties traditioneel te maken krijgen: "Stel, je wilt een waterstroming met luchtbellen simuleren. In het eenvoudigste geval kijk je alleen naar een enkele bel in een klein kubusje water. Als je de kubus groter maakt of als je meer bellen wilt simuleren, dan wordt het

Jok Tang

probleem al gauw zeer gecompliceerd. De rekentijd loopt uit de hand wanneer je realistische stromingen wilt simuleren."

Gokken

Vele tweefasestromingen met bellen en druppels kun je beschrijven met de zogeheten Navier-Stokes-vergelijkingen. Deze beschrijven bij gegeven beginvoorwaarden en vloeistofeigenschappen op elk punt in de ruimte en op elk tijdstip wat de lokale snelheid en druk zijn. In een computersimulatie van een stroming kun je de snelheid en de druk niet op alle punten oplossen, maar op een geselecteerd aantal punten, die worden vastgelegd door een rekenrooster. Tang koos voor een kubusvormig rekenrooster dat er in alle drie de ruimtelijke richtingen hetzelfde uitziet. Het rooster kan meer dan een miljoen roosterpunten tellen, wat voor de computer betekent dat hij meer dan een miljoen wiskundige vergelijkingen met evenzoveel onbekenden moet oplossen.

In de door Tang gebruikte methode om de stromingsvergelijkingen numeriek op te lossen worden ze in twee delen opgesplitst: eentje voor de druk en eentje voor de snelheid. Het drukverloop vertelt ook of er op een bepaalde plek vloeistof zit of gas. "Vervolgens lossen we de druk en de snelheid apart van elkaar op", zegt Tang. "Daarbij kost het oplossen van de drukvergelijking verreweg de meeste tijd – zo'n zestig tot zeventig procent voor problemen die ik bekeek. Mijn doel was daarom om te zoeken naar een slimme oplosmethode die de drukvergelijking veel sneller oplost."

De oplosmethode van Tang is gebaseerd op een iteratief proces, waarbij je stap voor stap de numerieke oplossing verbetert. Je begint met het doen van een gok voor de oplossing. Stapje voor stapje bereken je vervolgens met de oplosmethode een betere schatting van de oplossing, tot je uiteindelijk een aanvaardbare oplossing van het probleem hebt gevonden. De oplossingstijd wordt bepaald door het aantal iteratiestappen en de rekentijd per iteratiestap.

Met een slimme truc lukte het Tang inderdaad om de oplossingstijd van de drukvergelijking aanzienlijk terug te brengen (zie kader p. 100). Hij kon zowel in een

theoretische analyse als in numerieke simulaties laten zien dat het aantal iteraties van zijn zogeheten deflatiemethode nauwelijks afhangt van het aantal bellen in de stroming, terwijl deze in methoden uit de literatuur doorgaans flink toeneemt met het aantal bellen. Bovendien kon hij, vergeleken met traditionele iteratieve methoden, het aantal benodigde iteraties met de deflatiemethode aanzienlijk terugbrengen en de rekentijd significant verkorten.

Uiteenspattende druppel

Tang paste zijn numerieke methode zowel toe op de stroming van luchtbellen in water als op het uiteenspatten van druppels die op een wateroppervlak vallen. Globaal gezien werkt de methode in beide gevallen even goed.

Maar naast de deflatiemethode zijn in de wetenschappelijke literatuur ook andere methoden voorgesteld om de drukvergelijkingen, afgeleid van tweefasestromingen van druppels of bellen, op te los-

Hogesnelheidsopname van een vallende druppel water die op een wateroppervlak uiteenspat

Deflatietruc verkleint het aantal iteratiestappen

Voor het numeriek oplossen van een stroming met bellen of druppels moest promovendus Jok Tang voor de drukvergelijking een groot lineair stelsel van vergelijkingen oplossen. Dit stelsel kun je schrijven als een matrix-vector-vergelijking. Tang kon aantonen dat de duur van de simulatie vooral wordt bepaald door de eigenwaarden van de matrix. In de literatuur en bij zijn promotor Kees Vuik was al bekend dat een aantal 'slechte' eigenwaarden voor een sterke vertraging kunnen zorgen. Tang onderzocht samen met Vuik of hij dit probleem kon vermijden door de slechte eigenwaarden met een truc uit het probleem te werken. Via de zogeheten deflatiemethode zocht hij naar een nieuwe matrix zonder de 'slechte' eigenwaarden. In deze methode vermenigvuldigt hij zowel de linker- als de rechterkant van de matrix-vector-vergelijking met een zogenaamde deflatiematrix. Door een slimme keuze van deze deflatiematrix krijgt het product van de twee matrices aan de linkerkant van de vergelijking 'betere' eigenwaarden. Nadat Vuik deze methode uitgebreid had onderzocht in andere contexten, heeft Tang als eerste laten zien dat de deflatiemethode ook werkt voor de simulatie van bellen en druppels. In eerste instantie lijkt het erop alsof je informatie weggooit door een aantal van de eigenwaarden van de oorspronkelijke matrix weg te werken. Maar Tang heeft aangetoond dat de fysica van het probleem niet verandert en dat je de numerieke oplossing wel zestig tot zeventig procent sneller vindt dan met traditionele methoden.

sen. "Ik heb mijn methode vergeleken met diverse andere methoden", zegt Tang, "en ik heb geconcludeerd dat ze minstens zo goed werkt als deze methoden. Sommige methoden hebben minder iteraties nodig, maar dan gaat vaak wel de rekentijd per iteratie omhoog. Specifieke multigridmethoden doen het ongeveer even goed, maar de andere methoden vergen allemaal meer rekentijd. In de praktijk zal het afhangen van de specifieke toepassing of je het beste voor de multigridmethode of voor de deflatiemethode kunt kiezen. Ook heb ik theoretisch laten zien dat de relatie tussen deze verschillende methoden veel kleiner is dan men dacht."

Hoewel Tang zich heeft gericht op het fundamentele onderzoek van nieuwe numerieke methoden en

niet specifiek op de praktische toepassingen, denkt hij dat zijn resultaat wel degelijk in de praktijk belangrijk kan zijn: "Omdat computers steeds sneller rekenen, gaan mensen steeds moeilijkere en grotere stromingen simuleren, waarin steeds meer bellen of druppels zitten. Daarbij vergeten ze vaak dat de traditionele iteratieve methoden naar verhouding steeds duurder worden. Mijns inziens ontkom je daarom niet aan het gebruiken van betere en geavanceerde numerieke methoden om de drukvergelijking efficiënt te blijven oplossen. Het gebruik van ons type oplosmethoden is dan ook onontbeerlijk voor vele simulaties." ●

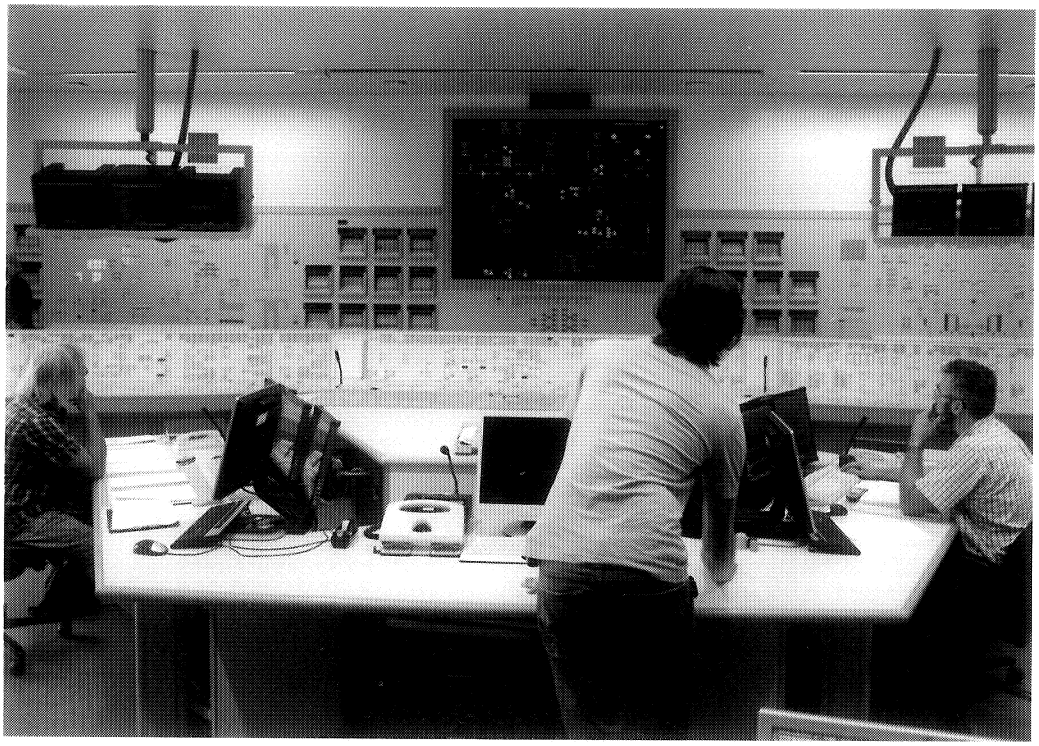
Faalkansen van softwaresystemen berekenen

Software maakt een steeds belangrijker deel uit van producten en diensten. Een nieuwe informaticatechniek kwantificeert het faalgedrag van softwaresystemen.

Elektriciteitscentrales worden met software bewaakt en geregeld. Hartbewaking, tumorbestraling of hersenscans zijn afhankelijk van software. Zelfs een auto bevat steeds meer digitale regeling en controle. Het aantal diensten en producten waarin software een cruciale rol speelt, is de afgelopen twee decennia sterk toegenomen. De kwaliteit van deze producten en diensten wordt traditioneel vrijwel alleen kwalitatief beoordeeld: of het systeem werkt, of het werkt niet.

Mariëlle Stoelinga van de Universiteit Twente onderzoekt met haar collega's of de kwaliteit van een product of dienst waarin software zit ook kwantitatief kan worden geanalyseerd: "Bij elk systeem kun je de vraag stellen wat de kans is dat het na een bepaalde tijd faalt. En vervolgens is dan de vraag: Vinden we die tijd wel of niet acceptabel en moet het systeem al dan niet verbeterd worden?"

Een belangrijk uitgangspunt voor het kwantitatief modelleren en analyseren van systemen met software, is het denkbeeldig uiteenrafelen van het gehele systeem in zijn belangrijkste onderdelen. Het doel



Controlekamer van de kerncentrale Borssele

is om de betrouwbaarheid van het geheel vervolgens te analyseren uit de betrouwbaarheid van de onderdelen. Neem een relatief eenvoudig systeem als een hartpompsysteem in een ziekenhuis, dat vaak wordt ingezet om de tijd tot een harttransplantatie te overbruggen. Dit systeem kun je opgebouwd denken uit een module met pompen, eentje met motoren en eentje met rekenprocessoren, die de software van het systeem aansturen. Elk van deze drie modules kun je weer opgebouwd denken uit submodules.

