University of Amsterdam

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Bachelor Thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

RoCKIn @ Work Visual Servoing

Active vision using template matching
on RGB-D sensor images

Author
Sébastien Negrijn

10340912

Supervisor
Arnoud Visser

Intelligent Systems Lab Amsterdam
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

Friday 26th June, 2015

Abstract

The intention of this work is to show the combination of depth information and colour information in correlation coefficient template matching to drive an active vision algorithm with the goal to optimise the amount of perspectives that are required to correctly identify an object from the RoCKIn@Work competitions. The resulting search spaces from the template matching algorithm of both type of images (depth and colour), obtained from a Creative Senz3D RGB-D camera, will be fused. The results from the fused search space will be compared to those of the search space created from the colour images. The templates, that are labelled with an object pose, are also validated for their maximum score. This maximum score is used to generate the new perspective to which the camera will move and from where it will most likely obtain a higher score then from the current perspective. The results show that although the depth information does increase the score that represents the presence of the object in the image, which provides a safety margin and a score that is more robust against changes in lighting conditions, that a reduction in amount of perspectives required to correctly identify an object can not be made.

Contents

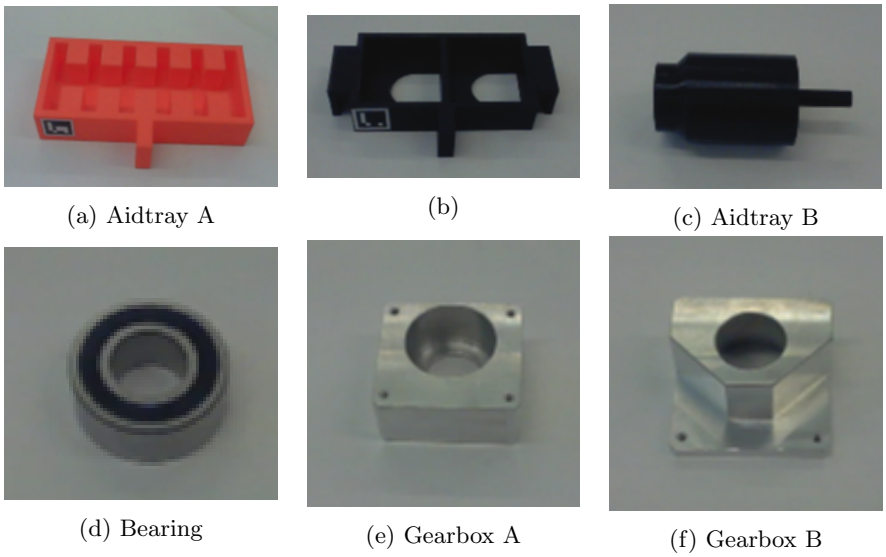
1 Introduction

In both the RoCKIn@Work competition and the RoboCup tournaments different robotic teams, one of which from the University of Amsterdam [1] [2], aim to complete several challenges that are designed to motivate the teams to develop new algorithms that go beyond the current state of the art [3]. A typical assembly warehouse is created where different parts need to be collected from different locations, assembled and modified while avoiding obstacles. The challenges are designed to simulate a section of such an assembly process. By improving the capabilities of robots in such environments more complex tasks that are currently being completed by human workers can be handed over to robots, bringing back mass production to Europe and providing a more reliable method of assembly.

One of these challenges lies in the research field of active vision which is the field that focuses on steering actuators, such as the motors in a robotic arm or the wheels of a moving platform, using visual feedback while attempting to recognise and manipulate an object. There are two situations that are distinguished in this field: hand-to-eye and eye-in-hand. In the first situation the camera is placed in the world and observes the actions of the actuators as well as the position of the object from a static position whereas in the second situation, the camera is placed in line with the actuators and moves relative to the object. This second method is often used in mobile robots that have the ability to move around in contrast to robotic arms which are mounted on a fixed position and thus the arms have a limited area that they can reach (which is also known as the work envelope).

