NHL STENDEN computer vision & data science

Van De Loosdrecht Machine Vision

Kinect for Windows in VisionLab

Johan van Althuis
Martin Dijkstra
Bart van Apeldoorn

20 January 2017

j.van.de.loosdrecht@nhl.nl, jaap@vdlmv.nl

Kinect for Windows in VisionLab

Overview

- **Kinect for Windows, the sensor**
- **How to use the Kinect with VisionLab**
- **Kinect parameters**
- **Examples**
- **Operators for Kinect**
- **Demonstration**

Kinect for Windows, the sensor

Specifications

- ~90 latency with processing
- Resolutions
  - Color
    - 30 FPS: 640 x 480 RGB camera
    - 12 FPS: 1280 x 960 RGB camera
  - Depth
    - 30 FPS: 640 x 480 'Depth' camera
- 57° horizontal field of view
- 43° vertical field of view

Kinect for windows, the sensor – Kinect2

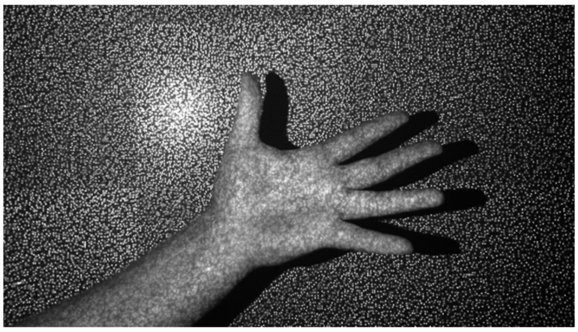
Specifications

- ~60 ms latency with processing
- Resolutions
  - Color
    - 30 FPS: 1920 x 1080 x 16bpp YUY2 Camera
  - Depth
    - 512 x 424 x 16bpp, 13-bit depth
  - IR
    - 512 x 424, 11 bit
- 70° horizontal field of view
- 60° vertical field of view

Kinect for Windows, the sensor

How does the depth camera work?

Infrared dots are projected into the scene by a laser. All objects in scene are reflecting a constellation of dots. How these dots are transformed into a depth measurement is unknown.



27-aug-18 Using Microsoft Kinect within VisionLab 5

Kinect for Windows, the sensor – Kinect2

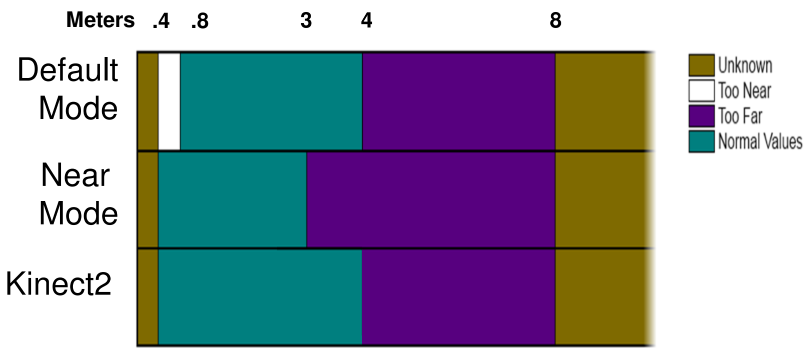
The Kinect2 uses a different depth technique instead.

It uses Time of Flight to create a depth image of the scene. An IR pulse is emitted and reflected by objects in the room. Then it's measured by the delay in the reflected light.

27-aug-18 Using Microsoft Kinect within VisionLab 6

Kinect for Windows, the sensor

Near Mode is a function in Kinect for Windows that enables the depth camera to see objects as close as 40 centimeters in front of the sensor while maintaining accuracy and precision.



27-aug-18 Using Microsoft Kinect within VisionLab 7

How to use the Kinect with VisionLab

The Kinect interface for VisionLab is ONLY to be used with the Microsoft Kinect for Windows sensor.

You are ONLY allowed to use this interface if you agree with the license terms of Microsoft Kinect for Windows Software Development Kit (SDK).

These license terms are described in Microsoft's SDKEula.rtf. The document is distributed with your copy of VisionLab and can also be found on the website: www.microsoft.com/en-us/kinectforwindows

Here you have to download and install the Kinect for Windows Runtime environment which is required for using the Kinect.