Reserveonderdelen

Schematisch kun je alle modules en submodules in een omgekeerde boomstructuur tekenen, met bovenaan de stam (het hele systeem) en daaronder de drie grote takken (pompen, motoren en rekenprocessoren) en daar weer onder de kleinere takken (de subeenheden). Zo kan de pompmodule bijvoorbeeld bestaan uit een gewone pomp en een reservepomp, die allebei een

eigen batterij hebben. Het systeem wordt nog ingewikkelder als we bijvoorbeeld aannemen dat als een van deze batterijen het begeeft, ze allebei terugvallen op dezelfde reservebatterij.

Juist reserveonderdelen, zoals een noodaggregaat of een databackupsysteem, maken de faalanalyse van het systeem extra complex. Typisch voor reserveonderdelen is dat hun faalkans sterk afhangt van of ze wel of niet worden gebruikt. Een reserveband die achter in de auto ligt, gaat zelden zomaar stuk. Maar zodra je een kapotte band vervangt door de reserveband, wordt zijn faalkans een stuk groter en gelijk aan die van een gewone band.

Informatie in zo'n foutenboom stroomt als het ware van de kleinste takken, via de grotere takken naar de stam. Elke foutenanalyse van een complex systeem gebruikt zo'n boomstructuur als basismodel. De faalanalyse zelf gebeurt met een kansmodel, omdat je nooit exact weet wanneer een onderdeel faalt, maar wel kunt kijken wat de faalkans is op grond van de gegevens uit het verleden. De cruciale vraag is vervolgens hoe je de boomstructuur kwantitatief kunt analyseren. Stel dat er ergens in een tak iets misgaat, hoe wordt die informatie dan doorgegeven aan de andere takken en wat is uiteindelijk het gevolg voor het functioneren van het hele systeem?

Een hartpompsysteem is nog relatief simpel. Complexere systemen kunnen wel honderden hardware- en softwarecomponenten hebben, die allemaal een faalkans hebben. "Het grote probleem van traditionele foutboomanalyses", vertelt Stoelinga, "is dat het aantal mogelijke toestanden waarin het systeem kan voorkomen, explodeert. Dat maakt het onmogelijk om alle toestanden die kunnen optreden te analyseren. Wij hebben een techniek ontwikkeld die het aantal toestanden beperkt, maar toch een betrouwbare analyse van de kans op falen maakt."

Architectuurtaal

Deze nieuwe techniek maakt een succesvolle faalanalyse mogelijk door alleen de belangrijkste eigenschappen van het systeem mee te nemen (zie kader p. 106).

Interactieve Markov-ketens

De nieuwe techniek die Stoelinga en haar collega's hebben gemaakt, is gebaseerd op interactieve *Markov-ketens* met een input-outputkarakter. Een Markov-keten beschrijft een systeem dat stapsgewijs van de ene naar de andere toestand gaat. Voor elke overgang is een bepaalde kans gegeven. Verder is elke overgang niet afhankelijk van wat er daarvoor in het systeem gebeurde. Traditioneel zijn Markov-ketens niet ontworpen om interactie met andere Markov-ketens te hebben. Het model van Stoelinga beschrijft juist gekoppelde Markov-ketens die wel informatie aan elkaar doorgeven. Elke Markov-keten representeert in het model een onderdeel van het systeem. Stel, dat in een bepaalde Markov-keten een onderdeel het begeeft, dan krijgen de andere Markov-ketens deze informatie door en kunnen ze zelf naar een andere toestand gaan. Om het aantal systeemtoestanden niet uit de hand te laten lopen, gebruiken de onderzoekers een aantal technieken. Ze kijken welke Markov-ketens ze kunnen samenvoegen tot een enkele Markov-keten en ook naar hoe ze het aantal schakels in elke individuele Markov-keten kunnen minimaliseren. Hiervoor nemen ze alleen de wezenlijke systeemeigenschappen mee, zoals of een reserveonderdeel wel of niet in gebruik is. In bepaalde gevallen kunnen ze het langzaam falen van twee systemen samenvatten door het snel falen van één systeem. Slimme algoritmen beslissen wanneer zoiets kan.

Stoelinga: "Voor een aantal eenvoudige voorbeeldsystemen, zoals het hartpompsysteem en een computersysteem met vier groepen van elk vier processoren, hebben we laten zien dat deze aanpak werkt. Onze techniek maakt de faalanalyse sneller en beter dan die van de traditionele technieken. Een grote winst is dat wij submodules veel flexibeler aan elkaar koppelen. Voor een ander deel zit de winst in het feit dat wij equivalente toestanden maar eenmaal analyseren. Voor sommige toepassingen blijkt het aantal toestanden dan met een factor tien af te nemen."

Om deze faalanalyse toe te passen, moet je eerst het originele systeemontwerp vertalen in een foutenboom. Op deze foutenboom kun je de faalanalyse loslaten. Nadeel van deze aanpak is dat je elke verandering in het systeem eerst moet vertalen in een veranderde

foutenboom. Het zou veel handiger zijn als je de faalanalyse meteen op het systeemontwerp kunt loslaten. Het systeemontwerp bevat immers al veel informatie, bijvoorbeeld over welke reserveonderdelen erin zitten.

Stoelinga: "Voor dit doel hebben wij de architectuurtaal *Arcade* ontwikkeld. Arcade is een taal die op basis van de systeemarchitectuur meteen de faalanalyse uitvoert. Om in de toekomst meer inzicht te krijgen in de kwaliteit van een op software gebaseerd systeem, kun je eraan denken om een dergelijke architectuurtaal als een industriële kwaliteitsstandaard in te voeren. Met dat idee in ons achterhoofd, gaan we onze architectuurtaal in vervolgonderzoek toepassen op een watermanagementsysteem en een elektriciteitsdistributiesysteem." ●

Logica legt systeemfouten bloot

Sommige softwarefouten komen helaas pas in de praktijk aan het licht. Nieuwe analysemethoden kunnen die fouten al bij het softwareontwerp ontdekken.

Hoe krachtiger de computerhardware in de loop van de decennia is geworden, hoe uitgebreider en gecompliceerder ook de software. Immers, als je meer computerkracht ter beschikking hebt, wil je die ook ten volle benutten. Daarnaast wordt software steeds meer in alledaagse apparaten en vervoersmiddelen geïntegreerd, denk aan mobiele telefoons, auto's en vliegtuigen.

Maar kunnen we wel vertrouwen op software die op zulke kritische plekken zit? Iedere computergebruiker heeft wel eens de ervaring gehad dat het besturingssysteem blijft hangen en dat er niets anders op zit dan het systeem eerst uit te schakelen en daarna weer in te schakelen. Nu kun je op je eigen computer vaak wel even wachten, maar software in een auto of een vliegtuig moet precies op tijd zijn werk doen. Je wilt dan absoluut zeker weten dat de software niet vastloopt doordat het meerdere opdrachten tegelijk niet aan kan en je wilt absoluut zeker weten dat de besturing van de wielen in je auto of de motoren van het vliegtuig op tijd en correct reageren.

In het eerste geval gaat het om reactieve systemen; in het tweede geval om tijdkritische reactieve

Tim Willemse

systemen. Tim Willemse van de Technische Universiteit Eindhoven heeft samen met zijn collega's een methode ontwikkeld om reactieve en tijdkritische software te analyseren en mogelijke systeemfouten bloot te leggen.

Deadlock

Een vaak voorkomend softwareprobleem is dat twee componenten op elkaar staan te wachten, zonder dat er iets gebeurt. De software is dan in een impasse terechtgekomen en de gebruiker kan niets meer doen. Dat is een *deadlock*. Een tweede vaak voorkomend probleem treedt op wanneer het systeem blijft hangen in een zoekopdracht. Dat heet een *livelock*. Het systeem is dan wel bezig, maar voor de gebruiker is dit niet zichtbaar: het systeem hangt. Op het scherm verschijnt dan bijvoorbeeld een zandlopertje dat zich maar blijft omkeren, of een zaklampje dat maar blijft zoeken. "Bij de analyse van reactieve systemen zijn dit twee van de meest rudimentaire checks die we op de software uitvoeren", zegt Willemse.

Willemse gebruikt een procesalgebra (mCRL2) waarmee de software beknopt wordt beschreven en efficiënt kan worden geanalyseerd. De procesalgebra bevat operatoren die bijvoorbeeld parallelle processen, communicatie tussen processen en volgorde van processen kan weergeven. Daarnaast kun je processen beschrijven waarvan je niet weet of A gebeurt of B. En je weet zelfs niet wat de kans is dat A gebeurt of B. Zulke processen heten niet-deterministische processen.

Willemse: "Systemen die we in deze procesalgebra beschrijven, kunnen we op een wiskundige manier onderwerpen aan eisen zoals 'het systeem moet vrij zijn van deadlocks' of 'als de gebruiker op een startknop drukt, moet het systeem een reactie geven'. Wij hebben een methode ontwikkeld waarmee we dit soort eigenschappen voor steeds complexere software automatisch kunnen verifiëren, iets wat voorheen niet kon." (zie kader)

Pacemaker

Samen met enkele afstudeerstudenten heeft Willemse deze formele analyse toegepast op commerciële software, die door verschillende bedrijven ter beschikking is gesteld. Zo analyseerden ze bijvoorbeeld de software

Formele softwareanalyse

Procesalgebra is een wiskundige taal waarmee je kunt redeneren over wat er gebeurt in een willekeurig systeem met aan elkaar gekoppelde processoren of computers. De algebra beschrijft hoe toestanden in zo'n systeem veranderen. De procesalgebra mCRL2 is een uitbreiding van de algebra voor communicatieprocessen (ACP) door rekening te houden met data en met het tijdsaspect. De filosofie van mCRL2 is gebaseerd op processen die acties kunnen uitvoeren. De toestand van een proces beïnvloedt de mogelijke acties die het proces kan uitvoeren en de uitvoering van een actie kan op zijn beurt de toestand veranderen. Je kunt verschillende processen met elkaar combineren via algebraïsche operatoren om zo nieuwe processen te vormen. De rijke datataal wordt gebruikt om het werkelijke systeemgedrag doeltreffend en beknopt te beschrijven. Elk proces correspondeert met een potentieel oneindig grote toestandsruimte, die alle toestanden voorstelt waarin het proces zich kan bevinden.

mCRL2 vertaalt elk complex systeem van honderden of zelfs duizenden processen in een enkel lineair proces. Gegeven een lineair proces en een serie wiskundig omschreven eisen waaraan een proces moet voldoen, genereer je vervolgens een stelsel van vergelijkingen die je formuleert in een uitbreiding van de predicaatlogica (Parameter Boolean Equation System). In het algemeen is het oplossen van zo'n stelsel van vergelijkingen onbeslisbaar, wat wil zeggen dat je niet kunt weten of je überhaupt wel een oplossing kunt berekenen. De uitdaging is nu om dit systeem van vergelijkingen toch op te lossen. Willemse en zijn collega's gebruiken een aantal trucs om toch een oplossing te vinden, bijvoorbeeld het identificeren van parameters die de oplossing niet beïnvloeden, of het opdelen van oneindige domeinen zoals natuurlijke en reële getallen. Wanneer je het stelsel van vergelijkingen uiteindelijk toch hebt opgelost, vertelt de oplossing of het proces wel of niet aan de gestelde eisen voldoet.

die geïntegreerd in een pacemaker zit. Dit stukje software moet adequaat reageren op alle mogelijke hartritmen. "Toen we onze methode op de software van de pacemaker toepasten, kwam er een fout aan het licht", vertelt Willemse over de resultaten. "De pacemaker kon in een deadlock terechtkomen en dat is wel het laatste wat je bij een pacemaker wilt. Die fout was weliswaar al bekend bij de fabrikant, maar onze methode kon ook laten zien waardoor de fout ontstond en dat wist de fabrikant nog niet."

