

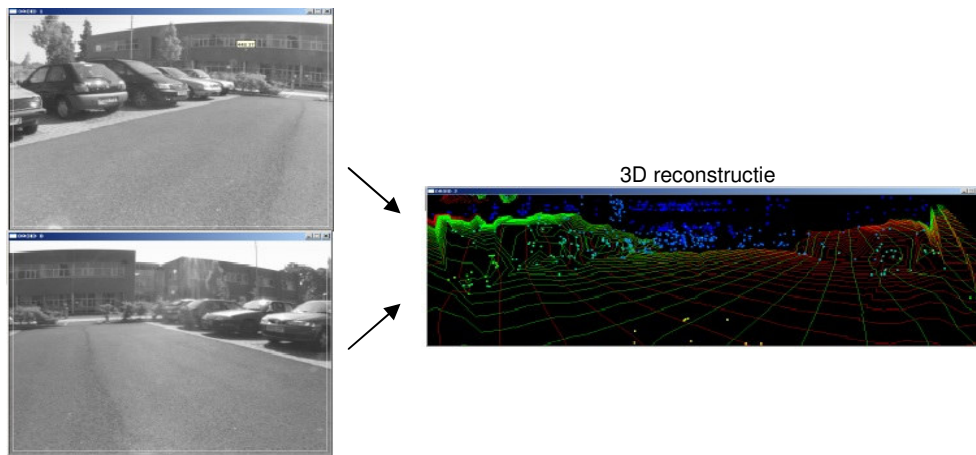
3D reconstructie op basis van stereovisie

1. Inleiding

Een camera maakt een projectie van de 3D wereld op een 2D vlak. Dit beeld bevat onvoldoende informatie om de driedimensionale structuur van de omgeving te achterhalen. Observeert men echter ook vanuit een ander gezichtspunt dezelfde scène, dan wordt dit wel mogelijk. Dit principe van stereovisie is analoog aan het principe waarop het menselijke dieptezicht is gebaseerd.

Mogelijke toepassingen van stereovisie zijn o.a.:

- industriële inspectie van 3D objecten (kwaliteitscontrole, vervorminganalyse, ...)
- autonome voertuigen, robotica
- scanning elektron microscope



Een illustratie van 3D reconstructie op basis van stereovisie.

Materiaal dat ter beschikking staat:

Artikels:

- [1] Falkenhagen, L. "Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces", Proc. of European Workshop on combined Real and Synthetic Image Processing for Broadcast and Video Production, Hamburg, November 1994
- [2] Principe van stereovisie, Linda Tessens, IPI, TELIN, oktober 2005

Software: de benodigde matlabscripts in deze opgave *harris.m*, *matchbycorrelation.m* and *nonmaxsuppts.m* kunnen gevonden worden op de website <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/index.html> (P. D. Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing. - School of Computer Science & Software Engineering, - The University of Western Australia).

De gecalibreerde testbeelden en de calibratiedata zijn beschikbaar op <http://www.robots.ox.ac.uk/~vgg/data/data-mview.html>.

2. Opgave

Onder *Model House* vind je 10 beelden van een modelbouwhuis, opgenomen vanuit 10 verschillende gezichtspunten. Aan de hand van deze beelden wensen we een idee te krijgen van de 3D vorm van het huis.

2.1 Kies 2 beelden. Kies twee overeenkomstige punten (bv. een bepaalde hoek van het dak) en lees de pixelcoördinaten ervan af in een geschikt programma. Leg uit hoe je de 3D coördinaten van dit punt t.o.v. het wereldcoördinatenstelsel reconstrueert (zie [2]).

Kies nu een derde beeld waarin dit 3D punt zichtbaar is. Reconstrueer de 3D coördinaten m.b.v. dit beeld en een van de voorgaande. Vind je twee keer hetzelfde resultaat? Verklaar mogelijke afwijkingen.

Maak een programma dat de lijst met overeenkomstige punten (bestand *house.nview-corners*) inleest en deze punten dan uit minimaal twee afbeeldingen 3D reconstrueert. Hoe ga je om met punten die op meerdere afbeeldingen zichtbaar zijn?