Current robotic teams in the RoCKin@Work challenge do not take advantage of the possibility of a mobile RGB camera as they only place the camera in a top-down view position [4] [5][1]. This limits the amount of information that the camera can acquire and in turn the probability of recognising the object. The top down view of two different objects could appear the same while from a specific angle, the two objects could easily be distinguished. As this method only relies on colour images that are provided by the RGB camera it is also subject to changes in illumination which complicates the recognition process and ultimately the act of manipulation. The objects provided by the RoCKIn@Work competition can be seen in Figure 1.

[1] `https://github.com/WFWolves/wolves-at-work/blob/master/youbot_manipulation_vision/src/Vision.cpp#L179`



(a) Aidtray A (b) (c) Aidtray B

(d) Bearing (e) Gearbox A (f) Gearbox B

Figure 1: The object set provided by RoCKIn on which the recognition algorithm has to be performed.

By equipping the robot with the ability to move its camera around an object instead of limiting it to an aerial view and to not only rely on colour information but to also use depth information, a more robust method of recognising and manipulating objects can be created. In order to compare the results, an object recognition method must be found that can be both applied to colour as well as depth images. One of these methods is template matching which uses descriptions of the objects. As this method provides a probability measure of an object being perceived given an image, it is suitable to compare different objects being perceived from different locations (as different locations result in different images). These locations are the candidate positions the camera can move to by steering the arm and can be given a score based on the image that the camera will perceive from this location. By iteratively choosing the position that provides the most information and moving the arm to this position ultimately a reliable probability will be reached that represents whether or not the object that is being perceived is the target object.

In the next section will be shown that in the current literature active vision with the purpose of recognising objects has been underwhelmed by research that focuses on the creation of models and by research that only uses RGB data instead of RGB-D data. Although the resulting developed methods can be used they will have to be altered to fit the task of recognition. From this the following research question is composed:

How much does RGB-D data impact the performance of active vision while trying to recognise a known object compared to RGB data?

The expectation is that the addition of depth data to the object recognition part will have a significant boost in the accuracy (which is defined as number of times the object is recognised with the right pose). This is partly due to the fact that depth information is not subjective to illumination. The increase in accuracy of object detection in separate images will most likely translate to a better trajectory of the camera towards a position in which it can recognise the object with a high enough certainty. A trajectory is considered better when it is shorter then another trajectory while still reaching a recognition certainty that is equal or higher then the other trajectory. This more optimal trajectory translates to a shorter period of time that is required to recognise the object and ultimately leads to a shorter time that is required for the more high level task that is at hand such as the assembly of a product.

In summary, the advantages of using depth information in addition to colour information will be tested by comparing the fused information of depth and colour, to just colour by using template matching. The expectation is that the addition of depth information will provide a method that is capable to recognise objects under different lighting conditions, due to the fact that the depth information remains unaffected by such changes.

In the next section a more detailed view of the research field, as well as the setting in which the resulting algorithm will be used, will be given.

2 Related Work

2.1 Robot competitions and Challenges

As mentioned in the introduction, RoboCup@Work and RoCKIn@Work are competitions that focus on providing challenges in the robotic research field which aim to drive the current state of the art forward. The challenges vary across the different aspects of robotics and include subjects from both the scientific and industrial field. The subjects that are used in the scientific field focus on improving algorithms that involve: perception of the surroundings, path planning, robot human interaction and others. The industrial challenges aim to solve more practical tasks such as loading parts from machines onto palettes, unpacking items from boxes and moving objects from one location to the other. Both are equally important, as without the supporting algorithms a task could not be successfully taken care of. On the other side, without the practical tasks, no benchmarks would be available to test the performance of the algorithms.