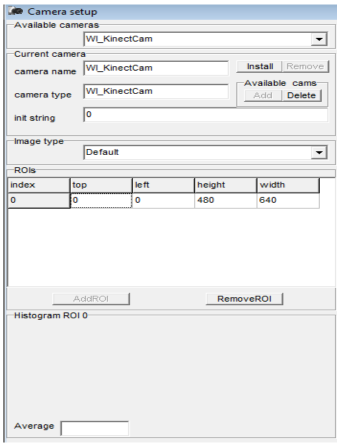
27-aug-18 Using Microsoft Kinect within VisionLab 8

How to use the Kinect with VisionLab

When drivers and software for the Kinect are correct installed, the Kinect can be used within VisionLab.

At the Camera setup (F12) select:

- **WI_KinectCam**
- **The Init string is needed to select which kinect to use when there are multiple Kinects connected. When only one Kinect is connected, the default init string can be used.**



27-aug-18 Using Microsoft Kinect within VisionLab 9

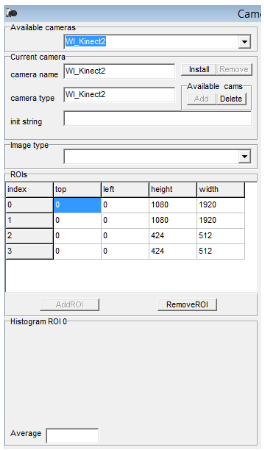
How to use the Kinect2 with VisionLab

When drivers and software for the Kinect2 are correct installed, the Kinect2 can be used within Vision Lab.

At the Camera setup (F12) select:

- **WI_Kinect2**
- **And press Install**

Note: No warning is shown when there is no Kinect camera connected.



27-aug-18 Using Microsoft Kinect within VisionLab 10

General Kinect camera interface info

The camera interface can generate different images.
To specify which image to generate, the ROI index number is used:

	Kinect	Kinect 2
ROI 0:	**WI_Snapshot_RGB**	**WI_Snapshot_RGB**
	(unless specified otherwise by ROI0MapMode)	
ROI 1:	**WI_Snapshot_Depth**	**WI_Snapshot_Depth**
ROI 2:	**WI_Snapshot_Users**	**WI_Snapshot_IR**
ROI 3:	**WI_Snapshot_Skeletons**	**WI_Snapshot_Skeletons**
ROI 4:	**WI_Snapshot_IR**	

Kinect camera parameters within VisionLab

- **ImageOutput <…>**
  Specify which image generators need to be set active.
  The Kinect camera can deliver up to 5 different image types, these are: RGB, Depth, Users, Skeleton and IR.
  For performance reasons it's possible to select only the needed image output. Simultaneously generating of an RGB & IR image is not possible, therefore this combination cannot be selected as ImageOutput.

- **MirrorMode <WI_NoMirror WI_MirrorX WI_MirrorY WI_MirrorXY>**
  Sets the global mirrormode for the image output.

Kinect camera parameters within VisionLab

- **NearMode <WI_Near_Disabled WI_Near_Enabled>**
  Sets the near mode used for the depth image output. The default sensor range has a minimum of 800mm and a maximum of 4000mm. Near mode handles a minimum of 400mm and maximum of 3000mm.

- **OverlayMode <WI_Overlay_Disabled WI_Overlay_Enabled>**
  Green channel of the RGB image output is replaced by the depth image.

- **ROI0MapMode <WI_Map_RGB WI_Map_Depth WI_Map_Users WI_Map_Skeleton WI_Map_IR>**
  Map ROI 0 to a different image output. Used for debugging purposes of Continious ROI 0. Default to: RGB.

27-aug-18 Using Microsoft Kinect within VisionLab 13

Kinect camera parameters within VisionLab

- **RgbResolution <WI_Resolution_640x480 WI_Resolution_1280x960>**
  Sets the resolution of the RGB image. The ROI height and widht in the VisionLab interface needs to be adjusted to the chosen resolution.

- **Angle<-27 <> 27>**
  Angle of the Kinect camera in degrees which can be set between -27 and 27 degrees.

- **SeatedMode<WI_Seated_Disabled WI_Seated_Enabled>**
  Enables or disables the seated mode for the skeleton image. The default mode tracks twenty skeletal joints, reporting their position as tracked or inferred. The seated mode only tracks the ten upper-body joints it reports the lower-body joints as not tracked.