2.2 In de meeste realistische toepassingen gebeurt de selectie van corresponderende punten automatisch en niet met de hand. Dit proces wordt *matching* genoemd.

- Omdat het zeer rekenintensief is om voor alle pixels van een beeld de corresponderende pixel in de andere beelden te bepalen, is een 1^e stap in het matchingproces het selecteren van een aantal goedgekozen punten. Dit is geen triviale opdracht, omdat het wenselijk is dat in de andere beelden de overeenkomstige punten ook automatisch geselecteerd worden. Een methode die hiervoor vaak gebruikt wordt is de Harris corner detector. In het matlabscript *harris.m* is deze methode geïmplementeerd. Zoek op welke principes deze methode gebaseerd is en leg kort uit in je eigen woorden.

Detecteer de hoeken in een beeld. Schakel zeker de optie ``disp'` aan. Merk op dat met ``hoeken'` in deze context niet hetzelfde bedoeld wordt als wat we in de omgangstaal onder ``hoeken'` verstaan. Stemt de lijst met gevonden hoeken overeen met de lijst die onder *2D geometry* van *Model House* teruggevonden kan worden? Waarom? Leg de invloed van de parameters van *harris* uit.

- Een tweede stap is het opsporen van overeenkomstige punten (het eigenlijke matchen). Werk vanaf nu verder met de lijsten van hoeken die online staan. Kies 2 beelden uit. Voor elke hoek gedetecteerd in het ene beeld, zijn alle hoeken gedetecteerd in het 2e beeld mogelijke ``matches'`. Overeenkomstige hoeken worden bijvoorbeeld gevonden door voor elke hoek uit het 1e beeld een omgeving van deze hoek te correleren met een omgeving van alle hoeken uit het 2e beeld. De combinatie van punten die tot de grootste correlatiewaarde leidt, wordt beschouwd als de gezochte combinatie. Deze correlatiemethode is geïmplementeerd in het matlabscript *matchbycorrelation.m*. Pas deze methode toe om het matchingproces tussen beide beelden te voltooien. Zijn alle gedetecteerd hoeken correct ``gematched'`? Visualiseer het resultaat. Controleer je resultaten a.d.h.v. de lijst van matches online. Bespreek.

2.3. De eerder geschetste methode om overeenkomstige punten te detecteren, geeft geen garantie op een foutloos resultaat. Voeg Gaussiaanse ruis toe aan beide beelden, en controleer of voorgaande methode nog even goede resultaten levert. Bespreek. Welke voorwaarden voor de matches worden in [1] vermeld om tot een robuustere matching te komen? Leg ze elk in eigen woorden uit.

2.4. We verbeteren de resultaten van het matchen nu m.b.v. de *epipolaire* voorwaarde.

- Waarom kunnen wij ons tot deze voorwaarde beperken, m.a.w. wat is het voornaamste verschil tussen onze doelstelling en de doelstelling in [1]?
- Kies nu 1 gedetecteerd hoek in het 1e beeld. Dit punt is de afbeelding van een punt in de ruimte. De projectie van het punt in de ruimte in het 2e beeld moet *op* de epipolaire lijn liggen. In de praktijk ligt de projectie dikwijls niet exact op, maar wel in de buurt van de epipolar lijn. Dit omwille van onnauwkeurigheden in de calibratie, de lokalisatie van de gedetecteerde features,...
 - Teken de epipolar lijn van het punt in het 1e beeld in het 2e beeld.
 - Controleer of voor alle punten uit de lijst met eerder gevonden overeenkomstige punten tussen de twee beelden, het overeenkomstige punt voldoende dicht bij de epipolaire lijn ligt (waarbij de afstand ``voldoende dicht'` empirisch vastgesteld

dient te worden). Worden alle eventuele fouten ontdekt? Worden ook correcte overeenkomsten verkeerdelijk als fout bestempeld?

- Pas nu 3D reconstructie toe op de puntenkoppels die je na puntje 2.4 bekomen hebt. Vergelijk met de 3D reconstructie uit 2.1.