2.2 Depth Cameras

In order to perceive the surroundings a camera is required that is able to provide both colour and depth information. A number of such cameras are available with the most popular one in the scientific field being the Kinect 1 by Microsoft [6]. This camera is one of the first depth cameras that was used in consumer electronics and then quickly adopted in different research fields. It is however

(a) The Microsoft Kinect 1.

(b) The Microsoft Kinect 2.

(c) The Creative Senz3D.

(d) The Asus Xtion Pro Live.

Figure 2: The Kinect 1, Kinect 2, Creative Senz3D and Asus Xtion Pro Live.

not suitable as a camera which has to be mounted on the end of a robotic arm for three reasons: its weight, size and range. The payload of the robotic arm that will be used in this application has a payload limit of half a kilo and the weight of the Kinect 1 is 1.4 kilograms. More important, the range of the Kinect 1 is limited to a minimum of 0.7 metres while in this application objects will be perceived at a distance as close 0.2 metres. This is also the reason why the new Kinect 2 by Microsoft can not be used as its range is between 0.8 and 4.0 metres for depth information.

The two remaining cameras are the Asus Xtion Pro Live and the Creative Senz3D. The first camera does fit the weight requirements but still has limited range for the application, this does however not make it unfit for other scientific applications [7]. Finally, the Creative Senz3D [2] has a range between 0.2 and 1.0 metres for depth information, making it a close range sensor. Although its intended application is the tracking of hand gestures [8], there is no reason why this camera would not be suitable for object recognition.

2.3 Methods used by other robotic teams

Different solutions towards visual servoing are suggested by a number of teams which participate in the robotic leagues. One of which (Swarmlab) uses the following features: length of principle axis (pixel), length of second principle axis (pixel), size of the area (pixel^2) and finally the average intensity [4]. These features are then fed into a J4.8 decision tree [9]. These features are derived

[2] `http://en.europe.creative.com/p/web-cameras/creative-senz3d/`

from images that are captured by a camera that is positioned directly above the scene. The WFWolves use a similar method in which they instead combine the height and width of the separated objects to calculate the ratios (height/width) of the objects in order to recognise them [5][3]. It is worth noting that these methods are not necessarily state of art but still perform adequately in the current challenges provided by the different competitions.

The major downsides of these similar approaches is that the features are very sensitive to changes in lighting conditions: as a result from this the decision tree has to be trained on-site and can not be prepared beforehand because the colour values change depending on the location. These changes can be made irrelevant by mainly relying on depth information to recognise the objects. These methods are however sensitive to changes in perspective of the perceived object. When the camera would be placed at an angle instead of the top-view all of the features would no longer match the values learnt in the decision tree or the ratios that are provided, proving the current methods of the different robotics teams to be not as reliable as they could be. A possible alternative solution towards this change of perspective, is to apply a correction for this shift.

2.4 Active vision

As stated by Roy et. all [10] a single view of a RGB image may not be enough to distinguish between two objects, the objects can even appear exactly the same. In this paper a reactive planning method to decide where the end effector is to move next is proposed. By constructing a partial 3D model of the object during recognition less (resource intensive) features can be used in comparison to when the object is to be recognised from a single image resulting in an online recognition application. A database of pre-made models is then compared to this partial model from which a move is determined that will result in the most distinguishable next image that can be combined with the partial model. In contrast to this paper, the use of depth images instead of RGB images to construct a partial model will result in a more complete partial model. Not only depth features that have to be extracted from the previously RGB images can be used but instead all points in the depth image can be used. The depth images will also not be influenced by lighting conditions.

The previous planning algorithm has been redesigned and improved in recent years [11] by applying information maximisation. Although this new method is optimised for model reconstruction it can be redesigned towards object recognition. This method is particularly suited as it allows for both refinement and new exploration by using voxels to represent the perceived features and their uncertainties. In order to calculate the best next pose a number of candidate

[3] `https://github.com/WFWolves/wolves-at-work/blob/master/youbot_manipulation_vision/src/Vision.cpp#L179`

poses are generated based on the boundaries of the current generated model. The best new position would be the position that previously generated the highest score for the object that is to be recognised.