27-aug-18 Using Microsoft Kinect within VisionLab 14

Kinect2 camera parameters within VisionLab

- ROI0MapMode <Image_Color,Image_Depth,Image_IR,Image_Body>
  Map ROI 0 to a different image output. Used for debugging purposes of Continious ROI 0. Default to: RGB.
- ColorFormat <RGB888,YUV888,HSV888>
  Sets the color format of the color Snapshot.
  HSV888 is converted from the RGB888 image.
  the YUV888 is the raw color format from the Kinect.
- ColorProjectMode <Off,DepthNearestNeighbour,DepthBilinear>
  If not set to ‘Off’ the color frame is mapped to depth space. With NearestNeighbour the color frame is mapped using NearestNeighbour interpolation, with Bilinear it is mapped using Bilinear interpolation. The resulting image will be the size of the depth frame.

Kinect2 camera parameters within VisionLab

- DepthProjectMode <Off,DepthNearestNeighbour,DepthBilinear>
  If not set to ‘Off’ the depth frame is mapped to color space. With NearestNeighbour the color frame is mapped using NearestNeighbour interpolation, with Bilinear it is mapped using Bilinear interpolation. The resulting image will be the size of the color frame.

Depth data

- **Returns the <u>distance and player</u> for every pixel**
  - **Ex: 640x480 = 307.200 pixels**
- **Distance**
  - **Distance in mm from Kinect**
- **Player**
  - **Kinect: Recognize 6 players track 2 players**
  - **Kinect2: more than the Kinect**

Examples (Depth images)

Normal color image

Depth image

Jaap van de Loosdrecht, NHL, vdLMV,
j.van.de.loosdrecht@tech.nhl.nl

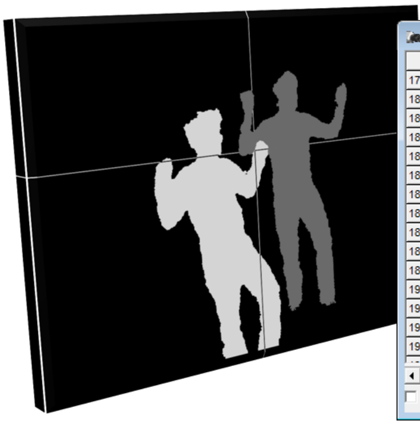
Examples (Depth images)

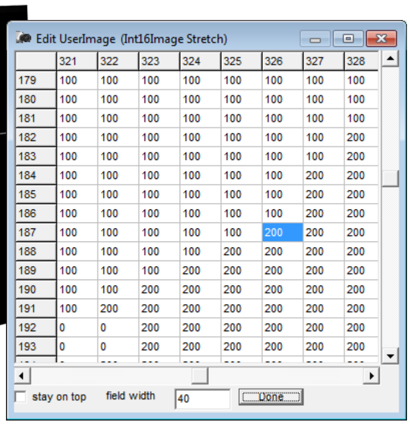
Near mode disabled (too near)

Near mode enabled

User image

- **Users correspond with skeleton image**
- **Userpixels are multiplied (value = userID * 100)**