Automatische parkeergarage in Chinatown, New York

Een andere toepassing die de onderzoekers met hun methode tegen het licht hielden, was de software van een automatische parkeergarage. Hierbij verplaatst een automatisch systeem je auto van een parkeerplatform naar een vrije plaats in een efficiënt geordende parkeerflat. Bij de analyse kwam een aantal softwarefouten aan het licht dat nog niet bekend was bij de fabrikant. Willemse: "Zo kan de software in een toestand terechtkomen waarbij de auto door twee uit elkaar bewegende platforms in tweeën wordt getrokken. Ook bleek het mogelijk dat een lift twee auto's tegelijk kon beschadigen."

De toepassingen die de onderzoekers tot nu toe hebben geanalyseerd gaan alleen over reactieve systemen. De volgende stap is om de methode ook toe te passen op tijdkritische reactieve systemen, waarin een bepaalde actie met honderd procent zekerheid vóór een bepaald moment moet gebeuren. "We hebben onze methoden zo ontworpen dat we zowel tijdkritische als niet-tijdkritische reactieve systemen kunnen analyseren", besluit Willemse. ◉

Alledaagse informatica

Hoe werkt Peer2Peer-bestandsuitwisseling?

Wie via de website uitzendinggemist.nl een radio- of tv-uitzending beluistert of bekijkt, wordt bediend door één centrale server. Zo gaat het bijna altijd met surfen op het web. Maar het kan ook anders. Een *Peer-2-Peer*-netwerk (P2P) heeft geen centrale server met daarop alle bestanden, maar alle bij het netwerk aangesloten computers vormen samen één groot, decentraal netwerk. Elke computer is een *peer* – een gelijke – voor elke andere computer, vandaar de naam Peer-2-Peer.

Via een P2P-netwerk kun je bestanden downloaden van alle andere computers, zonder dat je zelf weet van welke computer de informatie wordt opgehaald en zonder dat er een de leiding heeft. Om P2P-bestandsuitwisseling goed te laten functioneren, moeten er wel genoeg mensen informatie uploaden. Met alleen mensen die wel downloaden maar nooit uploaden lukt het niet.

Doordat wereldwijd nu zo'n één miljard mensen op het internet zijn aangesloten, kun je via een P2P-netwerk heel veel informatie uitwisselen. P2P-netwerken worden vooral gebruikt voor het delen van muziek en films, maar het kan met alle digitale bestanden. Deels gebeurt dat legaal, maar deels ook illegaal, bijvoorbeeld met muziek en films waarop copyright berust (zoals bij de bekende website *The Pirate Bay*).

Al jarenlang groeit het aandeel van P2P-bestandsuitwisseling. In 2006 bestond maar liefst tweederde van al het internetverkeer uit P2P-verkeer, driemaal zoveel als het verkeer door webbrowsing. Ruim zeventig procent van dit P2P-verkeer bestond uit het downloaden van films en video's.

Onderzoekers van de Technische Universiteit Delft hebben laten zien dat een effectief P2P-platform aan vier kenmerken moet voldoen. Ten eerste moet het goede van slechte bijdragen kunnen onderscheiden. Sommige bijdragen zullen immers fouten bevatten en andere zijn niet meer dan spam of sabotagepogingen. Een van de manieren waarop dat scheiden gebeurt, is het benutten van het commentaar dat mensen op een bijdrage leveren of van een kwantitatieve beoordeling. Ten tweede moet het P2P-platform goed kunnen omgaan met de beschikbare processorsnelheden, diskruimte en bandbreedte. Ten derde moet de groepscommunicatie technisch gezien effectief verlopen en ten vierde moet er bij die groepscommunicatie ook in sociaal opzicht een groepsgevoel ontstaan dat sociale controle kan uitoefenen.

Pirate Bay

P2P-netwerken komen in twee smaken voor: gestructureerde en ongestructureerde netwerken. In een gestructureerd

P2P-netwerk is er een centrale locatie (de *tracker*) die het downloaden van bestanden coördineert. De tracker zelf levert geen bestanden, maar houdt alleen bij wie welk bestand al heeft en wie nog bezig is met downloaden. De populaire maar illegale downloadsite *The Pirate Bay* is een voorbeeld van een gestructureerd P2P-netwerk dat een zogeheten *Bittorrent-tracker* gebruikt.

Stel, Alice wil een film downloaden in een gestructureerd P2P-netwerk. Dan vertelt de tracker aan de computer van Alice in hoeveel stukken de film is opgedeeld en welke aangesloten computers welke stukken al in hun bezit hebben. Stel dat de tracker ziet dat de computers van Bob en Charlie al bezig zijn de film te downloaden, maar dat ze allebei nog stukken missen. De computer van Alice gaat dan eerst naar een computer waarop delen staan die Bob en Charlie nog niet hebben en deze worden gedownload. Nu heeft Alice iets wat Bob en Charlie niet hebben, terwijl Bob en Charlie iets hebben wat Alice niet heeft. Vervolgens geeft de computer van Alice aan Bob en Charlie het stuk dat zij nog niet hebben en de computers van Bob en Charlie geven aan de computer van Alice de stukken van de film die zij nog niet heeft. Zo hebben ze aan het eind van de bestandsuitwisseling alle drie de gehele film.

Freenet

Een ongestructureerd P2P-netwerk heeft geen tracker die het downloadproces coordineert. Twee voorbeelden van ongestructureerde P2P-netwerken zijn *Gnutella* (veel gebruikt voor muziekdownloads) en *Freenet* (populair om in landen waar de

vrije pers onder druk staat onwelgevallige informatie onder de aandacht te brengen).
Stel dat Alice een MP3-liedje wil downloaden in een ongestructureerd P2P-netwerk. Dan moet haar verzoek eerst bij zoveel mogelijk andere computers in het netwerk terechtkomen. Haar computer moet wel al een of meer andere computers in het netwerk kennen. Als dat het geval is, gaat haar verzoek naar elk van deze computers. Deze computers zoeken op hun harde schijf naar het liedje. Als ze het liedje vinden, melden ze – met vermelding van IP-adres en bestandsnaam – aan de computer van Alice waar deze het liedje kan ophalen.
Tegelijkertijd sturen deze computers het verzoek door naar de computers waar zij zelf in het netwerk mee zijn verbonden, voor het geval dat ze het liedje zelf namelijk niet hebben. Dit doorstuurproces herhaalt zich een keer of zeven. Als elke computer dan met vier andere is verbonden, kan de computer van Alice 4^7 (16.384) computers doorzoeken. Hoewel dit een simpele en effectieve strategie is, bestaat er geen garantie dat het gezochte bestand wordt gevonden. Populaire bestanden uit het netwerk zal de computer van Alice gemakkelijk vinden, maar de kans is groot dat zeldzame bestanden niet opduiken.
De ontwikkeling van geavanceerdere P2P-netwerken gaat nog steeds door. Zo passen pionierende bedrijven het P2P-principe van zelforganisatie toe in de geldleensector. P2P-*lending* is een aanpak waarbij geldschieters en leners elkaar vinden zonder tussenkomst van banken. Verder ondersteunt de Europese Unie het onderzoeksprogramma P2P-*next*, dat volgende generatie P2P-systemen ontwikkelt met openbare broncodes. P2P-netwerken brengen de productie van informatie steeds meer in handen van de consumenten zelf, in plaats van in de handen van enkele commerciële producenten. Ze maken het steeds gemakkelijker voor individuen om hun creaties – of het nu gaat om muziek, film, video, computerprogramma's, tekst of wat dan ook – aan een breed publiek te verspreiden zonder tussenkomst van uitgevers en distributeurs.

In 2006 bestond maar liefst tweederde van al het internetverkeer uit P2P-verkeer, driemaal zoveel als het verkeer door webbrowsing.

Met dank aan dr. ir. Johan Pouwelse, onderzoeker Peer-2-Peer-technologie van de Technische Universiteit Delft.


VISA

Rekenen aan de gedroomde kwantumcomputer

Fundamenteel onderzoek naar de grondslagen van de kwantummechanica levert onverwacht praktisch inzicht in de bouweisen van een toekomstige kwantumcomputer.

Een van de ultieme dromen van natuurkundigen en informatici is het bouwen van een kwantumcomputer. Zo'n kwantumcomputer zou sommige rekenproblemen veel sneller kunnen oplossen dan willekeurig welke klassieke computer ooit voor elkaar kan krijgen. Een gewone computer rekent met een bit als eenheid van informatie: een 0 of een 1. Een kwantumcomputer is een fundamenteel nieuw type computer, die rekent volgens de wetten van de kwantummechanica.

Het equivalent van een bit in de kwantumwereld is een kwantumbit. Een kwantumbit is niet 0 of 1, maar kan tegelijk 0 en 1 zijn. Preciezer gezegd: een kwantumbit kan zich in een *superpositie* van 0 en 1 bevinden. Als er een kans p is dat het kwantumbit bij de meting 0 is, dan is er een kans $(1-p)$ dat het kwantumbit bij meting een 1 is. Deze kwantumeigenschap biedt ongekende rekenmogelijkheden. Met twee klassieke bits kun je vier combinaties vormen: 00, 01, 10 en 11, maar nooit tegelijk. Twee kwantumbits kunnen zich echter *tegelijk* in een superpositie van die vier toestanden bevinden. Dus waar N klassieke bits 2^N toestanden kunnen maken, maar

niet tegelijk, kunnen N kwantumbits 2^N toestanden *tegelijk* maken.