The use of RGB-D cameras in the field of active vision seems to be sparse as the main focus of these cameras is mapping of either environments or modelling of objects [12] [13] along side with the recognition of human gestures [14].

The next section will discuss various methods towards object recognition based on RGB-D data.

2.5 Object recognition

Classical methods based on SIFT [15] which use multiple histograms based on gradient descriptors, are not suitable for objects that do not have a lot of texture as without texture there are less gradients. This method is therefore not suitable for objects in the industrial field as provided by RoCKIn. This spawned the field of texture-less tracking. Most of the methods that do not depend on texture, are based on depth cues. These depth cues can be extracted from RGB-D data as well as RGB data and will be discussed in closer detail in the next section.

2.5.1 Object Recognition from RGB-D data

Templates Multiple methods for recognising objects from RGB-D data have been suggested. In one particular method templates are used [16]. Here the detection and tracking of objects is based on online learning which results in multiple templates for the object with different assigned poses. The motivation for online learning is that for complex models it is cumbersome to acquire such a model and that edge based detection methods are typically too time consuming. Fortunately, the objects in the proposed industrial setting are not complex and there are other methods to speed up the edge based detection. Still, the features proposed in Park's paper which consist of combining contour information from both the RGB en depth images will be used. By using contours the sensitivity towards illumination will be decreased.

3 Method

In order to create a trajectory that the camera can follow based on the visual feedback, a number of different methods need to be used. Before these different methods are explained, a short description of the hardware that is used will be given.

Figure 3: A colour image from the Creative Senz3D.

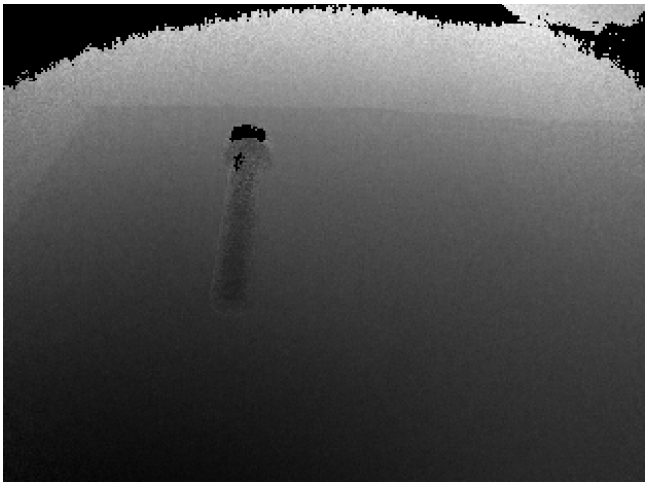


Figure 4: A depth image from the Creative Senz3D. Values with a higher intensity represent points in the real world that are further away from the camera sensor than points with a lower intensity.

3.1 Hardware

3.1.1 Creative Senz3D

In order to provide the visual feedback required for an active vision system, the Creative Senz3D was chosen. As described earlier in section 2.2, the camera has both a colour (RGB) and depth sensor usable for ranges from 20 centimetres up to 1 metre, making it suitable for close counter interaction with objects. When the camera is mounted on top of the robotic arm of the Kuka youBot, data from both the RGB and depth sensor can be received simultaneously, resulting in images such as those in Figures 3 and 4. The depth image is to be interpreted as follows: pixels with higher intensities are further away from the camera then values with a lower intensity. Pixels with an intensity of zero (black) lay outside the range of the camera, these values are also visible inside of the object caused by reflections of the infrared (IR) projection made by the Creative Senz3D. These reflections are similar to partial obstruction of the object as no usable information is gained from that region.

