

2019 • 2020

Faculteit Industriële ingenieurswetenschappen
master in de industriële wetenschappen: energie

Masterthesis

Automatische detectie van de slijtage op freeskopsnijplaatjes met een 2D visiesysteem

PROMOTOR :

Prof. dr. ir. Eric DEMEESTER

PROMOTOR :

ir. Tom JACOBS

BEGELEIDER :

ing. Maarten VERHEYEN

Sven Boden, Sander Vandebon

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: energie,
afstudeerrichting automatisering

Gezamenlijke opleiding UHasselt en KU Leuven



KU LEUVEN



KU LEUVEN

2019 • 2020

Faculteit Industriële ingenieurswetenschappen
master in de industriële wetenschappen: energie

Masterthesis

Automatische detectie van de slijtage op freeskopsnijplaatjes met een 2D visiesysteem

PROMOTOR :

Prof. dr. ir. Eric DEMEESTER

PROMOTOR :

ir. Tom JACOBS

BEGELEIDER :

ing. Maarten VERHEYEN

Sven Boden, Sander Vandenbon

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: energie,
afstudeerrichting automatisering



KU LEUVEN

Deze masterproef werd geschreven tijdens de COVID-19 crisis in 2020. Deze wereldwijde gezondheids crisis heeft mogelijk een impact gehad op de opdracht, de onderzoekshandelingen en de onderzoeksresultaten.

Woord vooraf

De opleiding tot industrieel ingenieur in de energie met focus automatisering leert studenten naast het verwerven van grote hoeveelheden kennis en informatie, ook hoe deze geïmplementeerd moeten worden in de praktijk. Vooral in de masterproef passen wij ons gehele kennen en kunnen toe in één groot eindproject waarmee we onze opleiding afronden. We hebben gekozen om een masterproef omtrent de automatische slijtagedetectie met behulp van een visiesysteem te realiseren omdat dit tegelijk nauw aansluit bij onze opleiding maar ons tegelijk ook de mogelijkheid biedt om nieuwe kennis te vergaren op vlak van visiesystemen.

Hiervoor willen wij eerst graag een woord van dank uitbrengen naar alle betrokken personen. In de eerste plaats is dit Ing. Maarten Verheyen, onze interne begeleider verantwoordelijk voor de dagelijkse opvolging van de masterproef. Ook stond hij altijd paraat om bij te springen wanneer onvoorziene en grote moeilijkheden opdoken. Daarnaast ook bijgestaan door onze interne promotor Prof. Dr. Ir. Eric Demeester, die ons werk en onze resultaten steeds met een kritische visie beoordeelde en hierop feedback gaf. Vervolgens bedanken wij ook Ir. Tom Jacobs, onze promotor binnen het onderzoekscentrum van Sirris. Niet alleen voorzag hij ons van het benodigde testmateriaal, ook kon hij ons inzicht geven in het theoretische proces achter de slijtagedetectie. Tot slot bedanken we iedereen die advies en respons heeft gegeven op onze verslaggeving, planning, uitvoering ...

Bedankt

Inhoudsopgave

Woord vooraf	3
Lijst van tabellen	7
Lijst van figuren	9
Abstract	11
Abstract in English	13
1 Inleiding	15
1.1 Situering.....	15
1.2 Probleemstelling/onderzoeksvraag.....	15
1.3 Doelstellingen.....	15
1.4 Materiaal en methode	16
1.5 Vooruitblik.....	16
2 Bronnenstudie	17
2.1 Gereedschapsslijtage	17
2.1.1 Kwantificeren gereedschapsslijtage	17
2.1.2 Soorten gereedschapsslijtagemetingen.....	19
2.2 Machinevisie.....	20
2.2.1 Camera	20
2.2.2 Lens	23
2.2.3 Belichting	27
2.3 Beeldverwerking.....	29
2.3.1 Image pre-processing	29
2.3.2 Randdetectie	30
2.3.3 Morfologische operaties	32
2.3.4 Kenmerkherkenning	33
2.4 Machine Learning.....	34
2.4.1 Supervised learning	35
2.4.2 Unsupervised learning.....	37
2.4.3 Reinforcement learning	38
2.4.4 Machine Learning algoritmes.....	38
2.4.5 Neural network.....	38
2.4.6 Neuro-fuzzy.....	41
2.4.7 Hidden Markov Model	42
3 Proefopstelling.....	43
3.1 Referentie opstelling/ microscoop	43
3.2 Strategie slijtagedetectie	44
3.3 Opbouw proefopstelling	45
3.4 Beperkingen proefopstelling.....	47

4	Algoritme voor slijtagedetectie	49
4.1	Voorbewerking van het beeld.....	49
4.2	Positie en oriëntatie bepalen	50
4.3	Slijtage afzonderen	53
4.4	Schatting slijtage.....	54
4.5	Finaal algoritme	57
5	Prestaties algoritme	59
5.1	Vergelijking algoritme met microscoop	59
5.2	Nauwkeurigheid algoritme	60
5.3	Moeilijkheden bij algoritme	61
6	Economische analyse	63
6.1	Berekeningen camera	63
6.2	Berekeningen lens.....	65
6.3	Berekening gebruikte camera- en lensopstelling.....	66
6.4	Resultaat economische analyse	67
6.4.1	Andere cameravoorstellen.....	68
6.4.2	Voorstel van consultant	70
7	Besluit.....	71
	Literatuurlijst.....	73
	Bijlagen	77
	Bijlage 1: Python code van het algoritme	77
	Bijlage 2: Vergelijking output van het algoritme met resultaten van de microscoop	84
	Bijlage 3: Datasheet van de gebruikte camera	88
	Bijlage 4: Datasheet van de gebruikte lens	93
	Bijlage 5: Berekeningen voor de benodigde lens van de cameravoorstellen	94

Lijst van tabellen

Tabel 1: Sensorspecificaties gebruikte camera: MV1-D2048x1088-3D03-760-G2.....	66
Tabel 2: Boven- en ondergrens gebruikte camera.....	67
Tabel 3: Productinformatie gebruikte lens: Optem MVZL 34-11-10 25 mm 1:1.4 TV Lens..	67
Tabel 4: Cameravoorstellen en eisen lens	68
Tabel 5: Lensvoorstellen	69
Tabel 6: Compatibiliteit tussen camera's en lenzen.....	69
Tabel 7: Specificaties camera Allied Vision Manta G-507B	70
Tabel 8: Specificaties telecentrische lens KOWA LM1123TC 0.69~0.88x	70

Lijst van figuren

Figuur 1: Soorten gereedschapsslijtage (links) en typische profiel van flank wear (rechts)....	17
Figuur 2: Typische profiel van crater wear	18
Figuur 3: Flank wear karakteristiek voor 3 snijsnelheden (V1 tot V3 in stijgende volgorde).	18
Figuur 4: Slijtagemechanisme bij stijgende snijsnelheid	19
Figuur 5: Directe en indirecte sensoren voor gereedschapsslijtage te meten.....	19
Figuur 6: Procesomstandigheden bij het indirect bepalen van de gereedschapsslijtag.....	20
Figuur 7: Directe methode om gereedschapsslijtage te meten m.b.v. een visiesysteem.....	20
Figuur 8: De opbouw van een CCD-chip	21
Figuur 9: De opbouw van een CMOS-chip.....	21
Figuur 10: Effect van stijgende resolutie	22
Figuur 11: Opstelling van 3D-beeldscanning met twee 2D camera's en triangulatie.....	23
Figuur 12: Resolutie bepalen aan de hand van een lijnenpaar	23
Figuur 13: Brandpuntsafstand	24
Figuur 14: Focus van een lens bij verschillende diafragmaopeningen	24
Figuur 15: Scherptediepte en circle of confusion.....	25
Figuur 16: Defect aberraties (links) en sferische aberraties (rechts).....	26
Figuur 17: Chromatische aberratie	26
Figuur 18: Interactie tussen het licht en het object.....	27
Figuur 19: Ringbelichting	28
Figuur 20: Directionele belichting	28
Figuur 21: Gepolariseerd licht.....	28
Figuur 22: Masker operatie	29
Figuur 23: Mean blur en Gaussian blur.....	29
Figuur 24: Anisotropic diffusion en Crimmins smoothing	30
Figuur 25: Randdetectie	32
Figuur 26: Morfologische operaties	32
Figuur 27: Histogram van anisotropic diffusion	33
Figuur 28: Hough lijn transformatie.....	34
Figuur 29: Soorten machine learning	35
Figuur 30: Voorbeeld huizenprijs aan de hand van regressie	35
Figuur 31: Verdeling van de dataset	36
Figuur 32: Bias en variantie	37
Figuur 33: Links: goed gefit, midden: onderfit (hoge bias), rechts: overfit (hoge variantie) ..	37
Figuur 34: Opbouw van een Artificial neural network	39
Figuur 35: Berekening voor de input van een node	39
Figuur 36: Sigmoid en tanh activatie functies.....	39
Figuur 37: Backpropagation.....	40
Figuur 38: Beeldverwerking om gebied van de slijtage te bepalen	40
Figuur 39: ANN om de gereedschapsslijtage te voorspellen.....	41
Figuur 40: Opbouw ANFIS netwerk.....	42
Figuur 41: Markovketen.....	42
Figuur 42: Beeld van snijgereedschap onder de microscoop.....	43

Figuur 43: Meten van de slijtage onder de microscoop	44
Figuur 44: Reflectie van de slijtage op het snijgereedschap	44
Figuur 45: Contrast tussen slijtage en omgeving vergroten door omgeving te overbelichten .	45
Figuur 46: Proefopstelling.....	46
Figuur 47: Belicht snijplaatje	46
Figuur 48: Freeshouder proefopstelling	46
Figuur 49: Origineel beeld van snijplaatje	49
Figuur 50: Verkleind en Gaussian blur	49
Figuur 51: Anisotropic diffusion met iteraties = 15, kappa = 70 en gamma = 0,2	50
Figuur 52: Laplace filter, Sobel edge dedector, Canny edge dedector op basis van Gaussian blur	50
Figuur 53: Laplace filter, Sobel edge dedector, Canny edge dedector op basis van anisotropic diffusion	51
Figuur 54: Laplace filter na adaptive thresholding met kernel = 501	51
Figuur 55: Morfologische operatie openen (links) en erosie (rechts)	52
Figuur 56: Hough lijn transformatie met grenswaarde = 500	52
Figuur 57: Probabilistische Hough lijn transformatie	53
Figuur 58: Geroteerd snijplaatje met aangrijppunt	53
Figuur 59: Slijtage afzonderen uit snijplaatje	54
Figuur 60: Slijtage van snijplaatje.....	54
Figuur 61: Afvlakfilters voor slijtage.....	54
Figuur 62: Randdetectie om slijtage te detecteren	55
Figuur 63: Thresholding om slijtage te detecteren.....	55
Figuur 64: Histogram van anisotropic diffusion	56
Figuur 65: Masker na morfologische operaties.....	56
Figuur 66: Contouren van slijtage.....	56
Figuur 67: Boord detecteren met Hough lijn transformatie	57
Figuur 68: VB_{max} van de slijtage berekenen.....	57
Figuur 69: Vergelijking algoritme met microscoop.....	59
Figuur 70: Spreidingsplot van nauwkeurigheid algoritme	60
Figuur 71: Absolute fout van het algoritme in μm	60
Figuur 72: Relatieve fout van het algoritme in %	61
Figuur 73: Beelden van slecht gedefinieerde slijtage.....	61
Figuur 74: Afwijkingen en uitschieters algoritme.....	62
Figuur 75: werkelijke slijtage vertaald naar aantal pixels.....	63

Abstract

Sirris is een onderzoeksinstituut dat bedrijven ondersteunt bij de optimalisatie van productieprocessen. Hun recentste onderzoeken besteden veel aandacht aan de verhoging van de productienauwkeurigheid en snelheid. Omwille van lage kostprijs en eenvoud wordt steeds vaker beroep gedaan op visiesystemen. Deze masterproef heeft als doel een praktische cameraopstelling en algoritme op te bouwen waarmee de gereedschapsslijtage van de snijplaatjes op freeskoppen gedetecteerd kan worden tot op minstens 10 μm nauwkeurig.

Een economische analyse geeft een voorstel voor een geschikte camera en lens voor de praktische opstelling. Als belichting wordt gediffuseerd en omgevingslicht gebruikt. Daarnaast wordt in Python het algoritme geschreven op basis van beeldverwerkingsbibliotheek OpenCV. De input van het algoritme is een beeld afkomstig van de camera. Verschillende maskeroperaties optimaliseren vervolgens het beeld. Hierna worden berekeningen op het beeld uitgevoerd met een schatting van de slijtage als resultaat.

De economische analyse licht twaalf camera's en twaalf lenzen toe die tot de gewenste nauwkeurigheid meten. Verder is het algoritme op 42 beelden getest. Hierin kon het slechts bij 9 beelden uit de dataset de slijtage tot de vooropgestelde 10 μm nauwkeurig meten. Op 24 van de 42 beelden is de slijtage tot op 30 μm nauwkeurig gemeten. De slijtage op het snijgereedschap varieerde hierbij tussen de 127 en 537 μm . De oorzaak hiervan is voornamelijk te wijten aan de gebruikte proefopstelling.

Abstract in English

Sirris, a Belgian research institute, supports companies in the optimisation of production processes. Their latest research focuses on upscaling the production accuracy and speed. For reasons of cost and simplicity, the use of vision systems gains interest. Therefore the scope of this master's thesis is to create a practical camera set-up and algorithm to detect tool wear on cutting inserts, with an accuracy of at least 10 μm .

An economic analysis provides a proposal for a suitable camera and lens. Diffusive and ambient light is used as illumination. Furthermore an algorithm is written in Python using OpenCV as computer vision library. The input of the algorithm is an image from the camera. The camera image is optimized using multiple mask operations for further tool wear calculations. Finally, the output of the algorithm is an estimation of the actual tool wear.

The economic analysis motivates the use of twelve potential cameras and twelve lenses that achieve the desired accuracy. Furthermore, experiments with the developed tool wear detection algorithm show that it was only able to measure the flank wear up to the required 10 μm accuracy for 9 out of 42 images from the data set. On 24 of the 42 images, the wear was measured to an accuracy of 30 μm . The wear on the cutting inserts varied between 127 and 537 μm . The cause of this is mainly due to the test setup used.

1 Inleiding

1.1 Situering

Als laatstejaarsstudenten van de gemeenschappelijke opleiding van de UHasselt en KU Leuven tot industrieel ingenieur in de energie met focus op automatisering onderzoeken wij hoe, met behulp van visietechnologie, de gereedschapsslijtage bepaald kan worden. Hiervoor worden wij ondersteund door de onderzoeksgroep ACRO van de KU Leuven en het onderzoekscentrum Sirris.

Sirris is een Belgisch onderzoeksinstituut dat technologische innovaties binnen bedrijven implementeert en onderzoek doet naar de optimalisatie van productieprocessen. In de recentste onderzoeken van Sirris wordt vooral aandacht besteed aan de verhoging van de productienauwkeurigheid en snelheid. Hierbij wordt steeds vaker beroep gedaan op visiesystemen omdat deze een goed alternatief zijn ten opzicht van directe methodes om de slijtage te meten.

Deze masterproef onderzoekt bijgevolg hoe automatisch de gereedschapsslijtage op freessnijplaatjes gedetecteerd kan worden met behulp van een 2D-visiesysteem.

1.2 Probleemstelling/onderzoeksvraag

In de huidige productie-industrie wordt gereedschap zoals snijplaatjes, freeskoppen en boren vervangen wanneer de operator denkt dat de maximaal toegelaten slijtage overschreden is. Dit is geen objectieve methode. Vaak wordt gereedschap daardoor vroegtijdig vervangen, wat economisch niet opportuun is, of soms te laat, waardoor sommige eindproducten slechtere nauwkeurigheden bezitten. Daarnaast ontstaat, door het ontbreken van een eenduidige vervangingsmoment, ook een grotere variatie op de afwerkingskwaliteit van het eindproduct.

Om een meer algemeen en objectief beeld te krijgen van de slijtage van gereedschappen kan er gebruik gemaakt worden van visietechnologie. Dit gebeurt aan de hand van beelden van het snijgereedschap. Tot op heden wordt vaak offline, met behulp van een microscoop als visiesysteem, de gereedschapsslijtage gemeten maar hiervoor moet het snijgereedschap uit de machine genomen worden. Dit leidt tot een grote vertraging waardoor dit vaak niet wordt toegepast in de huidige productie-industrie. In deze masterproef wordt daarom onderzocht hoe een 2D-visiesysteem, dat is ingebouwd in een productieproces, kan gebruikt worden om de gereedschapsslijtage te meten en welke nauwkeurigheden hierbij mogelijk zijn.

1.3 Doelstellingen

De masterproef stelt als doel te onderzoeken of met behulp van visietechnologie de hoeveelheid gereedschapsslijtage op de snijkant van snijplaatjes bepaald kan worden. Aan de hand van (al dan niet real-time) beelden dient ook het type slijtage bepaald te worden. De einddoelstelling kan dus opgesplitst worden in twee deeldoelstellingen.

Eerst moet door middel van een camera en lens een beeld van de gereedschapsslijtage worden genomen. De keuze van het type camera, lens en type belichting alsook de plaatsing hiervan

spelen een uitermate belangrijke rol voor een goede beeldvorming. De keuze van de benodigde camera en lens dient daarom ook berekend te worden. Het camerabeeld dient gefilterd en geoptimaliseerd te worden voor een betere beeldvorming van de slijtage. Vervolgens wordt op het verbeterde beeld de slijtage van het werkstuk bepaald.

Als tweede onderdeel moet uit de foto het type en de hoeveelheid slijtage bepaald worden. De oplossing van deze deeldoelstelling volgt hoofdzakelijk uit de literatuurstudie.

1.4 Materiaal en methode

De resultaten worden sterk beïnvloed door de opstelling. Bijgevolg worden verschillende opstellingen getest met verschillende camera's en belichtingsvormen. Deze opstellingen worden ter beschikking gesteld door de onderzoeksgroep ACRO van de KU Leuven. De opstelling waar de gereedschapsslijtage het duidelijkste zichtbaar is, zal vervolgens gebruikt worden om een grote hoeveelheid snijplaatjes te analyseren. Deze worden aan ons ter beschikking gesteld door onze externe promotor bij Sirris.

Voor het analyseren van de snijplaatjes wordt gebruik gemaakt van OpenCV. Dit is een bibliotheek (library) in Python en C++ voor beeldverwerking. De meest geschikte methode voor het analyseren van de snijplaatjes volgt uit de literatuurstudie.

1.5 Vooruitblik

De structuur van deze thesis is als volgt. Vertrekkende van een bronnenstudie in hoofdstuk 2 worden verschillende bestaande oplossingen en ideeën die tot op heden al getest zijn, met elkaar vergeleken. De bronnenstudie geeft daarnaast ook een betere verklaring waarom de voorkeur is gegeven voor bepaalde keuzes in het verdere verloop van de masterproef.

Vervolgens wordt in hoofdstuk 3 de gebruikte proefopstelling toegelicht waarbij ook de gebruikte slijtagedetectietechniek uitgelegd wordt. In hoofdstuk 4 wordt het algoritme uitgelegd. Meerdere technieken worden met elkaar vergeleken en de meest optimale techniek zal op het einde van ieder deelhoofdstuk verduidelijkt worden.

In hoofdstuk 5 wordt het algoritme op verschillende beelden getest. Zowel de sterktes als de zwaktes van het algoritme zullen hierin verklaard worden. Omdat zoals eerder genoemd de gebruikte camera en lens een grote invloed hebben op het verkregen resultaat zal in hoofdstuk 6 via een economische analyse een voorstel worden gedaan naar de benodigde camera- en lensconfiguratie. Tot slot wordt in het laatste hoofdstuk een besluit gemaakt rond de gehele masterproef.

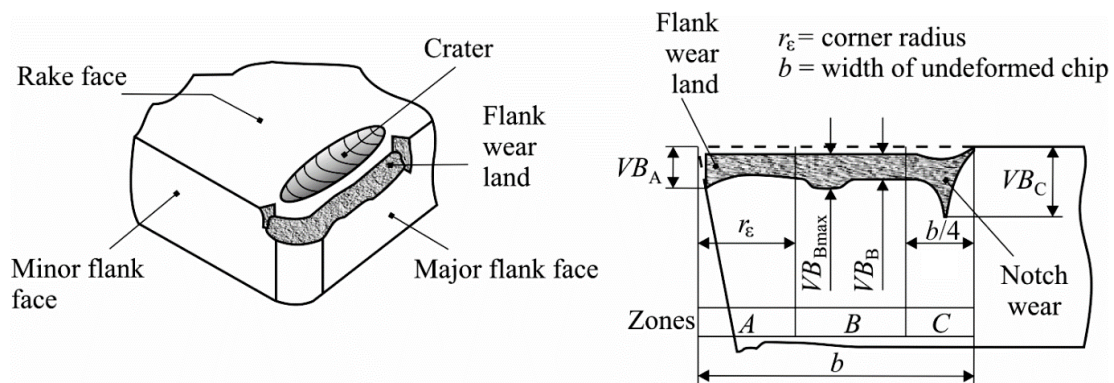
2 Bronnenstudie

In de bronnenstudie wordt eerst beschreven hoe gereedschapsslijtage gekwantificeerd kan worden en welke methodes gebruikt kunnen worden om deze te detecteren. Vervolgens worden de voornaamste componenten rond machinevisie uitgelegd omdat deze masterproef de gereedschapsslijtage hiermee waarneemt. Daarna worden de beelden verwerkt en wordt een overzicht gegeven van de meest gebruikte beeldverwerkingstechnieken. Tot slot worden de laatste jaren steeds meer machine learning methodes toegepast om de gereedschapsslijtage te kwantificeren. Daarom wordt een introductie omtrent machine learning gegeven en vervolgens enkele toegepaste methodes uit andere papers geïllustreerd. Het gedeelte rond machine learning is binnen deze masterproef nergens verder toegepast omwille van onverwachte moeilijkheden die teveel tijd in beslag namen. Toch wordt het vermeld omdat later onderzoek hierop verder kan werken.

2.1 Gereedschapsslijtage

2.1.1 Kwantificeren gereedschapsslijtage

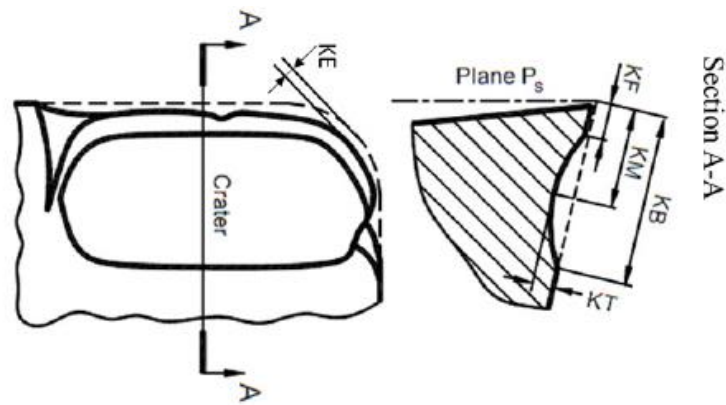
Snijplaatjes voor draaien en frezen moeten op het juiste tijdstip vervangen worden want 20% van de uitval is te wijten aan gereedschapsdefecten [1]. Verder kan de snijsnelheid met 10% tot 50% verhogen door gebruik te maken van een gepaste *tool condition monitoring* (TCM) techniek [2]. Hiervoor is er nood aan een objectieve manier om de gereedschapsslijtage te kwantificeren. De gereedschapsslijtage van snijplaatjes kan opgesplitst worden in *flank*-, *notch*- en *crater wear* zoals op Figuur 1 [3].



Figuur 1: Soorten gereedschapsslijtage (links) en typische profiel van flank wear (rechts) [4, p. 450]

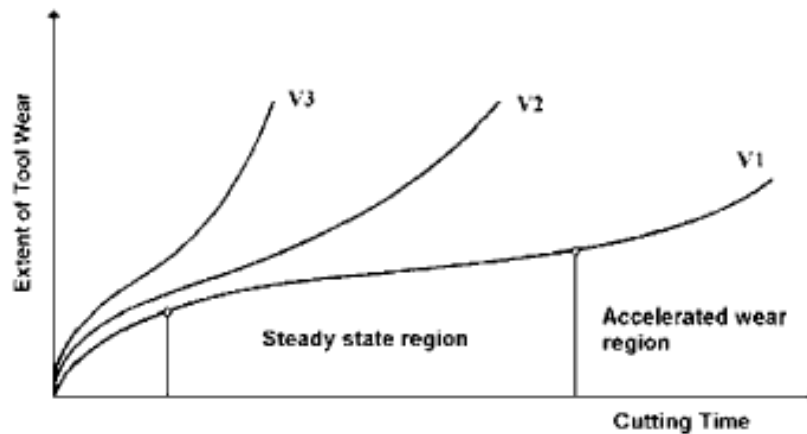
Flank wear wordt voornamelijk veroorzaakt door de hoge temperaturen en de wrijving van het snijplaatje met het oppervlak van het werkstuk [3]. Hierdoor verkleint de snijkant en daalt de nauwkeurigheid en de oppervlaktekwaliteit van het werkstuk [5]. Notch wear ontstaat ter hoogte van het vlak waar het snijplaatje het werkstuk snijdt. Deze slijtage is het gevolg door wrijving van de rake face en de major flank face met het werkstuk.

Crater slijtage vindt plaats op de rake face en verandert de snijhoek van de rake face waardoor de snijkant verzwakt ook de snijkant van het snijplaatje [3]. De crater wordt meestal gekwantificeerd door de diepte KT of door de afstanden KF en KB zoals op Figuur 2 [1].



Figuur 2: Typische profiel van crater wear [5, p. 490]

De hoeveelheid slijtage kan zoals op Figuur 3 in drie stadia gedeeld worden: *break-in*, *steady-state* en *accelerated*. De snijsnelheid V bepaalt hoe snel het snijplaatje naar het volgende stadium evolueert [6].

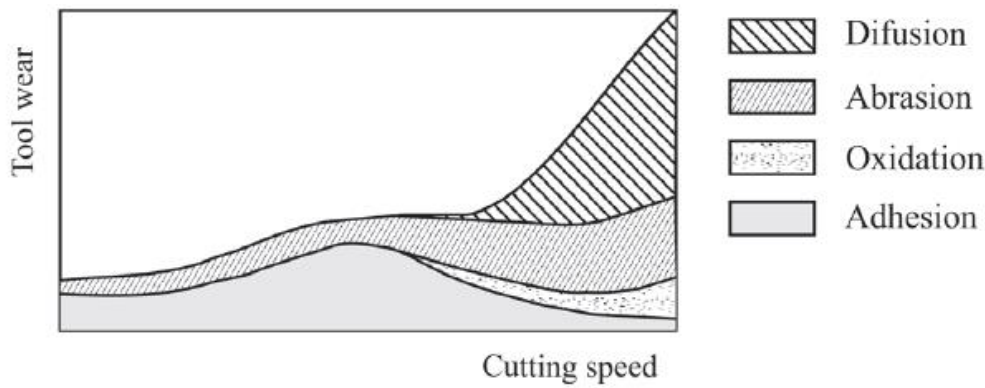


Figuur 3: Flank wear karakteristiek voor 3 snijsnelheden (V1 tot V3 in stijgende volgorde) [6, p. 782]

Het slijtageproces zelf verandert onder invloed van de snijsnelheid zoals weergegeven op Figuur 4. De slijtage wordt veroorzaakt door verschillende fysische verschijnselen [7]:

- adhesie is het proces waar het snijplaatje versmelt met het werkstuk door de warmte van de wrijving en door het breken van de lassen breken er kleine stukken af van het snijgereedschap;
- mechanische erosie is het resultaat van de wrijving waar het snijplaatje slijpt over het oppervlak van het werkstuk;
- diffusie is het gevolg van atomen die van het snijgereedschap naar het werkstuk overgaan;
- oxidatie treed op wanneer het gereedschap met zuurstof uit de lucht reageert door de hoge temperatuur en een oxidatie laag vormt [4].

Bij lage snijsnelheden treedt er voornamelijk adhesie en mechanische erosie op. Bij hoge snijsnelheden verlaagt de adhesie, verhoogt diffusie en mechanische erosie en ontstaat er ook oxidatie.

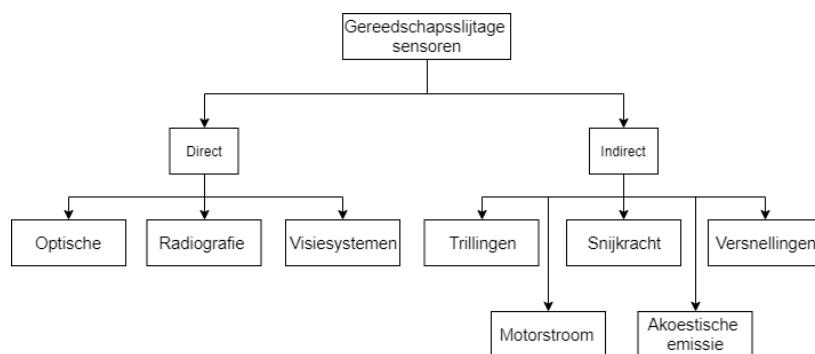


Figuur 4: Slijtagemechanisme bij stijgende snijnsnelheid [4, p. 450]

De meest gebruikte manier om de gereedschapsslijtage te kwantificeren is door het meten van VB_B en VB_{Bmax} van de flank omdat deze soort slijtage altijd aanwezig is en makkelijk te identificeren. Maar uit experimenten van de universiteit van Iowa blijkt dat snijplaatjes gebruikt onder verschillende snijnsnelheden en verschillende voedingssnelheden een verschillende patroon van slijtage kunnen ontwikkelen en nog steeds dezelfde VB_B hebben [6]. Verder is het in praktijk moeilijk om de grenzen van de slijtage nauwkeurig te definiëren door de hoge onregelmatigheden. Hierdoor wordt de oppervlakte van de slijtage een meer relevante parameter [11].

2.1.2 Soorten gereedschapsslijtagemetingen

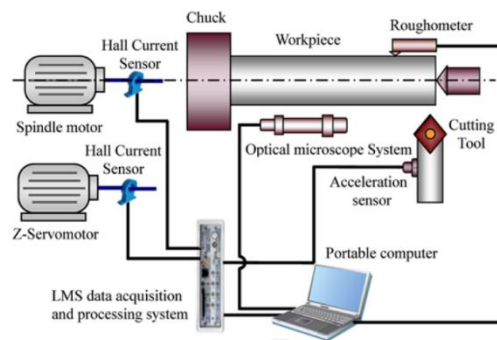
Gereedschapsslijtagemetingen kunnen, zoals in Figuur 5 wordt afgebeeld, opgedeeld worden in twee categorieën: direct en indirect. De directe methode maakt gebruik van directe metingen zoals optische technieken waarbij naar het uiteindelijke uitgangspunt wordt gekeken, radiografie of visiesystemen die de slijtage op het snijgereedschap beoordelen. Het voordeel van directe metingen is de hoge nauwkeurigheid en dat bijkomende geometrische veranderingen gemonitord kunnen worden. Maar het grote nadeel is dat deze technieken vaak moeilijk te implementeren zijn bij online machines zeker wanneer koelvloeistof wordt gebruikt tijdens het snijproces. Dit probleem doet zich niet voor bij indirecte metingen, hier wordt met behulp van machineparameters, zoals draaisnelheid, snijkraft, temperaturen, versnellingen, trillingen, akoestische emissie, motorstroom enz., de slijtage bepaald.