Verstrengeling

Wat ook nieuw is aan een kwantumcomputer ten opzichte van een klassieke computer, is het fenomeen *verstrengeling*. Bij twee klassieke bits heeft de waarde van het ene bit geen invloed op de waarde van het andere. Kwantumbits kunnen daarentegen verstrengeld zijn. Stel, je genereert twee kwantumbits die zich in een superpositie van de twee toestanden 00 en 11 bevinden. Dat paar heet een EPR-paar. Vervolgens verwijder je de twee kwantumbits van elkaar, waarna je ze allebei tegelijk gaat meten. Als het eerste kwantumbit dan een 1 is, moet volgens de kwantummechanica het andere kwantumbit ook een 1 zijn. En als het eerste een 0 is, dan is het tweede dat ook.

De foutenmarge van de kwantumcomputer

Rekenprocessoren bestaan uit logische poorten en er is altijd een kans dat een poort stuk is. Bij een kwantumcomputer speelt de extra complicatie dat kwantumbits in het kwantumgeheugen hun superpositie kunnen verliezen (decoherentie), bijvoorbeeld als ze onvoldoende van de omgeving zijn afgeschermd. Om te zorgen dat foute poorten en decoherente kwantumbits geen invloed hebben op de uitkomsten van de computer, wil je weten hoeveel fouten de computer tolereert. De ondergrens voor de fouttolerantie van een kwantumcomputer is ongeveer een tienduizendste. Dat betekent dat een kwantumcomputer die in minder dan 1 op de 10.000 logische poorten een fout bevat nog steeds vrijwel zeker de goede uitkomsten levert. Nu kun je ook een bovengrens definiëren. Als de foutenmarge boven deze grens uitkomt, worden de uitkomsten van de kwantumcomputer volkomen onbetrouwbaar. Tot voor kort lag die bovengrens bij 55 procent. Buhrman en zijn collega's hebben met hun theoretische verkenningen van de super-kwantumwereld aangetoond dat de bovengrens verlaagd kan worden naar 40 procent. De uitdaging is om zo precies mogelijk te achterhalen wat de onder- en bovengrenzen zijn voor de fouttolerantie van een kwantumcomputer. Deze grenzen vertellen een experimentator hoe goed hij kwantumbits moet opslaan en bewerken om een werkende kwantumcomputer te bouwen.

De meting van het ene kwantumbit verandert onmiddellijk de toestand van het andere kwantumbit, hoe ver de twee ook van elkaar verwijderd zijn.

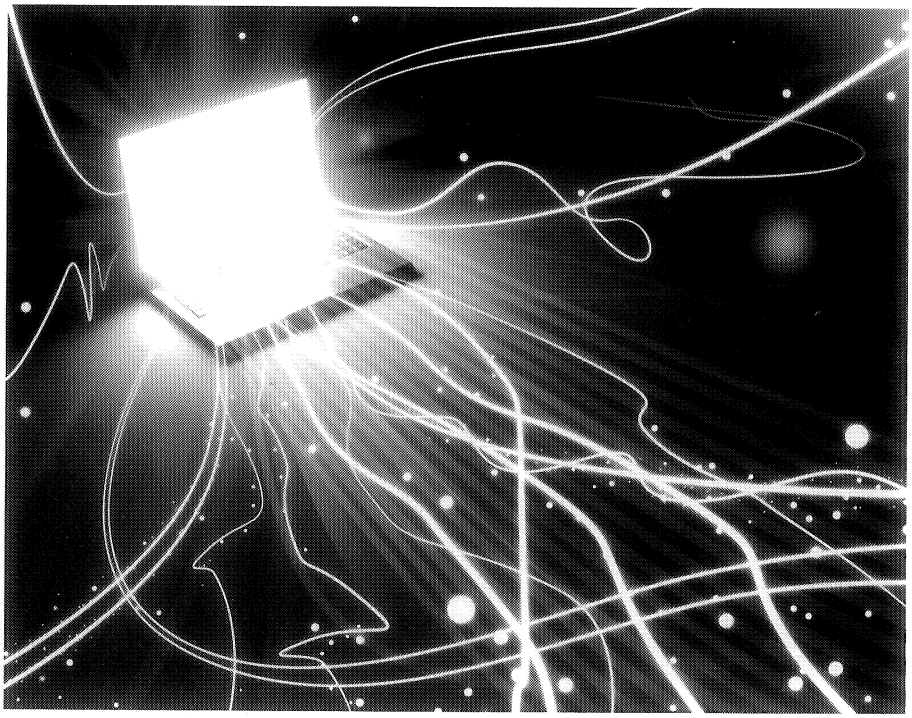
"Verstrengeling en superpositie zijn zo contra-intuïtief", vertelt hoogleraar Harry Buhrman van het Centrum Wiskunde & Informatica (CWI), "dat natuurkundigen nog steeds worstelen met de vraag waarom de kwantummechanica zo in elkaar zit als ze zit. Nu onderzoek ik zelf problemen die een toekomstige kwantumcomputer veel sneller kan oplossen dan een klassieke computer. Het interessante is dat mijn werk automatisch raakt aan de vraag naar de fundamentele aard van de kwantummechanica."

Een van de problemen die een kwantumcomputer veel sneller kan oplossen dan een klassieke computer, is het agendaprobleem. Stel, Alice woont in Amsterdam en Bob in New York. Ze proberen een afspraak te maken door hun agenda's te raadplegen. Als de agenda N bits bevat, dan moeten ze in het algemeen al die N bits uitwisselen om een geschikte dag voor een afspraak te vinden. Maar als ze gebruik zouden maken van de EPR-paren uit de kwantumwereld, hoeven ze maar √N bits uit te wisselen. "Door gebruik te maken van verstrengeling", zegt Buhrman, "kun je sommige communicatieproblemen met de uitwisseling van veel minder bits oplossen."

Super-kwantumwereld

Om de kwantummechanica op een nieuwe manier te testen, verzonnen Clauser, Horne, Shimony en Holt in 1969 het theoretische CHSH-spel. Het spel kent twee spelers: Alice en Bob. Zij hebben van tevoren een strategie met elkaar afgesproken, maar mogen tijdens het spel niet met elkaar communiceren. Zij krijgen ieder één bit als invoer, zeg x en y. Vervolgens moeten ze één bit als uitvoer genereren, zeg a en b. Zij winnen het spel als ze erin slagen dat '$x \otimes y$= a XOR b' (a XOR b = 0 als a en b allebei 0 of allebei 1 zijn, anders is de uitkomst 1).

In de klassieke wereld kunnen Alice en Bob dit spel met een kans van 75 procent winnen en in de kwantumwereld – met gebruik van een EPR-paar – met een kans van iets meer dan 85 procent. "Nu kun je ook een soort super-kwantumwereld definiëren", vertelt Buhrman, "waarin je het spel altijd wint, dus met een kans van 100



procent. Het bijzondere van deze super-kwantumwereld is dat je daarin sommige moeilijke communicatieproblemen met de uitwisseling van slechts één bit kunt oplossen. En in deze super-kwantumwereld geldt nog steeds de regel dat je informatie niet sneller dan het licht kunt versturen, zoals dat in de echte kwantumwereld ook niet kan."

Buhrman heeft met zijn collega's het regime onderzocht waarin Alice en Bob het CHSH-spel kunnen winnen met een kans die tussen de 85 procent van de echte kwantummechanica en de 100 procent van de super-kwantummechanica in ligt. Zij bewezen dat als je de kans om het spel te winnen verlaagt van 100 naar 90 procent, het oplossen van sommige moeilijke communicatieproblemen nog steeds met een enkel bit kan, zoals in de ideale super-kwantumwereld. Buhrman: "Bij een winstkans van 85 procent, zoals in de echte kwantumwereld, heb je veel communicatie nodig om sommige van die moeilijke problemen op te lossen. Maar bij een winstkans van 90 procent volstaat altijd slechts dat ene bit. Dit resultaat laat zien dat er een scherpe overgang is van een wereld waarin

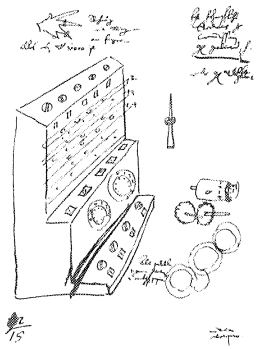
sommige communicatieproblemen moeilijk zijn op te lossen en een wereld waarin ze juist triviaal zijn."

Het bewijs dat tot dit resultaat heeft geleid lijkt in eerste instantie alleen van fundamenteel belang voor een beter begrip van de kwantumwereld. Toch blijkt het onverwachte toepassingen te hebben. Buhrman en zijn collega's hebben het gebruikt om een beter inzicht te krijgen in de bouweisen van een toekomstige kwantumcomputer (zie kader p. 118). Desondanks zijn we echter nog ver weg van de bouw van een echte kwantumcomputer. Voorlopig is het in laboratoria alleen nog maar gelukt om simpele berekeningen met enkele kwantumbits uit te voeren. ●

Mijlpalen van de informatica en haar toepassingen

De informatica heeft haar wortels in de eeuwenlange zoektocht naar geautomatiseerde manieren om te rekenen en informatie te verwerken. Als wetenschap ontstond ze in de jaren vijftig van de twintigste eeuw. Sindsdien heeft het vakgebied zich snel uitgebreid. Ontwikkeling van hardware, software, computerarchitectuur, algoritmen, netwerken, kunstmatige intelligentie, multimedia, cryptografie, datamining – het behoort allemaal tot de informatica.
Dit uitgebreide historisch overzicht geeft belangrijke ontwikkelingen in de informatica en haar toepassingen weer, met speciale aandacht voor Nederlandse bijdragen (aangegeven met (NL)). Het laat zien hoe de ontwikkeling van kennis en kunde in de informatieverwerking uiteindelijk kan leiden tot alledaagse toepassingen. Op dezelfde manier kan het fundamentele informaticaonderzoek dat in dit boek staat beschreven aan de basis staan van toekomstige toepassingen.

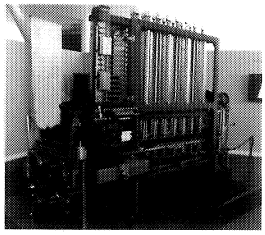
1623
De Duitser Wilhelm Schickard bouwt de eerste **rekenmachine**. Later volgen de rekenmachines van de Fransman Blaise Pascal (1642) en de Duitser Gottfried Wilhelm Leibniz (1672). Deze rekenmachines waren niet erg praktisch en geavanceerd. Ingewikkelde berekeningen werden vanaf de achttiende eeuw soms door een zaal met tussen de zestig en tachtig mensen uitgevoerd, als een soort parallelle menselijke computer. Tot in de eerste helft van de twintigste eeuw bleef het rekenen grotendeels iets wat door mensen werd gedaan, met gebruik van mechanische rekenhulpmiddelen zoals de latere kantoorrekenmachines.



1801
De Fransman Joseph-Marie Jacquard ontwikkelt een manier om weefmachines aan te sturen met behulp van een rij ponskaarten. Het is een van de vroegste voorbeelden van het **idee van programmeren**.