Although images can be observed from both sensors at the same time, the sensors are not perfectly synchronised. The example images in the previously mentioned Figures also show the slight shift in perspective, as both sensors are mounted approximately 3 centimetres apart horizontally. This shift in perspective is corrected by calibrating both the sensors to correct the warping, as well as providing corresponding features in both the depth and colour image from which the shift can be estimated. Both the camera calibration and perspective estimation are described in further detail in section 3.3.1. The colour images have a resolution of 1280 by 720 pixels while the depth images have a resolution of 320 by 240 pixels both providing images at thirty frames per second (FPS). The colour images are first scaled to a similar resolution of the depth image while maintaining the original aspect ratio, resulting in a image of 430 by 240 pixels. This reduces the amount of pixels from the colour image to be processed from 921600 to 103200, a 88.80% reduction of pixels. As each pixel of the image needs to be processed to determine if the object is visible in that section of the image and both the depth and colour image will be combined, the reduced amount of pixels results in shorter execution time per image.

3.1.2 Kuka youbot

The camera mentioned in the previous section is mounted to the end-effector of the youBot developed by Kuka [4]. This robot is, although not officially, the standard platform used in the RoCKIn competition as well as the RoboCup@Work and is an educational platform for robotics [17] [18] [19]. The robot composes of two parts: an omnidirectional platform and a robotic arm consisting of 5 joints as can be seen in Figure 5. Due to its relative low price compared to industrial standard robots, it is used by a large number of research institutes and universities. It also comes fully assembled without the need of any knowledge of

[4] `http://www.kuka.com/`

Figure 5: The KUKA youBot with the arm attached to the omnidirectional platform.

electrical components, this makes it suitable for universities that focus on software rather than hardware such as the University of Amsterdam. The software drivers used to communicate with the actuators are integrated into ROS, which is described in the next section.

3.2 Robot Operating System (ROS)

ROS is a broadly used framework for robotic application development. It supplies an easily extensible environment of basic components (nodes) which can be combined flexibly to form applications. Furthermore, it comes with a range of packages, libraries, drivers and simulation programs that simplify the use of standard platform robots. These nodes communicate with each other using topics based on the publisher-subscriber model. In this model, multiple nodes with different functions can be subscribed to the same topic (e.g. multiple image processors using the same camera). The Kuka youBot has multiple nodes that represent parts of the robot: the arm and the platform. By sending messages to these nodes over the specified topics the arm or platform will move according to the contents of those messages.

3.3 Image Processing

As a number of preexisting algorithms are required for the generation of a trajectory for the camera, the Open Computer Vision Library (OpenCV[5]) is used as it contains a large number of implementations for these algorithms.

[5] `http://opencv.org/`

First, both the camera sensors are calibrated after which, a form of template matching is applied to the produced images.

3.3.1 Calibration of Camera Sensors

Before data from the two sensors on the camera can be used, both the sensors have to be calibrated. Low cost cameras often suffer from distortion also known as the "barrel" or "fish-eye" effect. The calibration is done by taking a known pattern, as can be found on a chessboard, and moving this pattern in front of the camera spread around its field of view. By detecting the crossings on the chessboard and comparing their relative location to each other with the relative position of the crossings in the known pattern, the distortion matrix can be calculated [20]. When this distortion matrix is first applied to the images they are suitable for use for further processing as they are no longer warped.

3.3.2 Template Matching

For the template matching of both the colour and depth images, the correlation coefficient between a template and the image is calculated. Pixels in the template are described by the symbol $R(x, y)$, pixels from the template as $T(x, y)$, w and h are the width and height of the image and finally $I(x, y)$ is the resulting search space or matrix.

$$R(x,y) = \sum T'(x',y') \cdot I'(x+x', y+y') \tag{1}$$

Where:

$$T'(x',y') = T(x',y') - \frac{1}{w \cdot h} \cdot \sum_{x^n, y^n} T(x^n, y^n) \tag{2}$$

And:

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{w \cdot h} \cdot \sum_{x^n, y^n} I(x+x^n, y+y^n) \tag{3}$$

The result from this equation is then normalised across the entire image resulting in the final formula of:

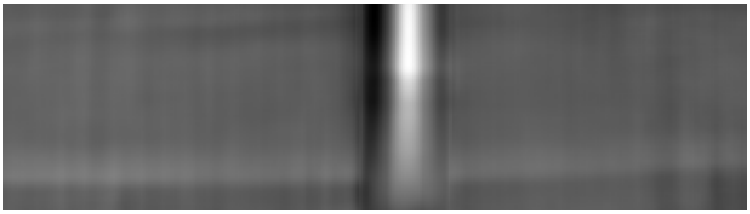
$$R(x,y) = \frac{\sum T'(x',y') \cdot I'(x+x', y+y')}{\sqrt{\sum T'(x',y')^2 \cdot \sum_{x^n, y^n} I'(x+x', y+y')^2}} \tag{4}$$

When applied to an image, for example the colour one, the result is a matrix (search space) containing scores for every pixel in the original image that represents the odds of the template being in that position of the original image. An example of such a matrix can be seen in Figure 6(c).

As can be seen, the resulting matrix, also referred to as the search space, no longer maintains the same aspect ratio as the original input colour image. Only

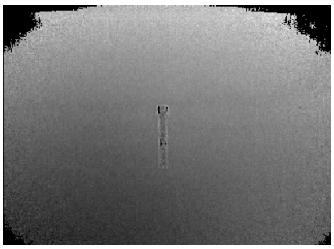
(a) An example of a colour image.

(b) A template of the bolt object from the RoCKIn@Work challenges.



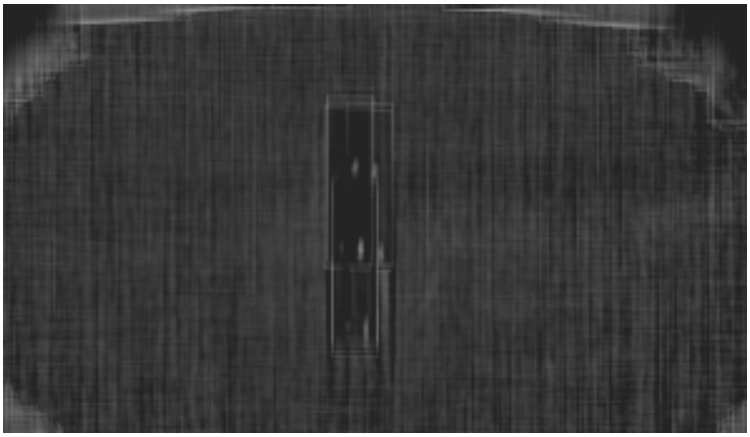
(c) The resulting search space from the above colour image and template. The image is normalised in such a way the the minimal value of the pixels in the image equals to zero and the maximum value equals to one in order to better visualise the different values in the search space.

Figure 6: An overview of the different images required to match a single template to a single input colour image. The template is matched with the colour image which results in the search space matrix containing probabilities of the template being present in that position in the original image.



(a) An example of a depth image.

(b) A template of the bolt object from the RoCKIn@Work challenges.



(c) The resulting search space from the above depth image and template. The values in this image have been normalised to better visualise the different values in the matrix.

Figure 7: An example of the search space from a depth image and the teplate used to create the search space.

the positions of the pixels that could possibly match a full template are searched, meaning that both from the bottom and the right side of the image a small part is left away in the remaining search space matrix. In the search space image, the most likely position of the template is represented by the highest value in the search space.

The same can be done with the depth images with their corresponding templates as shown in Figure 7.