Figuur 5: Directe en indirecte sensoren voor gereedschapsslijtage te meten

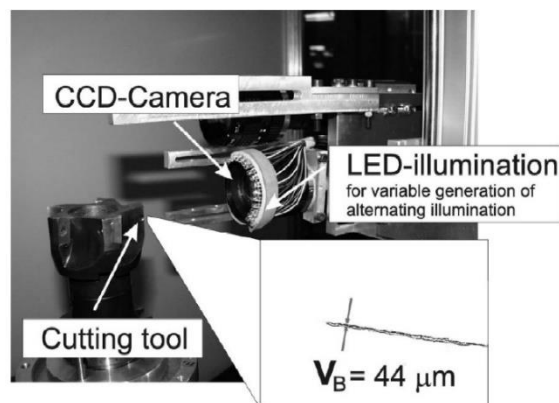
Figuur 6 geeft een voorbeeld van een opstelling die gebruik maakt van de indirecte methode. Tijdens het hier toegepaste draaiproces worden met behulp van allerlei sensoren procesparameters opgemeten, deze signalen worden via een data acquisitie systeem verzameld

en geoptimaliseerd. Tot slot simuleert een computer uit deze verkregen informatie de gereedschapsslijtage [3].



Figuur 6: Procesomstandigheden bij het indirect bepalen van de gereedschapsslijtage [3, p. 50]

Een groot nadeel van indirecte methodes is dat de simulatie volledig vertrouwt op de machineparameters. De afwijkingen op de sensoren en onnauwkeurigheden bij data acquisitie leiden tot ophoping van foutieve inputs voor het simulatieprogramma. Dit heeft als gevolg dat de schatting van de gereedschapsslijtage niet altijd accuraat is. Een veel voorkomende oplossing is om meerdere indirecte parameters te gebruiken om zo de nauwkeurigheid te verhogen. Dit probleem doet zich niet voor bij de directe methode (Figuur 7) omdat het visiesysteem onafhankelijk werkt van het productieproces [8].



Figuur 7: Directe methode om gereedschapsslijtage te meten m.b.v. een visiesysteem [9, p. 219]

2.2 Machinevisie

De opbouw van een machinevisiesysteem beperkt zich niet enkel tot een camera, maar bestaat daarnaast ook nog uit belichting en een lens. Dit deelhoofdstuk licht elk onderdeel verder toe aan de hand van de cursus “Machinevisie” van E. Claesen [10], presentaties van E. Demeester [11], presentaties van J. Baeten [12] en het handboek “Imaging & Vision” van Stemmer Imaging [13].

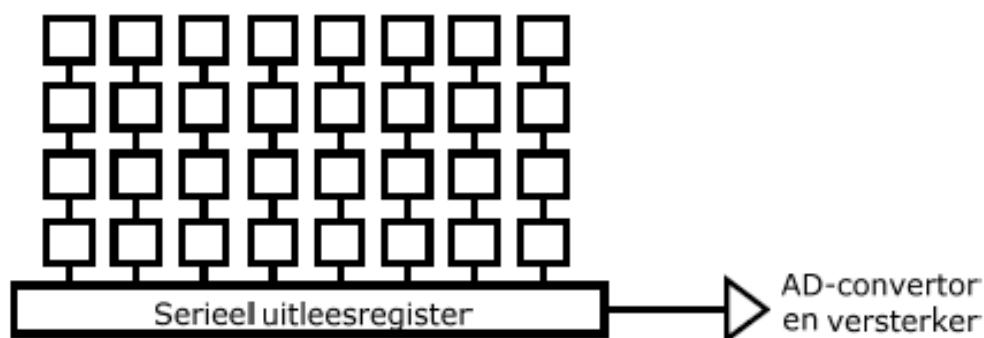
2.2.1 Camera

De camera is het element van het visiesysteem dat het reflecterende licht registreert. Meer specifiek wordt het licht opgevangen op een chip. Met oog op het doel van deze masterproef zijn er maar beperkte eisen voor de camera, daarom zullen hier alleen de belangrijkste kenmerken toegelicht worden. Deze zijn het werkingsprincipe van de chiptechnologie en de resolutie. Voor meer specifieke cameraspecificaties wordt verwezen naar [13].

2.2.1.1 Sensortechnologie

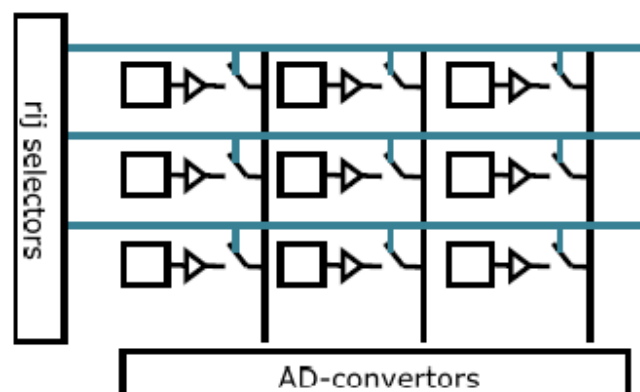
De meeste hedendaagse camera's zijn Charge-Coupled Devices (CCDs) of gebaseerd op CMOS-technologie. Beide vertrekken volgens hetzelfde basisprincipe: de chip, bestaande uit fotodetectoren, zal het invallende licht omzetten in een elektrische spanning. Het verschil zit in de opbouw en de uitlezing van de fotodetectoren.

Met een CCD-chip zitten alle fotodetectoren in een raster en worden hun waardes elk omgezet met behulp van een AD-converter (Figuur 8). Het ruimteverlies tussen de verschillende fotodetectoren is beperkt, hierdoor ontstaat een zeer zuiver beeld. Maar doordat iedere pixel afzonderlijk door de AD-converter wordt uitgelezen is de framerate laag. Daarnaast kan ook niet zomaar één willekeurige pixel uit het raster geselecteerd worden, dit maakt het moeilijk om een *region of interest* te definiëren.



Figuur 8: De opbouw van een CCD-chip [10, p. 15]

De CMOS-chip is uitgevoerd met rijselectoren waardoor het mogelijk is om iedere fotodetector afzonderlijk uit te lezen (Figuur 9). Hierdoor kan tegelijk een hoge framerate en een kleine region of interest (ROI) behaald worden. Vroeger was het nadeel van een CMOS-chip dat het ruimteverlies groter was in vergelijking met de CCD-chip. Tegenwoordig is het verschil in ruimteverlies kleiner, daarom worden hedendaagse camera's voor machinevisie meestal uitgerust met een CMOS-chip.

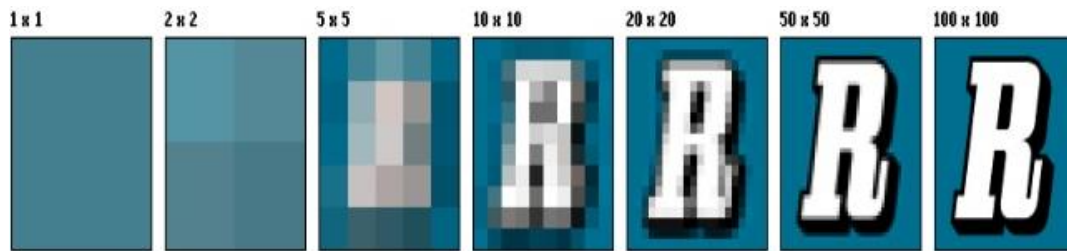


Figuur 9: De opbouw van een CMOS-chip [10, p. 15]

2.2.1.2 Resolutie

De resolutie van de camera geeft het aantal pixels van de output van de chip weer. Een hogere resolutie betekent dus dat eenzelfde beeld meer gedetailleerd geconstrueerd kan worden op de

camerachip. Figuur 10 toont het effect van een stijgende waarde voor de resolutie. Hoe meer pixels de chip bevat, hoe meer detail zichtbaar wordt.



Figuur 10: Effect van stijgende resolutie [14]

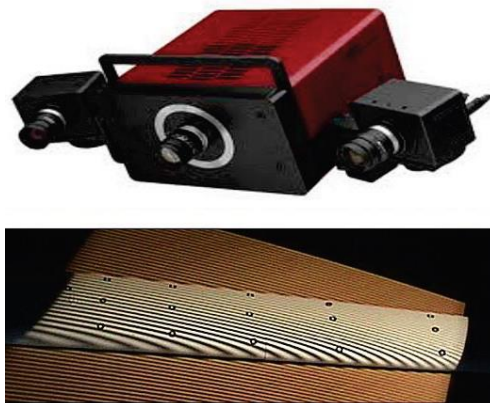
Voor gereedschapsslijtagedetectie met visiesystemen maakt een goede cameraresolutie het enerzijds mogelijk om de contouren beter te herkennen. Dit heeft het voordeel dat de software achter de slijtagedetectie minder fouten zal geven. Anderzijds, indien de resolutie hoog genoeg is, kan de slijtage rechtstreeks gemeten worden aan de hand van de pixels. Door het aantal pixels tussen de uiterste grenzen te tellen en deze te vermenigvuldigen met de hoogte van iedere pixel, krijgt men de afstand tussen de twee grenzen. De afwijking op deze methode is gelijk aan één pixel indien verondersteld wordt dat de contouren perfect getekend zijn.

Zoals hierboven reeds is vermeld, is een goede resolutie cruciaal; het nadeel is echter dat de kostprijs voor de camera hierdoor sterk oploopt. Daarom is het noodzakelijk om, alvorens een camera te kiezen, eerst een schatting te maken van de benodigde resolutie. Een combinatie van het kleinste detail dat waargenomen moet kunnen worden en het gewenste gezichtsveld (*field of view*) zijn de voornaamste parameters voor het schatten van de benodigde resolutie bij monochromatische beelden. Kleurencamera's gebruiken vaak informatie van aangrenzende pixels om nauwkeuriger te werken, een hogere resolutie is dan aanbevolen [13].

2.2.1.3 3D-camera's

Duurdere en nauwkeurige systemen voor het bepalen van de gereedschapsslijtage maken gebruik van 3D-camera technieken. Hiermee wordt het ook mogelijk om naast de flank wear ook de diepte van de crater exact te meten. Twee veel gebruikte methodes om 3D beelden te verkrijgen zijn 'Time-of-flight'-methode en de triangulatiemethode [4].

Verschillende principes zijn mogelijk via de triangulatiemethode. Figuur 11 toont de opstelling bestaande uit 2 camera's en een centrale projector. De centrale projector zorgt voor een gestructureerd lichtpatroon op het object. Uit de licht- en schaduwlijnen berekenen de camera's de afstanden en hoeken. Om beide camera's met elkaar te uit te lijnen worden ook referentiepunten gereflecteerd op het werkstuk. Voor de achterliggende geometrische verbanden wordt verwezen naar [4].



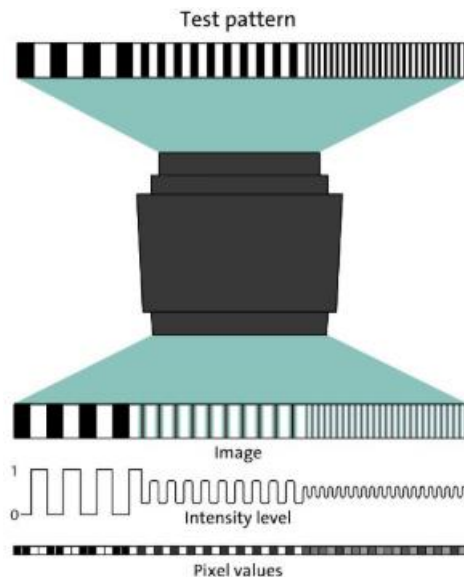
Figuur 11: Opstelling van 3D-beeldscanning met twee 2D camera's en triangulatie [4, p. 452]

2.2.2 Lens

Elk camerasysteem beschikt over een lens. Een lens focust het gereflecteerde licht van een object naar de beeldchip van een camera. De eigenschappen van lenzen en hun gevolgen op beelden worden hieronder uitgelegd.

2.2.2.1 Resolutie

De lens moet in staat zijn om alle lichtstralen nauwkeurig naar het sensoroppervlak van de camera te brengen. De optische resolutie van een lens, ook wel het scheidend vermogen genoemd, bepaalt hierin het kleinste waarneembare detail in beelden. Deze definitie sluit nauw aan bij de definitie voor de resolutie van een camera. Figuur 12 toont dit praktisch als de minimale detecteerbare afstand tussen een lijnenpaar.

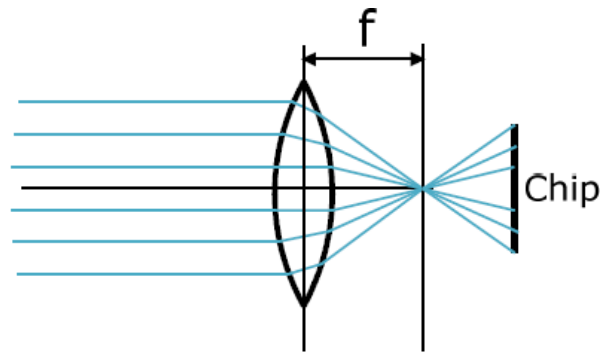


Figuur 12: Resolutie bepalen aan de hand van een lijnenpaar [13, p. 156]

De kwaliteit van een lens wordt uitgedrukt in een *modulation transfer function* (MTF). Deze beschrijving houdt tevens ook rekening met de optische *distortion* (vervorming) en alle vormen van aberraties.

2.2.2.2 Brandpuntsafstand

De brandpuntsafstand bepaalt de grootte van het werkgebied. Figuur 13 illustreert dat het vergroten van de brandpuntsafstand f het gevolg heeft dat het werkingsgebied verkleint.



Figuur 13: Brandpuntsafstand [10, p. 9]

De brandpuntsafstand is rechtstreeks in verband met de vergrotingsfactor van het beeld. Grotere brandpuntsafstanden geven grotere vergrotingsfactoren ten koste van het gezichtsveld.

Volgende formules verduidelijken het effect van de brandpuntsafstand op het werkgebied en de vergrotingsfactor:

Legende:

do afstand van het object tot de lens (distance to object)

di afstand van het beeld tot de lens (distance to image)

f brandpuntsafstand (focal length)

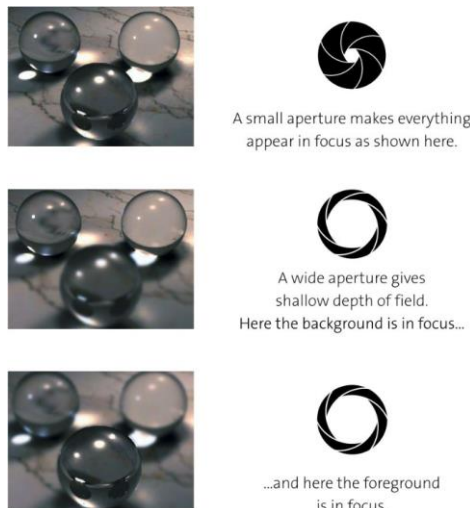
m vergroting (magnification)

Dunne lens vergelijking:
$$\frac{1}{f} = \frac{1}{do} + \frac{1}{di} \quad (1)$$

Vergrotingsfactor:
$$m = \frac{di}{do} \quad (2)$$

2.2.2.3 Diafragma, F-getal en scherptediepte

Het diafragma bepaalt hoeveel licht wordt doorgelaten naar de beeldchip (Figuur 14). De hoeveelheid licht doorgelaten door de lens wordt beschreven aan de hand van het F-getal. Specificatiebundels van lenzen geven altijd het F-getal bij de grootst mogelijke diafragmaopening.



Figuur 14: Focus van een lens bij verschillende diafragmaopeningen [13, p. 153]

Legende:

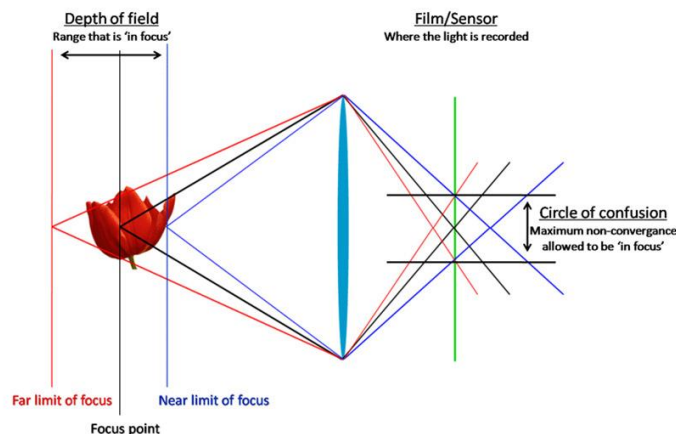
f brandpuntsafstand

D diafragmaopening

F-getal:

$$F = \frac{f}{D} \quad (3)$$

Het F-getal is dimensieloos en krijgt betekenis in de context van de scherptediepte. De scherptediepte is het bereik waarbinnen het beeld scherp en gefocust is. Figuur 15 verduidelijkt de achterliggende gedachte. Om een punt in de wereld volledig scherp op het beeldvlak te krijgen moeten alle lichtstralen, die door de lens op de sensorvlak binnen vallen, binnen de *circle of confusion* samen komen. De lichtstralen die het meest afgebogen worden, zijn deze aan de rand van het lensoppervlak. Deze hebben bijgevolg het meeste invloed op de scherpte van het beeld. Door het diafragma kleiner te maken zullen de buitenste lichtstralen wegvallen. Hierdoor wordt de scherptediepte groter.



Figuur 15: Scherptediepte en circle of confusion [11, p. 39]

Een groot F-getal, en dus een kleine diafragmaopening, betekent dat er meer lichtstralen binnen de circle of confusion vallen. Het beeld is bijgevolg scherper, maar door de beperkte hoeveelheid licht kan een extra lichtbron nodig zijn. Kleine F-getallen daarentegen zorgen dat slechts een deel van het beeld gefocust is. De scherptediepte is bijgevolg beperkt en er moet gekozen worden of er op de achtergrond of op de voorgrond gefocust wordt.

2.2.2.4 Werkrimte

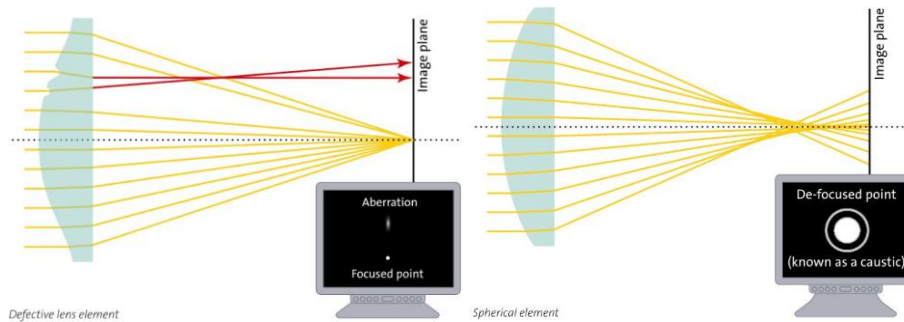
De werkrimte is de vrije ruimte tussen het object en het lensoppervlak. Lenzen focussen meestal van een oneindig bereik tot een minimale afstand tot het object (*minimum object distance* of *MOD*). Wanneer de afstand tussen de lens en de camera vergroot wordt, dan daalt de MOD. Bijgevolg is het dus mogelijk om de MOD te verkleinen door gebruik te maken van verlengstukken [13].

2.2.2.5 Lensafwijkingen

Bij een perfecte lens gaan alle lichtstralen van één punt op een object allemaal gefocust naar één enkel punt op de sensorchip. Afwijkingen van dit ideaal noemen aberraties. Lenzen ondervinden drie belangrijke vormen van aberraties. Deze worden weergegeven in Figuur 16 en Figuur 17.

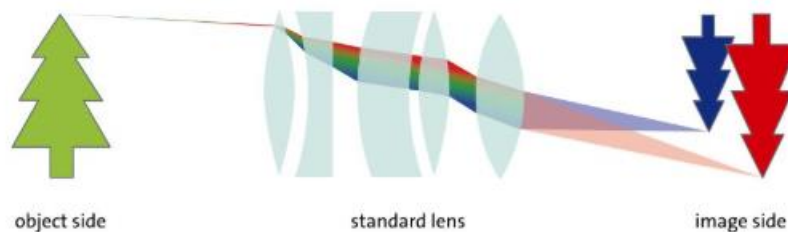
Defect aberraties worden veroorzaakt doordat het oppervlak van de lens beschadigt geraakt en degradeert in de tijd. De lens is hierdoor niet meer in staat om het licht correct te focussen.

Sferische aberraties zijn te wijten aan de vorm van lenzen. Ideale lenzen zijn asferisch, hierdoor is de brandpuntsafstand over het gehele lensoppervlak constant. Doordat dit veel moeilijker te fabriceren is, worden lenzen sferisch gemaakt, dit introduceert een kleine afwijkingen rond het centrum van de lens en grotere afwijkingen dicht bij de rand van de lens. [13]



Figuur 16: Defect aberraties (links) en sferische aberraties (rechts) [13, p. 156]

Tot slot is er chromatische aberratie. Wit licht is opgebouwd uit verschillende golflengtes. Doordat de brekingsindex tussen de omgeving en de lens verschillen zal het wit licht in meerdere kleuren breken. De lichtstralen afkomstig van eenzelfde punt, maar gebroken in verschillende golflengtes achter de lens, focussen hierdoor niet op hetzelfde punt op de sensorchip. Door het object met monochromatisch licht, zoals LED verlichting, te belichten kan dit probleem sterk verminderd worden. Indien wit licht toch in sterke mate aanwezig is in de omgeving, dan kan ook een kleuren correctie lens gebruikt worden.



Figuur 17: Chromatische aberratie [13, p. 155]

2.2.2.6 Lenstypes en filters

Dit tussenhoofdstuk geeft een kort overzicht van de verschillende soorten lenzen en filters die tot op heden beschikbaar zijn op de markt. De nadruk ligt vooral op de verschillen tussen de hoofdcategorieën, meer specifieke details en kenmerken zijn terug te vinden in het handboek “Imaging & Vision” van Stemmer Imaging [13].

De meeste lenzen hebben een resolutie tot één megapixel. Deze zijn beschikbaar met een brandpuntsafstand van 4,5 tot 100 mm. Boven één megapixel spreekt men over hoog resolutie of precisie lenzen. Macrolenzen behalen grote vergrotingsfactoren. Telecentrische lenzen elimineren het perspectief beeld en de geometrische distortie wanneer ze even groot gedimensioneerd zijn als het object dat ze willen vastleggen. Indien de lens snel van focus moet kunnen veranderen, dan bestaan hiervoor elektrisch gestuurde, vloeibare lenzen. [13]

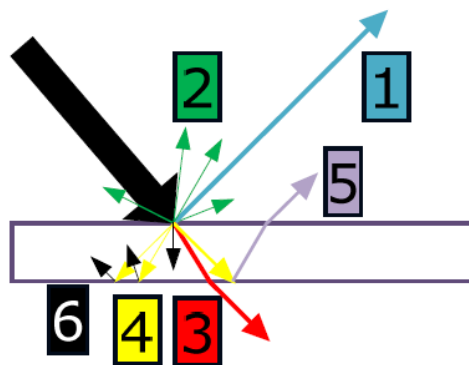
Op het lensoppervlak kan ook een filter aangebracht worden. Dit is een dun laagje van enkele micrometers dat bepaalde golflengtes elimineert. Voorbeelden hiervan zijn: shortpass-, highpass-, bandpass-, daglicht-, UV- en kleurenfilters. Daarnaast bestaan ook polarisatiefilters, deze blokkeren de elektromagnetische golven van het licht indien dit onder een verkeerde hoek invalt en verhinderen het licht dat loodrecht ten opzichte van deze hoek staat niet.

2.2.3 Belichting

Door gebruik te maken van de juiste belichting kunnen details van objecten sterk geaccentueerd worden en omgevingsinvloeden uitgeschakeld worden. Toch is het niet voor de hand liggend om een correcte belichting te kiezen omdat enkel de reflectie van licht op objecten, en niet het licht zelf, voor het menselijk oog zichtbaar is. Het doel van de belichting is om een ideale reflectie te creëren vanuit het standpunt van de camera.

Figuur 18 illustreert de verschillende componenten waaruit de interactie tussen licht en een object zijn opgebouwd:

1. directe reflectie,
2. diffuse reflectie of scattering,
3. directe transmissie,
4. diffuse transmissie,
5. achterzijderefectie,
6. absorptie.

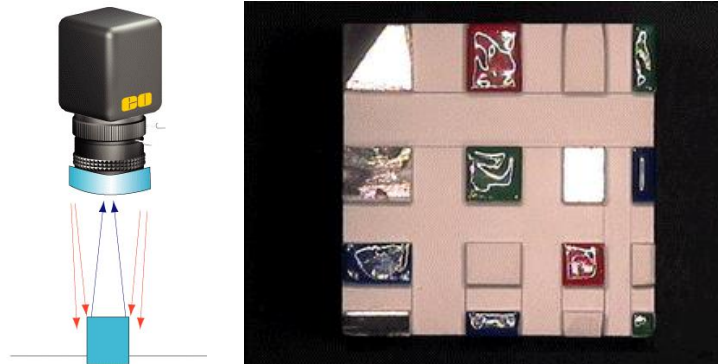


Figuur 18: Interactie tussen het licht en het object [10, p. 7]

Verschillende soorten van belichting spelen elk op een andere manier in op de reflectie van het licht. De belangrijkste en meest gebruikelijke vormen van belichting voor het bepalen van gereedschapsslijtage worden hieronder verduidelijkt.

2.2.3.1 Ringbelichting

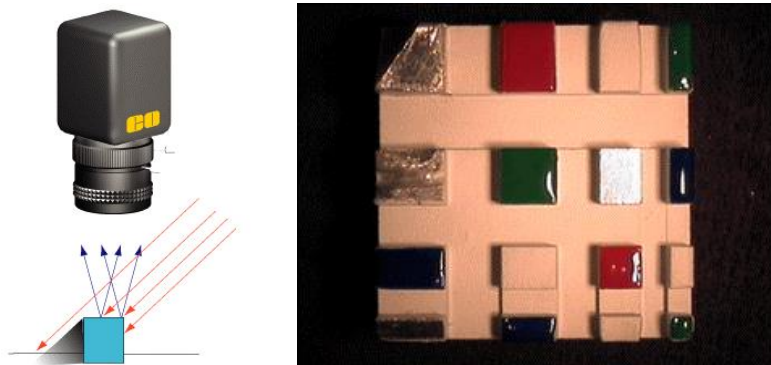
Bij ringbelichting (Figuur 19) lijkt de bron van het licht zich gelijkmatig rondom de camera te bevinden. Hierdoor wordt schaduw sterk verminderd maar is er wel kans op een ringvormige glans ten gevolge van de directe lichtreflectie op het oppervlakte



Figuur 19: Ringbelichting [12, p. 3]

2.2.3.2 Directionele belichting

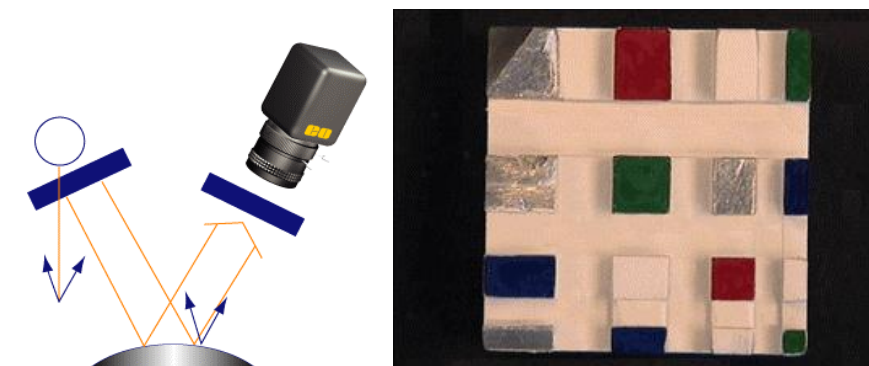
Directionele belichting (Figuur 20) zorgt voor een sterke en gelijkmatige belichting op het oppervlak. Doordat de lichtstralen van de bron niet rondom het hele voorwerp kunnen is schaduwvorming mogelijk. Ook is er kans op glans ten gevolge van de directe reflectie.



Figuur 20: Directionele belichting [12, p. 3]

2.2.3.3 Gepolariseerd licht

Gepolariseerd licht (Figuur 21) geeft een oplossing voor de nadelen van de bovenstaande types van belichting. De gepolariseerde bron verlicht het oppervlak gelijkmatig en de camera ondervindt geen last van glans op het oppervlak. Maar doordat de filters een gedeelte van het licht blokkeren zorgt dit dat de helderheid aanzienlijk verlaagd wordt.



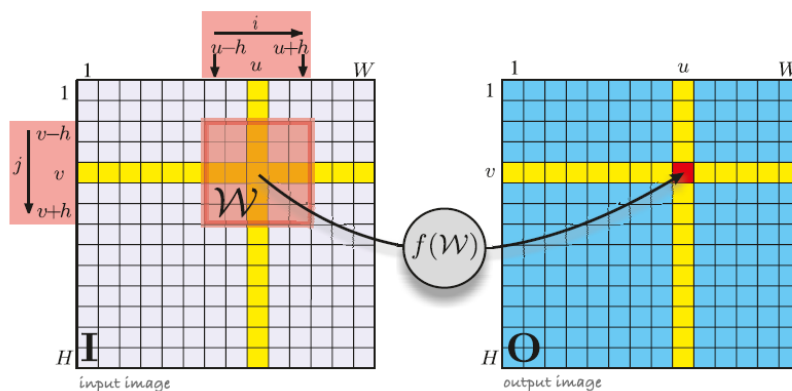
Figuur 21: Gepolariseerd licht [12, p. 4]

2.3 Beeldverwerking

De review paper van Dutta et al. [2] geeft een overzicht van verschillende beeldverwerkingstechnieken die gebruikt worden om gereedschapsslijtage te detecteren. Deze technieken en alternatieven worden hieronder in meer detail uitgewerkt. Ook worden deze toegepast op enkele beelden om een beter idee te krijgen van hun effect.

2.3.1 Image pre-processing

In *image pre-processing* worden voornamelijk laagdoorlaatfilters toegepast met als doel het beeld af te vlakken waardoor de ruis wordt onderdrukt [2]. De meest gebruikte filters zijn masker operaties zoals op Figuur 22. Hierbij wordt op iedere pixel van het originele beeld een functie $f(w)$ uitgevoerd die afhankelijk is van de naburige pixels, dit wordt ook wel een masker genoemd .



Figuur 22: Masker operatie [11, p. 21]

Bij *mean blur* zorgt de functie $f(w)$ ervoor dat er een gemiddelde wordt gemaakt van de aangrenzende pixels. Hierdoor wordt de ruis onderdrukt maar vervagen de randen. Anderzijds is het masker van *Gaussian blur* bepaald op basis van formule 4 en is gebaseerd op een normale verdeling. Hierdoor hebben de pixels dicht bij het center meer effect dan de pixels aan de rand. Deze afvlakfilters worden vergeleken in Figuur 23 [11].

Gaussian blur:
$$f(w) = G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \quad (4)$$

Legende:

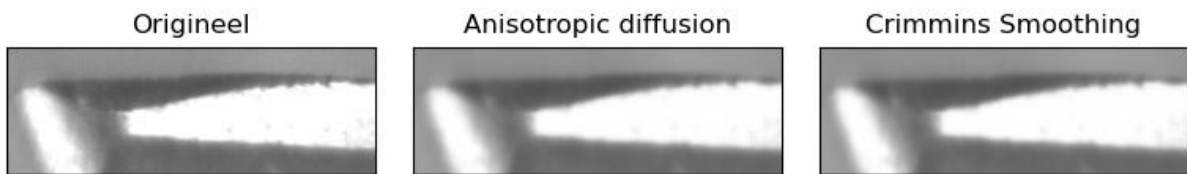
- u pixel t.o.v. oorsprong horizontaal
- v pixel t.o.v. oorsprong verticaal
- μ gemiddelde
- σ Standaardafwijking



Figuur 23: Mean blur en Gaussian blur

Bij afvlakfilters moet er een afweging gemaakt worden tussen het afvlakken van het beeld en het behouden van de randen. In sommige gevallen zijn de randen door Gaussian blur ook vervaagd en moet er gebruikt gemaakt worden van een masker die de randen behoud zoals *anisotropic diffusion*. Deze filter (zie Figuur 24) gebruikt de gradiëntinformatie van de Laplaciaan (zie 2.3.2) om het vervagen te promoten in de richting met een lage gradiënt (pixels verschillen weinig van elkaar). Hierdoor vervagen de gebieden met kleine verschillen (meestal afkomstig van ruis) en blijven de grenzen van de verschillende gebieden behouden. Bij deze filter moet ingesteld worden hoeveel iteraties er uitgevoerd moeten worden, de conductiecoëfficiënt (κ) en de diffusiecoëfficiënt (γ). De conductiecoëfficiënt (κ) bepaalt hoe sterk een rand moet zijn om vervagen tegen te houden en de diffusiecoëfficiënt (γ) bepaalt de snelheid van het vervagen [15].