1832
De Engelsman Charles Babbage bouwt een prototype van de **Difference Engine**, een mechanische rekenmachine. Het prototype was te klein om praktisch bruikbaar

te zijn. De eerste werkende versie werd pas in 1991 gebouwd bij het London Science Museum.

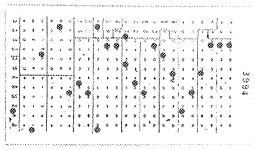


1834
Charles Babbage ontwerpt de **Analytical Engine**, een apparaat dat wordt gezien als de eerste mechanische computer. De *Analytical Engine* heeft een processor, een geheugen en een manier voor informatie-input (ponskaarten) en informatie-output. De technische realisatie bleek echter te moeilijk en Babbage slaagde er niet in de machine te bouwen.

1842/1843
De Engelse Lady Ada Lovelace beschrijft hoe de *Analytical Engine* Bernoulli-getallen kan berekenen. Dit werk wordt tegenwoordig beschouwd als het **eerste computerprogramma** en Ada Lovelace als de eerste computerprogrammeur, hoewel het idee van programmeren in de tijd van Babbage en Lovelace niet bestond. De programmeertaal Ada is naar haar vernoemd.

1847
De Engelsman George Boole zet in de publicatie *Mathematical Analysis of Logic* uiteen wat nu **Boole-algebra** heet. Deze nieuwe algebra werkt met de getallen 0 en 1 en met logische operatoren die in de elektronica AND, OR en NOT kwamen te heten. Pas in de jaren 1930 zag Claude Shannon het praktische belang van de Boole-algebra voor het moderne compute rrekenen.

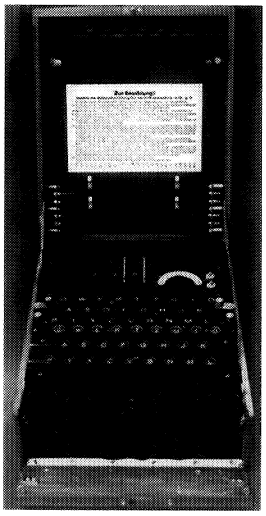
1890
De Amerikaan Hermann Hollerith bouwt een ponskaartmachine om de volkstelling in de VS efficiënt te verwerken. Waar de verwerking van de volkstelling in 1880 nog zeven jaar kostte, klaart de **Tabulating Machine** van Hollerith de klus in zes weken. (Uitkomst: iets meer dan 62 miljoen inwoners.)



1896
Hollerith richt de *Tabulating Machine Company* op. Dit bedrijf wordt in 1924 omgedoopt tot **IBM: International Business Machines**. Eind jaren 1920 herontwerpt IBM de bestaande ponskaart tot een nieuwe IBM-ponskaart, die populair zou worden. Tot 2004 bleef IBM zelf nog computers fabriceren. De grote verdienste van IBM voor de informatica is vooral de ontwikkeling van het systeemdenken in de computerarchitectuur.

1918
De Duitser Arthur Scherbius patenteert de beroemdste codeermachine aller tijden, de elektromechanische **Enigma** – een rotormachine. De Duitsers gebruikten de Enigma tijdens de Tweede Wereldoorlog om gecodeerde berichten te versturen. In

1915 hadden de wetenschappers T. A. van Hengel en R.P.C. Spengler van de Nederlandse marine trouwens al een concept van een rotor-codeermachine uitgewerkt, voor zover bekend als eerste.



1929
De Amerikaanse ingenieur Vannevar Bush voltooit de bouw van de **Differential Analyzer**, een analoge computer die tijdrovende, gewone differentiaalvergelijkingen kan oplossen.

1932
De Pool Marian Rejewski **kraakt** als eerste de Duitse Enigma-codering. Tijdens de Tweede Wereldoorlog speelt Alan Turing een grote rol bij het kraken van de door de Duitsers verbeterde Enigma-codering. De prestaties van Rejewski en Turing vormen twee hoogtepunten uit de cryptografie, een specialisatie binnen de informatica.

1936
De Engelsman Alan Turing schrijft de sleutelpublicatie *On computable numbers, with an application to the Entscheidungsproblem*. Turing bedenkt hierin een abstract rekenmodel waarop alle bekende rekenprocessen zijn na te bootsen: de **Turingmachine**. Hij toont ook het bestaan aan van een universele computer: een Turingmachine die in staat is om elke andere Turingmachine na te doen, gegeven het programma van die machine. Het is de basis van ons moderne begrip van de '*general purpose computer*'. Ook bewees Turing het bestaan van problemen die niet in eindige rekentijd beslisbaar of oplosbaar zijn, zoals het stopprobleem (*halting problem*) voor Turingmachines zelf. Verder beweert de Church-Turing-hypothese dat elke functie die wij intuïtief berekenbaar zouden noemen, berekenbaar is met een Turingmachine.

1937
De Amerikaan Claude Shannon beschrijft in zijn afstudeerscriptie *A symbolic analysis of relay and switching circuits* hoe de principes van de Boole-algebra praktisch kunnen worden toegepast. Dit werk staat aan de basis van de bouw van **digitale circuits**.

1939-1945
Tijdens de Tweede Wereldoorlog werken allerlei onderzoeksteams aan de ontwikkeling van de eerste **elektronische, programmeerbare computer**. Bekende machines uit deze pioniertijd zijn de Duitse elektromechanische Zuse Z3 (1941), de Engelse Colossus (1943 – voor het kraken van codes), de Amerikaanse ENIAC (1946 – voor ballistische berekeningen) en de Engelse Mark 1 (1948). Toch ontbreekt bij defensieorganisaties nog het besef van de grote mogelijkheden van een *general purpose stored-program computer.*

1945
De Amerikaanse Hongaar John von Neumann legt in het *First Draft of a Report on the EDVAC* een belangrijke theoretische basis voor de universele computerarchitectuur bestaande uit een rekeneenheid, een controle-eenheid een input-output-eenheid en een geheugen. Deze **Von Neumann-architectuur** maakt van

de computer een universele machine in plaats van een machine die maar één type computerprogramma kan verwerken.

1945
In het essay *As we may think* beschrijft de Amerikaan Vannevar Bush het informatiemanagementsysteem '**memex**'. Dit is het prototype van een hypertext computersysteem zoals we dat nu kennen met het internet.

1946
J. Presper Eckert en John Mauchly voltooien de **ENIAC**, de eerste praktisch volledig elektronische computer. De ENIAC was ontworpen voor het oplossen van gewone differentiaalvergelijkingen. Het apparaat woog 30.000 kilogram, verbruikte 150 kilowatt per uur (ruim duizendmaal zoveel als die van een moderne pc) en kon vijfduizend optellingen per seconde uitvoeren.

1946
John von Neumann, Stanislaw Ulam en Nick Metropolis bedenken de **Monte Carlo-simulatiemethode**, een wiskundige methode die later in veel computersimulaties toegepast zal worden.

1947
De uitvinding van de **transistor** door John Bardeen, Walter Brattain en William Shockley maakt de ontwikkeling van de microprocessor mogelijk. Essentieel voor het produceren van de geminiaturiseerde rekeneenheid van de moderne computer.

1947
De Amerikaan George Dantzig ontwerpt **het simplex-algoritme** voor de oplossing van problemen uit de lineaire programmering. Het simplex-algoritme is sindsdien de basis voor vele praktische LP-methoden en softwarepakketten zoals CPLEX.

1948
De Amerikaan Claude Shannon legt in het artikel *A mathematical theory of communication* de wiskundige basis voor de **informatietheorie**.

1949
Maurice Wilkes en William Renwick voltooien aan Cambridge University in Engeland de bouw van de eerste stored-program elektronische computer: **de EDSAC** (Electronic Delay Storage Automatic Calculator). Een *stored-program* computer houdt de programmainstructies in een RAM-geheugen, in tegenstelling tot bijvoorbeeld de ENIAC die handmatig met schakelaars 'geprogrammeerd' moest worden. Wilkes en anderen ontwikkelden ook een methode om computerinstructies in een symbolische vorm te noteren.

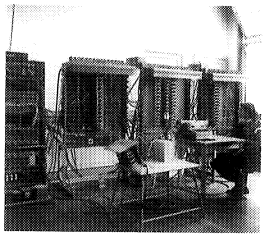
1949
Claude Shannon legt in het artikel *Communication Theory of Secrecy Systems* de basis voor de **moderne cryptografie** en **cryptoanalyse**.

1950
Alan Turing beschrijft in het artikel *Computing, Machinery and Intelligence* de **Turingtest** om te bepalen of een machine kan denken.

1951
J. Presper Eckert en John Mauchly vol-

tooien de **UNIVAC I**, de eerste elektronische computer voor 'massaproductie' (er werden er 46 van gemaakt).

1952 (NL)
Carel Scholten en Jan Loopstra bouwen in opdracht van Aad van Wijngaarden op het toenmalige Mathematisch Centrum in Amsterdam (het huidige CWI) de *eerste Nederlandse computer: de* **ARRA**. Op 21 juni 1952 wordt de eerste computer van Nederland officieel in gebruik genomen.



1953 (NL)
Carel Scholten, Jan Loopstra en Gerrit Blaauw bouwen de **ARRA II**, ook op het Mathematisch Centrum in Amsterdam. Hiermee worden bijvoorbeeld de vliegtuigvleugels van de F27 Fokker Friendship doorgerekend. Het ontwerp van de ARRA II was van Blaauw.

1953 (NL)
Onder leiding van Leen Kosten voltooit Willem van der Poel bij het Centraal Laboratorium van de PTT de bouw van de **PTERA**: de PTT Elektronische RekenAutomaat. Kosten was in 1948 de computerontwikkeling bij het Centraal Laboratorium van de PTT gestart.

1956
De *Dartmouth Summer Research Conference on Artificial Intelligence* vormt het begin van het vakgebied **kunstmatige intelligentie** (Artificial Intelligence, AI). John McCarthy, Marvin Minsky, Allen Newell en Herbert Simon behoren tot de pioniers van de AI.

1957
Alan Newell, Cliff Shaw en Herbert Simon creëren de **General Problem Solver** (GPS), een van de eerste grote computerprogramma's in de kunstmatige intelligentie, dat in beginsel elk probleem in geformaliseerde vorm zou moeten kunnen oplossen.

1957
Willem van der Poel ontwerpt de **ZEBRA**, die in Engeland wordt geproduceerd en in 1958 in Nederland op de markt komt.

1957
IBM brengt de eerste hogere programmeertaal op de markt: **FORTRAN** (afkorting van *Formula Translator*). In de jaren zestig verschijnen de programmeertalen ALGOL, COBOL, PL/I en BASIC en in de jaren zeventig Pascal.