3.3.3 Template Creation

In order to match a template, these templates first need to be created. This is done by hand for each object and for each perspective of these objects. In order to determine how many different perspectives of each object were needed,

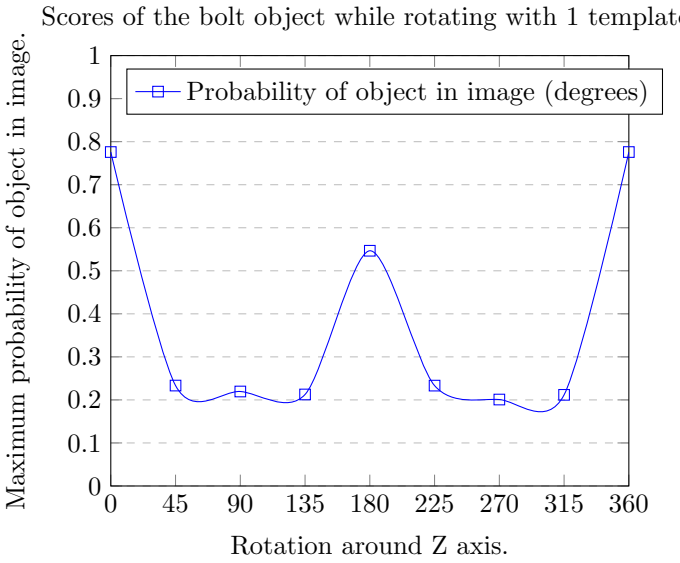


Figure 8: The highest scores that are obtained are plotted as the object (bolt) is rotated while there is only one template being used. This template is of the object under a rotation of zero degrees. The spike at a rotation of 180 degrees is caused by the near mirrored appearance with only the head being different.

a small test was conducted by matching a single object to a template under different orientations. The object was rotated 45 degrees around the Z axis while maintaining a profile view of the object such as in Figure 6. The results from this test can be seen in plot 8.

The plot shows that the maximum score obtained for this object reduces from 0.78 to 0.21. When templates are created of the object at 45 degrees intervals the score becomes no less then 0.34, meaning that the object is clearly distinguished from the background as can be seen in Figure 9 and 10. Ideally more templates would be added at a smaller interval than 45 degrees, this would however introduce a lot of templates that would need to be processed for each image in turn increasing the computational time required to process each image.

To partly reduce the amount of templates that are required to describe an object, only templates are created of the object from a perspective that can be reached by the camera relative to the object. For example: the bolt object is not able to balance itself at a 45 degree angle from a vertical position, therefore this perspective and all the possible derived perspectives from this position are disregarded. The possible perspectives are further reduced by the available positions of the camera as these are limited to the possible positions of the various joints of the youBot robotic arm.

The remaining views of the bolt object therefore look as can be seen in Figure 11. The set of templates for the other objects in the RoCKIn@Work

The highest scores as the object is rotated with available templates at 45 degree intervals.

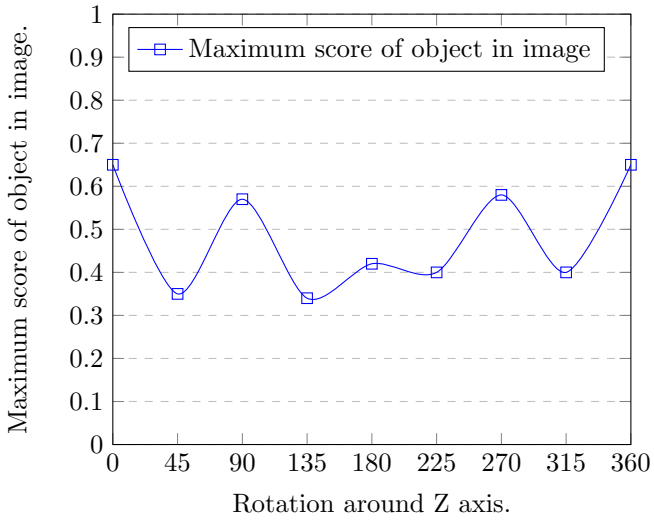


Figure 9: While the object is rotated, some templates receive higher maximum scores than others. This means that these templates provide more information which will later on be used to generate the path of the camera towards a position that returns the maximum amount of information.