Daarnaast kan er ook gebruik gemaakt worden van *Crimmins smoothing* zoals op Figuur 24. Hier wordt de intensiteit van iedere pixel vergeleken met deze van zijn burens en bijgesteld om zo meer op zijn burens te lijken. Deze vergelijkingen gebeuren in paren (N-Z, W-O, NW- ZO, NO- ZW) en worden verschillende keren herhaald. Door deze vergelijkingen voor iedere pixel meerdere maal te maken zorgt dit ervoor dat het algoritme zeer rekenintensief is [16].



Figuur 24: Anisotropic diffusion en Crimmins smoothing

2.3.2 Randdetectie

Na het afvlakken van de beelden kunnen randdetectie algoritmes gebruikt worden om de slijtage te detecteren. Er zijn verschillende randdetectiemethodes die eerst de rand in de x-richting en y-richting bepalen en daarna optellen, zoals *Sobel*, *Scarr* en *Prewitt*. De maskers voor de x-richting van deze methodes worden hieronder opgesomd [17], [18], [19] :

$$Sobel = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Scarr = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \quad Prewitt = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

De *Roberts* randdetectie methode zoekt ook de rand ook in twee richtingen maar dan diagonaal [20], [21]:

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ en } G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

De *Krisch compass mask* en *Robinson compass edge detection* zijn gelijkaardig aan bovenstaande methodes maar hier moet opgegeven worden in welke kompasrichting (N, NW, W,...) de rand gezocht moet worden [22], [23]:

$$Krisch_{NW} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

$$Robinson_{NW} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

Door gebruik te maken van de *Nevatia Babu* randdetectie kunnen er randen gevonden worden onder eender welke hoek [24]. Hieronder wordt het masker weergegeven voor lijnen te zoeken onder 30° .

$$Nevatia Babu 30^\circ = \begin{bmatrix} -100 & 32 & 100 & 100 & 100 \\ -100 & -78 & 92 & 100 & 100 \\ -100 & -100 & 0 & 100 & 100 \\ -100 & -100 & -92 & 78 & 100 \\ -100 & -100 & 100 & -32 & 100 \end{bmatrix}$$

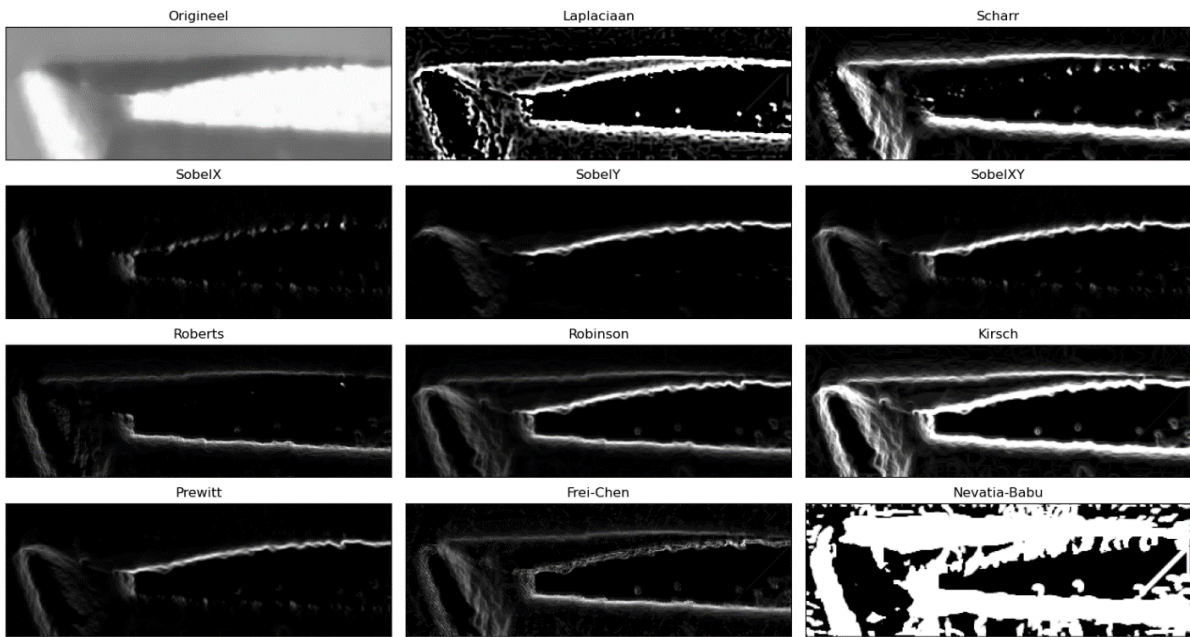
In de *Frei-Chen* filter worden er negen (3x3) maskers gebruikt, vier voor de randen, vier voor lijnen en de laatste als gemiddelde [25]. Deze negen maskers worden opgeteld in een gewogen som.

$$G_1 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad G_5 = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad G_9 = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

De *Canny edge detector* is een veel gebruikt algoritme om randen te detecteren omwille van de immuniteit tegen ruis en de eigenschap om de randen zeer nauwkeurig te vinden. Eerst voert de Canny edge detector een *Gaussian blur* uit om het beeld af te vlakken. Vervolgens worden, op basis van de gradiënt, de randen gezocht op dit afgevlakte beeld. Daarna wordt een niet-lokale maxima-onderdrukking toegepast om de randen te verdunnen. Tot slot worden de zwakke randen (afkomstig van ruis of reflectie) weg gefilterd via een *hysteresis thresholding* principe waardoor er enkel dunne randen overblijven [11], [26].

Alternatief kan ook de Laplacefilter gebruikt worden om randen te zoeken. Deze filter gebruikt een masker zoals weergegeven hieronder, dit komt overeen met twee keer afleiden. Hierdoor is deze filter isotroop en kan deze dus veranderingen in alle richtingen dedecteren. De Laplacefilter geeft dus weer hoe groot de verandering is [26].

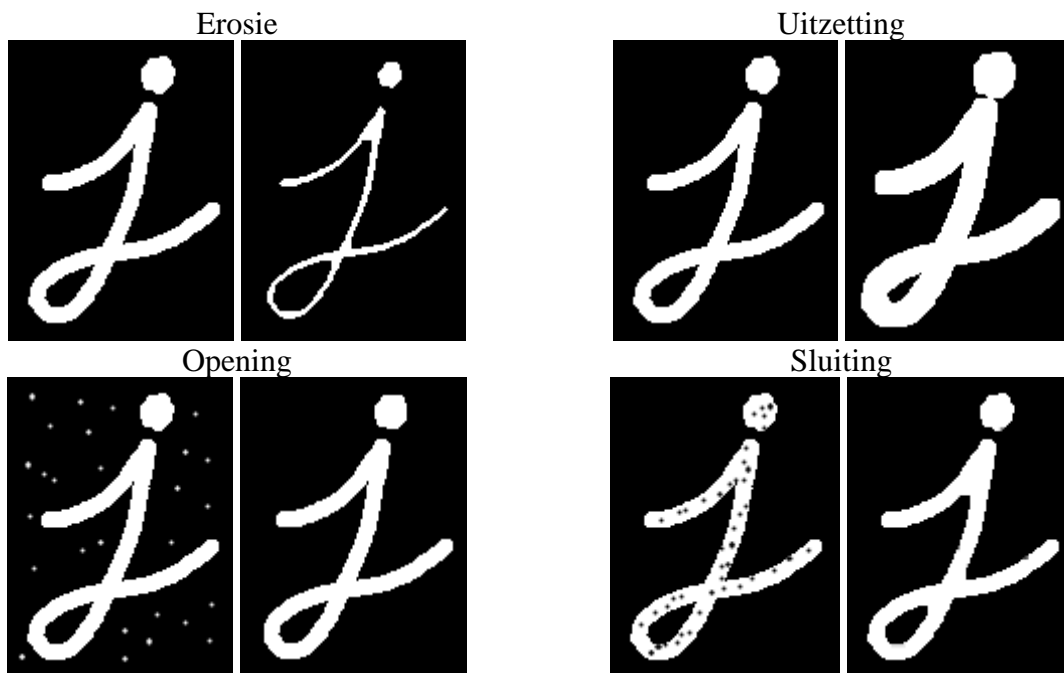
$$Laplacefilter = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \text{ of } \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



Figuur 25: Randdetectie

2.3.3 Morfologische operaties

Na de randdetectie worden morfologische operaties uitgevoerd om het profiel te vervolledigen. Deze maskeroperaties worden uitsluitend toegepast op binaire beelden. De meest gebruikte morfologische operaties zijn erosie (*erosion*), uitzetting (*dilation*), opening (*opening*) en sluiting (*closing*). Bij erosie wordt de pixel zwart als er minstens één zwarte pixel in het masker is. Uitzetting werkt omgekeerd, hier wordt de pixel wit als er minstens één witte pixel in het masker is. De openingsoperatie is erosie gevolgd door uitzetting (met hetzelfde masker) en de sluitingsoperatie is uitzetting gevolgd door erosie ([11], [27]). Het effect van deze morfologische operaties wordt geïllustreerd met Figuur 26.



Figuur 26: Morfologische operaties [27]

2.3.4 Kenmerkherkenning

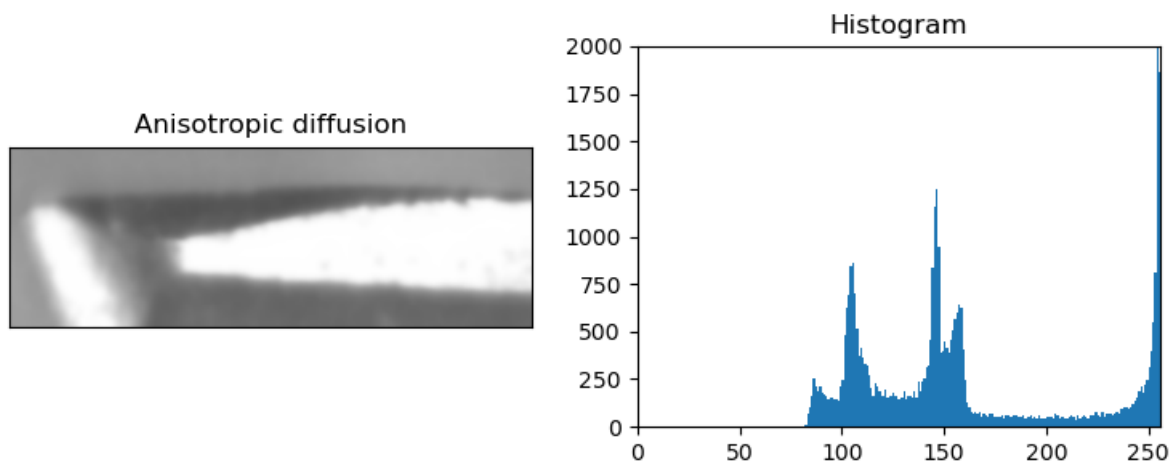
Een eenvoudige manier om kenmerken te herkennen, is door gebruik te maken van thresholding, hierbij wordt iedere pixel $f(x,y)$ vergeleken met een grenswaarde T [17]. In formule 5 wordt de functie gegeven voor binaire thresholding [28] [29].

$$g(x,y) = \begin{cases} 1 & \text{als } f(x,y) > T \\ 0 & \text{als } f(x,y) \leq T \end{cases} \quad (5)$$

Het instellen van de juiste grenswaarde is niet altijd even gemakkelijk, zeker wanneer de belichting in het beeld varieert. Met behulp van globale thresholding (ook wel dynamische treshold genoemd) kan er voor iedere pixel individueel een grenswaarde $T(x,y)$ berekend worden zoals in formule X. De grenswaarde $T(x,y)$ is het gemiddelde (of Gaussian gewogen gemiddelde) van een ingestelde kernel min een constante waarde C [29] [30]

$$g(x,y) = \begin{cases} 1 & \text{als } f(x,y) > T(x,y) \\ 0 & \text{als } f(x,y) \leq T(x,y) \end{cases}$$

Een histogram geeft weer hoeveel pixels dezelfde intensiteit hebben zoals op Figuur 27 [11]. De histogram is een andere manier om het beeld te begrijpen en kan inzicht geven in de contrast, helderheid en intensiteitsverdeling [31]. De pieken geven weer welke waardes veel voorkomen en kunnen gebruikt worden om grenswaardes te selecteren voor thresholding.



Figuur 27: Histogram van anisotropic diffusion

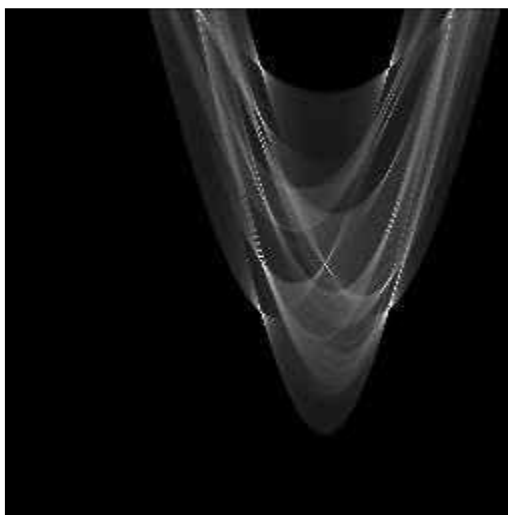
Met behulp van de Hough lijn transformatie kunnen lijnen gezocht worden in een beeld. De lijnen worden voorgesteld in polaire coördinaten zoals formule 6.

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (6)$$

Voor Ieder punt (x_0, y_0) is er een verzameling van lijnen die door dat punt gaat:

$$\rho = x_0 \cdot \cos(\theta) + y_0 \cdot \sin(\theta)$$

Dit betekent dat paren van (ρ, θ) lijnen voorstellen door het punt (x_0, y_0) . Door alle combinaties van (ρ, θ) weer te geven voor alle punten waar lijnen door kunnen gaan ontstaat er Figuur 28. De witte snijpunten stellen lijnen voor die door meerdere punten gaan en zijn dus de werkelijke lijnen va het originele beeld [11], [32].



Figuur 28: Hough lijn transformatie [11, p. 34]

Een contour is een verzameling van punten die alle punten van een grens met elkaar verbinden. Het zoeken van contouren wordt meestal uitgevoerd op binaire afbeeldingen [33].

Op basis van de contour kunnen er momenten uitgerekend worden volgens formule 7 [11]. Deze momenten kunnen gebruikt worden om het massamiddelpunt van de contour te vinden [34].

$$m_{ji} = \sum_{x,y} (array(x,y) \cdot x^j \cdot y^i) \quad (7)$$

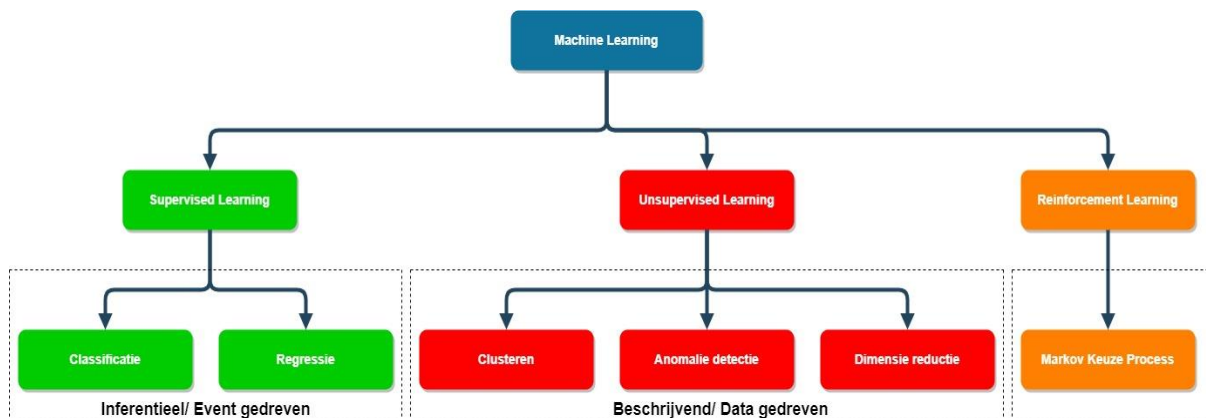
Met massamiddelpunt:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{en} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (8)$$

2.4 Machine Learning

Machine learning is een manier van programmeren voor iets dat zeer moeilijk sequentieel te beschrijven is. Twee goede omschrijvingen van wat machine learning precies inhoudt, zijn die van Arthur Samuel in 1956 die het beschrijft als “the ability for computers to learn without being explicitly programmed” en deze van Tom Mitchell in 1997 die machine learning definieert als “the process of teaching a computer to perform a particular task by improving its measure of performance with experience” [35, p. 169].

Drie principes van machine learning zijn te onderscheiden afhankelijk van de inputs en de gewenste outputs. Deze drie categorieën zijn: *supervised learning*, *unsupervised learning* en *reinforcement learning* (Figuur 29). Dit hoofdstuk is vooral gebaseerd op de boeken van [35] en [36].



Figuur 29: Soorten machine learning [36, p. 83]

2.4.1 Supervised learning

Supervised learning vertrekt vanuit een dataset van inputs met hun bijhorende gekende outputs. Vervolgens zal het algoritme uit deze data leren generaliseren en patronen zoeken. Het gevolg is dat het algoritme dan bij een nieuwe input zelf het resultaat kan schatten. Twee vaak voorkomende modelleermethoden zijn classificatie en regressie.

Bij classificatie is de output een label of variabele. De keuze van de output van een variabele volgt uit eigenschappen van de inputdata. Het algoritme leert tijdens de trainingsfase probabilistische verbanden te leggen tussen de inputvariabelen en de bijhorende outputs. Een voorbeeld van machine learning op basis van classificatie is het filteren van spam e-mails: door de meest voorkomende woorden in een spam e-mail aan het algoritme mee te geven kan het algoritme met een welbepaalde zekerheid nieuwe e-mails beoordelen naar kans op spam.

Regressie daarentegen geeft als output reële continue getallen. Uit een gegeven dataset van inputs en outputs legt het algoritme een verband tussen de beide. Figuur 30 beschouwt als voorbeeld dat de prijs van huizen uit een wijk bepaald wordt door het aantal kamers. Het algoritme leert uit een dataset van de inputkenmerken de gewichtsfactor op de prijs van elke specifieke kamer. Vervolgens kan het algoritme op een nieuwe input dan zelf voorspellen wat de huizenprijs zal zijn.

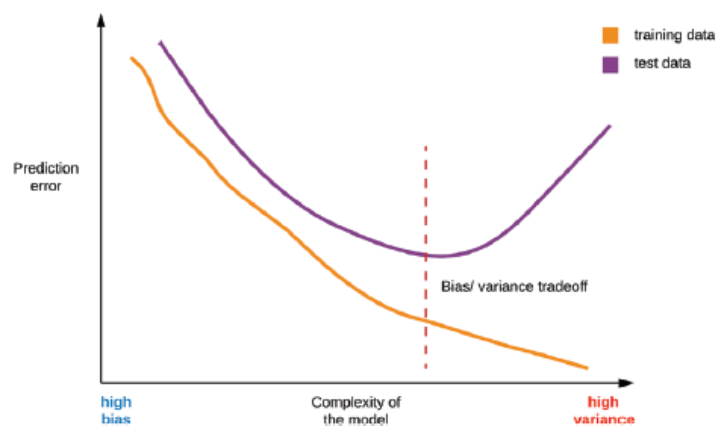
Features			Target feature
#bedrooms	#bathrooms	sq. ft	price
4	6	3	18.3
2	4	1	15.2
...
5	8	5	24.7

} samples

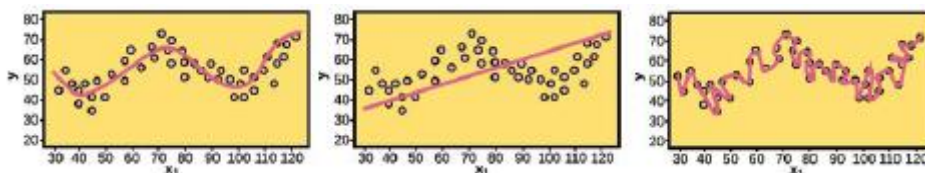
Figuur 30: Voorbeeld huizenprijs aan de hand van regressie [35, p. 173]

Anderzijds kan de dataset ook te nauwkeurig geanalyseerd worden. Het algoritme leert de training dataset te specifiek en kan hierdoor de belangrijke input features niet onderscheiden van de ruis. Figuur 32 toont dat het algoritme wel goed reageert op de trainingsdataset, maar niet op de testdata.

Uit Figuur 32 volgt dat er een optimaal punt bestaat waar de voorspellingsfout minimaal is. Het algoritme heeft in dit punt de belangrijkste inputfeatures uitgefilterd en de ruis verworpen. Figuur 33 toont links de curve hoe het algoritme in dit optimum hoort te reageren bij een nieuwe dataset. Centraal en recht wordt respectievelijk een onderfitting (te hoge bias) en overfitting (te hoge variantie) weergegeven.



Figuur 32: Bias en variantie [35, p. 179]



Figuur 33: Links: goed gefit, midden: onderfit (hoge bias), rechts: overfit (hoge variantie) [35, p. 179]

2.4.2 Unsupervised learning

In sommige situaties is het niet mogelijk om een eenduidige output te bepalen aan de hand van een voorafgaande dataset. De opdracht bij dit soort situaties is om de patronen van de dataset te analyseren en vervolgens de elementen te groeperen op basis van specifieke eigenschappen. Bij unsupervised learning is het dus niet de bedoeling dat er één specifieke oplossing wordt gevonden, maar eerder wordt gezocht naar de correlatie tussen verschillende elementen. Er bestaan verschillende methodes voor unsupervised machine learning, deze literatuurstudie beperkt zich tot drie van de veelgebruikte methodes.

Clustering vertrekt van een dataset met willekeurige elementen. Het doel is om de dataset op te delen in groepen waarin elementen met verwante eigenschappen bij elkaar zitten.

Anomalie detectie veronderstelt een homogene dataset en doorzoekt deze op uitschieters. Het doel van deze methode is om een grote dataset te reduceren naar een kleinere. Hierdoor wordt

de data minder intensief om door te rekenen en kan een betrouwbaarder beeld gemaakt worden van de overeenkomstige kenmerken van de elementen in de dataset.

Dimensie reductie heeft, net zoals anomalie detectie, het doel om een dataset in omvang te reduceren. In deze methode wordt de dataset verkleint op een manier zodat de voornaamste data nog altijd beschikbaar blijft. Een dataset met veel repeterende elementen kan hierdoor sterk vereenvoudigd worden en moet hierbij niet inboeten op betrouwbaarheid. Het grote verschil met anomalie detectie is dat uitschieters wel behouden zullen blijven omdat deze sterke verschillen hebben ten opzichte van de overige data.

2.4.3 Reinforcement learning

Het basisdoel van reinforcement machine learning is het in kaart brengen van situaties met daaropvolgende acties die de maximale beloning opleveren. Bij het in kaart brengen van de actie moet het algoritme niet alleen rekening houden met de onmiddellijke beloning, maar ook met de volgende en alle verder beloningen. Bij reinforcement learning zit het model in een feedbackloop waarin alle acties rechtstreeks beoordeeld worden. Omdat dit niet van toepassing is in ons onderzoek wordt voor meer informatie verwezen naar [35] en [36].

2.4.4 Machine Learning algoritmes

Er bestaan verschillende algoritmes om machine learning toe te passen, deze worden globaal ingedeeld in lineaire en niet-lineaire methodes. lineaire methodes worden toegepast als er verwacht wordt dat een lineair verband bestaat tussen de inputdata en outputinformatie. Deze methodes worden verder niet besproken omdat gereedschapsslijtage een niet-lineair proces is zoals zichtbaar in Figuur 3.

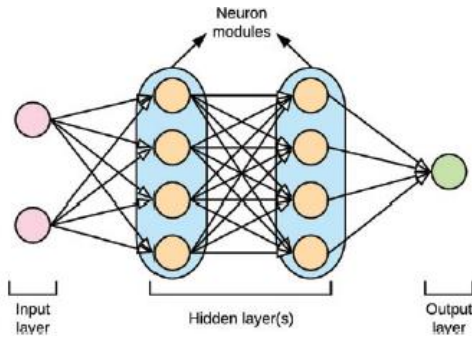
Niet-lineaire machine learning algoritmes proberen een verband te leren tussen de belangrijkste inputfeatures en de output. De meest gebruikte niet-lineaire methodes zijn:

- K-nearest neighbours,
- Classification and regression trees (they form the foundation for ensemble methods such as boosting and bagging),
- Support vector machines,
- Artificial neural network (ANN).

Van deze niet-lineaire machine learning algoritmes worden neural networks in het volgende hoofdstuk verder uitgelegd omdat deze in de literatuur vaak worden toegepast.

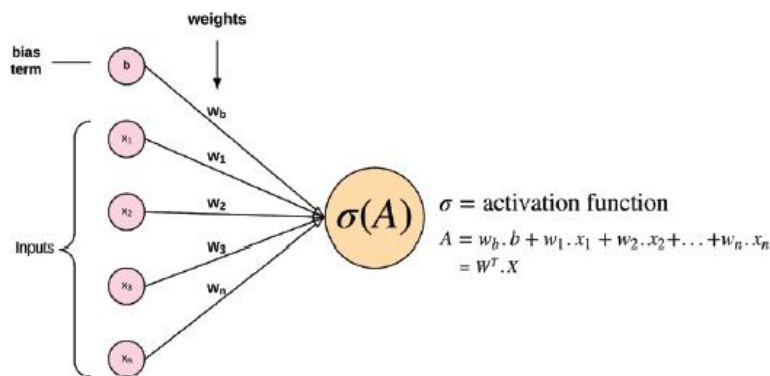
2.4.5 Neural network

Een Artificial neural network (ANN) is een collectie van nodes die geïnspireerd zijn op biologische neuronen, deze nodes zijn verbonden met elkaar zoals de synapsen in een biologische brain. Figuur 34 geeft de opbouw van een ANN weer, deze bestaat uit een input laag, verborgen lagen en een output laag. Het idee achter een ANN is dat dat iedere verborgen laag features leren te herkennen uit de inputdata. Door het gebruiken van meerdere verborgen lagen achter elkaar kunnen de nodes in de verdere verborgen lagen meer complexe features leren herkennen uit de vorige laag, hierdoor wordt deze techniek ook wel deep learning genoemd.

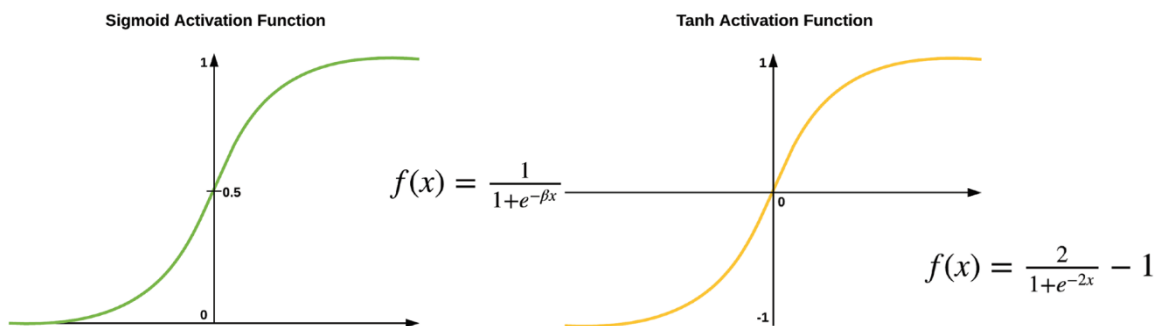


Figuur 34: Opbouw van een Artificial neural network [35, p. 332]

Figuur 35 toont hoe de informatie van een vorige laag naar een node in de volgende laag gaat. Iedere connectie heeft een gewicht toegewezen, hierdoor is de input van de node de som van de uitgangen van de vorige laag maal de gewichten plus een bias b . Op deze som wordt een niet-lineaire activatie functie σ toegepast. Figuur 36 geeft twee mogelijk activatie functies weer. Deze functies zetten de input om naar een waarde tussen $[0,1]$ of $[-1,1]$. Deze waarde is de uitgang van de node en wordt vervolgens gebruikt als input voor de nodes in de volgende laag

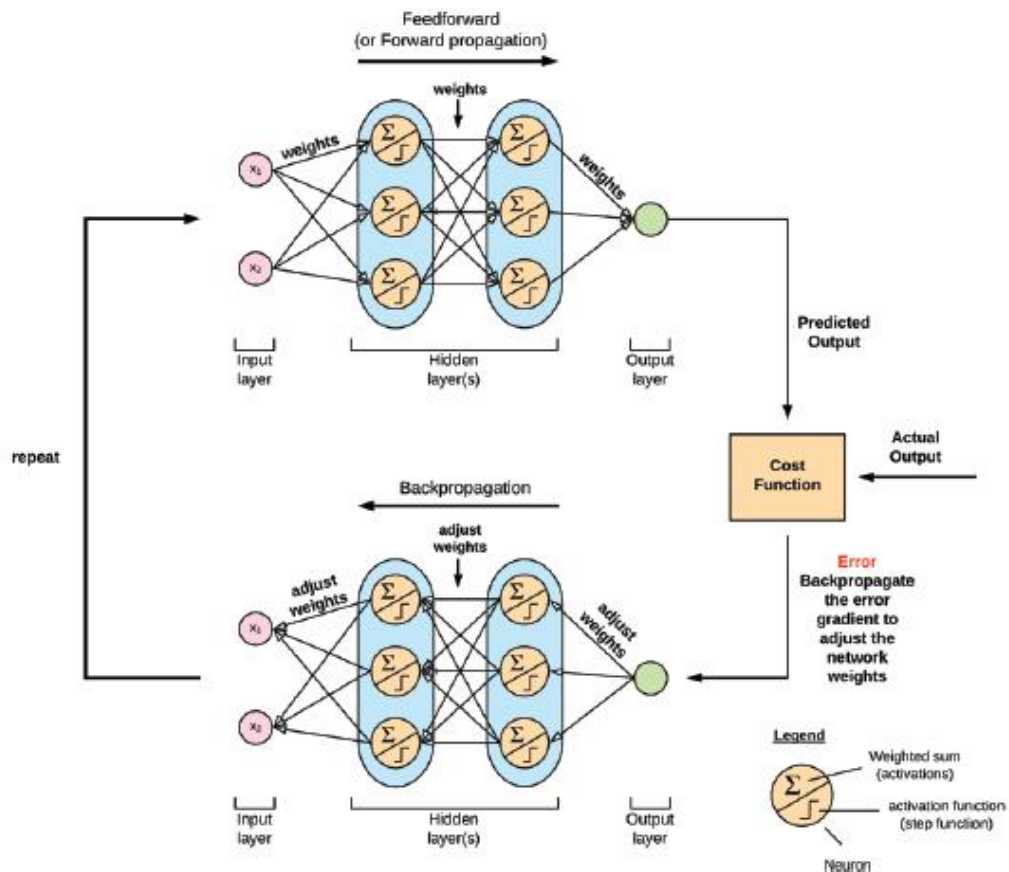


Figuur 35: Berekening voor de input van een node [35, p. 335]



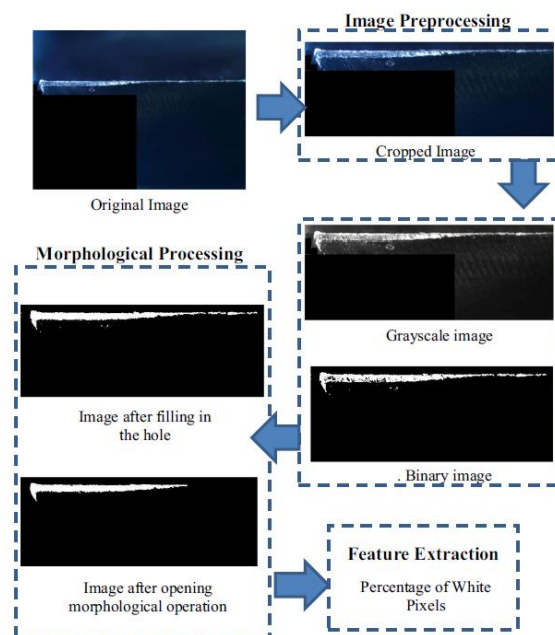
Figuur 36: Sigmoid en tanh activatie functies [35, pp. 340-341]

De waarden van de gewichten in Figuur 35 zijn initieel willekeurig. Hierdoor zijn de uitkomsten initieel ook willekeurige waarden. Met deze uitkomsten wordt er een kostenfunctie berekend (*mean squared error*), deze bepaalt hoe goed of slecht een ANN de taak uitvoert. Met behulp van *backpropagation* wordt de kostenfunctie van de uitkomst en de werkelijke uitkomst van de dataset vergeleken en worden de gewichten van verbinden aangepast om de kostenfunctie te verkleinen. Door deze stappen te herhalen zoals in Figuur 37 wordt de kostenfunctie geminimaliseerd en zal het ANN steeds nauwkeuriger worden.