1958 (NL)
Electrologica, opgericht door levensverzekeringsmaatschappij Nillmij en het Mathematisch Centrum, brengt de eerste commerciële Nederlandse computer op de markt: de **X1**. Het bedrijf zou tot 1968 bijna dertig computers bouwen, tot en met de X8. Vooral universitaire rekencentra werkten nog lang met de X8.

1959
Jack Kilby (Texas Instruments) en Robert Noyce (Fairchild Semiconductor Corp) vinden onafhankelijk van elkaar de geïntegreerde schakeling ofwel **chip** uit.

1959 (NL)
De Nederlander Edsger Dijkstra publiceert zijn **kortste-pad-algoritme**, een efficiente methode om de snelste route van A naar B te bepalen. Dit algoritme ligt aan de basis van moderne navigatiesystemen als de TomTom.

1960 (NL)
Introductie van de **hogere program-**

meertaal ALGOL 60, waaraan een team van vijf Nederlandse informaticapioniers onder leiding van Aad van Wijngaarden een bijdrage hebben geleverd. Jaap Zonneveld en Edsger Dijkstra schrijven ook de eerste ALGOL-compiler ter wereld.

1960 (NL)
Nederland telt **37 computers**; de VS al meer dan zesduizend.

1962
De Brit Tony Hoare bedenkt het algoritme **Quicksort**, een van de meest gebruikte sorteeralgoritmen in de wereld.

1964 (NL)
In zijn inaugurele rede aan de Rijksuniversiteit Leiden ('Informatieverwerking en Wiskunde'), introduceert hoogleraar numerieke wiskunde Guus Zoutendijk voor het eerst de term '**informatica**' in de Nederlandse taal.

1964
IBM introduceert **System/360** en het bedrijfssysteem OS/360, een omvangrijk computersysteem met vele innovaties. Wetenschappers als F.B. Brooks, maar ook de Nederlander Gerrit Blaauw droegen bij aan het ontwerp. IBM-mainframes zouden jarenlang toonaangevend zijn, ook in de computerarchitectuur.

1965
Gorden Moore voorspelt dat de computertechniek zo snel zal voortschrijden dat elke twee jaar het aantal componenten op een chip zal verdubbelen. Deze voorspelling staat sindsdien bekend als de **Wet van Moore**. Zijn voorspelling is redelijk juist gebleken, deels omdat de 'wet' voor chipfabrikanten een doel op zichzelf werd.

1965
James Cooley en John Tukey bedenken het **Fast Fourier Transform-algoritme** (FFT), een van de meest gebruikte algoritmen in de toegepaste wiskunde.

1965
Juris Hartmanis en Richard Stearns ontwikkelen een model voor de studie van tijd- en geheugencomplexiteit in berekeningen met een Turingmachine. Zij leggen hiermee de basis voor de **moderne complexiteitstheorie** van berekeningen.

1967 (NL)
Het begin van de ontwikkeling van **AUTOMATH**, onder leiding van de wiskundige N.G. de Bruijn van de Technische Universiteit Eindhoven. AUTOMATH is het eerste prototype van een bewijsverificator, een formalisme voor de weergave van wiskundige bewijzen dat door zijn vorm de wiskundige correctheid automatisch door een computer verifieerbaar maakt.

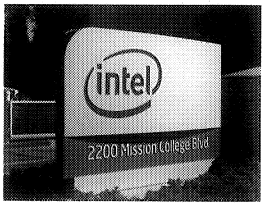
1968 (NL)
De programmeertaal **ALGOL 68** wordt gepresenteerd. Het Mathematisch Centrum in Amsterdam heeft een grote bijdrage geleverd aan de taal. Ook aan de latere herziening en implementatie van de taal is een belangrijke bijdrage vanuit Nederland geleverd (onder andere door Ch.A. Koster, L. Meertens D. Grune en J.C. van Vliet). De definitie van de taal maakt gebruik van de zogenaamde tweeniveaugrammatica's, bedacht door Aad van Wijngaarden.

1968
Peter Hart, Nils Nilsson en Bertram Raphael presenteren het **A*-zoekalgoritme**. Het wordt een van de bekendste zoekheuristieken in de kunstmatige intelligentie,

ontworpen om gebruik te maken van veelbelovende zoekrichtingen op elke moment in het zoekproces.

1968
In Garmisch (Duitsland) vindt van 7 tot 11 oktober de eerste conferentie plaats gewijd aan '**software engineering**'. De conferentie, gefinancierd door de NAVO, wordt algemeen gezien als het begin van de wetenschappelijke ontwikkeling van technieken en methoden voor het bouwen van software.

1968
Gordon Moore en Robert Noyce richten **Intel Corporation** op, dat zal uitgroeien tot de succesvolste chipfabrikant ter wereld.



1968
In een ingezonden brief in de *Communications of the* ACM – getiteld 'Go to statement considered harmful' – pleit Edsger W. Dijkstra voor een **gestructureerde aanpak van het programmeren** en tegen het gebruik van *go-to*-opdrachten, die juist ongestructureerde codes in de hand werken. Dit pleidooi (met andere proponenten zoals D. Gries, C.A.R. Hoare, M.A. Jackson, en N. Wirth) luidde het begin in van de ontwikkeling van vele systematische programmeeraanpakken.

1969
Het eerste boekdeel *Fundamental algorithms* van de invloedrijke serie **The art of computer programming** van Donald Knuth verschijnt. Knuth maakt een rigoureuze wiskundige analyse van algoritmen, probleemoplossende rekenmethoden die door een computer uitvoerbaar zijn.

1969
Bij Bell Labs in de VS wordt het invloedrijke **Unix-besturingssysteem** ontworpen.

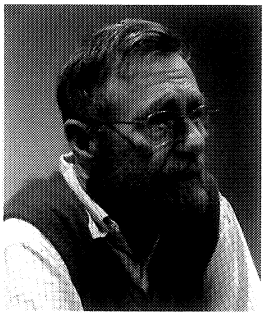
1969
De allereerste versie van het ARPANET wordt aangelegd, ontwikkeld door het *United States Department of Defence*. Het ARPANET zou uitgroeien tot het eerste toonaangevende computernetwerk en de voorloper van het huidige internet. Het legt de basis voor veel latere netwerkprotocollen.

1971
Ray Tomlinson verstuurt 's werelds **eerste e-mail** tussen twee computers op het ARPANET. Hij gebruikt ook als eerste het @-teken in een e-mailadres.

1971/1972
Stephen Cook, Richard Karp en Leonid Levin ontwikkelen het begin van de complexiteitstheorie met NP-compleetheid als centrale begrip en P ≠ NP? als open vraag. Problemen in P zijn snel op te lossen; voor problemen in NP is een gegeven oplossing snel te verifiëren. Het **P-versus-NP-probleem** is een van de belangrijkste open problemen in de complexiteitstheorie van algoritmen.

1971
Intel lanceert de **eerste microprocessor**: de **Intel 4004**. Het is de eerste processor die volledig op één chip is gebouwd.

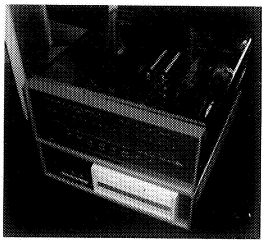
1972 (NL)
Edsger W. Dijkstra wint de Turing Award



als enige Nederlander tot nu toe. De Turing Award wordt beschouwd als de Nobelprijs van de informatica.

1972
Bij Bell Labs in de VS wordt de **programmeertaal C** ontwikkeld, een taal die hand in hand gaat met het Unix-besturingssysteem.

1975
De Altair 8800-microcomputer komt op de markt, het begin van het microcomputertijdperk. De Altair werd ontwikkeld door Ed Roberts en zijn bedrijf MITS. Voor 397 Amerikaanse dollars werd het als bouwpakket verkocht door het Amerikaanse blad *Popular Electronics*.



1975
John Holland publiceert in het boek **Adaptation in Natural and Artificial Systems** baanbrekend werk over genetische algoritmen, een speciale klasse van evolutionaire algoritmen.

1975
Bill Gates en Paul Allen richten **Microsoft** op. Het bedrijf zal een grote rol gaan spelen in de ontwikkeling van software voor pc's en de acceptatie daarvan door particulieren, bedrijven en instellingen over de hele wereld.

1976
Steven Jobs en Steven Wozniak richten **Apple Computer Inc.** op.

1976
Kenneth Appel en Wolfgang Haken gebruiken een computer om het beroemde **vierkleurentheorema** te bewijzen. Het eerste computerbewijs van een grote wiskundige stelling.

1976
De **eerste commerciële supercomputer, de Cray-1,** komt op de markt. Supercomputers gaan een grote rol spelen in de ontwikkeling van de *computational science*, de toepassing van computers op grote fysische modellen zoals de simulatie van weer en klimaat.

1976
Whitfield Diffie en Martin Hellman bedenken de **cryptografie met publieke sleutels**. Het jaar erna presenteren Ron Rivest, Adi Shamir en Len Adleman een implementatie van dit idee met behulp van elementaire getaltheorie, het RSA-schema. Het is nu een van de standaarden voor veilige gegevensuitwisseling, hoewel het voor de veiligheid wel sleutels vereist van ten minste 512 bits.

1976 (NL)
De eerste editie van Andrew Tanenbaums boek **Structured Computer Organization** verschijnt. Het is het eerste in een reeks van zeer succesvolle leerboeken van de hand van Tanenbaum die over de hele wereld gebruikt worden.

1977
Donald Knuth, James Morris en Vaughan Pratt laten zien dat grote tekstbestanden veel sneller op gegeven **patroonteksten** zijn te **doorzoeken** dan ooit werd aangenomen. Een praktische variant werd ontwikkeld door Boyer en J. Strother Moore. Het KMP-algoritme was een doorbraak in de algoritmiek voor teksten.

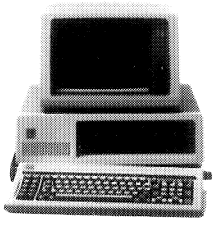
1979
De Russische wiskundige Leonid Khachian bereikt een doorbraak met de **ellipsoïde-methode**, een geheel nieuwe methode

voor het oplossen van problemen uit de lineaire programmering (LP). Deze methode bewijst dat LP-problemen in ieder geval theoretisch in polynomiale rekentijd oplosbaar zijn.

1980
Carver Mead en Lynn Conway presenteren hun **Structured VLSI Design-methodologie**. Door de toenemende complexiteit van geïntegreerde circuits worden computers ingeschakeld voor het ontwerp van hun eigen chips.

1980 (NL)
Jaco de Bakkers boek **Mathematical proof of program correctness** verschijnt, een van de vele resultaten van het onderzoek onder zijn leiding naar de theorie van programma's en de semantiek van programmeertalen bij het Mathematisch Centrum in Amsterdam.

1981
IBM introduceert de **personal computer (pc)**.



1981 (NL)
De Nederlandse overheid erkent de **studierichting informatica**. In de VS en Engeland was dat al veel eerder gebeurd.