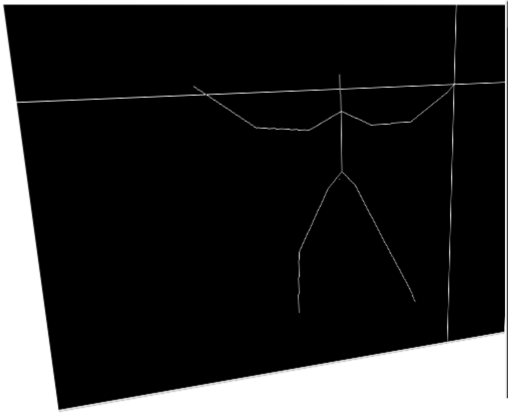


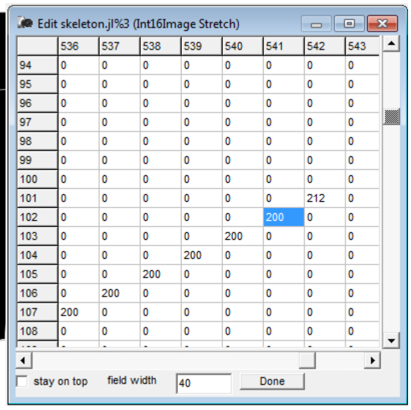
	321	322	323	324	325	326	327	328
179	100	100	100	100	100	100	100	100
180	100	100	100	100	100	100	100	100
181	100	100	100	100	100	100	100	100
182	100	100	100	100	100	100	100	200
183	100	100	100	100	100	100	100	200
184	100	100	100	100	100	100	200	200
185	100	100	100	100	100	100	200	200
186	100	100	100	100	100	100	200	200
187	100	100	100	100	100	200	200	200
188	100	100	100	100	200	200	200	200
189	100	100	100	200	200	200	200	200
190	100	100	200	200	200	200	200	200
191	100	200	200	200	200	200	200	200
192	0	0	200	200	200	200	200	200
193	0	0	200	200	200	200	200	200

Jaap van de Loosdrecht, NHL, vdLMV,
j.van.de.loosdrecht@tech.nhl.nl

Skeleton image

- Skeletons correspond to the user image
- Every joint has it's unique value (userID * 100 + jointID + 1)





Edit skeleton.jl%3 (Int16Image Stretch)

	536	537	538	539	540	541	542	543
94	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0
101	0	0	0	0	0	0	212	0
102	0	0	0	0	0	200	0	0
103	0	0	0	0	200	0	0	0
104	0	0	0	200	0	0	0	0
105	0	0	200	0	0	0	0	0
106	0	200	0	0	0	0	0	0
107	200	0	0	0	0	0	0	0
108	0	0	0	0	0	0	0	0

stay on top field width 40 Done

27-aug-18 Using Microsoft Kinect within VisionLab 21

Operators for the Kinect camera interface

- WI_ConvertProjectiveToWorld(CameraName, x, y, z)
  Convert a point of the depth image to real world coordinates.
  The x and y coordinates are defined by the pixel position in depth image, where z is the depth value.
  Returns offset from optical axis of the Kinect in millimetres.

- WI_ConvertWorldToProjective(CameraName, x, y, z)
  Convert a world coordinate back to the projective depth image plane. The x and y coordinates are defined by the offset from the optical axis of the kinect in millimetres, where z is the distance in millimetres.
  Returns position of pixel in depth image plane.

27-aug-18 Using Microsoft Kinect within VisionLab 22

Operators for the Kinect camera interface

- **WI_GetNrOfUsers(CameraName)**
  Returns the number of users.
- **WI_GetUsersInfo(CameraName)**
  Operator returns a vector with the center of mass for each user based on the user ID.
  Returns id1 (x1,y1,z1) id2 (x2,y2,z2) … idn (xn,yn,zn)
- **WI_GetUsersInfoArray(CameraName, &$tab)**
  Equal to WI_GetUsersInfo but result is stored in an array
  Returns array length.

We need a skeleton snapshot to successfully update user information.

27-aug-18 Using Microsoft Kinect within VisionLab 23

Operators for the Kinect camera interface

- **WI_GetSkeletonJoint (CameraName, UserID, SkeletonJoint)**
  Get the coordinates of a skeleton joint of the user identified by the UserID. Returns the confidence of the given skeleton joint followed by a vector which holds the coordinates for this joint. Possible skeleton joint values are:

27-aug-18 Using Microsoft Kinect within VisionLab 24

Skeleton Joint Values

Torso

Hip center	1
Spine	2
Shoulder center	3
Head	4
Center	21

Left		**Right**	
Shoulder	5	Shoulder	9
Elbow	6	Elbow	10
Wrist	7	Wrist	11
Hand	8	Hand	12
Hip	13	Hip	17
Knee	14	Knee	18
Ankle	15	Ankle	19
Foot	16	Foot	20

Kinect2 Skeleton Joint Values

Torso

Spinebase	1
SpineMid	2
Neck	3
Head	4
SpineShoulder	21

	Left	Right
Shoulder	5	9
Elbow	6	10
Wrist	7	11
Hand	8	12

	Left	Right
Hip	13	17
Knee	14	18
Ankle	15	19
Foot	16	20
Handtip	22	24
Thumb	23	25

Operators for the Kinect camera interface

- **WI_GetSkeletonJointMulti (CameraName, UserID, SkeletonJoint, [Multiple selected skeleton joints])**

Get the coordinates of multiple selected skeleton joints of the user identified by the UserID.
Returns the confidence of each given skeleton joint followed by the vector which hold the coordinates for this joint.