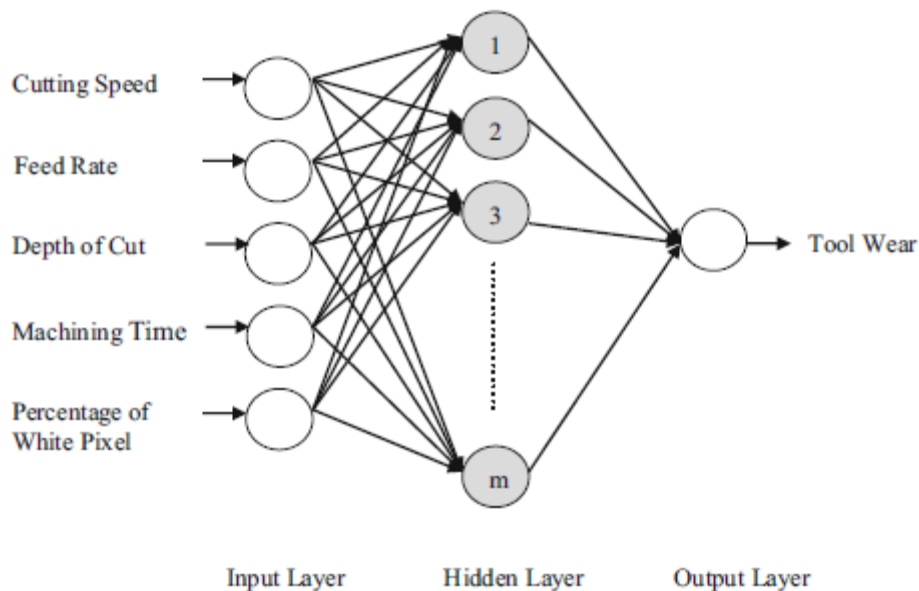


Figuur 37: Backpropagation [35, p. 338]

In [37] wordt er een ANN toegepast op parameters die verkregen zijn door beeldverwerking om hieruit de flank wear te bepalen met 0.1mm nauwkeurigheid. Anderzijds kan een ANN ook gebruikt worden om de slijtage te voorspellen. Zo wordt er in [38] eerst een percentage witte pixels bepaald die overeenkomen met de huidige slijtage volgens Figuur 38. Vervolgens wordt dit gebruikt samen met de machine parameters om de slijtage te voorspellen (Figuur 39).



Figuur 38: Beeldverwerking om gebied van de slijtage te bepalen [38, p. 1372]

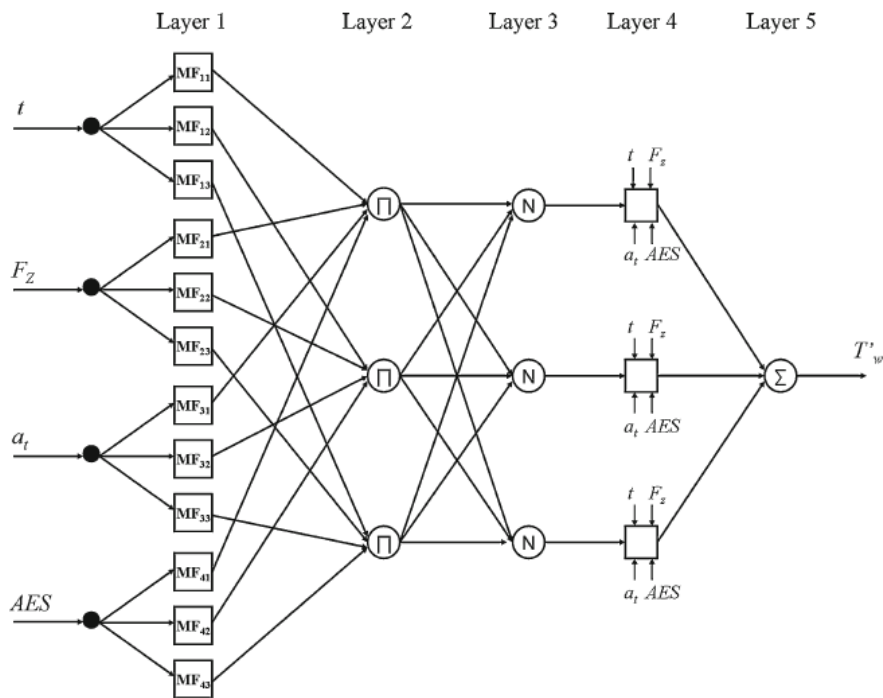


Figuur 39: ANN om de gereedschapslijtage te voorspellen [38, p. 1372]

2.4.6 Neuro-fuzzy

Door het combineren van de zelflerende eigenschappen van een neural network en de ervaring van de operator in de vorm van een fuzzy systeem ontstaan er neuro-fuzzy hybride systemen [39]. Deze systemen zijn ook te vinden onder de naam *Adaptive Neuro Fuzzy Inference System* (ANFIS). Uit de conclusie van [2] volgt dat ANFIS een zeer robuuste manier is voor het voorspellen van gereedschapslijtage uit indirecte sensoren.

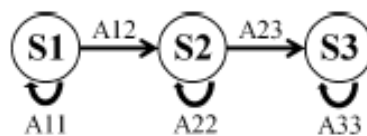
De werking van een ANFIS netwerk wordt uitgelegd aan de hand van Figuur 40. De vierkante nodes zijn nodes met aanpasbare parameters, de ronde nodes zijn voorgedefinieerd. In de eerste laag worden de inputs omgezet naar lidmaatschapsgraad (of waarheidsgraad). Laag twee en drie worden ook wel de rule layer genoemd omdat de nodes in deze lagen de fuzzy regels voorstellen. De vierde laag is de defuzicatielaag waar de waarheidsgraad wordt omgezet in een getal. De laatste laag maakt een gewogen gemiddelde van de vierde laag om één getal bij de uitgang te bekomen. ANFIS gebruikt error back propagation zoals een neural network om de parameters in het netwerk aan te passen om zo een beter resultaat uit te komen [39]. In [40] wordt er een ANFIS netwerk gebruikt om de slijtage van de flank te voorspellen met een nauwkeurigheid van 92,42%



Figuur 40: Opbouw ANFIS netwerk [39, p. 872]

2.4.7 Hidden Markov Model

In een Hidden Markov Model wordt de gereedschapsslijtage opgedeeld in verschillende toestanden en wordt er van uitgegaan dat de slijtage verloopt volgens een markovketen. Meestal worden er drie toestanden gedefinieerd in de aard van scherp, semi onscherp en bot. In een markovketen worden de toestanden weergegeven door cirkels zoals in Figuur 41, de pijlen geven de kans aan dat de gereedschapsslijtage overgaat naar een andere toestand (of terug naar zichzelf) [41].



Figuur 41: Markovketen [41, p. 504]

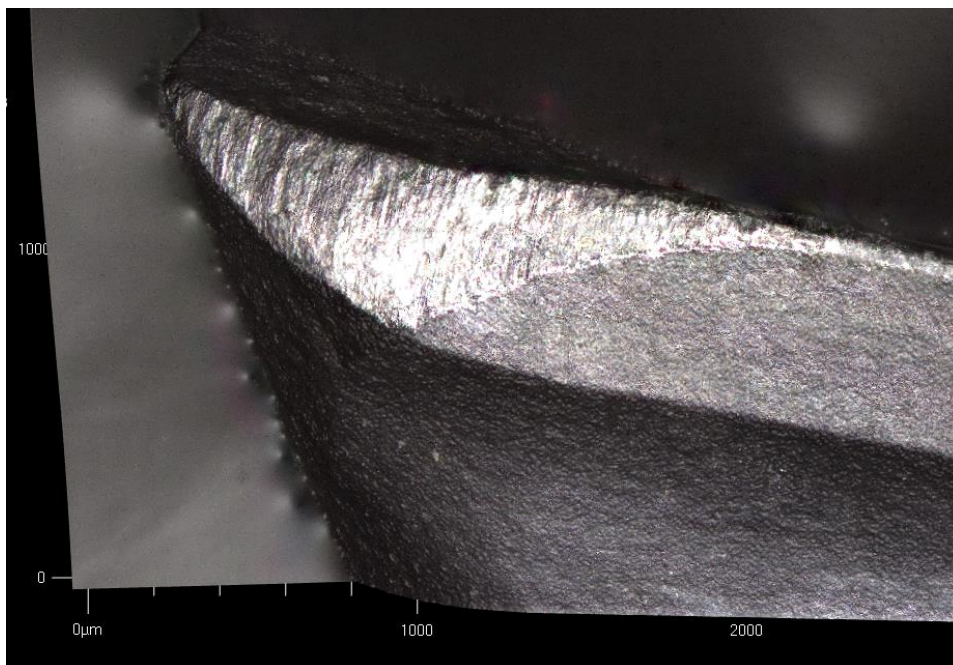
Deze methode wordt gebruikt om gereedschappen te classificeren volgens slijtage met grote nauwkeurigheid, zoals in [41] met 95% nauwkeurigheid of in [42] met 97% nauwkeurigheid.

3 Proefopstelling

Dit hoofdstuk heeft als opzet om de technische opstelling van de masterproef te omschrijven. Eerst wordt beschreven hoe de dataset van snijplaatjes is verkregen en vervolgens de strategie die gebruikt is om de slijtage te detecteren. Hierna volgt een omschrijving van de werkelijk gebruikte proefopstelling. Uit dit deelhoofdstuk zal blijken dat de opstelling nog enkele beperkingen heeft, deze worden in het laatste deelhoofdstuk toegelicht en enkele mogelijke oplossingen worden aangeboden.

3.1 Referentie opstelling/ microscoop

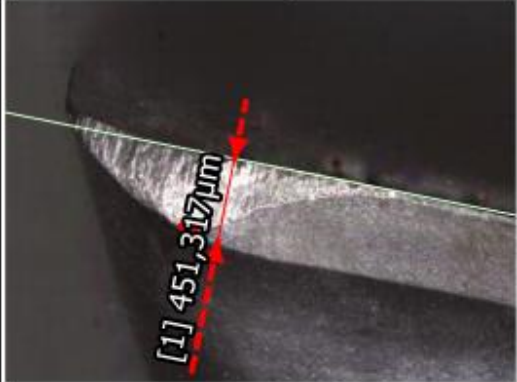
De dataset van snijplaatjes is aangeboden door het onderzoekscentrum SIRRIS te Diepenbeek dat mede deze masterproef ondersteunt. De dataset bevat 50 snijplaatjes die langs beide snijvlakken bewerkt zijn. Hieruit volgen dus 100 beelden waarop de slijtage varieert tussen de 28 μm en de 537 μm . De slijtage van deze snijplaatjes is onder een microscoop gemeten. Figuur 42 toont het outputbeeld van onder de microscoop.



Figuur 42: Beeld van snijgereedschap onder de microscoop

Het microscoopbeeld maakt zowel gebruik van een optisch visiesysteem als een lasersysteem. Dankzij het lasersysteem wordt ook de diepte van het snijgereedschap in beeld gebracht en wordt dus een driedimensionaal beeld verkregen. Merk op dat dit beeld offline gemaakt is. Het snijplaatje werd uit de freeskop gehaald en onder de microscoop geplaatst.

Figuur 43 toont hoe de slijtage gemeten is op het microscoopbeeld. De groene lijn benadert het originele, niet beschadigde snijvlak. Vanuit deze lijn wordt gezocht naar het diepste punt van de slijtage. De afstand tussen de groene lijn en het diepste punt van de slijtage is de maximale slijtage op de flank (flankwear oftewel VB_{Bmax}).

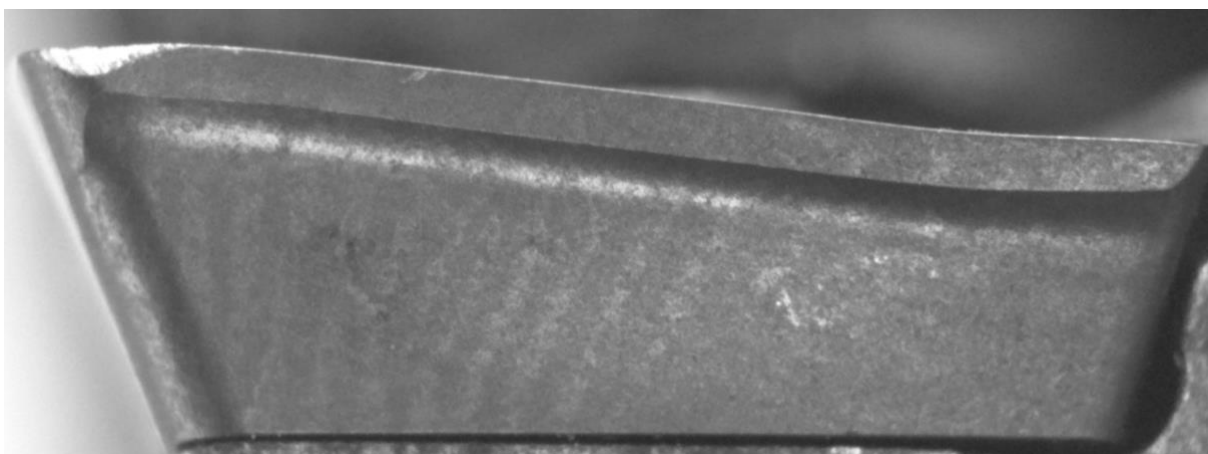
File name	Plane							
	Laser+Optical	Measured values						
MeasureData202 00115144102		<table border="1"> <thead> <tr> <th>No.</th> <th>Measurement name</th> <th>Measured val</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Perpendicular1</td> <td>451,3</td> </tr> </tbody> </table>	No.	Measurement name	Measured val	1	Perpendicular1	451,3
		No.	Measurement name	Measured val				
1	Perpendicular1	451,3						

Figuur 43: Meten van de slijtage onder de microscoop

Op bovenstaande figuur is zowel de positie en oriëntatie van het originele snijvlak als het diepste punt van de slijtage manueel gemeten. Het algoritme dat in deze masterproef wordt uitgewerkt heeft als doel om volledig automatisch te handelen en verwacht dus geen input van de gebruiker. Om dit te behalen moet het snijgereedschap altijd op een consistente manier aan het visiesysteem aangeboden worden. Volgend deelhoofdstuk licht hiervoor een gepaste strategie toe.

3.2 Strategie slijtagedetectie

Het streefdoel is om online de slijtage van snijgereedschap op een freeskop te detecteren en te meten. Om dit mogelijk te maken moet de slijtage sterk opvallen op de gemaakte beelden. Uit de literatuurstudie volgen meerdere mogelijke oplossingen, maar de voorkeur in deze masterproef gaat uit naar een techniek op basis van reflectie. Het snijplaatje wordt met deze methode belicht zodat oftewel de slijtage oplicht ten opzichte van de achtergrond, oftewel de slijtage donker kleurt doordat de omgeving errond het licht van de bron sterk reflecteert. Beide principes worden respectievelijke weergegeven in Figuur 44 en Figuur 45.



Figuur 44: Reflectie van de slijtage op het snijgereedschap



Figuur 45: Contrast tussen slijtage en omgeving vergroten door omgeving te overbelichten

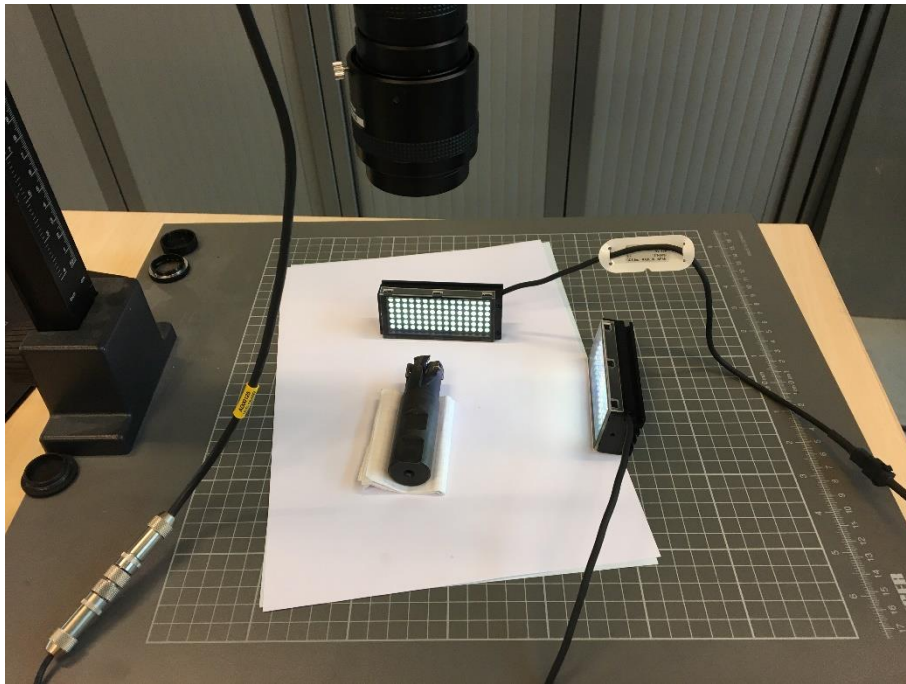
In beide beelden hierboven is hetzelfde snijplaatje weergegeven. Toch gaat de voorkeur gaat uit naar het onderste beeld waar het snijplaatje extra belicht wordt met ledbars. Het belichten van de slijtage zoals dit op Figuur 44 is gebeurd is praktisch moeilijk te realiseren in een freesmachine. Omdat de slijtage niet bij alle snijplaatjes onder dezelfde hoek voorkomt, en dus dieper in het hier geprojecteerde vlak ligt, dient de lichtbron steeds anders opgesteld te worden. Omdat het snijplaatje altijd in dezelfde positie wordt aangeboden, laat de methode van Figuur 45 toe om de lichtbron vast op te stellen in de proefopstelling.

De lichtbron wordt ingesteld op een lage intensiteit waardoor er slechts een beperkt contrast tussen slijtage en snijplaatje ontstaat. De lichtbron instellen op hoge intensiteit introduceert in sommige gevallen fouten. Zo wordt onder andere de slijtage soms toch opgelicht waardoor geen slijtagedetectie op het beeld uitgevoerd kan worden.

3.3 Opbouw proefopstelling

De proefopstelling die gebruikt is voor de masterproef is weergegeven in Figuur 46. Twee ledbars worden gebruikt om zowel het snijvlak als het kopvlak (aangeduid met de rode pijl op Figuur 47) op te lichten. Dit vergroot de kans dat het algoritme de slijtage kan herkennen.

De belichting is zo opgesteld dat één ledbar evenwijdig met as van de frees ligt en de andere ledbar evenwijdig met het kopvlak van de frees. Dit laat toe op de opstelling eenvoudig terug op te bouwen. In hoofdstuk 5 wordt aan de hand van het algoritme uitgelegd welke rol de camera en lens hebben binnen het bepalen van de beelden.



Figuur 46: Proefopstelling



Figuur 47: Belicht snijplaatje

Omdat de proefopstelling een zo goed mogelijke benadering van een CNC-freesmachine moet weergeven, is een freeshouder geprint. Deze is weergegeven in Figuur 48. CNC-machines laten namelijk toe op de robotassen nauwkeurig aan te sturen. Hiermee kan de freeskop, zowel in de praktijk als in de testopstelling, bijna altijd op eenzelfde manier aan het visiesysteem aangeboden worden.



Figuur 48: Freeshouder proefopstelling

3.4 Beperkingen proefopstelling

De proefopstelling gebruikt in deze masterproef is verre van ideaal, deze verwaarloost enkele belangrijke elementen die in de meeste hedendaagse freesprocessen aanwezig zijn. Dit hoofdstuk licht deze kort toe.

De meeste hedendaagse freesprocessen maken bij het vervaardigen van werkstukken gebruik van een koelvloeistof. Deze dient om zowel het werkstuk als het snijgereedschap te koelen. De gebruikte testopstelling vertrekt steeds van beelden waarop de snijplaatjes droog zijn. In de praktijk zal dit niet altijd het geval zijn. De testopstelling en het algoritme houden bijgevolg geen rekening met mogelijke reflecties van olie op het snijgereedschap.

Een tweede veronderstelling die de proefopstelling maakt, is dat het snijgereedschap altijd op eenzelfde manier wordt aangeboden aan het visiesysteem. Er is slechts in beperkte mate rekening gehouden met uitlijnfouten van de freeskop ten opzichte van de camera.

Een derde beperking van de proefopstelling is dat omgevingslicht aanwezig is. Vooral bij de kleinste vormen van slijtage zorgt dit voor een probleem. In sommige gevallen volstaat het invallende omgevingslicht om, ongewenst, de slijtage sterk te laten reflecteren. Dit is mede één van de oorzaken waarom het algoritme moeite heeft om de hoeveelheid slijtage te bepalen indien weinig slijtage aanwezig is. Deze beperking had eenvoudig opgelost kunnen worden, maar is niet in voldoende mate getest in de gebruikte proefopstelling.

De laatste beperking en tevens de tweede oorzaak waarom de slijtagedetectie bij snijplaatjes met beperkte slijtage faalt, is te wijten aan de camera en lensconfiguratie. Uit hoofdstuk 5 zal blijken dat 1 pixel op het beeld overeen komt met 7 μm in werkelijkheid. Dit volstaat in een theoretisch ideale situatie waar de lens zéér duidelijk de kenmerken (meer specifiek de rand tussen het snijgereedschap, de slijtage en de omgeving) van het snijplaatje kan onderscheiden. In de praktijk is dit echter niet het geval, daarom moeten de camera en lens beter op elkaar afgestemd worden. In hoofdstuk 6 wordt daarom een economische analyse uitgevoerd waaruit een betere camera- en lensconfiguratie volgt.

4 Algoritme voor slijtagedetectie

Het algoritme werkt in vier globale stappen, de eerste stap voert een voorbewerking uit om de ruis in de beelden te onderdrukken. De tweede stap bepaalt hierop de positie en de oriëntatie van het snijplaatje. Dit is noodzakelijk omdat deze niet constant is in de proefopstelling. Vervolgens wordt met deze informatie de tip van het snijplaatje geïsoleerd want enkel hier treedt slijtage op. Als laatste stap wordt uit het vorige beeld de contour van de slijtage gezocht en de flank-slijtage $v_{b \max}$ berekend.

4.1 Voorbewerking van het beeld

Het doel van de voorbewerking van het beeld is om de ruis te onderdrukken om zo makkelijker de kenmerken van het snijplaatje te kunnen detecteren. De belangrijkste kenmerken voor de volgende stappen zijn de onderste en de linker rand van het snijplaatje (zoals weergegeven in Figuur 49) én het gebied waar de slijtage zich voordoet. Een makkelijke en veelgebruikte manier om ruis weg te filteren is het toepassen van een Gaussian blur filter. Deze is toegepast met een grote kernel van 15 bij 15 pixels om de ruis voldoende te onderdrukken, maar hierdoor vervagen de randen die waargenomen moeten worden in de volgende stap.



Figuur 49: Origineel beeld van snijplaatje

Een eenvoudige oplossing om deze randen beter te behouden is om het beeld eerst te verkleinen naar de helft van de originele resolutie (Figuur 50). Hierdoor is de rand over minder pixels verdeeld waardoor deze meer gedefinieerd is. Daarnaast vermindert deze stap gedeeltelijk de ruis op het beeld. Anderzijds zorgt deze methode ervoor dat de nauwkeurigheid met de helft daalt.



Figuur 50: Verkleind en Gaussian blur

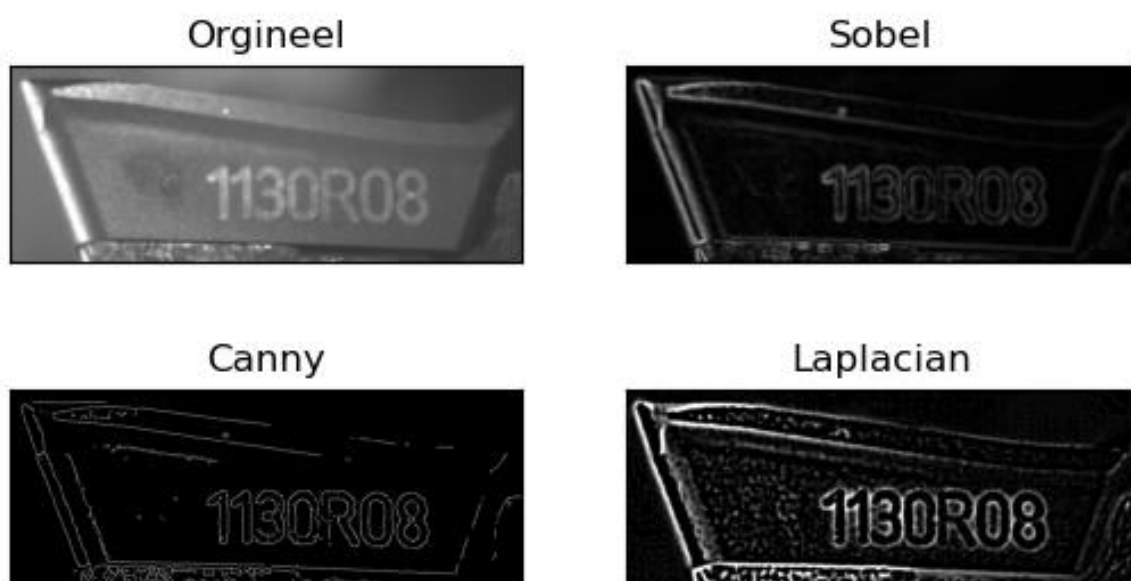
Figuur 51 geeft een alternatieve methode om de ruis te verminderen en tegelijkertijd de randen te behouden, namelijk *anisotropic diffusion*. Deze filter vervaagt het beeld alleen in de richting met een kleine gradiënt (waar de pixels weinig verschillen van elkaar), waardoor de randen behouden blijven. Hierdoor is het niet nodig om de foto eerst te verkleinen. Het nadeel is dat deze methode meerdere iteraties nodig heeft (Figuur 51 past 15 iteraties toe) en dus meer tijd kost. Als vergelijking het uitrekenen van Gaussian blur duurt gemiddeld 0,50 ms en van anisotropic diffusion 350 ms.



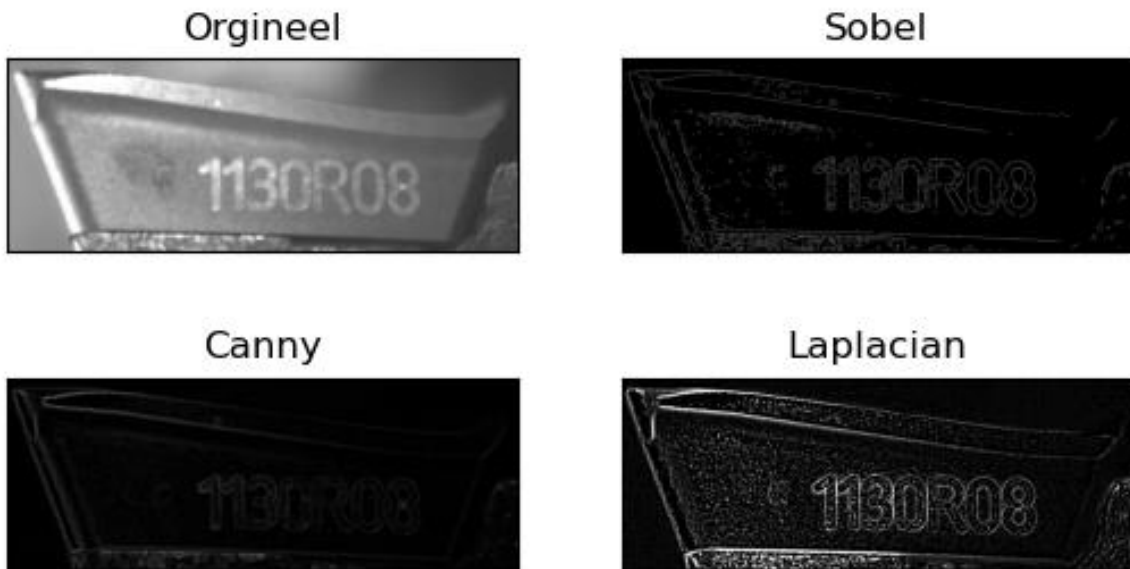
Figuur 51: Anisotropic diffusion met iteraties = 15, kappa = 70 en gamma = 0,2

4.2 Positie en oriëntatie bepalen

Omwille van de opstelling is de oriëntatie en positie van het snijplaatje niet direct af te leiden uit de beelden. Om deze toch te bepalen wordt de onderkant en linkerflank van het snijplaatje gezocht op de beelden. Verschillende methodes zijn mogelijk om de randen waar te nemen, veelgebruikte technieken zijn: Laplace filter, Sobel edge detector en Canny edge detector. Deze worden vergeleken in Figuur 52 met Gaussian blur als voorbereiding om de ruis te onderdrukken. In Figuur 53 worden deze methodes opnieuw vergeleken maar met anisotropic diffusion als voorbereiding. Het besluit is dat de randdetectie op basis van Gaussian blur en op basis van anisotropic diffusion vergelijkbare resultaten leveren.



Figuur 52: Laplace filter, Sobel edge dedector, Canny edge dedector op basis van Gaussian blur



Figuur 53: Laplace filter, Sobel edge dedector, Canny edge dedector op basis van anisotropic diffusion

Op basis van Figuur 52 is er voor gekozen om de Canny edge dedector niet te gebruiken omdat de te detecteren randen niet volledig weergegeven zijn. Verder moeten voor deze functie twee parameters opgegeven worden die voor ieder beeld licht verschillend en moeilijk automatische te bepalen zijn. Bij de Sobel edge dedector is de intensiteit van de randen niet overal even sterk. Vooral aan de onderkant van het snijplaatje is de rand moeilijk waarneembaar met deze techniek. De Laplace filter toont de randen in sterk contrast, maar de intensiteit is niet homogeen. Daarnaast reageert deze filter hevig op de ruis, daarom wordt een kernel van 7 gebruikt.