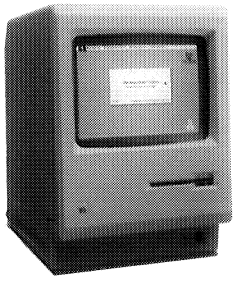
1981
De Amerikaanse natuurkundige Richard Feynman is de eerste die een **kwantumcomputer** voorziet. Charles Bennett en Paul Benioff spelen in de jaren daarna een vooraanstaande rol bij de eerste theoretische verkenningen van een kwantumcomputer.

1982
Het Amerikaanse tijdschrift **'Time' roept de pc uit tot 'Man of the Year 1982'**, de eerste keer dat niet een mens maar een machine wordt gekozen.

1982 (NL)
Jan Bergstra en Jan Willem Klop ontwikkelen het concept van **procesalgebra's** voor het beschrijven, analyseren en ontwerpen van het gedrag van softwaresystemen bestaande uit vele, met elkaar communicerende processen.

1983
Introductie van de objectgeoriënteerde **programmeertaal C++**, ontworpen door Bjarne Stroustrup.

1984
Apple brengt de **Apple Macintosh** op de markt. Deze onderscheidt zich van de IBM-pc door een gebruikersvriendelijk grafisch interface.



1985
Microsoft lanceert de eerste versie van besturingssysteem **Microsoft Windows**.

1985
David Deutsch beschrijft als eerste een **universele kwantumcomputer**. Hij bedenkt ook de kwantumequivalenten voor de klassieke logische poorten AND, OR en NOT.

1988 (NL)
De **eerste trans-Atlantische internetver-**

binding (64 kilobits per seconde) wordt tot stand gebracht door Piet Beertema bij het CWI. Beertema stelde ook als eerste voor om landextensies als .nl voor domeinnamen te gebruiken (1986).

1990
Geboorte van het **World Wide Web**. Tim Berners-Lee toont bij het Europese laboratorium voor deeltjesfysica CERN de eerste World Wide Web-browser en -editor. Deze uitvinding zorgt ervoor dat het internet niet langer alleen een militair of academisch doel heeft, maar een integraal onderdeel van het alledaagse leven gaat worden.

1990
Gene Myers en anderen publiceren het **BLAST-algoritme** (Basic Local Alignment Search Tool), het meest gebruikte algoritme ter wereld. BLAST vergelijkt DNA- of eiwitsequenties met elkaar.

1990 (NL)
Het omvangrijke tweedelige **Handbook of Theoretical Computer Science** onder redactie van Jan van Leeuwen verschijnt. Het is de eerste grote integrale presentatie van de vondsten in de theoretische informatica sinds haar begin.

1991 (NL)
De Nederlander Vic Hayes bedenkt de voorloper van **Wi-Fi** voor draadloze communicatie en werkt daarna aan de acceptatie van Wi-Fi als wereldwijde standaard.

1991
De Fin Linus Torvalds ontwikkelt de **Linux Kernel** vanuit het instructieve Minix-systeem (ontwikkeld door de in Amsterdam werkzame Andrew Tanenbaum). Linux heeft zich ontwikkeld tot een volwaardig bedrijfssysteem voor vele soorten van computers en is een veelgeroemd alternatief voor bijvoorbeeld Microsoft Windows op pc's.

1993
Het computervideospel **DOOM** verschijnt op de markt. Hoewel er al *first-person shooter games* bestonden (zoals Wolfenstein 3D uit 1992), maakt DOOM het genre op wereldschaal populair.

1994
Peter Shor ontdekt een algoritme om grote getallen snel te **factoriseren** met een (nog niet bestaande) kwantumcomputer. Voor gewone computers is zo'n algoritme nog niet gevonden.

1994
James Gosling introduceert de veel gebruikte objectgeoriënteerde programmeertaal **Java**.

1994
De Amerikaan Len Adleman laat experimenteel zien dat grote collecties geschikt DNA-materiaal in beginsel de oplossing van combinatorische problemen kunnen berekenen. Het is het begin van de zogeheten **DNA-computing** en van de meer algemene studie van de 'natural computing' waaraan in Nederland onder andere Grzegorz Rozenberg veel heeft bijgedragen.

1995
Jeff Bezos richt internetbedrijf **Amazon.com** op, een van de eerste internetbedrijven. In hetzelfde jaar wordt ook de elektronische veilingsite eBay opgericht. Het begin van softwareontwikkeling voor webgebaseerde diensten en ontwikkeling van *e-commerce security*.

1997 (NL)
Oprichting van de **Amsterdam Internet**

Exchange (AMS-IX). Hier hebben anno 2009 meer dan driehonderd internetaanbieders, telecombedrijven en mediabedrijven hun netwerken op aangesloten. Daarmee is Amsterdam de internethoofdstad van de wereld (piek: 600 gigabits per seconde).

1997 (NL)
De Nederlander Jaap Haartsen vindt **Bluetooth** uit: de standaard voor draadloze communicatie over korte afstand tussen apparaten zoals mobiele telefoons, laptops, pc's, printers en camera's.

1997
IBM-schaakcomputer **Deep Blue** verslaat wereldkampioen schaken Gary Kasparov, een mijlpaal in de intellectuele strijd tussen mens en machine.

1998
Larry Page en Sergey Brin richten **Google** op. In eerste instantie is Google alleen een internetzoekmachine die met een vernieuwende zoekstrategie al snel de beste en meest gebruikte ter wereld wordt. Google breidt haar diensten in de jaren daarna gestaag uit met e-mail, online kaarten, nieuwsdiensten, sociale netwerken en foto- en videodiensten.

1999
Er ontstaat veel ophef over het '**millenniumprobleem**' dat dreigt te ontstaan omdat in veel pc-software de jaartelling niet automatisch van 1999 op 2000 overgaat, waardoor er chaos in veel computersystemen over de hele wereld kan ontstaan. Tegen het eind van 1999 zijn vrijwel alle systemen aangepast en problemen blijven uit.

2000
Het Clay Mathematics Institute (Cambridge, Massachusetts) noemt het P-versus-NP-probleem bij de zeven belangrijkste problemen die in de eenentwintigste eeuw opgelost zouden moeten worden. Het instituut looft een beloning van 1 miljoen Amerikaanse dollars uit voor de correcte oplossing van elk van de **zeven millenniumproblemen**, dus ook voor het nog altijd openstaande P-versus-NP-probleem.

2001
Christos Papadimitriou laat zien dat er een nauw verband bestaat tussen wiskundige speltheorie, waarin spelstrategieën tussen onafhankelijke spelers worden onderzocht en de ontwikkeling van algoritmen voor internettoepassingen. De **algoritmische speltheorie** heeft sindsdien een grote vlucht genomen en staat model voor bijvoorbeeld het ontwerp van internetveilingen.

2003
Het Amerikaanse internetbedrijf Linden Lab introduceert de virtuele wereld **Second Life**, gebaseerd op een geavanceerd driedimensionaal modelleringsysteem. Deelnemers van over de gehele wereld kunnen hierin een virtueel tweede leven opbouwen en met elkaar omgaan via 'avatars'.

2004
Blizzard Entertainment brengt het computerspel '**World of Warcaft**' uit, het thans meest gespeelde *massively multiplayer online role-playing game*, met zeer geavanceerde 3D- graphics.

2004 (NL)
Introductie van het navigatiesysteem **TomTom** door het gelijknamige Nederlandse bedrijf.

2004 (NL)
Het Nederlandse bedrijf Guerilla Games ontwikkelt het succesvolle spel **Killzone**

voor de Playstation 2, de grootste Nederlandse multimediaproductie aller tijden.

2004
Oprichting van de sociale netwerksite **Facebook**.

2005
Drie Nederlandse banken – ING/Postbank, Rabobank en ABN AMRO – ontwikkelen **iDEAL**, de leidende standaard voor veilige online betalingen bij webwinkels.

2005
Oprichting van **YouTube** en lancering van **Google Earth**.

2006 (NL)
Lex Schrijver en Adri Steenbeek van het CWI voltooien de ontwikkeling van een wiskundig model en een snel algoritme voor de berekening van dienstregelingen van de Nederlandse Spoorwegen, in het bijzonder voor de geheel vernieuwde **NS-dienstregeling 2007**.

2006
Oprichting van microbloggingsite **Twitter**.

2006
Tim Berners-Lee, Wendy Hall, James Hendler, Nigel Shadbolt en Daniel Weitzner nemen het initiatief tot de ontwikkeling van **Web Science**, de eigen informatica van het World Wide Web en haar toepassing in alle sectoren van de maatschappij.

2007
Apple brengt de **iPhone** op de markt, een *smartphone* met een multifunctioneel aanraakgevoelig scherm. De gebruiker kan bellen, e-mailen, internetbrowsen, muziek luisteren, video's bekijken, Google maps raadplegen en gebruikmaken van een steeds groeiende bibliotheek van programmaatjes (apps).

2008 (NL)
Computerdeskundigen onder leiding van Bart Jacobs van de Radboud Universiteit in Nijmegen kraken de Mifare Classic, de wereldwijd meest gebruikte contactloze chipkaart, die ook de basis vormt van de Nederlandse **ov-chipkaart** van dat moment.

2008 (NL)
Wiskundige Marc Stevens (CWI) ontdekt met een internationaal team onderzoekers een zwakke plek in de https-internetbeveiliging. Door het lek was het mogelijk alle beveiligde websites en mailservers na te bootsen.

2009
Stephen Wolfram lanceert antwoordmachine **WolframAlpha** als een aanvulling op de zoekmachine van Google. WolframAlpha is een *computational knowledge* engine, die antwoorden geeft op vragen door te zoeken in een grote database met informatie.

Begin 21e eeuw
De informatica vindt nieuwe uitdagingen in het omgaan met de steeds maar toenemende hoeveelheden geproduceerde data (zoals in wetenschappelijke experimenten, medische imaging en digitale bibliotheken), de complexe problemen uit de aard- en levenswetenschappen (zoals weer- en klimaatmodellen, bio-informatica en systeembiologie), maatschappelijke logistieke problemen (zoals in transport, industrie en gezondheidszorg) en het gebruik van software voor de ontwikkeling van moderne elektronische diensten en elektronische veiligheid. Computers en software worden steeds slimmer.

Beide ontwikkelingen samen stuwen de informatica als wetenschap voort.

Dit overzicht is tot stand gekomen in samenwerking met Jan van Leeuwen, hoogleraar informatica aan de Universiteit Utrecht en met Gerard Alberts, docent geschiedenis van de wiskunde en de informatica aan de Universiteit van Amsterdam. De uiteindelijke keuze van de hoogtepunten komt voor rekening van de auteur.