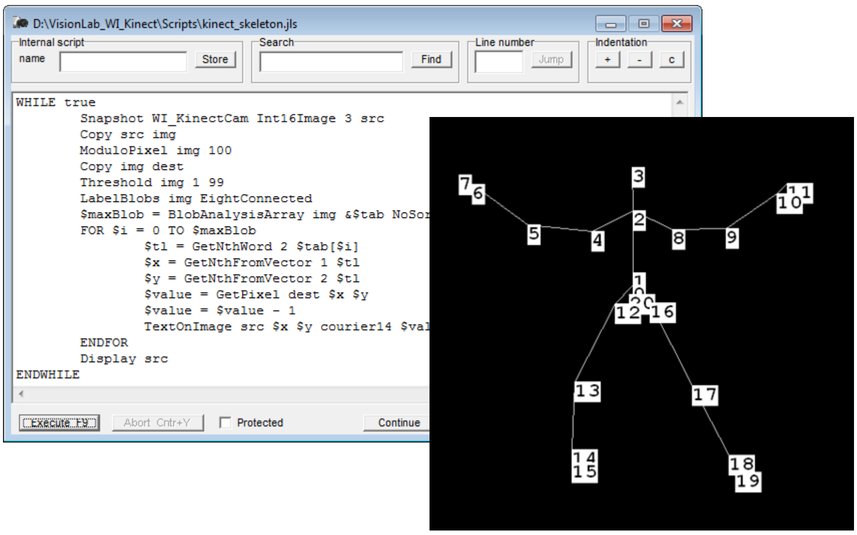
The confidence of a skeleton joint appear in three levels:

- **0 Tracking of skeleton joint is lost;**
- **0.5 Skeleton joint is occluded/in shade of other object;**
- **1 Tracking of skeleton joint is successful.**

In practise a confidence of 0.5 never appears, therefore a confidence of 1(one) is only reliable.

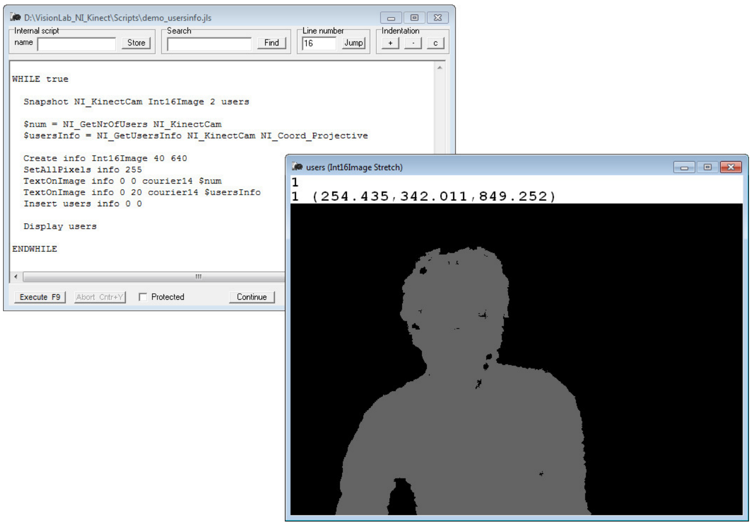
27-aug-18 Using Microsoft Kinect within VisionLab 27

Examples (kinect_skeleton.jls)



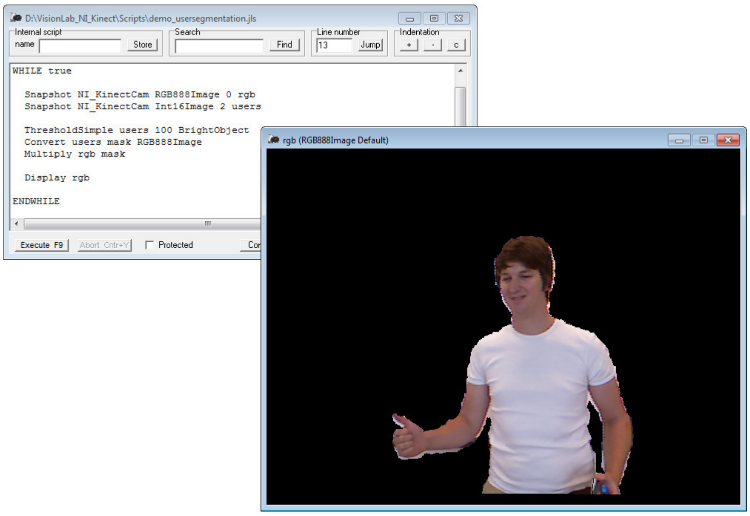
27-aug-18 Using Microsoft Kinect within VisionLab 28