In de opstelling zijn de beelden niet gelijkmatig belicht waardoor de gradiënt van de onderkant van het snijplaatje niet overal even sterk is. Hierdoor is het toepassen van gewone thresholding niet mogelijk. Bijgevolg gaat de voorkeur naar adaptive thresholding. Deze functie kan automatisch voor kleinere gebieden afzonderlijk de grenswaarde bepalen [29]. Het beste resultaat werd bekomen door gebruik te maken van Gaussian thresholding met een zeer grote kernel van 501 (Figuur 54). Bij een te kleine kernel wordt de ruis nog meer gedetecteerd.



Figuur 54: Laplace filter na adaptive thresholding met kernel = 501

Figuur 54 toont dat de Laplace filter hevig reageert op de aanwezige ruis na adaptive thresholding. De morfologische operatie ‘openen’ verwijdert een groot deel van de ruis op de

achtergrond. De morfologische operatie ‘erosie’ werd ook getest maar deze creëert gaten in de randen. Beide resultaten zijn weergegeven in Figuur 55



Figuur 55: Morfologische operatie openen (links) en erosie (rechts)

De Hough lijntransformatie vindt de lijnen in het beeld terug. De grenswaarde is hoog ingesteld zodat enkel de meest dominante lijnen, met de meeste stemmen, gedetecteerd worden [32]. De functie stelt namelijk de lijnen voor volgens formule X en geeft als resultaat de ρ en θ van de gevonden lijnen

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (9)$$

Figuur 56 visualiseert de gevonden lijnen op het beeld met originele resolutie.

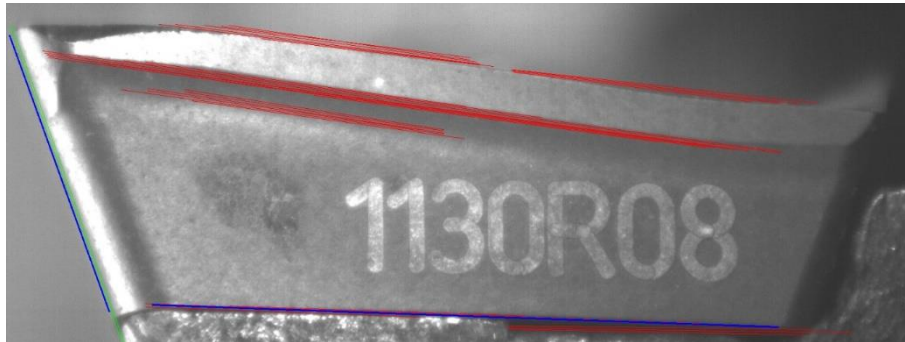


Figuur 56: Hough lijn transformatie met grenswaarde = 500

In tegenstelling tot de normale Hough lijn transformatie geeft de probabilistische Hough lijn transformatie twee punten terug $[x_1, y_1, x_2, y_2]$ die het begin- en eindpunt van het lijnstuk voorstellen. Een extra parameter bij deze functie, bovenop de grenswaarde, is de minimum lengte van het lijnstuk en de maximale afstand tussen twee opeenvolgende pixels. Met deze informatie en extra oplegbare criteria is het makkelijker om te filteren op de juiste lijnen. Op Figuur 57 worden in het rood alle gevonden lijnen weergegeven met een minimum lengte van 200 pixels, grenswaarde van 180 pixels en een maximale afstand tussen twee opeenvolgende punten van 12 pixels. Door de keuze van deze parameters worden meer lijnen gevonden dan gewenst is. Zo staan de grenswaarde en minimum lengte laag om de flank van het snijplaatje te detecteren. Verder staat de maximale afstand tussen twee opeenvolgende punten hoog omdat de basis regelmatig openingen vertoonde. Vandaar dat ook lijnen gevonden worden op plaatsen met veel ruis. Ten slotte zijn de randen in Figuur 55 behoorlijk groot en kunnen er makkelijker meerdere lijnen gevonden worden in dezelfde rand.

In een volgende stap moet uit alle gevonden lijnen, de twee geselecteerd worden die de basis en de linkerflank voorstellen. Voor de lijn die de basis van het snijplaatje representeert is de langste lijn gekozen. De groene lijnen op Figuur 57 zijn de lijnen die een dezelfde hoek hebben

tussen linkerflank en basis. Uit deze lijnen wordt de meest linkse gekozen. De twee geselecteerde lijnen zijn op Figuur 57 in het blauw afgebeeld.



Figuur 57: Probabilistische Hough lijn transformatie

Met de informatie van de basis van het snijplaatje kan het beeld zoals op Figuur 58 geroteerd worden zodat deze horizontaal is. Het snijpunt tussen deze twee lijnen is het aangrijppunt. Dit punt definieert de volledige uitwendige vorm van alle snijplaatjes.



Figuur 58: Geroteerd snijplaatje met aangrijppunt

Alternatief is ook de Harris corner detector getest om de positie en oriëntatie van het snijplaatje te bepalen. Deze methode is niet bruikbaar omdat de plaats van de gevonden punten sterk verschilt tussen de beelden.

Bijkomend is ook een techniek van *template matching* getest. Template matching gebruikt een vooraf gekozen beeld (template) als masker en probeert dit masker terug te vinden in een ander beeld. De pixels van het masker worden overal op het beeld vergeleken. Doordat de proefopstelling niet voor alle beelden gelijk was, zijn sterke intensiteitsverschillen zichtbaar tussen het beeld en het masker. Hierdoor geeft het masker vaak vals positieve resultaten.

4.3 Slijtage afzonderen

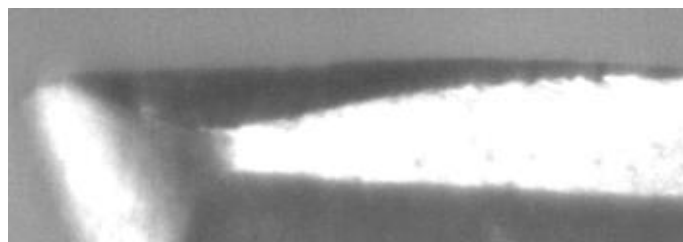
Doordat alle beelden door de vorige stap dezelfde oriëntatie en er een aangrijppunt hebben kan de slijtage afgezonderd worden van de rest van het snijplaatje. De grootte en vorm van deze slijtage varieert sterk in de dataset waardoor er gekozen is om deze uit te knippen met inclusief extra speling. Bijgevolg is de nauwkeurigheid van het aangrijppunt uit de vorige stap minder belangrijk. Figuur 59 geeft weer hoe de slijtage wordt geïsoleerd.



Figuur 59: Slijtage afzonderen uit snijplaatje

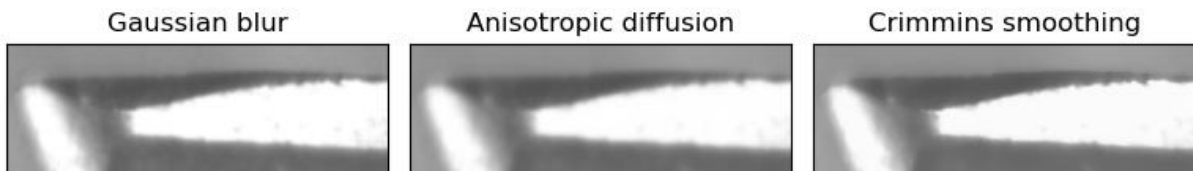
4.4 Schatting slijtage

Deze stap start met de slijtage afgezonderd uit de originele foto zoals wordt weergegeven op Figuur 60.



Figuur 60: Slijtage van snijplaatje

Voor het bepalen van de slijtage is het belangrijk om bij het afvlakken van het beeld de randen zo veel mogelijk te behouden. Daarom zijn verschillende afvlakfilters getest en de meest veelbelovende weergegeven in Figuur 61.

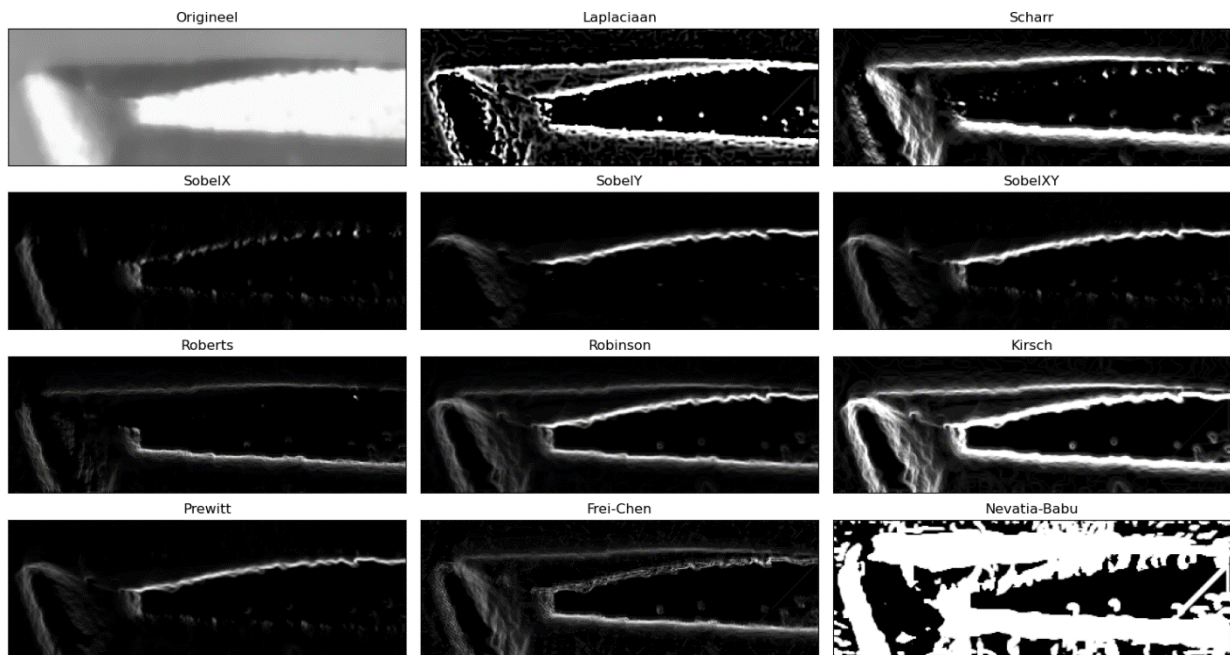


Figuur 61: Afvlakfilters voor slijtage

Om de randen te behouden met Gaussian blur wordt gebruik gemaakt van een kleine kernel van 5 bij 5 pixels, maar hierdoor wordt de ruis slechts beperkt onderdrukt. Bij anisotropic diffusion wordt meer ruis verwijderd en blijven de randen relatief goed behouden. Crimmins smoothing verwijdert zo goed als alle ruis en behoudt de zichtbaarheid van de randen. Maar het nadeel van dit algoritme is dat het zeer traag is. Het uitrekenen van de anisotropic diffusion voor bovenstaande figuur duurde 0,005 s terwijl het uitrekenen van de Crimmins smoothing 2,825 s duurde.

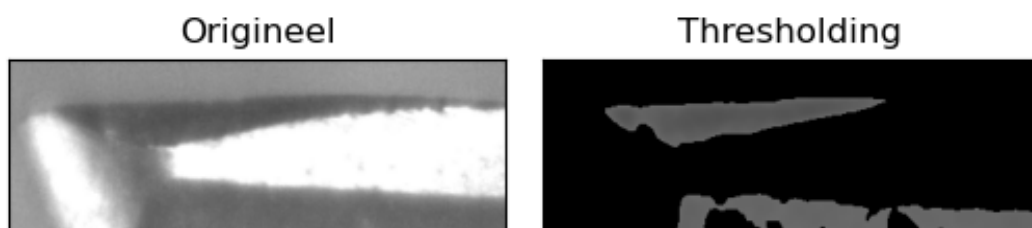
Doordat de snelheid van algoritme slechts in beperkte mate van belang is, wordt de tragere Crimmins smoothing geopteerd als afvlakfilter voor het inputbeeld van de randdetectie omdat deze de beste resultaten terug geeft. In Figuur 62 worden 11 verschillende randdetectietechnieken weergegeven, hiervan leveren de Laplace filter en Kirsch filter het beste resultaat. Het probleem met het gebruik van een randdetectietechniek om de slijtage te

detecteren is dat de belichting rond de slijtage gelijkmatig moet zijn. Indien dit niet het geval is, is de gradiënt van de slijtage niet overal even sterk en worden valse randen gevonden. Deze ongelijkmatige belichting is het gevolg van de vorm van het snijplaatje en de slijtage aan de punt, die moeilijk gelijkmatig te belichten zijn in de proefopstelling.



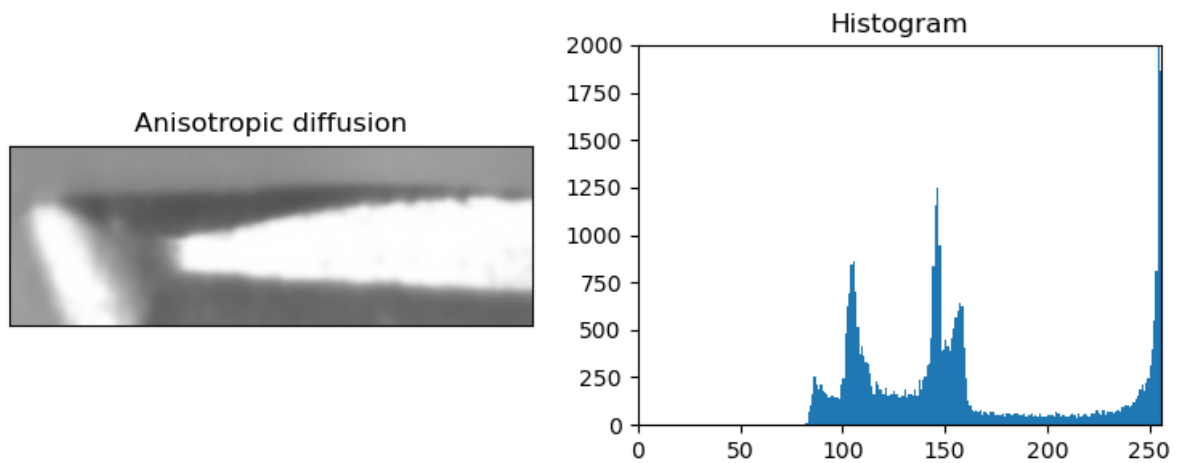
Figuur 62: Randdetectie om slijtage te detecteren

Door gebruik te maken van anisotropic diffusion met 20 iteraties, $\kappa = 50$ en $\gamma = 0,1$ zoals op Figuur 61, heeft het gebied van slijtage ongeveer dezelfde grijswaarde en kan thresholding toegepast worden om de slijtage te detecteren. In Figuur 63 is handmatig een grenswaarde gekozen om de slijtage te vinden. Dit masker, afkomstig van de thresholding, wordt gecombineerd met het originele beeld als visuele controle. Hieruit volgt dat met thresholding de slijtage gevonden kan worden. Maar de grenswaarde die hiervoor ingesteld is, is voor ieder beeld anders.



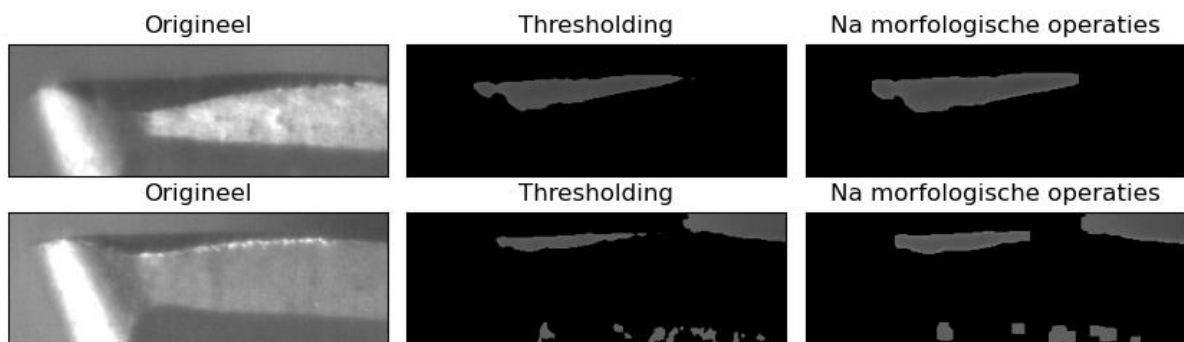
Figuur 63: Thresholding om slijtage te detecteren

Doordat anisotropic diffusion de slijtage omvormt naar grotendeels dezelfde grijswaarde, zal dit ook zichtbaar zijn in de vorm van een piek in het histogram. In Figuur 64 bevindt de piek zich rond de waarde 75. De slijtage is altijd de donkerste vlek op het beeld en is bijgevolg de eerste piek in de histogram. Hierdoor is deze makkelijk te detecteren door het lokale maximum te zoeken. Aan de hand van de plaats van de eerste piek kan er automatisch een grenswaarde voor de thresholding gekozen worden.



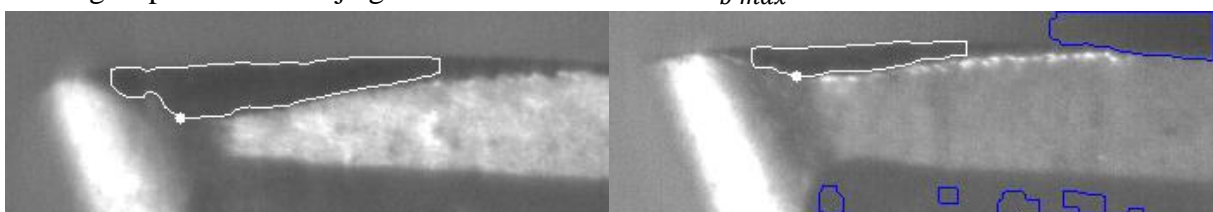
Figuur 64: Histogram van anisotropie diffusie

De thresholding selecteert alle pixels die een grijswaarde hebben onder de grenswaarde, ongeacht waar deze zich bevinden in het beeld. Als gevolg worden er soms ook gebieden geselecteerd die geen slijtage zijn zoals op Figuur 63. De morfologische operatie ‘openen’ verwijdert kleine gebieden die geen slijtage zijn. Vervolgens breidt de morfologische operatie ‘uitzetting’ het masker uit en komt deze meer overeen met de werkelijke slijtage. Het effect van de morfologische operaties is geïllustreerd in Figuur 65.



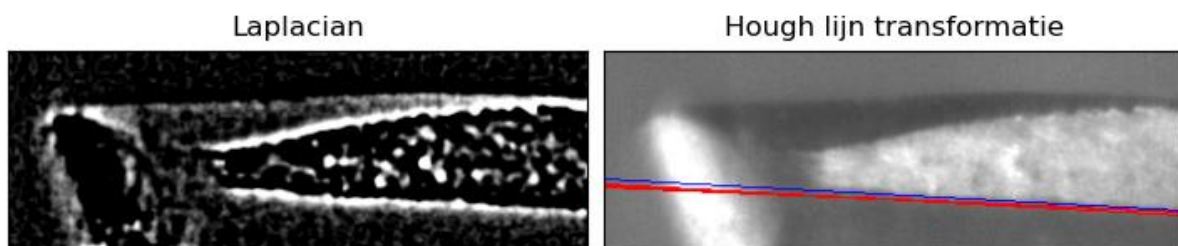
Figuur 65: Masker na morfologische operaties

Het masker na de morfologische operaties wordt gebruikt om de contouren te vinden. De gevonden contouren worden afgebeeld op Figuur 66. Wanneer er meer dan één contour aanwezig is, moet de contour die overeenkomt met de slijtage gefilterd worden. Dit wordt gedaan door van iedere contour het massamiddelpunt uit te rekenen. Door de manier waarop de slijtage is uitgesneden uit het originele beeld is ook geweten dat de slijtage zich in de linkerbovenhoek gaat bevinden van de beelden. Daarom wordt de meest complexe contour (meeste punten om contour te beschrijven) die zich in deze zone bevindt genomen als contour die de slijtage voorstelt. Deze contour, in het wit afgebeeld op Figuur 66, wordt gebruikt om het laagste punt van de slijtage te vinden waaruit later $v_{b \max}$ wordt berekend.



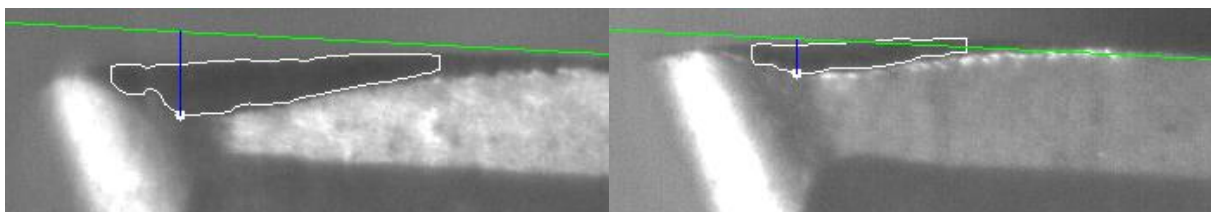
Figuur 66: Contouren van slijtage

Om de $v_{b \max}$ te berekenen moet het snijvlak van het snijplaatje doorgetrokken worden, maar omdat hier juist de slijtage optreedt kan deze ook vervormen. Op Figuur 49 is te zien dat het snijplaatje een boord heeft die evenwijdig loopt met het snijvlak. Deze boord kan gedetecteerd worden en vervolgens omhoog geschoven worden om als denkbeeldig snijvlak gebruikt te worden. Figuur 62 geeft alle verschillende randdetectie methodes weer, hieruit is afgeleid dat de Laplace filter de boord goed kan vinden. Daarom wordt er gebruik gemaakt van een Laplace filter met een kernel van 7 bij 7 pixels en Hough lijntransformatie met een drempelwaarde van 275. Deze functie vindt veel lijnen maar er is ook geweten dat de boord onder een bepaalde hoek ligt waardoor er veel lijnen genegeerd kunnen worden. Op Figuur 67 wordt de gebruikte Laplace filter en de gevonden lijnen weergegeven. Omdat er meestal meerdere lijnen worden gevonden gaat de voorkeur naar de bovenste lijn (blauwe lijn).



Figuur 67: Boord detecteren met Hough lijn transformatie

Door deze lijn naar boven te verplaatsen kan met behulp van het laagste punt van de contour de $v_{b \max}$ berekend worden. Op Figuur 68 wordt visueel deze berekening weergegeven voor twee verschillende snijplaatjes. De groene lijn loopt evenwijdig met het niet beschadigde snijvlak, van hieruit stel de blauwe lijn de $v_{b \max}$ voor.



Figuur 68: $V_{B_{\max}}$ van de slijtage berekenen

4.5 Finaal algoritme

In de eerste stap wordt het beeld verkleind en Gaussian blur toegepast om het beeld af te vlakken en tegelijkertijd de randen te behouden, zoals in Figuur 50. Vervolgens wordt de Laplace filter toegepast (Figuur 52) en globale thresholding met de morfologische operatie openen (Figuur 55) om de randen te detecteren. Op basis van dit beeld wordt er met de probabilistische Hough lijn transformatie de lijnen van het beeld gevonden (Figuur 57). Door het selecteren van de juiste lijnen kan de oriëntatie en positie van het snijplaatje bepaald worden zoals in Figuur 58. Hieruit kan de slijtage uit het origineel beeld afgezonderd worden (Figuur 59).

In de laatste stap wordt er anisotropic diffusion toegepast op de slijtage, waardoor de grijswaarde van alle pixels van de slijtage ongeveer dezelfde grijswaarde krijgen (Figuur 61). Op basis van het histogram (Figuur 64) kan de eerste piek gedetecteerd worden en hieruit een

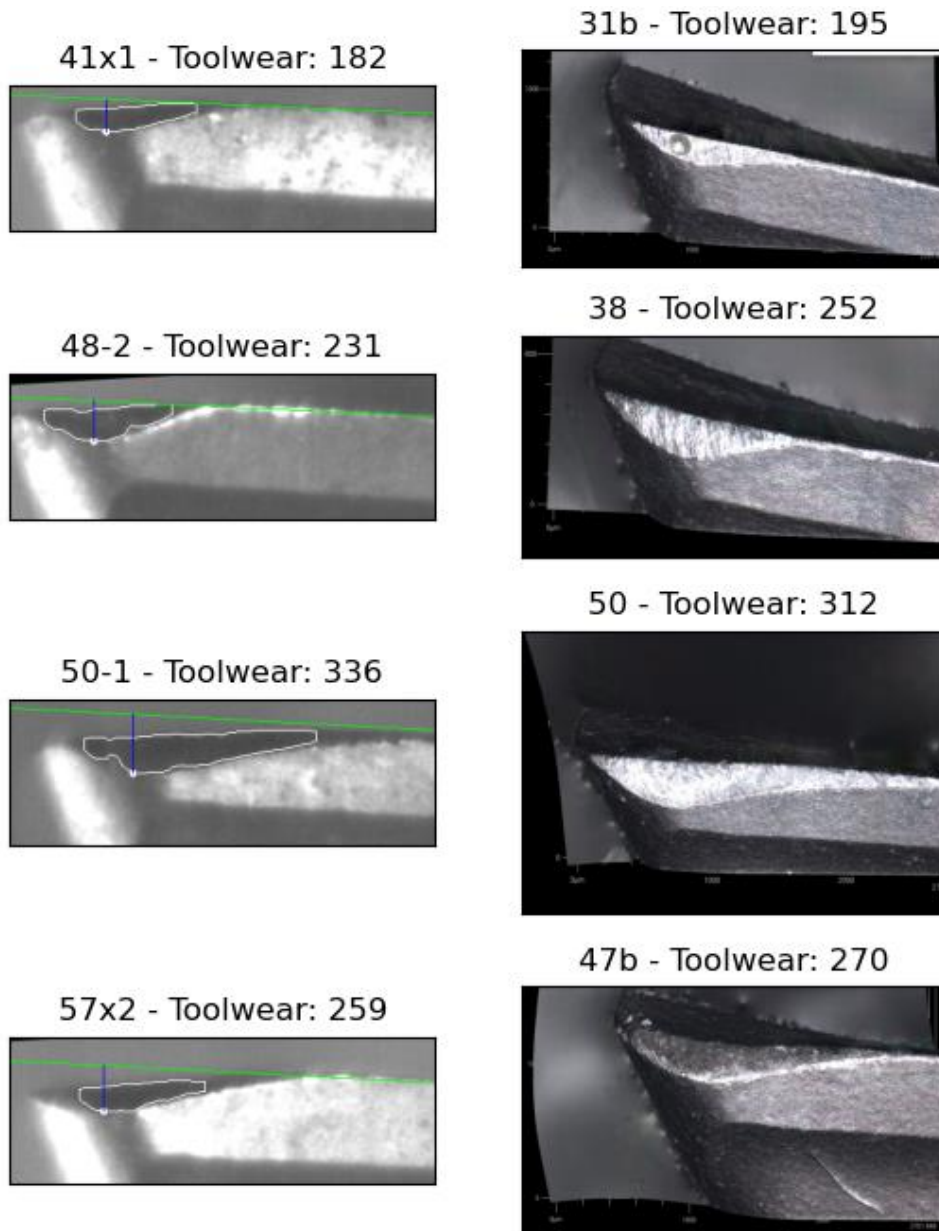
goede grenswaarde geselecteerd worden om thresholding mee uit te voeren zoals Figuur 63. Op deze gevonden slijtage worden er morfologische operaties uitgevoerd om zo de slijtage beter te selecteren (Figuur 65). Daarna kan op basis van dit masker de contouren van de slijtage gevonden worden en het laagste punt van de slijtage (Figuur 66). Om de $VB_{B_{max}}$ te berekenen wordt eerst nog de boord van het snijplaatje gevonden met Hough lijn transformatie op basis van de Laplace filter (Figuur 67). Door deze lijn te verplaatsen kan de $VB_{B_{max}}$ gemeten worden van het laagste punt van de slijtage tot het snijvlak zoals op Figuur 68. De finale code van het algoritme is toegevoegd in Bijlage 1: Python code van het algoritme.

5 Prestaties algoritme

In dit hoofdstuk wordt de prestatie van het algoritme besproken. Eerst worden de beelden van het algoritme vergeleken met deze van de microscoop. In het hoofdstuk worden slechts zes voorbeelden getoond. De andere vergelijkingen tussen microscoopbeeld en het beeld uit het visiesysteem zijn terug te vinden in Bijlage 2: Vergelijking output van het algoritme met resultaten van de microscoop. Vervolgens wordt de nauwkeurigheid van het algoritme vastgesteld aan de hand van een vergelijking van de slijtage op de flank $v_{b\ max}$. Ten laatste wordt er geëvalueerd waar het algoritme moeilijkheden mee heeft.

5.1 Vergelijking algoritme met microscoop

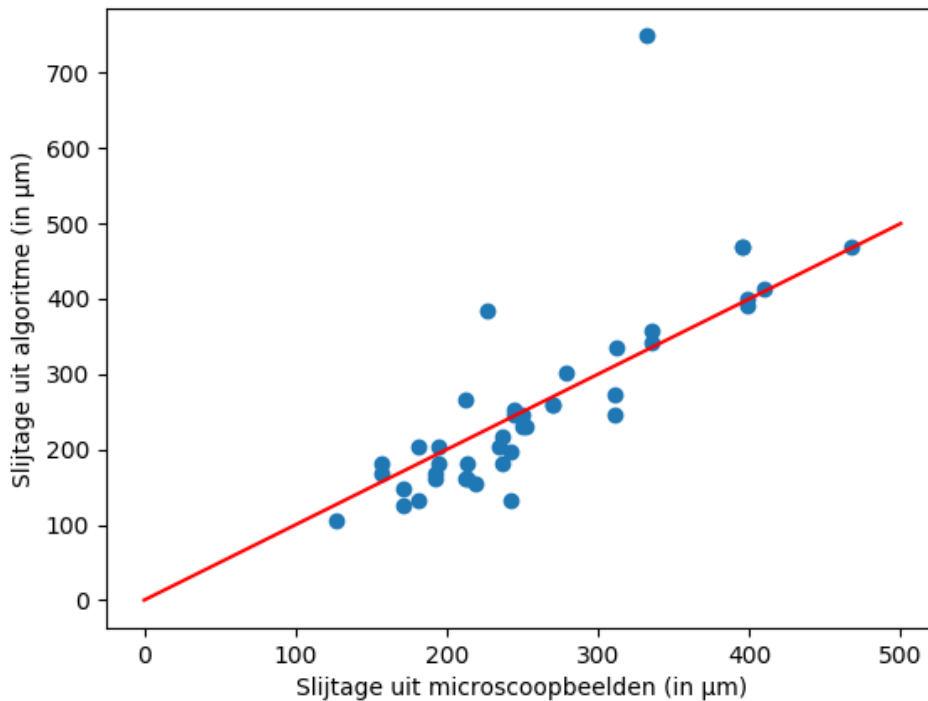
In Figuur 69 worden twee snijplaatjes vergeleken met de microscoop, waar de slijtage wel altijd duidelijk zichtbaar is. Hieruit is af te leiden dat de contour de slijtage niet volledig volgt maar dat deze wel het laagste punt van de slijtage omsluit. Hierdoor kan de $v_{b\ max}$ uitgerekend worden (blauwe lijn).