Meer informatie

Boeken

- Martin Campbell-Kelly en William Aspray. *Computer – A History Of The Information Machine*, Westview Press, 2nd edition, 2004, ISBN 9780813342641
- Gerard O'Regan. *A brief history of computing*. Springer, 2008, ISBN 9781848000834
- Martin Rem. *Tegen de stroom in – De Nederlandse rol in de ICT*. ICT Regie, 2009, ISBN 9789090239477
- Cordula Rooijendijk. *Alles moest nog worden uitgevonden*. Atlas, 2007, ISBN 9789045013671
- A. Nijholt, J. van den Ende, *Geschiedenis van de rekenkunst, van kerfstok tot computer*, Academic Service, 1994, ISBN 9039500487
- A. van den Boogaard (red.), *De eeuw van de computer – De geschiedenis van de informatietechnologie in Nederland*, Kluwer, 2008, ISBN 9013060072

Internet

- De Turing Award wordt beschouwd als de Nobelprijs voor informatica. De prijs wordt sinds 1966 jaarlijks uitgereikt en het historische overzicht van de prijswinnaars geeft ook een interessant historisch overzicht van de informatica als wetenschap:
  http://www.awards.acm.org en http://en.wikipedia.org/wiki/Turing_Award
- 2012 is het Internationale Alan Turing-jaar: http://www.turingcentenary.eu
- Historisch overzicht van algoritme :
  http://en.wikipedia.org/wiki/Timeline_of_algorithms
- Historisch overzicht van de informatica:
  http://www.cs.uwaterloo.ca/~shallit/Courses/134/history.html
- 'What is computer science?':
  http://www.cse.buffalo.edu/~rapaport/584/S07/whatiscs.html
- http://en.wikipedia.org/wiki/Computer_science
- Meer over de computergeschiedenis:
  http://plato.stanford.edu/entries/computing-history

Tv-documentaires op internet

- VPRO Noorderlicht-interview met informaticus en Turing Award-winnaar Edsger Dijkstra: http://noorderlicht.vpro.nl/afleveringen/3502225/
- 'De pc revolutie' – In het VPRO-geschiedenisprogramma *Andere Tijden*:
  http://geschiedenis.vpro.nl/programmas/2899536/afleveringen/11306426/

Appendix

Gegevens over het BRICKS-project

BRICKS-onderzoeksconsortium

- Centrum Wiskunde & Informatica (penvoerder)
- Nederlandse Organisatie voor Wetenschappelijk Onderzoek
- Universiteit Utrecht
- Universiteit Twente
- Technische Universiteit Eindhoven
- Technische Universiteit Delft
- Universiteit Leiden
- Radboud Universiteit Nijmegen

BRICKS-projecten

Om invulling te geven aan het BRICKS-programma, is gekozen om de Nationale Onderzoeks Agenda voor Informatica (NOAG-I) te volgen. Deze agenda, die opgesteld is door het Informatica Platform Nederland (IPN), de Advies Commissie Informatica (ACI), en de Nederlandse organisatie voor Wetenschappelijk Onderzoek, afdeling Exacte Wetenschappen (NWO-EW), omvatte ten tijde van de planning van het BRICKS-project in 2003 zeven strategische onderzoeksthema's. Aangezien reeds drie van deze zeven thema's aan bod kwamen in andere Bsik projecten, werd gekozen om BRICKS in te vullen met de resterende vier thema's, te weten:

- Parallel and Distributed Computing (PDC)
- Modelling, Simulation and Visualization (MSV)
- Intelligent Systems (IS)
- Algorithms and Formal Methods (AFM)

Binnen het BRICKS-programma bestaan 21 projecten van diverse omvang die elk in een van bovenstaande thema's vallen. Van deze projecten zijn er elf aan het begin van het BRICKS-programma gestart. De overige 10 projecten zijn later gestart via het door NWO-EW ingebrachte deelprogramma FOCUS (reinFOrcing CompUter Science) waarvoor het budget 20% van het totale budget bedroeg. De 10 FOCUS-projecten zijn geselecteerd via twee open competities (één in 2005 en één in 2006) die beide ingericht waren door NWO-EW. Voor deze competities waren in totaal 32 voorstellen ingediend.

De totale lijst van BRICKS-projecten is als volgt:

PDC1	Security, identification, and authentication
PDC2	Quality of service in communication networks
PDC3	Network infrastructure support for convergent interactive media
MSV1	Scientific computing
MSV2	Interactive virtual environments
MSV3	Geometric algorithm design for geographic environments (FOCUS 2005)
IS1	Databases for personalized ubiquitous intelligent devices
IS2	The petabyte data-mining challenge
IS3	Decision support systems for logistic networks and supply chain optimization
IS4/5	Cracking a scientific database (FOCUS 2006)
IS6	Visual information retrieval based on synthetic imagery (FOCUS 2006)
IS7	Distributed implementations of adaptive collective decision making (FOCUS 2006)
IS8	Bayesian decision support in medical screening (FOCUS 2006)
AFM1	Quantum computing
AFM2	Algorithms in bio-informatics
AFM3	Formal methods for active networking
AFM4	Advancing the real use of proof assistants (FOCUS 2005)
AFM5	Infinite objects: computation, modelling and reasoning (FOCUS 2005)
AFM6	A verification grid for enhanced model checking (FOCUS 2005)
AFM7	Modeling and analysis of QoS for component-based designs (FOCUS 2005)
AFM8	A common framework for the analysis of reactive and timed systems (FOCUS 2006)

BRICKS-proefschriften

Binnen het BRICKS-programma hebben 36 promovendi de mogelijkheid gekregen om te werken aan innoverend wetenschappelijk onderzoek op het gebied van de informatica. De bijbehorende proefschriften behoren tot de belangrijkste wetenschappelijke BRICKS-publicaties en worden daarom hier apart vermeld. Omdat promotieonderzoek minimaal 4 jaar duurt in Nederland en de meeste BRICKS-promovendi tussentijds zijn begonnen, is een aantal promovendi ten tijde van dit schrijven nog niet klaar met het proefschrift. De titels van deze proefschriften, aangegeven met een asterisk (*), zijn voorlopig en kunnen afwijken van de uiteindelijke titels.

Parallel and Distributed Computing (PDC)

R. de Haan	Algebraic Methods for Low Communication Secure Protocols
J. Calamé	Testing Reactive Systems with Data-Enumerative Methods and Constraint Solving
T. Chen	Clocks, Dice and Processes
W. van der Weij	Queueing Networks with Shared Resources
P.M.D. Lieshout	Queueing Models for Bandwidth-Sharing Disciplines

I. Vaishnavi	Estimated Service: A Deadline and Estimate Based QoS Mechanism for Real-Time Media Distribution
H.L. Jonker	Security Matters: Privacy in Voting and Fairness in Digital Exchange
D. Miretskiy	Queueing Networks: Rare Events and Fast Simulations *
Y. Volkovich	Stochastic Analysis of Web Page Ranking

Modelling, Simulation and Visualization (MSV)

C. Li	Joining Particle and Fluid Aspects in Streamer Simulation
J. Wackers	Surface Capturing and Multigrid for Steady Free-Surface Water Flows
J.M. Tang	Two-Level Preconditioned Conjugate Gradient Methods with Applications to Bubbly-Flow Problems
G. de Haan	Techniques and Architectures for 3D Interaction
J. Rommes	Methods for Eigenvalue Problems with Applications in Model Order Reduction
D. Nieuwenhuisen	Path Planning in Changeable Environments
D. Sármány	High-order Discontinuous Gelerkin Methods for the Maxwell Equations *

Intelligent Systems (IS)

F.E. Groffen	Armada. An Evolving Database System
W.J. van Hoeve	Operations Research Techniques in Constraint Programming
G. Maroti	Operations Research Models for Railway Rolling Stock Planning
E.J.E.M. van Leeuwen	Optimization and Approximation on Systems of Geometric Objects
J.J. Paulus	Online Scheduling & Project Scheduling
G. Diepen	Column Generation Algorithms for Machine Scheduling and Integrated Airport Planning
S. Idreos	Database Cracking *
J. Vreeken	Making Pattern Mining Useful *
A.C.M. Koopman	Characteristic Relational Patterns *
E. Liarou	DataCell: Building a Data Stream Engine on top of a Relational Database Kernel *
J.J.J. van den Broek	MIP-based Approaches for Complex Planning Problems *
M.M. Jansen	Hierarchical Coupling Mechanisms for Supply Chain Operations Planning *
H.J.W. Heerde	Privacy Aware Data Management by Means of Data Degradation: Making Data Less Sensitive over Time *

Algorithms and Formal Methods (AFM)

R. Cilibrasi	Statistical Interference Through Data Compression
L. van Iersel	Algorithms, Haplotypes and Phylogenetic Networks
J. Markovski	Real and Stochastic Time in Process Algebras for Performance Evaluation
A. Mathijssen	Logical Calculi for Reasoning with Binding
W.M. Koolen-Wijkstra	Algorithmic Statistics in Bio-informatics *
S. Kemper	Formal Methods for Active Networking - Components and Connectors *
Z. Maraikar	Building and Reasoning about Circuits in the Reo Coordination Language *

Colofon

Tekst en interviews
Bennie Mols, wetenschapsjournalist
www.benniemols.blogspot.com

Redactie
eter Bosman
Anette Kik
Bennie Mols
Esther van Tienen
Jan Verwer

Eindredactie
Cyril Lansink

Grafisch ontwerp
Kitty Molenaar

Portretfotografie
Peter van Beek

Drukwerk
Spinhex & Industrie drukkerij

ISBN 979-90-6196-552-7

december 2009

Illustratieverantwoording

8 Marcel van den Bergh/HH
15 CERN
20 Tobias Baanders
28 Simon Norfolk/ Nb Pictures/HH
34 Rob Huibers/HH
37 Kitty Molenaar
43 Goos van der Veen/HH
45 NASA
56 Radboud University Nijmegen Medical Centre
60/61 AT&T Laboratories, Cambridge
64 The Opte Project, Barrett Lyon
71 Peter Hilz/HH
74 Peter Bosman
79 Guido Diepen
84 Mark Overmars
88 Kievit/HH
92 Jeroen Wackers
93 Transportcafe.cu.uk
99 Jacques Honvault/HH
104 Goos van der Veen/HH
112 David Lat/sxc
114 Jan Lankveld/HH
120 Shutterstock
122 Violet/HH
123 1623: Wilhelm Schickard
124 1832: Allan J. Cronin
124 1842/1843: The Ada Picture Gallery
124 1890: Herman Hollerith
126 1946: U.S. Army
126 1951: Violet/HH
127 1952: Centrum Wiskunde & Informatica (CWI)
128 1968: Centrum Wiskunde & Informatica (CWI)
129 1968: Scott Carson / Zumapress/HH
129 1972: Hamilton Richards
130 1975: Michael Holley
131 1984: Marco Mioli
132 1988: Centrum Wiskunde & Informatica (CWI)
133 2004: Gerard Til/HH
134 2008: Franssen/HH