Examples (kinect_userinfo.jls)



27-aug-18 Using Microsoft Kinect within VisionLab 29

Examples (kinect_usersegmentation.jls)



27-aug-18 Using Microsoft Kinect within VisionLab 30

Exercise: Hawaii

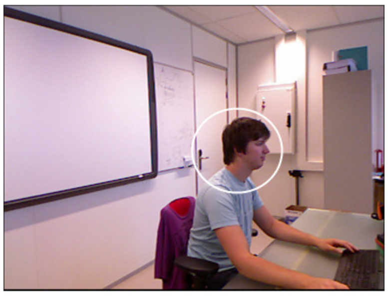
- **Use the depth image from the Kinect as 'green'-screen**
- **Try to smooth the edges!**

Answer: kinect_hawaii.jls

27-aug-18 Using Microsoft Kinect within VisionLab 31

Exercise: Face detection – part A

Draw a circle around the head of each user that's infront of the kinect

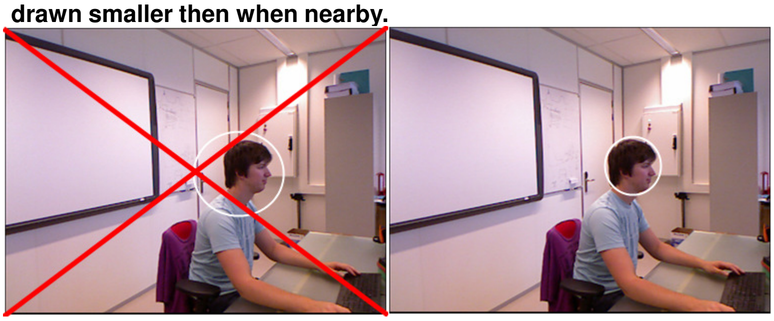


Hint: Use the skeleton image to find the head position of each user

27-aug-18 Using Microsoft Kinect within VisionLab 32

Exercise: Face detection – part B

Similar to A: but scale the circle to the size of the head.
If the user is farther away from the kinect the circle has to be drawn smaller then when nearby.



Hint: Use the depth image to calculate how far away the user is.

27-aug-18 Using Microsoft Kinect within VisionLab 33

Exercise: Face detection – part C

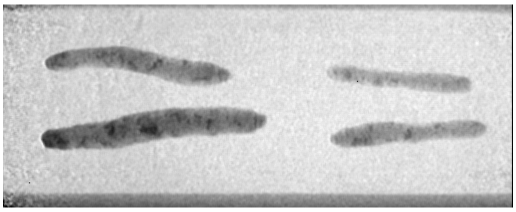
Cut-out the faces of 2 users and position them at the face spots of the kinect_photoboard.jl.
The faces have to be resized to mask size of the face spots.

Hint: Use the method of exercise B to create a correct head-mask.

27-aug-18 Using Microsoft Kinect within VisionLab 34

Exercise: Carrot Detection

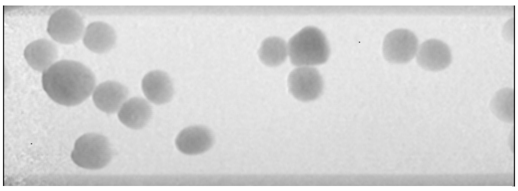
Do an accurate threshold of CarrotBelt.jl
So that there only blobs of carrots left.

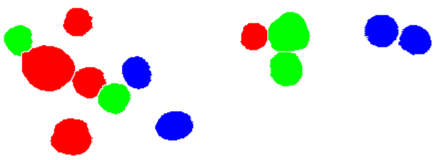


27-aug-18 Using Microsoft Kinect within VisionLab 35

Exercise: Potato Segmentation

Do a threshold so that you get a blob for each potato in PotatoBelt.jl.
Hint: Use the WatershedFeature to separate the potatoes.





27-aug-18 Using Microsoft Kinect within VisionLab 36