Figuur 69: Vergelijking algoritme met microscoop

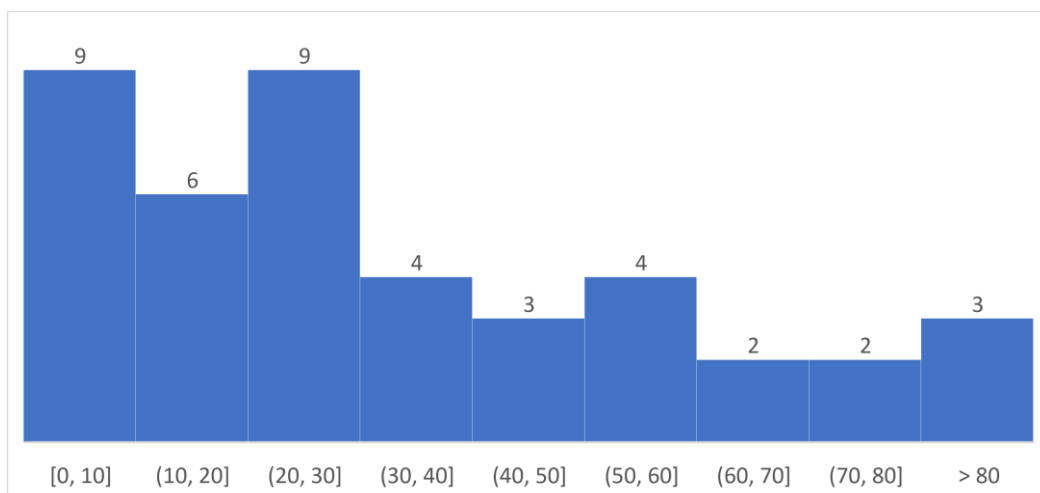
5.2 Nauwkeurigheid algoritme

Op Figuur 70 wordt de nauwkeurigheid van het algoritme weergegeven in een spreidingsplot. De rode lijn geeft weer waar de gedetecteerde waarde gelijk is aan de werkelijke waarde van $v_{b\ max}$. Het algoritme volgt deze trendlijn maar heeft een zeker spreiding. Verder geeft het algoritme eerder een kleinere slijtage terug dan werkelijk aanwezig is. Ook zijn er drie uitschieters waar een fout op de slijtagebepaling meer dan 80 μm bedraagt. Deze uitschieters worden in het volgende deelhoofdstuk verklaard. Door de drie uitschieters te negeren volgt dat de slijtage op de 40 snijplaatjes gemiddeld 28 μm bedraagt.



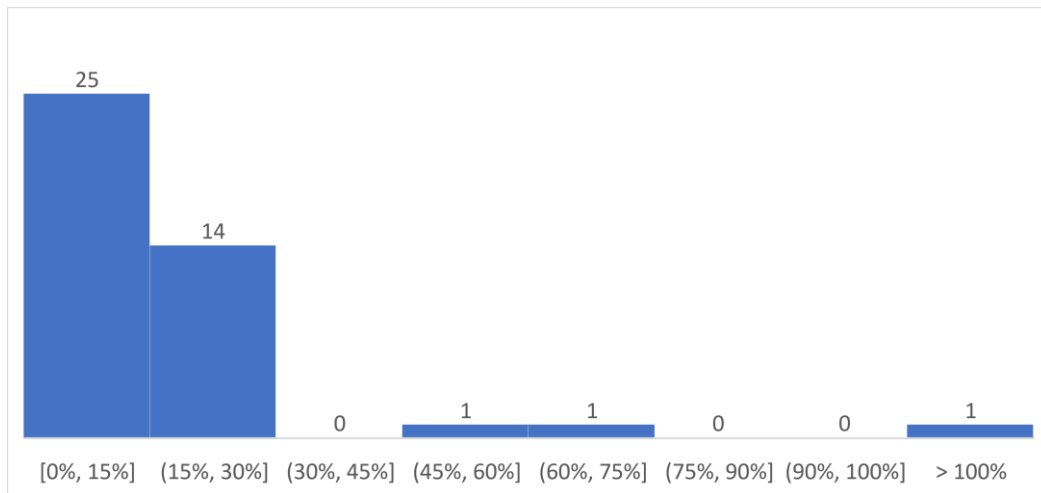
Figuur 70: Spreidingsplot van nauwkeurigheid algoritme

In Figuur 71 wordt de absolute fout weergegeven van het algoritme. Op slechts 9 van de 42 snijplaatjes kon de slijtage op minder dan 10 μm nauwkeurig gemeten worden en op de helft van de snijplaatjes op minder dan 30 μm . Doordat 1 pixel gelijk is aan 7 μm , zorgt een kleine fout in het algoritme voor een grote fout op de slijtage.



Figuur 71: Absolute fout van het algoritme in μm

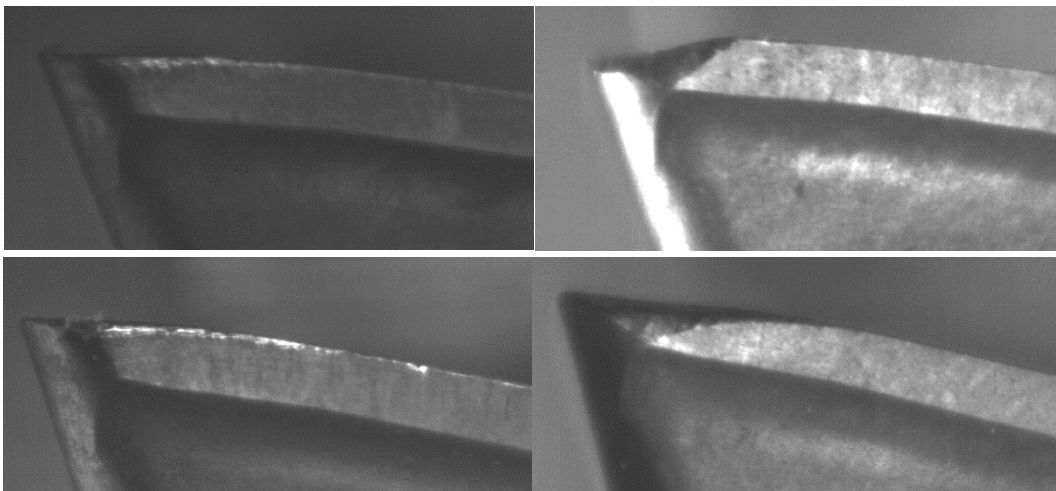
Figuur 72 toont de relatieve fout van het algoritme in procenten. Door Figuur 72 en Figuur 70 naast elkaar te vergelijken wordt het duidelijk dat er geen verband is tussen de afwijking van het algoritme en de grootteorde van de werkelijk aanwezig slijtage.



Figuur 72: Relatieve fout van het algoritme in %

5.3 Moeilijkheden bij algoritme

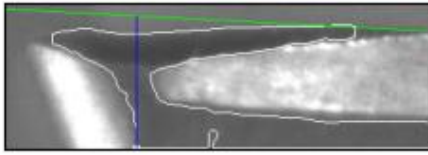
Omwillen van de complexe vorm van het snijplaatje en de proefopstelling is het moeilijk om de snijplaatjes gelijkmatig te belichten en scherpe beelden te verkrijgen. Hierdoor is er bij veel beelden geen eenduidige grens tussen de slijtage en het snijplaatje zoals op Figuur 73.



Figuur 73: Beelden van slecht gedefinieerde slijtage

Omdat er geen mogelijkheid was om deze beelden opnieuw te maken (eventueel met andere lens of camera) zijn deze beelden uit de dataset verwijderd. Hierdoor blijven er 42 van de 106 beelden over. Deze 42 beelden zijn gebruikt om de nauwkeurigheid van het algoritme te testen. De afwijkingen en uitschieters zoals op Figuur 74 zijn te wijten aan een foute grenswaarde voor de thresholding. Hierdoor wordt het foute gebied als slijtage beschouwd en de slijtage foutief uitgerekend. Daarnaast is er in OpenCV geen functie om randen of contouren uit te rekenen op subpixel niveau, waardoor deze nauwkeurigheid blijft op 1 pixel.

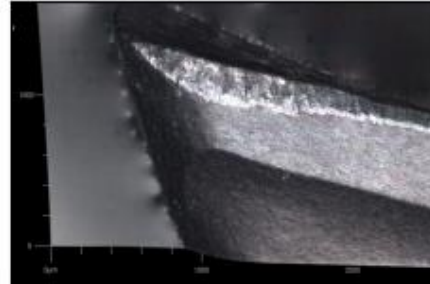
54x2 - Toolwear: 749



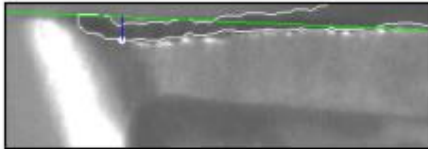
44b - Toolwear: 332



40b - Toolwear: 242



40x2 - Toolwear: 133



Figuur 74: Afwijkingen en uitschieters algoritme

6 Economische analyse

De camera en lens waarmee de beelden uit het vorige hoofdstuk zijn genomen, zijn niet optimaal voor de vooropgestelde doelstelling. Dit hoofdstuk toont de beperkingen van de huidige opstelling en geeft een voorstel naar een beter alternatief.

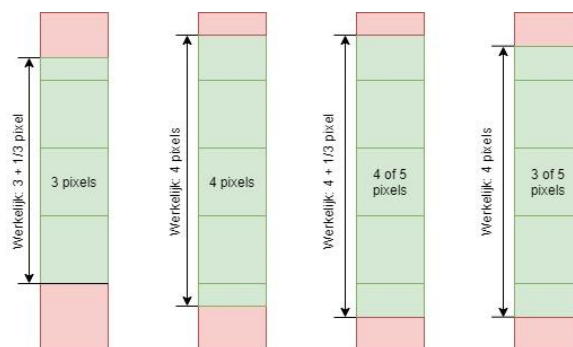
Eerst licht dit hoofdstuk de methode voor het berekenen van de camera en lens toe. Hieruit worden de selectiecriteria bepaald. Aan de hand van deze criteria wordt eerst gekeken naar de beperkende factoren van de huidige proefopstelling. Vervolgens wordt gezocht naar alternatieve camera- en lensconfiguraties. De gevonden alternatieven worden vergeleken met elkaar op basis van verschillende specificaties om zo een slotbesluit te maken voor een meer geschikte opstelling.

6.1 Berekeningen camera

Beginnende met de camera zijn er twee belangrijke kenmerken: de resolutie en de pixelgrootte. Deze zijn beide in rechtstreeks verband met elkaar en worden bepaald door de procesomstandigheden.

Het moment van vervanging van de snijplaatjes is afhankelijk van de toepassing. Bij fijne verspanende processen moet het snijgereedschap vervangen worden wanneer de slijtage 150 μm bedraagt. Indien de ruwheid van het werkstuk van minder belang is, dan volstaat het wanneer het snijgereedschap vervangen wordt tussen de 200 μm en 300 μm . Gewenst is dat het visiesysteem dit ook met een nauwkeurigheid van kleiner of gelijk aan 10 μm uitvoert. Dit om minstens beter te presteren dan gereedschapsslijtagebepalingen met indirecte methodes.

Daarnaast zijn de pixels van de sensorchip die gebruikt worden om de slijtage te bepalen gekwantificeerd. Dit wil zeggen dat wanneer de grens van slijtage willekeurig door een pixel loopt, er altijd een fout geïntroduceerd wordt. Figuur 75 toont grafisch dat de grootste fout zich kan voordoen wanneer de slijtage tweemaal exact door het midden van een pixel loopt én het algoritme dit tweemaal op een eenzelfde manier interpreteert. In dit geval zal één pixel meer of minder geteld worden dan er in werkelijkheid aan slijtage is. Bijgevolg moet de grootte van de pixels op de sensorchip kleiner of gelijk zijn aan 10 μm x 10 μm . Opmerking: deze benadering verwaarloost de circle of confusion omdat de slijtage in één vlak voor de camera wordt beschouwd.



Figuur 75: werkelijke slijtage vertaald naar aantal pixels

Afhankelijk van de gekozen pixelgrootte voor de camera zal dit invloed hebben op de benodigde resolutie. Het algoritme gebruikt het volledige snijplaatje om de slijtage te bepalen. Het oppervlak van 3,59 mm x 10 mm, afkomstig uit de datasheet [43], moet volledig op het sensorvlak geprojecteerd worden. De manier van projectie heeft ook gevolgen voor de lens omdat de gewenste vergrotingsfactor wijzigt. Hieronder worden drie methodes verklaart om de resolutie te bepalen samen met hun gevolgen op de vergrotingsfactor.

De eerste methode is zonder een vergroting of verkleining toe te passen (vergrotingsfactor $m = 1$). Een object dat exact één pixel vult op het beeld heeft in werkelijkheid dezelfde afmetingen als de pixel zelf. De resolutie moet dan minstens zo groot zijn dat het snijplaatje volledig op het sensorvlak past. In andere woorden, het aantal pixels vermenigvuldigt met de breedte van één pixel moet groter zijn dan de lengte van het snijplaatje (deze gelijkheid moet zowel in horizontale als in verticale richting voldaan zijn).

De tweede methode vertrekt vanuit de einddoelstelling om een nauwkeurigheid van 10 μm op de slijtagedetectie te verkrijgen. Het beeld op het sensorvlak verkleint ($m < 1$) door de afmetingen van de pixels uit te rekken naar de gewenste einddoelstelling. Hierdoor is een lagere resolutie vereist. De verkleining wordt in de volgende vergelijking geformuleerd:

$$m = \frac{\textit{lengte van 1 pixel}}{\textit{te behalen nauwkeurigheid}} \quad (10)$$

Dit is tegelijk ook de ondergrens van de minimale vergrotingsfactor die steeds behaalt moet worden. Uit de formule van de vergrotingsfactor en de dunne lens vergelijking, zoals uitgelegd in de bronnenstudie, volgt dat de lens hierdoor met een kleinere brandpuntsafstand gedimensioneerd wordt. Het nadeel van deze methode is echter dat er geen ruimte voor mogelijke fouten wordt voorzien. De camera en lens worden bijgevolg op de grens van de doelstellingen gekozen.

De derde en laatste methode probeert om het sensorvlak zo volledig mogelijk op te vullen met het snijplaatje. Hier wordt het beeld van de werkelijkheid maximaal vergroot ($m > 1$). Dit is de bovengrens van de vergrotingsfactor waarbij nog steeds het volledige snijplaatje op het sensorvlak wordt geprojecteerd. In tegenstelling tot de vorige methode leidt dit, zoals duidelijk zal worden in volgend deelhoofdstuk, tot een lens die groter is dan nodig om aan de einddoelstelling te voldoen. De optimale resolutie en vergrotingsfactor bevindt zich bijgevolg tussen de onder- en de bovengrens. Tabel 2 in rubriek 6.3 geeft de uitwerking van de bovenstaande methodes toegepast op de gebruikte camera.

6.2 Berekeningen lens

De keuze voor de lens is complexer dan deze voor de camera. Opnieuw moeten twee kenmerken optimaal gedimensioneerd worden. Dit zijn de brandpuntsafstand en de minimum object distance oftewel MOD. Samen met de vergrotingsfactor uit de camerakeuze zijn deze drie variabelen opnieuw in rechtstreeks verband met elkaar via de dunne lens vergelijking. Daarnaast moet ook de montage van de lens op de camera mogelijk zijn en moet vignettering vermeden worden door een voldoende grote lens te kiezen.

De dunne lens vergelijking geeft het verband tussen de brandpuntsafstand f en de afstanden van zowel het object do als het sensorvlak di tot de lens. Zoals eerder aangehaald in de bronnenstudie is de dunne lens vergelijking:

$$\frac{1}{f} = \frac{1}{do} + \frac{1}{di} \quad (1)$$

De afstand van het sensoroppervlak tot de lens is ongekend. Deze wordt daarom geschreven in functie van de vergrotingsfactor m en de afstand van het object tot de lens do :

$$m = \frac{di}{do} \leftrightarrow di = m * do \quad (2)$$

Herschrijven van de dunne lens vergelijking geeft dan:

$$\frac{1}{f} = \frac{1}{do} + \frac{1}{m*do} \quad (11)$$

$$f = \frac{do*m}{m+1} \quad (12)$$

In deze laatste vergelijking is de vergrotingsfactor m bepaald door de camerakeuze. De afstand van het object tot de lens do ligt niet vast. Maar omdat het snijgereedschap in de opstelling tot zeer dicht bij de lens gebracht kan worden, mag de kleinst mogelijk afstand voor do gekozen worden. Bijgevolg is do gelijk aan de minimum object distance (MOD), deze waarde verschilt van lens tot lens en neemt globaal genomen toe bij een toenemende brandpuntsafstand.

Een lens moet daarnaast ook altijd een groter oppervlak belichten dan de het sensorvlak alleen. De eenvoudige verklaring hiervoor is omdat de sensorchip vierhoekig is, en de lens rond. Indien de lens toch kleiner zou zijn dan het sensoroppervlak, dan zal het beeld lijden aan vignettering of lichtafval. In de hoekpunten van de sensorchip vallen veel minder lichtstralen binnen, de helderheid is hier daarom veel lager. Een voldoende grote lens is bijgevolg gewenst.

Tot slot dienen lenzen ook steeds gemonteerd te worden op een camera. De meest voorkomende bevestigingstypen zijn C- en CS- montringen. Het verschil tussen beide is dat een CS-montring een 5 mm kortere schroefdraaddiepte heeft in vergelijking met een C-montring. Hierdoor is de brandpuntsafstand ook 5 mm korter. Een C- montring lens kan wel op een CS-montring camera gekoppeld worden via een extension ring, omgekeerd om een CS-montring lens op een C-montring camera te koppelen is niet mogelijk.

6.3 Berekening gebruikte camera- en lensopstelling

Dit deelhoofdstuk geeft kort een overzicht van de specificaties van de gebruikte camera en lens, hieruit wordt duidelijk waarom deze niet aan de vereisten voldoen. Volgend deelhoofdstuk bouwt hierop verder door meerdere voorstellen te geven voor andere camera- en lensconfiguraties.

Tabel 1 toont de sensorspecificaties van de camera uit de huidige proefopstelling. De volledige datasheet van de camera is terug te vinden in Bijlage 3: Datasheet van de gebruikte camera. De pixelgrootte van $5,5 \mu\text{m} \times 5,5 \mu\text{m}$ is kleiner dan de benodigde $10 \mu\text{m} \times 10 \mu\text{m}$ en voldoet dus voor de toepassing. De datasheet toont ook dat het actieve optische sensoroppervlak gelijk is aan $11,26 \text{ mm} \times 5,98 \text{ mm}$, dit is voldoende om het snijplaatje van $3,59 \text{ mm} \times 10 \text{ mm}$ met de juiste vergroting op het sensorvlak te projecteren. Schijnbaar lijkt de camera geschikt voor de detectie van gereedschapsslijtage op voldoende nauwkeurige schaal volgens het eerder beschreven algoritme.

Tabel 1: Sensorspecificaties gebruikte camera: MV1-D2048x1088-3D03-760-G2 [44]

Manufacturer / Type	CMOSIS / CMV2000
Technology	CMOS
Optical format	2/3"
Optical diagonal	12.76mm
Resolution	2048 x 1088
Pixel size	$5.5\mu\text{m} \times 5.5\mu\text{m}$
Active optical area	$11.26\text{mm} \times 5.98\text{mm}$
Dark current	$125\text{e}^-/\text{s}$
Read out noise	13e^-
Full well capacity / SNR	$11\text{ke}^- / 105: 1$
Spectral range	Monochrome: < 350 to 950nm (to 10% of peak responsivity)
Responsivity	Monochrome: $1100 \times 10^9 \text{ DN} / (\text{J}/\text{m}^2) @ 520\text{nm} / 8\text{bit}$
Quantum Efficiency	Monochrome: > 60%
Optical fill factor	42% without micro lenses
Dynamic range	60dB
Characteristic curve	Linear, Piecewise linear
Shutter mode	Global shutter

De onder- en bovengrens waarbinnen het snijplaatje op de sensorchip geprojecteerd kan worden is weergegeven in respectievelijk kolom drie en kolom vier van Tabel 2. Verder toont kolom één ook de brandpuntsafstand wanneer de vergrotingsfactor m gelijk is aan 1. De minimum object distance in deze tabel is een schatting. Uit datasheets van verdelers van visiesystemen ([13]) blijkt dat lenzen met een brandpuntsafstand rond 75 mm een MOD rond 300 mm of hoger hebben. Omdat de MOD van de huidige gebruikte camera gekend is wordt deze in Tabel 2 gebruikt.

Tabel 2: Boven- en ondergrens gebruikte camera

Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor $m = \frac{h_i}{h_o}$	$m = 1$	$m = \frac{5,5 \mu m}{10 \mu m}$ $m = 0,55$	$m = \frac{11,26 mm}{10 mm}$ $m = 1,126$
Afstand in de wereld omgerekend naar één pixel	5,5 μm /pixel	10 μm /pixel	4,9 μm /pixel
Benodigde resolutie $\frac{\text{lengte snijplaatje}}{\text{lengte per pixel}}$	Horizontaal: $\frac{10\ 000 \mu m}{5,5 \mu m} = 1\ 819$ Verticaal: $\frac{3\ 590 \mu m}{5,5 \mu m} = 653$	Horizontaal: $\frac{10\ 000 \mu m}{10 \mu m} = 1\ 000$ Verticaal: $\frac{3\ 590 \mu m}{10 \mu m} = 359$	Horizontaal: $\frac{10\ 000 \mu m}{4,9 \mu m} \cong 2\ 048$ Verticaal: $\frac{3\ 590 \mu m}{4,9 \mu m} = 733$
Minimum object distance (MOD)	204 mm	204 mm	204 mm
Brandpuntsafstand $f = \frac{do * m}{m + 1}$	102 mm	73 mm	109 mm

Tabel 2 dat een lens met minimaal een brandpuntsafstand $f = 73$ mm nodig is om de gereedschapsslijtage met de gewenste nauwkeurigheid te bepalen. Het type lens dat gebruikt is in de proefopstelling beschikt over een lens met brandpuntsafstand $f = 108$ mm. De lens en camera-configuratie is bijgevolg dus geschikt voor de beoogde doelstelling. De details van de gebruikte lens zijn kort weergegeven in Tabel 3 en worden in Bijlage 4: Datasheet van de gebruikte lens meer uitgebreid getoond.

Tabel 3: Productinformatie gebruikte lens: Optem MVZL 34-11-10 25 mm 1:1.4 TV Lens [45]

Focal length	108 mm
Aperture	F2,5 - Closed
Max. sensor size	2/3"
Mounting	C-mount
Minimum Object Distance	204 mm
Focusing and shutter	Manual with locking screw

6.4 Resultaat economische analyse

Het vorige deelhoofdstuk vermeldde al dat de gebruikt camera en lens de mogelijkheid boden om deze masterproef met voldoende nauwkeurigheid te doen slagen. Praktische testen tonen aan dat één pixel overeen komt met 7 μm in de realiteit. Bijgevolg betekent dit dat wanneer het algoritme 2 pixels verkeerd meet, de fout op de slijtage direct 14 μm afwijking geeft. Omdat dit meer dan de vooropgestelde 10 μm is worden hieronder nog enkele andere camera- en lensvoorstellen gegeven die mogelijk betere eindresultaten geven.

6.4.1 Andere cameravoorstellen

De vereiste camera en lens moeten voldoen aan de eerder genoemde eisen en zijn bij voorkeur zo laag mogelijk in prijs. Online catalogussen vermelden meestal geen prijs, daarom worden hieronder meerdere camera- en lensvoorstellen gegeven. Eerst worden voorstellen van geschikte camera's gegeven, vervolgens volgen de lenzen die hierop compatibel zijn.

Tabel 4 toont geschikte camera's voor de toepassing. De keuze van de camera bepaalt direct ook de voorwaarden voor de bijhorende lenzen. Een betere camera betekent dat een goedkopere lens gebruikt kan worden. Omgekeerd geldt ook dat een zwakke camera een betere lens nodig zal hebben.

Tabel 4: Cameravoorstellen en eisen lens

Fabrikant	Model	Resolutie (MP)	Pixelgrootte (μm)	Sensorformaat
Photon Focus	MV1-D2048x1088-3D03-760-G2	2,2	5,5	2/3"
Allied Vision	MAKO U-503B	5	2,2	1/2,5"
Allied Vision	MAKO G-040B POE	1,6	3,45	1/2,9"
Allied Vision	MANTA G-158B POE	1,6	3,45	1/2,9"
Allied Vision	MANTA G-319B POE	3	3,45	1/1,8"
Allied Vision	GT2460/C	5,1	3,45	2/3"
Allied Vision	GUPPY PRO F503/B	5	2,2	1/2,5"
Allied Vision	STINGRAY-II F-125B	1,2	3,75	1/3"
Allied Vision	STINGRAY-II F-504B	5	3,45	2/3"
Jai	GO-5101M/C-PGE	5	3,45	2/3"
Jai	SP-12401M/C-PGE-C	12,3	3,45	1,1"
Teledyne Dalsa	GENIE NANO-M/C1450	1,6	3,45	1/2,9"
Teledyne Dalsa	GENIE NANO-C4900	18,1	1,25	1/2,3"

De eerste camera in Tabel 4 is de gebruikte camera voor de proefopstelling. Het valt direct op dat de andere cameravoorstellen over kleinere pixelgroottes beschikken. Hierdoor wordt de slijtage over meer pixels verspreid waardoor een fout minder invloed op het eindresultaat heeft.

In Bijlage 5: Berekeningen voor de benodigde lens van de cameravoorstellen is een uitgebreidere tabel geplaatst waarin de benodigde brandpuntsafstand van iedere camera berekend is. Hiervoor is voor iedere camera een extra parameter opgegeven, namelijk de MOD. (Minimum Object Distance). Dit is een specificatie van de lens en bepaalt de vergrotingsfactor. De MOD wordt bij aanvang op 300 mm geschat omdat dit ongeveer het gemiddelde is voor lenzen met een brandpuntsafstand van $f = 75$ mm.

Aan de hand van de camera's van Tabel 4 kunnen vervolgens lenzen gekozen worden. Enkele voorstellen voor lenzen worden gegeven in Tabel 5.

Tabel 5: Lensvoorstellen

Fabrikant	Model	f (mm)	Max sensor size	Min pixel size (µm)	MOD (mm)	Bevestiging
Kowa	LM35JC10M	35	2/3"	2,5	100	C-mount
Kowa	LM50JC10M	50	2/3"	2,5	100	C-mount
Qioptiq	MEVIS	12	2/3"	2	30	C-mount
Qioptiq	MAKRO CCD 1/4	50	2/3"	5,5	90	C-mount
Ricoh	FL-BC7528-9M	75	1"	3,69	250	C-mount
Schneider	KMP-IR XENOPLAN 35/1,9 - M30,5	35	2/3"	3,5	82	C-mount
Schneider	KMP-IR XENOPLAN 17/1,4 - M30,6	17	2/3"	3,5	42	C-mount
Schneider	KMP CINEGON 16/1,8 - M30,5	16	1"	3,5	20	C-mount
Schneider	KMP XENOPLAN 24/2,0	24	1,3"	2,5	76	C-mount
Tamron	M23FM50	50	2/3"	2,5	150	C-mount
Zeiss	DIMENSION 2,0/12	18	4/3"	2	91	C-mount

De voorgestelde lenzen uit Tabel 5 passen niet altijd op de camera's uit Tabel 4. Meestal ligt de oorzaak in de minimale pixel grootte van de lens. Tabel 6 toont daarom de mogelijke camera- en lensconfiguraties.

Tabel 6: Compatibiliteit tussen camera's en lenzen

	MV1-D20*8x1088-3D03-760-G2	MAKO U-5038	MAKO G-040B POE	MANTIA G-1588 POE	MANTIA G-3198 POE	GT2460/C	GUPPY PRO F503/B	STIN GRAY-II F-1258	STIN GRAY-II F-504B	GO-5101M/C-PGE	SP-12401M/C-PGE-C	GENIE NANO-M/C1450	GENIE NANO-C4900
LM35JC10M		x	x	x	x		x	x	x	x	x	x	
LM50JC10M	x		x	x	x		x	x	x	x	x	x	
MEVIS	x	x	x	x	x	x							
MAKRO CCD 1/4	x												
FL-BC7528-9M							x						
KMP-IR XENOPLAN 35/1,9 - M30,5	x		x	x	x	x		x	x	x	x	x	
KMP-IR XENOPLAN 17/1,4 - M30,6	x		x	x	x	x		x	x	x	x	x	
KMP CINEGON 16/1,8 - M30,5	x		x	x	x	x		x	x	x	x	x	
KMP XENOPLAN 24/2,0			x	x	x	x		x	x	x	x	x	
M23FM50			x	x	x	x		x	x	x	x	x	
DIMENSION 2,0/12		x					x						

6.4.2 Voorstel van consultant

Zoals eerder vermeld komen meerdere mogelijke oplossingen voort uit deze economische analyse. Daarom is de expertise en mening van de consultingsservice van Stemmer Imaging gevraagd voor een voorstel zodat hiermee vergeleken kan worden. Tabel 7 en Tabel 8 vermelden de voornaamste kenmerken voor de door Stemmer Imaging geselecteerde camera en lens.

Tabel 7: Specificaties camera Allied Vision Manta G-507B [13, p. 273]

Sensor model	Pregius IMX264
Sensor Type	CMOS
Resolution	5,1 Megapixel
Horizontal resolution	2464
Vertical resolution	2056
Pixel size	3,45 μm
Sensor format	2/3"
Price	€1 180,00

Tabel 8: Specificaties telecentrische lens KOWA LM1123TC | 0.69~0.88x [46]

Sensor size	2/3"
Pixel size	4,5 μm
Magnification	0,69
Mounting	C-mount
Iris Control	Manual
Lens Type	Telecentric
Wavelength Range	VIS (380nm - 780nm)
Focusing Range	111 mm
Focus Control	Manual
Working Distance	111 mm
Resolution, Center	120
Resolution, Corner	120
Distortion (TV)	-0.001 %
Size	Ø50 X 121,5mm
Weight	290g
Temperature Range	-10°C - +50°C
Price	€1 088

Opvallend in Tabel 8 is dat hier voor een telecentrische lens is gekozen. Telecentrische lenzen bieden het voordeel dat ze fouten ten gevolge van perspectiefprojecties en incorrecte beeldvershaling opvangen. Dit komt omdat de telecentrische lens de invallende lichtstralen parallel ten opzichte van elkaar uitlijnt. Over het volledige beeld is nu een homogene vergrotingsfactor en er zijn geen vertekeningen in het beeld.

7 Besluit

De hoofddoelstelling van deze masterproef was om de slijtage van snijgereedschap tot op 10 μm nauwkeurig te meten. Daarnaast moest de slijtage ook online gemeten kunnen worden door het visiesysteem zo te ontwerpen dat het implementeerbaar is binnen een CNC-freesmachine. Zo zou het mogelijk zijn om de slijtage online, te meten. Praktisch komt dit erop neer dat de slijtage van het snijgereedschap op de freeskop gemeten wordt tussen twee freesopdrachten in.

Het stadium van slijtage wordt benaderd door de slijtage op de flank te meten (VB_{Bmax}). Hiervoor is gekozen omdat dit zowel betrouwbaar als voorspelbaar in de tijd is. Hoofdstuk 2 heeft dit aangetoond aan de hand van een bronnenstudie.

In hoofdstuk 3 werd dit praktisch toegepast door een camera- en lens opstelling, voorzien van extra ledbars als belichtingsbronnen. Deze ledbars focuste licht op de flank van het snijgereedschap waardoor een sterk contrast ontstond tussen de slijtage en het onbeschadigde flankoppervlak. Door de aanwezigheid van omgevingslicht en enkele andere factoren, zoals vermeldt in het laatste deelhoofdstuk van hoofdstuk 3, zorgt de proefopstelling niet altijd voor goede beelden. Zo is het soms niet mogelijk om de slijtage te onderscheiden van de omgeving.

De goede beelden die uit de opstelling van hierboven kwamen werden vervolgens behandeld door een beeldverwerkingsalgoritme. Dit algoritme is geschreven in python en gebruikt als voornaamste library OpenCV. Hoofdzakelijk maakt het gebruik van verschillende maskeroperaties om het verkregen beeld te optimaliseren.

Het proces om uit het ingangsbeeld de slijtage te bepalen verloopt in vier stappen en is uitgelegd in hoofdstuk 4. Eerst wordt het beeld voorbehandeld zodat de ruis vermindert. Vervolgens wordt de positie en oriëntatie van het snijgereedschap in het beeld bepaald. Zo kan het beeld geroteerd worden opdat het altijd op eenzelfde manier geïnterpreteerd kan worden en bijgevolg het gebied waar de slijtage zich voordoet uitgeknipt worden. De stap nadien past een anisotropische diffusiefilter toe op het beeld waardoor het beeld vervaagt in de richting met kleine gradiënt. De rand is nu nog steeds goed zichtbaar waardoor de contour kan gezocht worden aan de hand van thresholding. Tot slot wordt hieruit de slijtage gemeten.

Het algoritme is getest op 42 snijplaatjes zoals hoofdstuk 5 beschrijft. Aan de hand van een microscoop als referentie blijkt dat slechts in 9 gevallen de slijtage tot kleiner dan 10 μm bepaald is kunnen worden. De gemiddelde afwijking ten opzichte van de microscoopbeelden bedraagt 28 μm indien drie uitschieters uit de dataset genegeerd worden. De oorzaak van de beperkte nauwkeurigheid is voornamelijk te wijten aan de gebruikte camera en lens. Bijgevolg kan dus toch gezegd worden dat het doel van de masterproef in zekere maten behaald is sinds de economische analyse een meer geschikte camera- en lensconfiguratie terug geeft.

De lens en belichting waren slechts in beperkte maten in staat om de grens van slijtage duidelijk in beeld te brengen. Daarnaast stelt 1 pixel in het beeld, 7 μm in de werkelijkheid voor. Hierdoor is weinig ruimte voor fouten voorzien. Uit de economische analyse van hoofdstuk 6 volgt dat

een lens met brandpuntsafstand van 35 mm en een MOD van 100 mm (of met een andere met gelijke prestaties) een beter resultaat zal leveren. Daarnaast heeft ook de pixelgrootte van de camera een belangrijke invloed op het eindresultaat. Door een camera te selecteren met kleinere pixels vermindert de afwijking van slijtage ten opzichte van de microscoopbeelden.

Indien verder gewerkt wordt op deze masterproef kan het interessant zijn om gebruik te maken van een meer geschikte camera en lens. Verder is de belichting in de proefopstelling ook slechts in beperkte mate geoptimaliseerd. Onderzoek doen naar een betere oriëntering van het visiesysteem t.o.v. het snijgereedschap kan ook helpen om de performantie van het algoritme te verhogen. Bijkomende kan een andere library dan OpenCV ook de mogelijkheid bieden om op subpixel niveau te werken. Hierdoor kunnen nog nauwkeurigere metingen van de slijtage gemaakt worden. Verder onderzoek kan ook proberen om de gereedschapsslijtage te bepalen op basis van machine learning en de invloeden van eventuele koelvloeistoffen onderzoeken. Deze masterproef geeft in de bronnenstudie al een korte inleiding van verschillende principes die hiervoor toegepast kunnen worden.

Literatuurlijst

- [1] S. Kurada en C. Bradley, „A review of machine vision sensors for tool condition monitoring,” *Computers in Industry*, nr. 34, pp. 55-72, 1996.
- [2] S. Dutta, P. S.K., M. S. en S. R., „Application of digital image processing in tool condition monitoring: a review,” *Manufacturing Science and Technology*, nr. 6, pp. 212-232, 2013.
- [3] W. Prashant, A. Nitin en S. Chinchankar, „A Review on Tool Wear Monitoring System,” *Journal of Mechanical Engineering and Automation*, Pune, India, 2016.
- [4] D. Vučina, D. Bajić, S. Jozić en I. Pehnc, „Evaluation of 3D tool wear in machining by successive stereo-photogrammetry and point cloud processing,” *Technical Gazette*, vol. 3, nr. 20, pp. 449-458, 2013.
- [5] L. Čerče, F. Pušavec en J. Kopač, „A New Approach to Spatial Tool Wear Analysis and Monitoring,” *Journal of Mechanical Engineering*, vol. 9, nr. 61, pp. 489-497, 2015.
- [6] D. Kerr, J. Pengilley en R. Garwood, „Assessment and visualisation of machine tool wear using computer vision,” Springer-Verslag, London, 2005.
- [7] T. Selvaraj, C. Balasubramani, S. Hari Vignesh en M. Prabakaran, „Tool Wear Monitoring By Image Processing,” *International Journal of Engineering Research & Technology (IJERT)*, nr. 2, p. Issue 8, 2013.
- [8] C. Zhang en J. Zhang, „On-line tool wear measurement for ball-end milling cutter based machine vision,” *Computers in Industry*, nr. 64, pp. 708-719, 2013.
- [9] T. Pfeifer en L. Wieggers, „Reliable tool wear monitoring by optimized image and illumination control in machine vision,” *Measurement*, nr. 28, pp. 209-218, 2000.
- [10] E. Claesen, „Machinevisie: praktische aspecten - cursus,” UHasselt/KU Leuven, Diepenbeek, 2019.
- [11] D. Eric, „Visie: beeldvorming - cursus,” UHasselt/KU Leuven, Diepenbeek, 2019.
- [12] J. Baeten, „Visietechnologie - cursus,” 12 Maart 2009. [Online]. Available: <https://perswww.kuleuven.be/~u0004696/>. [Geopend 20 November 2019].
- [13] Stemmer Imaging, *The Imaging and Vision Handbook*, Puchheim, Duitsland: Stemmer Imaging, 2018.
- [14] S. Abulhasan Quadri, „SlideShare,” 7 Mei 2012. [Online]. Available: <https://www.slideshare.net/reachquadri/what-is-spatial-resolution>. [Geopend 20 April 2020].
- [15] P. Perona en J. Malik, „Scale-Space and Edge Detection Using Anisotropic,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, nr. 12, pp. 629-639, 1990.
- [16] M. Sekhon, „Image Filters in Python,” towards data science, 10 Augustus 2019. [Online]. Available: <https://towardsdatascience.com/image-filters-in-python-26ee938e57d2>. [Geopend 6 Mei 2020].

- [17] R. E. W. Rafael C. Gonzalez, Digital image processing, New York: Pearson, 2018.
- [18] OpenCV, „Sobel(),” OpenCV, [Online]. Available: https://docs.opencv.org/master/d4/d86/group__imgproc__filter.html#gacea54f142e81b6758cb6f375ce782c8d. [Geopend 6 Mei 2020].
- [19] Tutorials point, „Prewitt Operator,” Tutorials point, [Online]. Available: https://www.tutorialspoint.com/dip/prewitt_operator.htm. [Geopend 6 Mei 2020].
- [20] Z. Dewei, Y. Lei, W. Xiaoyu, W. Nanxi en L. Hui, „An Improved Roberts Edge Detection Algorithm Based on Mean Filter and Wavelet Denoising,” in *Advances in information technology and industry applications*, Beijing, China , Springer, 2012, pp. 299-305.
- [21] S. P. A. W. a. E. W. R. Fisher, „Roberts Cross Edge Detector,” Image processing learning resources, 2003. [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/roberts.htm>. [Geopend 6 Mei 2020].
- [22] Tutorials point, „Krisch Compass Mask,” Tutorials point, [Online]. Available: https://www.tutorialspoint.com/dip/krisch_compass_mask.htm. [Geopend 6 Mei 2020].
- [23] G. S. Robinson, „Edge Detection by Compass Gradient Masks,” *Computer Graphics and Image Processing*, vol. 1977, nr. 6, pp. 192-501, 1977.
- [24] A. C. K. Juan Andrade-Cetto, „Object Recognition,” *John Wiley & Sons*, vol. 2000, nr. 1, pp. 449-470, 2000.
- [25] D. Rákos, „Frei-Chen edge detector,” Raster Grid, 30 Januari 2011. [Online]. Available: <http://rastergrid.com/blog/2011/01/frei-chen-edge-detector/>. [Geopend 6 Mei 2020].
- [26] P. Corke, Robotics, Vision and Control, Brisbane, Australia: Springer, 2013.
- [27] OpenCV, „Morphological Transformations,” OpenCV, [Online]. Available: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html. [Geopend 6 Mei 2020].
- [28] OpenCV, „Basic Thresholding Operations,” OpenCV, [Online]. Available: <https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>. [Geopend 6 Mei 2020].
- [29] OpenCV, „Image Thresholding,” OpenCV, [Online]. Available: https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html. [Geopend 5 Mei 2020].
- [30] OpenCV, „adaptiveThreshold(),” OpenCV, [Online]. Available: https://docs.opencv.org/3.4/d7/d1b/group__imgproc__misc.html#ga72b913f352e4a1b1b397736707afcd3. [Geopend 6 Mei 2020].
- [31] K. A. A. Mordvintsev, „Histograms - 1 : Find, Plot, Analyze !!!,” OpenCV-Python Tutorials, [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_begins/py_histogram_begins.html. [Geopend 6 Mei 2020].

- [32] OpenCV, „Hough Line Transform,” OpenCV, [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html. [Geopend 6 Mei 2020].
- [33] OpenCV, „Contours : Getting Started,” OpenCV, [Online]. Available: https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html. [Geopend 6 Mei 2020].
- [34] A. Rosebrock, „OpenCV center of contour,” pyimagesearch, 1 Februari 2016. [Online]. Available: <https://www.pyimagesearch.com/2016/02/01/opencv-center-of-contour/>. [Geopend 6 Mei 2020].
- [35] B. Ekaba, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Ottawa, Canada: Apress, 2019.
- [36] S. Manohar, *Mastering Machine Learning with Python in Six Steps*, Bangalore, Karnataka, India: Apress, 2019.
- [37] W. Weis, „Processing of Optical Sensor Data for Tool Monitoring with Neural Networks,” Institute of Machine Tools and Production Science, University of Karlsruhe, Karlsruhe, Duitsland.
- [38] P. Ong, W. K. Lee en R. Jit Hoo Lau, „Tool condition monitoring in CNC end milling using wavelet neural network based on machine vision,” *The International Journal of Advanced Manufacturing Technology*, nr. 104, pp. 1369-1379, 2019.
- [39] A. Gajate, R. Haber, R. del Toro, P. Vega en A. Bustillo, „Tool wear monitoring using neuro-fuzzy techniques: a comparative study in a turning process,” *J Intell Manuf*, nr. 23, pp. 869-882, 2012.
- [40] A. A. D. Sarhan, „Adaptive neuro-fuzzy approach to predict tool wear accurately in turning operations for maximum cutting tool utilization,” *IFAC-PapersOnLine*, nr. 48, pp. 93-98, 2015.
- [41] N. N. Bhat, S. Dutta, S. K. Pal en S. Pal, „Tool condition classification in turning process using hidden Markov model based on texture analysis of machined surface images,” *Measurement*, nr. 90, pp. 500-509, 2016.
- [42] L. Wang, M. G. Mehrabi en E. Kannatey-Asibu Jr., „Hidden Markov Model-based Tool Wear Monitoring in Turning,” *Journal of Manufacturing Science and Engineering*, nr. 124, pp. 651-658, 2002.
- [43] Sandvik, „Datasheet R390-11 T3 08M-PM 1030,” [Online]. Available: <https://www.sandvik.coromant.com/en-gb/products/pages/productdetails.aspx?c=R390-11%20T3%2008M-PM%20%20201030>. [Geopend 13 April 2020].
- [44] Photonfocus, „Datasheet MV1-D2048x1088-3D03-760-G2,” [Online]. Available: <https://www.photonfocus.com/>. [Geopend 6 November 2019].
- [45] Best Scientific, „OPTEM MVZL 18-108mm Macro Video Zoom Lens,” [Online]. Available: <https://www.bestscientific.co.uk/product/optem-macro-video-zoom-lens-34-11-10/>. [Geopend 20 April 2020].

- [46] KOWA, „LM1123TC | 0.69~0.88x Telecentric Lens,” [Online]. Available: <https://www.kowa-lenses.com/en/lm1123tc-5mp-industrial-lens-c-mount>. [Geopend 2 Mei 2020].
- [47] P. Bandyopadhyay, D. A. Blaser, B. D. Kaminski en Y. H. Lee, „Tool wear measurement by machine vision”. Verenigde Staten Patent 4,845,763, 4 Juli 1989.
- [48] J. Jurkovic, M. Korosec en J. Kopac, „New approach in tool wear measuring technique using CCD vision system,” *International Journal of Machine Tools & Manufacture*, nr. 45, pp. 1023-1030, 2005.

Bijlagen

Bijlage 1: Python code van het algoritme

Pythonfile 1: MP_Sven_Sander.py

```
import cv2 as cv
from matplotlib import pyplot as plt
import pandas as pd
# Self-made modules
import LoadData
import FindToolTip
import CalculateToolWear

# OPTIONS
show = True
pyplot = True
showMeasurement = True
saveCrop = False
saveComparison = False

# Load data from csv-file
# 0 Tool wear black
# 1 Tool wear white
# 2 All data
data = LoadData.load(0)

# Filepath to folders with all images
filePathCamera = 'C:\\Users\\Sander\\PycharmProjects\\MP_Sven_Sander\\Camera'
filePathMicro = 'C:\\Users\\Sander\\PycharmProjects\\MP_Sven_Sander\\Microscoop'

array = []

for i in range(len(data)):
    # Get data for this iteration
    titleMicro = data[i][1]
    toolWearMicro = data[i][2]
    titleCamera = data[i][3]

    # Load images
    img = cv.imread(filePathCamera + '\\ ' + titleCamera + '.png', cv.IMREAD_GRAYSCALE)
    imgMicro = cv.imread(filePathMicro + '\\ ' + titleMicro + '-img.png', cv.IMREAD_COLOR)

    # Current image
    print(titleCamera)

    # Find tooltip
    imgRotated, imgRotatedRGB, originPoint = FindToolTip.findTool(img)
    imgToolTip = FindToolTip.cropTool(imgRotated, originPoint)
    if saveCrop: cv.imwrite("Crop.png", imgToolTip)

    # Calculate toolwear
    toolWearCamera, imgToolTip, imgToolTipMeasurement = CalculateToolWear.toolWear(imgToolTip)
    array.append([toolWearMicro, toolWearCamera])

# Show img one by one
if show:
    nameCamera = str(titleCamera) + " - Toolwear: " + str(toolWearCamera)
    nameMicro = str(titleMicro) + " - Toolwear: " + str(toolWearMicro)
    if pyplot:
        # Show with pyplot
        plt.subplot(1, 2, 1)
        if showMeasurement:
            plt.imshow(cv.cvtColor(imgToolTipMeasurement, cv.COLOR_BGR2RGB))
        else:
            plt.imshow(cv.cvtColor(imgToolTip, cv.COLOR_GRAY2RGB))
        plt.title(nameCamera)
        plt.xticks([], plt.yticks([]))

        plt.subplot(1, 2, 2)
        plt.imshow(imgMicro, 'gray')
```



```

plt.title(nameMicro)
plt.xticks([], plt.yticks([]))
if saveComparison: plt.savefig("output\\" + str(titleCamera) + " vergelijking.png")
plt.show()
# Press close button of pyplot window for next img
else:
    # Show with openCV
    if showMeasurement:
        cv.imshow(nameCamera, imgToolTipMeasurement)
    else:
        cv.imshow(nameCamera, imgToolTip)
        cv.imshow(nameMicro, imgMicro)
        cv.waitKey(0)
        cv.destroyAllWindows()
        # Press esc for next img

pd.DataFrame(array).to_csv("CSV\\resultaten.csv", sep=";", header=["Micro", "Camera"])

```

Pythonfile 2: FindToolTip.py

```

import cv2 as cv
import numpy as np
import sys
import copy
import AnisotropicDiffusion
# Self-made modules
import MathFunctions as f

# OPTIONS
showCompact = False # Shows results (& scaled down)
showAll = False # Shows all steps
drawAllLines = False # Draws all found lines
save = False # Save all steps to png

# Finds Tool orientation & origin point
# Rotates image so the base of the tool is horizontal
# Returns rotated image (in gray), rotated image with lines to find origin (in RGB), origin point (on rotated image)
def findTool(img):
    # Converting img to RGB so that colored lines can be drawn on img
    imgRGB = cv.cvtColor(img, cv.COLOR_GRAY2RGB)
    #if showAll: cv.imshow("0 - original", img)

    # 0 - Downsizing so edges of the tool are more defined & reduces noise
    imgDown = cv.pyrDown(img)
    if showAll: cv.imshow("0 - original scalded down", imgDown)
    if save: cv.imwrite("0 - original scalded down.png", imgDown)

    # 1 - Gaussian blur to smooth the image with big kernel
    imgBlur = AnisotropicDiffusion.anisoDiff(img, niter=15, kappa=70, gamma=0.2, option=2)
    if save: cv.imwrite("1 - aniso diff.png", imgBlur)

    imgBlur = cv.GaussianBlur(imgDown, (15, 15), 0)
    if showAll: cv.imshow("1 - Gaussian blur", imgBlur)
    if save: cv.imwrite("1 - Gaussian blur.png", imgBlur)

    # 2 - Find edges by using laplacian
    lap = cv.Laplacian(imgBlur, ddepth=-1, ksize=7, scale=0.15)
    if showAll: cv.imshow("2 - Laplacian", lap)
    if save: cv.imwrite("2 - Laplacian.png", lap)

    sobelX = cv.Sobel(imgBlur, cv.CV_64F, 1, 0)
    sobelY = cv.Sobel(imgBlur, cv.CV_64F, 0, 1)
    sobelX = np.uint8(np.absolute(sobelX))
    sobelY = np.uint8(np.absolute(sobelY))
    sobelCombined = cv.addWeighted(sobelX, 0.5, sobelY, 0.5, 0)
    if save: cv.imwrite("2 - Sobel.png", sobelCombined)

```

```

sigma = 0.45
v = np.median(imgBlur)
# ---- apply automatic Canny edge detection using the computed median----
lower = int(max(0, (1.0 - sigma) * v)) # ---- lower threshold
upper = int(min(255, (1.0 + sigma) * v)) # ---- upper threshold
#edged = cv.Canny(imgBlur, lower, upper)
edged = cv.Canny(imgBlur, 23, 27)
if save: cv.imwrite("2 - Canny.png", edged)

# 3 - Adaptive thresholding because images are not evenly lit which makes the Laplacian inconsistent
# Very big kernel with gaussian thresholding to get the real edges instead of the noise
th = cv.adaptiveThreshold(lap, 255, cv.ADAPTIVE_THRESH_GAUSSIAN_C, cv.THRESH_BINARY, 501, 0)
if showAll: cv.imshow("3 - Thresholding", th)
if save: cv.imwrite("3 - Thresholding.png", th)

# 4 - Morphological Transformation: opening
# (Erosion leaves to many gaps)
kernel = np.ones((5, 5), np.uint8)
morph = cv.morphologyEx(th, cv.MORPH_OPEN, kernel)
morph = cv.morphologyEx(th, cv.MORPH_ERODE, kernel, iterations=1)
if showAll: cv.imshow("4 - Morphological Transformation: opening", morph)
if save: cv.imwrite("4 - Morphological Transformation opening.png", morph)

# 5 - Probabilistic Hough Transform
height, width = imgDown.shape[:2]
lines = cv.HoughLinesP(morph, 1, np.pi / 720, 180, minLineLength=200, maxLineGap=12)
# Exit program if there are no lines
if lines is None:
    print("ERROR: Tool base not found")
    sys.exit()

# Find tool base
# (Assuming that tool base is longest line, on the bottom half of the picture)
lineMaxDis = 0
toolBaseLine = [0, 0, 0, 0] #x1, y1, x2, y2
imgRGBLines = copy.copy(imgRGB)
for line in lines:
    x1, y1, x2, y2 = line[0]
    if f.calcDistanceLine(line[0]) > lineMaxDis:
        if y1 >= height / 2 and y2 >= height / 2:
            lineMaxDis = f.calcDistanceLine(line[0])
            toolBaseLine = line[0]
    if drawAllLines: cv.line(imgRGBLines, (x1*2, y1*2), (x2*2, y2*2), (0, 0, 255), 1)
    #if drawAllLines: cv.line(imgRGBLines, (x1, y1), (x2, y2), (0, 0, 255), 1)
if showAll: cv.line(imgRGBLines, (toolBaseLine[0]*2, toolBaseLine[1]*2), (toolBaseLine[2]*2, toolBaseLine[3]*2), (255, 0, 0), 2)
2)
#if showAll: cv.line(imgRGBLines, (toolBaseLine[0], toolBaseLine[1]),(toolBaseLine[2], toolBaseLine[3]), (255, 0, 0), 2)
angleToolBase = f.calcAngle(np.multiply(toolBaseLine, 2))

# Find flank
lineMinDis = width
flankLine = [width, height, width, height]
for line in lines:
    x1, y1, x2, y2 = line[0]
    angle = f.calcAngle(np.multiply(line[0], 2))
    # Flank has a specific angle
    if 65 < (angle - angleToolBase) < 71:
        intercept = f.lineIntersect(line[0], [0, height/2, width, height/2])

```

```

dis = f.calcDistance(0, height/2, intercept[0], intercept[1])
if dis < lineMinDis:
    flankLine = line[0]
    lineMinDis = dis
if drawAllLines: cv.line(imgRGBLines, (x1 * 2, y1 * 2), (x2 * 2, y2 * 2), (0, 255, 0), 1)
    #if drawAllLines: cv.line(imgRGBLines, (x1, y1), (x2, y2), (0, 255, 0), 1)
if showAll: cv.line(imgRGBLines, (flankLine[0] * 2, flankLine[1] * 2), (flankLine[2] * 2, flankLine[3] * 2), (255, 0, 0), 2)
    #if showAll: cv.line(imgRGBLines, (flankLine[0], flankLine[1]), (flankLine[2], flankLine[3]),(255, 0, 0), 2)

if showAll: cv.imshow("5 - Probabilistic Hough Transform", imgRGBLines)
    #if showCompact and not showAll: cv.imshow("5 - Probabilistic Hough Transform", cv.pyrDown(imgRGBLines))
if save: cv.imwrite("5 - Probabilistic Hough Transform.png", imgRGBLines)

# Rotate image
height, width = imgRGB.shape[:2] #shape of full image
rotationMatrix = cv.getRotationMatrix2D((width / 2, height / 2), angleToolBase, 1)
imgRotated = cv.warpAffine(img, rotationMatrix, (width, height))
imgRotatedRGB = cv.warpAffine(imgRGB, rotationMatrix, (width, height))

# Rotate lines
toolBaseLine = f.rotateLine((width / 2, height / 2), toolBaseLine*2, angleToolBase)
flankLine = f.rotateLine((width / 2, height / 2), flankLine*2, angleToolBase)
#toolBaseLine = f.rotateLine((width / 2, height / 2), toolBaseLine, angleToolBase)
#flankLine = f.rotateLine((width / 2, height / 2), flankLine, angleToolBase)
originPoint = f.lineIntersect(toolBaseLine, flankLine)
cv.line(imgRotatedRGB, (originPoint[0], originPoint[1]), (toolBaseLine[2], toolBaseLine[3]), (255, 0, 0), 2)
cv.line(imgRotatedRGB, (flankLine[0], flankLine[1]), (originPoint[0], originPoint[1]), (255, 0, 0), 2)
cv.circle(imgRotatedRGB, originPoint, 3, (255,0,0), thickness=3)
if showAll: cv.imshow('6 - Rotated image + origin point of tool', imgRotatedRGB)
if showCompact and not showAll: cv.imshow('6 - Rotated image + origin point of tool', cv.pyrDown(imgRotatedRGB))
if save: cv.imwrite('6 - Rotated image + origin point of tool.png', imgRotatedRGB)

if showAll or showCompact: cv.waitKey(0)
if showAll or showCompact: cv.destroyAllWindows()
cv.waitKey(0)
cv.destroyAllWindows()
# Returns rotated image (in gray), rotated image with lines to find origin (in RGB), origin point (on rotated image)
return imgRotated, imgRotatedRGB, originPoint

# Crop out tooltip based on the assumption that this is always around the same place relative to origin point
def cropTool(img, originPoint):
    height, width = img.shape[:2]

    # Relative place of tooltip & size of tool wear (can be tweaked)
    moveX = 210
    moveY = 530
    widthCrop = 350
    heightCrop = 120

    # Making sure all coordinates are inside image
    y1 = max(originPoint[1] - moveY, 0)
    y2 = min(y1 + heightCrop, height)
    x1 = max(originPoint[0] - moveX, 0)
    x2 = min(x1 + widthCrop, width)

    if save:
        imgRGB = cv.cvtColor(img, cv.COLOR_GRAY2BGR)
        cv.rectangle(imgRGB, (x1,y1), (x2,y2), (255,0,0), 2)
        cv.imwrite("crop.png", imgRGB)
    # Return crop
    return img[y1:y2, x1:x2]

```

Pythonfile 3: CalculateToolWear.py

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
import copy
import AnisotropicDiffusion
# Self-made modules
import MathFunctions as f

# OPTIONS
show = False
save = False

def toolWear(img):
    # Converting img to RGB so that colored lines can be drawn on img
    imgRGB = cv.cvtColor(img, cv.COLOR_GRAY2RGB)

    # Anisotropic diffusion to blur image and keep edges
    aniso = AnisotropicDiffusion.anisoDiff(img, niter=20, kappa=50, gamma=0.1, option=2)
    #Crimmins smoothing geeft gelijkaardig resultaat maar is veel trager
    #aniso = CrimminsSmoothing.crimmins(img)

    # Laplacian with big kernel to minimize the effect of noise
    lap2 = cv.Laplacian(aniso, ddepth=-1, ksize=7, scale=0.15)

    # Calculate the histogram of the toolTip
    histogram = plt.hist(aniso.ravel(), 256, [0, 256])

    # The threshold value is the first peak (between 5 and 90) + 10
    threshVal = max(histogram[0][5:90].argmax() + 5 + 10, 80)

    # Thresholding with value from histogram
    _, thresColour = cv.threshold(aniso, threshVal, 255, cv.THRESH_BINARY_INV)

    # Morphological operations
    kernel = np.ones((3, 3), dtype='uint8')
    thresColour2 = cv.morphologyEx(thresColour, cv.MORPH_OPEN, kernel, iterations=2)
    kernel = np.ones((5, 5), dtype='uint8')
    thresColour2 = cv.morphologyEx(thresColour2, cv.MORPH_DILATE, kernel, iterations=1)

    # Lay masks on images for visual conformation
    maskedFrame = cv.bitwise_and(aniso, aniso, mask=thresColour)
    maskedFrame2 = cv.bitwise_and(aniso, thresColour2)

    # Find contours & Draw ALL contours
    imgRGBcontour = copy.copy(imgRGB)
    imgRGBlijnen = copy.copy(imgRGB)
    contours, hierarchy = cv.findContours(thresColour2, cv.RETR_TREE, cv.CHAIN_APPROX_NONE)
    #print("Number of contours = " + str(len(contours)))
    cv.drawContours(imgRGBcontour, contours, -1, (255, 0, 0), 1)

    #init van contourWear
    contourWear = contours[0]

    # Find contour
    # When there are multiple contours, the center of mass has to be in a certain zone. In this zone the most complex
    # contour gets selected
    if len(contours) == 1:
        contourWear = contours[0]
    else:
        lenMax = 0
        for contour in contours:
            M = cv.moments(contour)
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])
            if 50 <= cX <= 310 and 16 <= cY <= 75:
                if len(contour) >= lenMax:
```

```

        contourWear = contour
        lenMax = len(contour)
# Draw contour that represents tool wear
if len(contourWear) > 0:
    cv.drawContours(imgRGBcontour, contourWear, -1, (255, 255, 255), 1)
    cv.drawContours(imgRGB, contourWear, -1, (255, 255, 255), 1)

# Find lowest point of tool wear
array = []
for point in contourWear:
    array.append(point[0])
array = np.array(array, dtype='int32')
index = np.argmax(array[:, 1])
lowestPoint = [array[index][0], array[index][1]]
cv.circle(imgRGBcontour, (lowestPoint[0], lowestPoint[1]), 3, (255, 255, 255), -1)
cv.circle(imgRGB, (lowestPoint[0], lowestPoint[1]), 3, (255, 255, 255), -1)

# Find lines with with Hough lines
lines = cv.HoughLines(lap2, 1, np.pi / 720, 275)

# Filter lines
bMin = 300
for line in lines:
    rho, theta = line[0]
    if 92 * (np.pi / 180) < theta < 93 * (np.pi / 180):
        a = np.cos(theta)
        b = np.sin(theta)
        x0 = a * rho
        y0 = b * rho
        x1 = int(x0 + 100 * (-b))
        y1 = int(y0 + 100 * (a))
        x2 = int(x0 - 400 * (-b))
        y2 = int(y0 - 400 * (a))
        line = [x1, y1, x2, y2]
        slope = f.slope(line)
        b = line[1] - slope * line[0]
        if 50 < b <= bMin:
            lineCut = line
            bMin = b
            cv.line(imgRGBlijnen, (x1, y1), (x2, y2), (0, 0, 255), 1)

# place line on cutting edge
height, width = img.shape[:2]
cv.line(imgRGBlijnen, (lineCut[0], lineCut[1]), (lineCut[2], lineCut[3]), (255, 0, 0), 1)
slopeCut = f.slope(lineCut)
bCut = (lineCut[1] - slopeCut * lineCut[0]) - 70
x1 = int(0)
y1 = int(x1 * slopeCut + bCut)
x2 = int(width)
y2 = int(x2 * slopeCut + bCut)
cv.line(imgRGB, (x1,y1), (x2,y2), (0,255,0),1)

# measure distance from lowest point to cutting edge
yMeasure = int(lowestPoint[0] * slopeCut + bCut)
cv.line(imgRGB, (lowestPoint[0], lowestPoint[1]), (lowestPoint[0], yMeasure), (255,0,0),1)
wear = lowestPoint[1] - yMeasure
wear2 = wear * 7

if save:
    cv.imwrite("origineel.png", img)
    cv.imwrite("mask after morphological operations.png", maskedFrame2)
    cv.imwrite("contour.png", imgRGBcontour)
    cv.imwrite("lap.png", lap2)
    cv.imwrite("lijnen.png", imgRGBlijnen)
    cv.imwrite("Tool wear.png", imgRGB)

if show:

```

```

cv.imshow("origineel", img)
cv.imshow("mask after morphological operations", maskedFrame2)
cv.imshow("contour", imgRGBcontour)
cv.imshow("lap", lap2)
cv.imshow("lines", imgRGBlijnen)
cv.imshow("Tool wear", imgRGB)
cv.waitKey(0)
cv.destroyAllWindows()

return wear2, img, imgRGB

```

Pythonfile 4: MathFunctions.py

```

import math

# Conventions for this module
# Point = [x1, y1]
# line = [x1, y1, x2, y2]
# angels in degrees

# Calculates distance based on pythagoras's rule
def calcDistance(x1, y1, x2, y2):
    return math.hypot(x2 - x1, y2 - y1)

# Calculates distance based on pythagoras's rule but with a line as input
def calcDistanceLine(line):
    return math.hypot(line[2] - line[0], line[3] - line[1])

# Calculates angle of a line relative to horizon (in degree)
def calcAngle(line):
    return math.atan((line[1] - line[3]) / (line[0] - line[2])) * (180 / math.pi)

# Rotates point around origin with angle (in degree)
def rotatePoint(origin, point, angle):
    xOrigin, yOrigin = origin
    xPoint, yPoint = point
    angel2 = - angle * (math.pi / 180)
    xRotated = int(round(xOrigin + math.cos(angel2) * (xPoint - xOrigin) - math.sin(angel2) * (yPoint - yOrigin)))
    yRotated = int(round(yOrigin + math.sin(angel2) * (xPoint - xOrigin) + math.cos(angel2) * (yPoint - yOrigin)))
    return xRotated, yRotated

# Rotates the 2 points of line around origin with angle (in degree)
def rotateLine(origin, line, angle):
    x1Rotated, y1Rotated = rotatePoint(origin, (line[0], line[1]), angle)
    x2Rotated, y2Rotated = rotatePoint(origin, (line[2], line[3]), angle)
    return x1Rotated, y1Rotated, x2Rotated, y2Rotated

# Calculates slope of a line
def slope(line):
    return (line[3] - line[1]) / (line[2] - line[0])

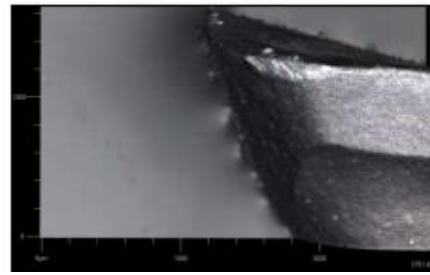
# Calculates intercept point of 2 lines (lines can NOT be vertical or parallel)
def lineIntersect(line1, line2):
    # b = y - mx
    r1 = slope(line1)
    b1 = line1[1] - r1 * line1[0]
    r2 = slope(line2)
    b2 = line2[1] - r2 * line2[0]

    if r1 == r2:
        return None
    # y = mx + b
    x = (b2 - b1) / (r1 - r2)
    y = r1 * x + b1
    return round(x), round(y)

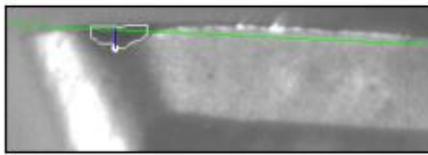
```

Bijlage 2: Vergelijking output van het algoritme met resultaten van de microscoop

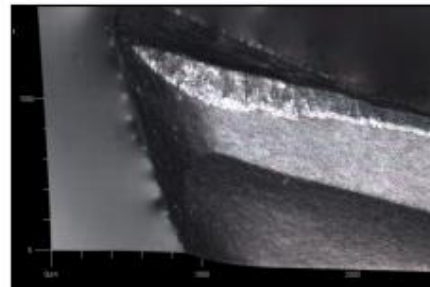
40 - Toolwear: 127



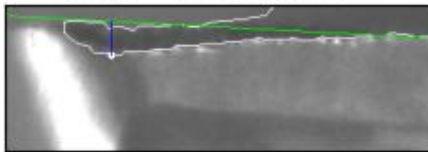
40-2 - Toolwear: 105



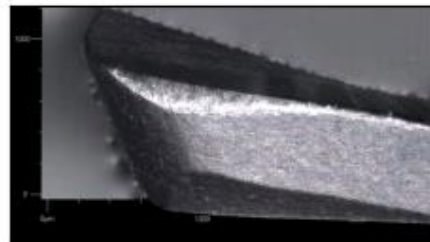
40b - Toolwear: 242



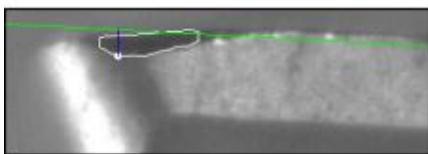
40x1 - Toolwear: 196



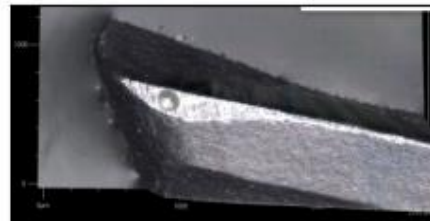
31 - Toolwear: 171



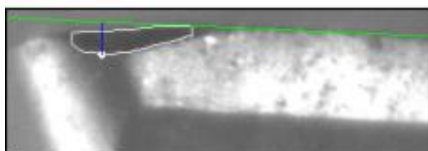
41-1 - Toolwear: 147



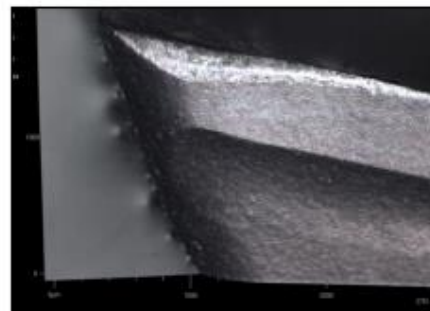
31b - Toolwear: 195



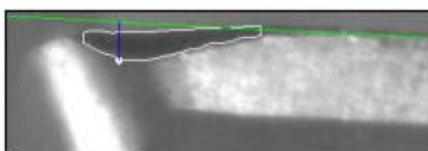
41x1 - Toolwear: 182



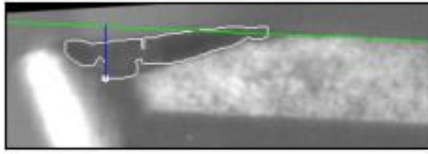
32 - Toolwear: 237



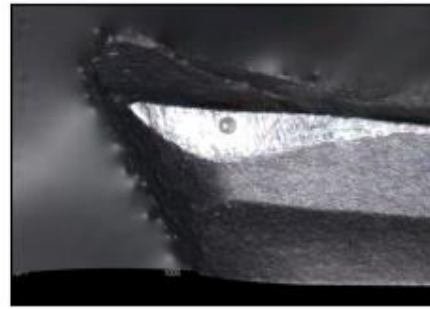
42-1 - Toolwear: 217



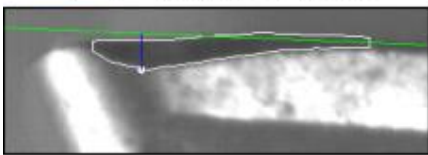
42x1 - Toolwear: 301



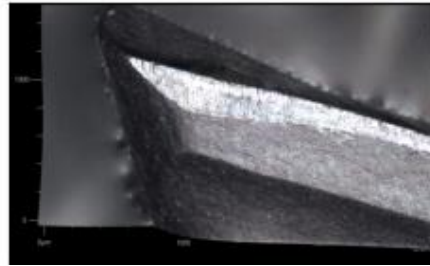
32b - Toolwear: 279



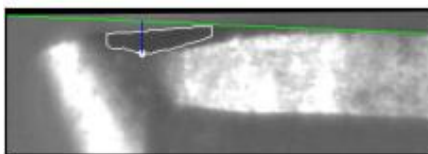
43-1 - Toolwear: 203



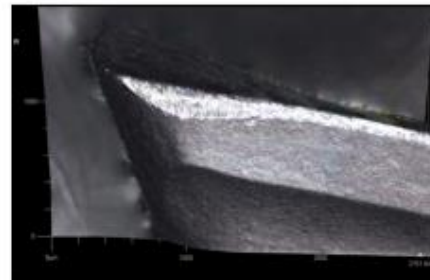
33 - Toolwear: 234



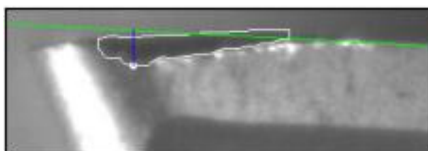
43x1 - Toolwear: 182



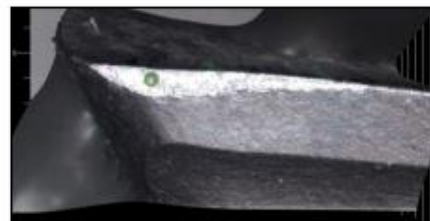
33b - Toolwear: 213



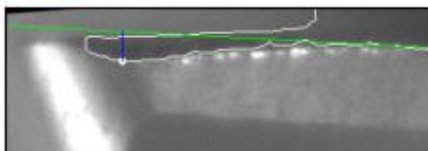
47-1 - Toolwear: 203



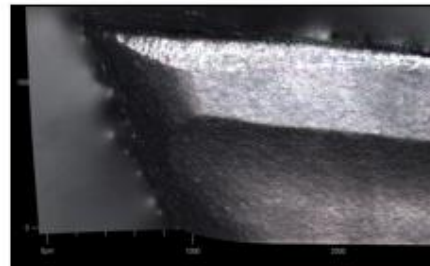
37 - Toolwear: 181



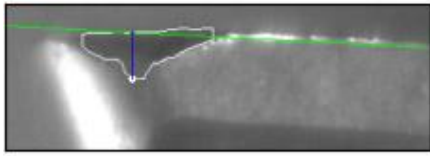
47x1 - Toolwear: 168



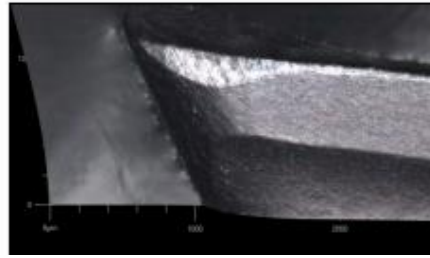
37b - Toolwear: 192



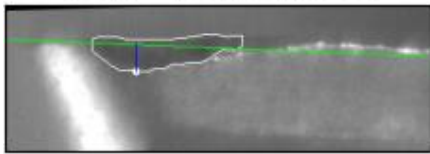
49-1 - Toolwear: 266



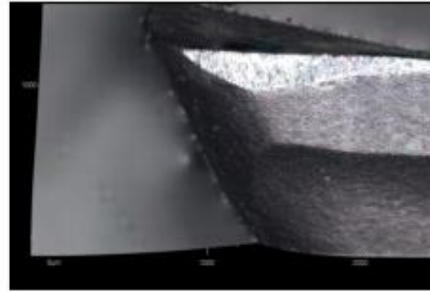
39 - Toolwear: 212



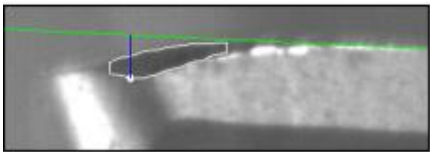
49x1 - Toolwear: 154



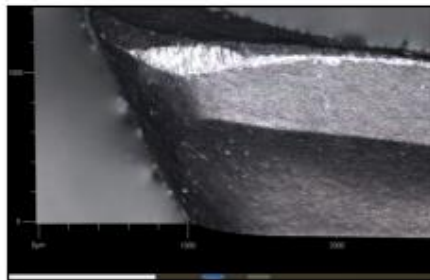
39b - Toolwear: 219



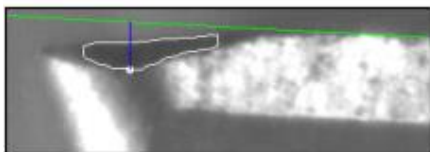
55-1 - Toolwear: 245



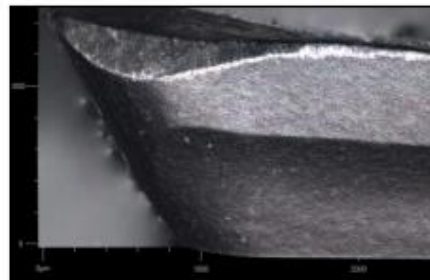
45 - Toolwear: 250



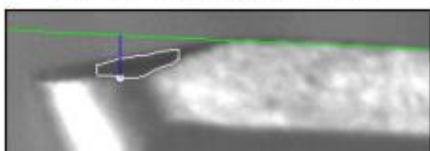
55x1 - Toolwear: 273



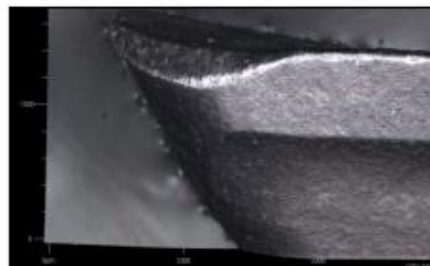
45b - Toolwear: 311



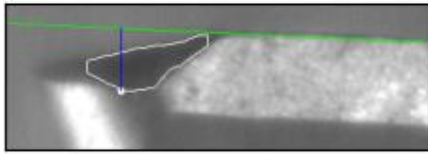
56-1 - Toolwear: 245



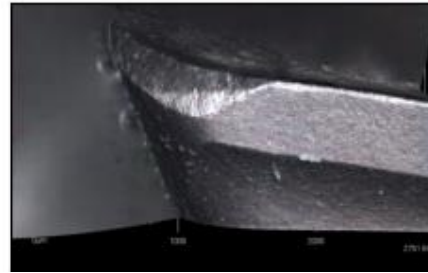
46 - Toolwear: 244



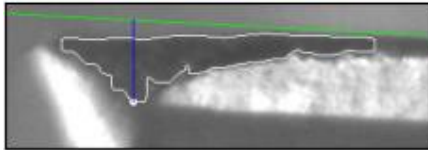
56x1 - Toolwear: 357



46b - Toolwear: 335



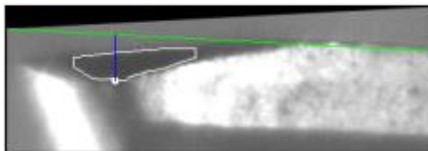
57-1 - Toolwear: 469



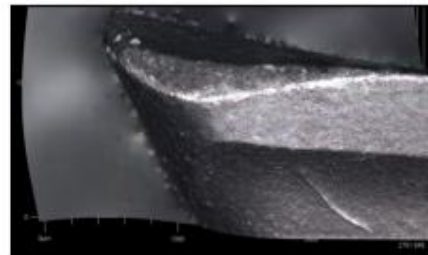
47 - Toolwear: 395



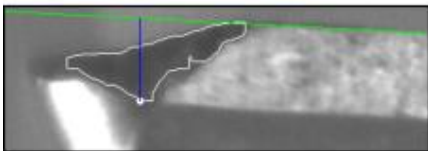
57x1 - Toolwear: 259



47b - Toolwear: 270



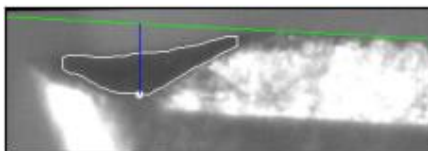
58-1 - Toolwear: 469



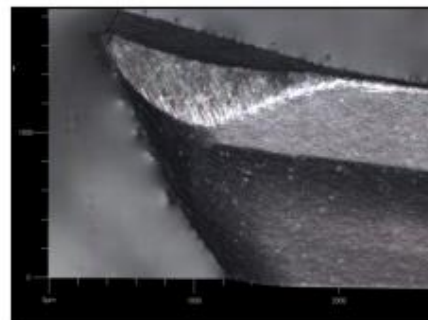
48 - Toolwear: 467



58x1 - Toolwear: 399



48b - Toolwear: 399





MV1-D2048x1088-3D03-760-G2

The 3D CMOS camera MV1-D2048x1088-3D03-760-G2 was developed for Laser triangulation systems with high triangulation rates

Features

- Detection of up to 2 laser lines (peak detector)
- 2D single line for 2D surface inspection and image overlay
- CMOSIS CMV2000 CMOS image sensor
- 2048 x 1088 pixel resolution
- Up to 10200fps @ 2048x23 pixels
- Global shutter
- Extended sensor and camera features
- A/B shaft encoder interface
- GigEVision interface
- Free GUI available (PF 3D Suite)



Quantum Efficiency Image Sensor

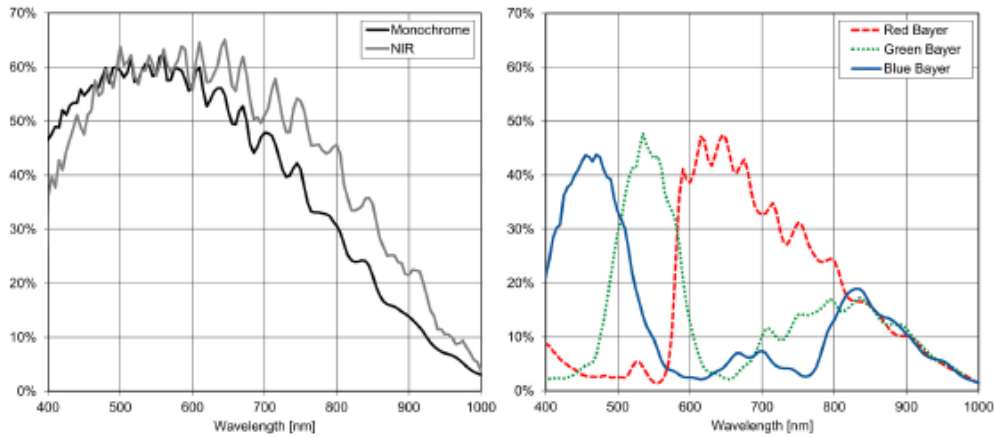


Image Sensor Specifications

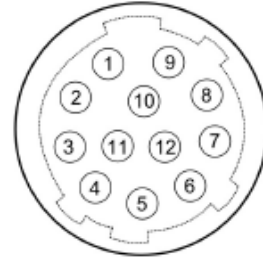
Manufacturer / Type	CMOSIS / CMV2000
Technology	CMOS
Optical format	2/3"
Optical diagonal	12.76mm
Resolution	2048 x 1088
Pixel size	5.5µm x 5.5µm
Active optical area	11.26mm x 5.98mm
Dark current	125e ⁻ /s
Read out noise	13e ⁻
Full well capacity / SNR	11ke ⁻ / 105: 1
Spectral range	Monochrome: < 350 to 950nm (to 10% of peak responsivity)
Responsivity	Monochrome: 1100 x 10 ³ DN / (J/m ²) @ 520nm / 8bit
Quantum Efficiency	Monochrome: > 60%
Optical fill factor	42% without micro lenses
Dynamic range	60dB
Characteristic curve	Linear, Piecewise linear
Shutter mode	Global shutter

Camera Specifications

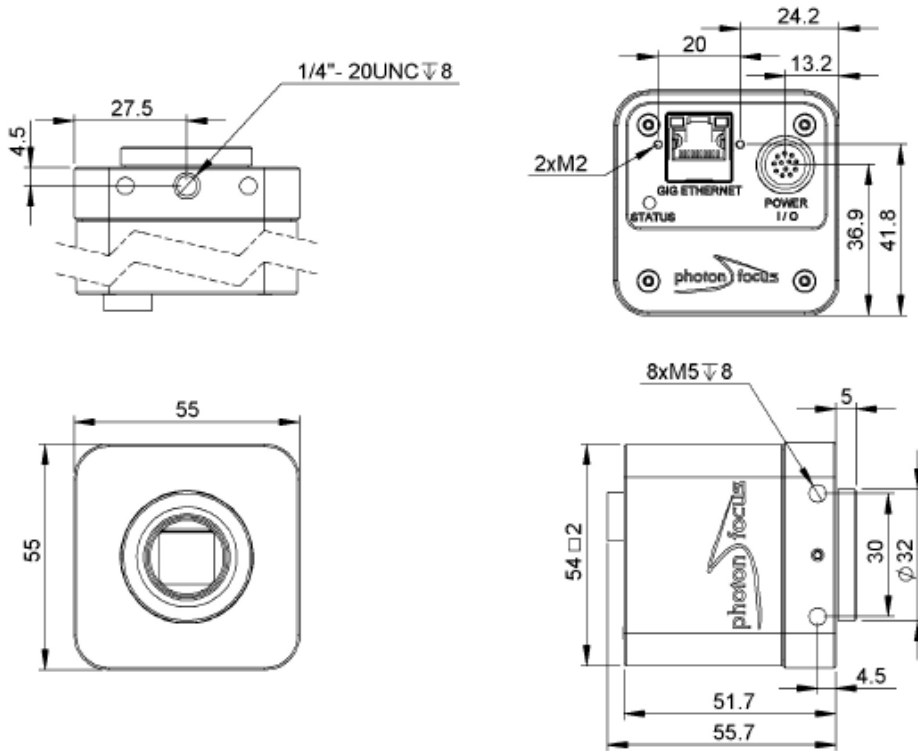
Interface	GigE
Frame rate	10200fps
Pixel clock	48MHz
Camera taps	2
Greyscale resolution	8Bit
Fixed pattern noise (FPN)	< 1DN RMS @ 8Bit
Exposure time range	13µs - 349ms
Analog gain	yes
Digital gain	0.1 to 15.99 (FineGain)
Trigger Modes	Free running (non triggered), external Trigger, SWTrigger, AB-Trigger
Features	Detection of up to 2 laser lines (peak detector), 2D single line for 2D surface inspection and image overlay, Linear Mode / multiple slope (High Dynamic Range), Configurable region of interest (ROI), Temperature monitoring of camera, Ultra low trigger delay and low trigger jitter, Extended trigger input and strobe output functionality, Isolated inputs (2 single ended, 2 differential) and outputs (2 single ended), A/B shaft encoder interface (RS-422 (G2 models) or HTL (H2 models)), Free GUI available (PF 3D Suite) for an easy system set up and visualisation of 3D scans
Operation temperature / moisture	0°C ... + 40°C / 20% ... 80%
Storage temperature / moisture	-25°C ... 60°C / 20% ... 95%
Power supply	+12VDC (-10%) ... +24VDC (+10%)
Power consumption	< 6W
Lens mount	C-Mount (CS-Mount optional)
I/O Inputs	2x Opto-isolated 2x RS-422 or HTL Opto-isolated for AB-Trigger
I/O Outputs	2x Opto-isolated
Dimensions	55 x 55 x 52mm ³
Mass	265g
Connector I/O (Power)	Hirose 12-pole (mating plug HR10A-10P-12S)
Connector Interface	RJ-45
Conformity	CE / RoHS / WEEE
IP Code	IP40

Connectors

Pin	I/O Type	Name	Description
1	PWR	CAMERA_GND	Camera GND 0V
2	PWR	CAMERA_PWR	Camera Power 12V... 24V
3	O	ISO_OUT0	Default Strobe out, internally Pulled up to ISO_PWR with 4k7 Resistor
4	I	ISO_INC0_N	INC0 differential input (G2: RS-422, H2: HTL), negative polarity
5	I	ISO_INC0_P	INC0 differential input (G2: RS-422, H2: HTL), positive polarity
6	PWR	ISO_PWR	Power supply 5V... 24V for output signals
7	I	ISO_IN0	IN0 input signal
8	O	ISO_OUT1 (MISC)	Q1 output from PLC, no Pull up to ISO_PWR; can be used as additional output (by adding Pull up) or as controllable switch (max. 100mA, no capacitive or inductive load)
9	I	ISO_IN1(Trigger IN)	Default Trigger IN
10	I	ISO_INC1_N	INC1 differential input (G2: RS-422, H2: HTL), negative polarity
11	I	ISO_INC1_P	INC1 differential input (G2: RS-422, H2: HTL), positive polarity
12	PWR	ISO_GND	I/O GND 0V



Dimensions



Explanation

DN	DigitalNumber (equals to LSB)
e ⁻	Electrons

Order Information

MV1-D2048x1088-3D03-760-G2-8	RS-422 Encoder Interface
MV1-D2048x1088-3D03-760-H2-8	HTL Encoder Interface
MV1-D2048x1088-3D03-760-G2-8-S10	RS-422, Scheimpflug Adapter
MV1-D2048x1088-3D03-760-H2-8-S10	HTL, Scheimpflug Adapter

Compatibility



Photonfocus AG
Bahnhofplatz 10
CH-8853 Lachen SZ
Switzerland

Phone: +41 55 451 00 00
www.photonfocus.com
info@photonfocus.com

Bijlage 4: Datasheet van de gebruikte lens

Optem[®] MVZL

18-108mm MACRO VIDEO ZOOM LENS



Specifications

Working Distance
 "close-up" lens 5.5 - 11"
 w/o "close-up" lens 11" - ∞

Focal Length 18 - 108mm

Aperture F/2.5 - Closed

Magnification w/close-up lens
 @Max. WD: 0.066 - 0.4X
 @Min. WD: 0.17 - 1.0X

Zoom Ratio 6:1 (parfocal)

Focus External, Manual w/ locking screw

Zoom Control External, Manual ring

Video Mount C-Mount

Camera Format 2/3" or smaller

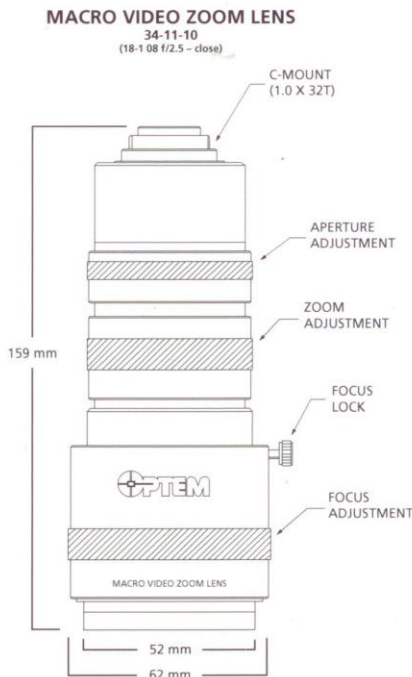
Weight 1 lb., 3 oz.

Length 159mm

Diameter 62mm maximum

Features/Advantages

- Exceptional Image clarity and color accuracy
- Robust engineering for durability & stability
- Parfocality maintains crisp focus throughout the entire 6:1 optical zoom range.
- Longer Working Distances - View a variety of specimens from 5.5 inches out through infinity.
- Magnification - With "close-up" lens = 1.0X - 0.066X. Without "close-up" lens = 0.47X - 0.084X
- Optimized for C-mount video cameras. 2/3" format or smaller recommended.
- Ideal for monitoring & Surveillance, machine vision, video inspection, in-line inspection, packaging inspection



FIELDS-OF-VIEW (FOV)

WORKING DISTANCE.	FOV (mm) @ LOW MAG.	FOV (mm) @ HIGH MAG.
CLOSE-UP LENS		
5.5" (13.97cm)	38 x 51 mm	6 x 8 mm
7" (17.78cm)	60 x 80 mm	10 x 14 mm
8" (20.32cm)	72 x 96 mm	12 x 16 mm
9" (22.86cm)	83 x 110 mm	14 x 19 mm
10" (25.40cm)	90 x 120 mm	16 x 22 mm
11" (27.94cm)	100 x 133 mm	17 x 23 mm
WITHOUT CLOSE-UP LENS		
11" (27.94cm)	79 x 105 mm	14 x 19 mm
24" (60.96cm)	198 x 264 mm	33 x 44 mm
36" (91.44cm)	311 x 415 mm	55 x 73 mm
48" (121.9cm)	415 x 553 mm	70 x 93 mm

Based upon a 2/3" camera format. For 1/2" camera format, multiply F.O.V. x 0.72. For 1/3" camera format multiply F.O.V.s x 0.54.

bestscientific.co.uk

For more information, contact Best Scientific
 t: +44 (0)1793 812477 | e: enq@bestscientificweb.com

Bijlage 5: Berekeningen voor de benodigde lens van de cameravoorstellen

Het paginanummer bij iedere camera verwijst naar de productcatalogus van Stemmer Imaging [13].

MAKO U-503B			
Cameraspecificaties			
Pixelgrootte	2,2 µm	pg 272	
Sensor formaat	1/2,5 inch		
Breedte sensor	5,7 mm		
Resolutie	5 Megapixel		
Horizontale resolutie	2592 pixels		
Verticale resolutie	1944 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,220	0,57
Overeenk afst pixel	2,2	10,00	3,86
Benodigde res HOR	4545	1000	2591
Benodigde res VER	1632	359	930
MOD	300	300	300
F	150	54,10	108,92

MAKO G-040B POE			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 272	
Sensor formaat	1/2,9 inch		
Breedte sensor	4,98 mm		
Resolutie	1,6 Megapixel		
Horizontale resolutie	1456 pixels		
Verticale resolutie	1088 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,50
Overeenk afst pixel	3,45	10,00	6,93
Benodigde res HOR	2899	1000	1443
Benodigde res VER	1041	359	518
MOD	300	300	300
F	150	76,95	99,73

MANTA G-158B POE			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 273	
Sensor formaat	1/2,9 inch		
Breedte sensor	4,98 mm		
Resolutie	1,6 Megapixel		
Horizontale resolutie	1456 pixels		
Verticale resolutie	1088 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,50
Overeenk afst pixel	3,45	10,00	6,93
Benodigde res HOR	2899	1000	1443
Benodigde res VER	1041	359	518
MOD	300	300	300
F	150	76,95	99,73

MANTA G-319B POE			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 273	
Sensor formaat	1/1,8 inch		
Breedte sensor	7,11 mm		
Resolutie	3 Megapixel		
Horizontale resolutie	2064 pixels		
Verticale resolutie	1544 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,71
Overeenk afst pixel	3,45	10,00	4,85
Benodigde res HOR	2899	1000	2061
Benodigde res VER	1041	359	740
MOD	300	300	300
F	150	76,95	124,66

GT2460/C			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 275	
Sensor formaat	2/3 inch		
Breedte sensor	8,6 mm		
Resolutie	5,1 Megapixel		
Horizontale resolutie	2464 pixels		
Verticale resolutie	2056 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,86
Overeenk afst pixel	3,45	10,00	4,01
Benodigde res HOR	2899	1000	2493
Benodigde res VER	1041	359	895
MOD	300	300	300
F	150	76,95	138,71

GUPPY PRO F503/B			
Cameraspecificaties			
Pixelgrootte	2,2 µm	pg 277	
Sensor formaat	1/2,5 inch		
Breedte sensor	5,7 mm		
Resolutie	5 Megapixel		
Horizontale resolutie	2588 pixels		
Verticale resolutie	1940 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,220	0,57
Overeenk afst pixel	2,2	10,00	3,86
Benodigde res HOR	4545	1000	2591
Benodigde res VER	1632	359	930
MOD	300	300	300
F	150	54,10	108,92

STINGRAY-II F-125B			
Cameraspecificaties			
Pixelgrootte	3,75 µm	pg 278	
Sensor formaat	1/3 inch		
Breedte sensor	4,8 mm		
Resolutie	1,2 Megapixel		
Horizontale resolutie	1292 pixels		
Verticale resolutie	964 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,375	0,48
Overeenk afst pixel	3,75	10,00	7,81
Benodigde res HOR	2667	1000	1280
Benodigde res VER	957	359	460
MOD	300	300	300
F	150	81,82	97,30

STINGRAY-II F-504B			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 278	
Sensor formaat	2/3 inch		
Breedte sensor	8,6 mm		
Resolutie	5 Megapixel		
Horizontale resolutie	2452 pixels		
Verticale resolutie	2056 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,86
Overeenk afst pixel	3,45	10,00	4,01
Benodigde res HOR	2899	1000	2493
Benodigde res VER	1041	359	895
MOD	300	300	300
F	150	76,95	138,71

GO-5101M/C-PGE			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 288	
Sensor formaat	2/3 inch		
Breedte sensor	8,6 mm		
Resolutie	5 Megapixel		
Horizontale resolutie	2454 pixels		
Verticale resolutie	2056 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,86
Overeenk afst pixel	3,45	10,00	4,01
Benodigde res HOR	2899	1000	2493
Benodigde res VER	1041	359	895
MOD	300	300	300
F	150	76,95	138,71

SP-12401M/C-PGE-C			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 289	
Sensor formaat	1,1 inch		
Breedte sensor	12,52 mm		
Resolutie	12,3 Megapixel		
Horizontale resolutie	4088 pixels		
Verticale resolutie	3000 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	1,25
Overeenk afst pixel	3,45	10,00	2,76
Benodigde res HOR	2899	1000	3629
Benodigde res VER	1041	359	1303
MOD	300	300	300
F	150	76,95	166,79

GENIE NANO-M/C1450			
Cameraspecificaties			
Pixelgrootte	3,45 µm	pg 308	
Sensor formaat	1/2,9 inch		
Breedte sensor	4,96 mm		
Resolutie	1,6 Megapixel		
Horizontale resolutie	1456 pixels		
Verticale resolutie	1088 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,345	0,50
Overeenk afst pixel	3,45	10,00	6,96
Benodigde res HOR	2899	1000	1438
Benodigde res VER	1041	359	516
MOD	300	300	300
F	150	76,95	99,47

GENIE NANO-C4900			
Cameraspecificaties			
Pixelgrootte	1,25 µm	pg 308	
Sensor formaat	1/2,3 inch		
Breedte sensor	6,16 mm		
Resolutie	18,1 Megapixel		
Horizontale resolutie	4912 pixels		
Verticale resolutie	3684 pixels		
MOD	300 mm		
Berekening benodigde brandpuntsafstand			
Selectiecriteria	Methode 1: Zonder vergroting	Methode 2: Kleinste lens	Methode 3: Maximale vergroting
Vergrotingsfactor m	1	0,125	0,62
Overeenk afst pixel	1,25	10,00	2,03
Benodigde res HOR	8000	1000	4928
Benodigde res VER	2872	359	1769
MOD	300	300	300
F	150	33,33	114,36