The score of an object found in image while the object is not present.

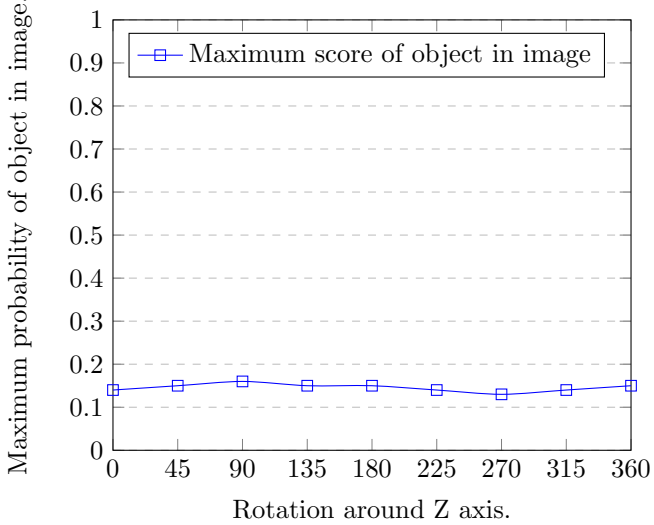


Figure 10: The maximum score obtained for a object while the object is not present.

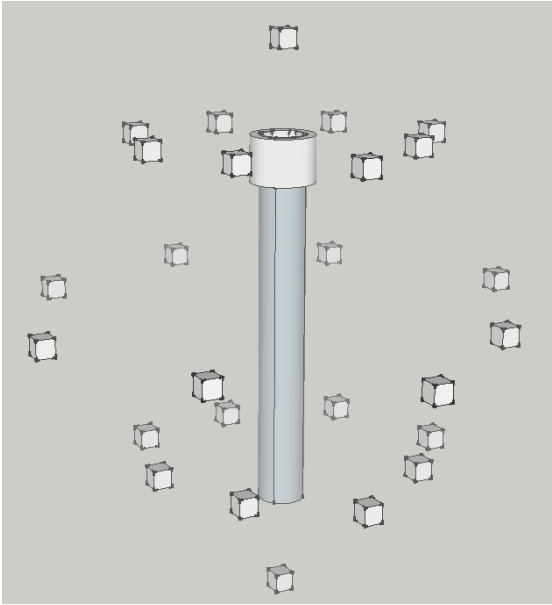


Figure 11: Image of the possible perspectives of the bolt object from which templates are created.

challenge are created in the same manner.

When all templates of the different objects are available, the next step is to fuse the images from the colour and depth sensors together to provide a alternate source of information that drives the active part of active vision in this algorithm.

3.4 Colour and Depth image Fusion

In order to fuse both the images, a transformation must be found between the two sensors that transforms the coordinate system of one of the images to the coordinate system of the other image. This is done by first manually picking the location of at least four features that are visible in both the colour image and the depth image. With these 4 pairs of features the transformation matrix can be calculated. First, two sets of linear equations are created from the features from both images (one for each image).

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \tau \end{bmatrix} = \begin{bmatrix} x_4 \\ y_4 \\ 1 \end{bmatrix} \tag{5}$$

Which for both images can be transformed into:

$$A = \begin{bmatrix} \lambda x_1 & \mu x_2 & \tau x_3 \\ \lambda y_1 & \mu y_2 & \tau y_3 \\ \lambda & \mu & \tau \end{bmatrix} \tag{6}$$

Now, in order to transform points from one image into the coordinate system of the other, the following matrix can be used:

$$C = B \cdot A^{-1} \tag{7}$$

Where C is the final transformation matrix from points in the depth image to points in the colour image, A the derived feature matrix from the depth image and B the derived matrix from the colour image. In order to perform these calculations, the implementation of projection transformation in the OpenCV library, as mentioned earlier, is used.

Now that both the images, colour and depth, have their points in the same coordinate system what remains is to combine the values into one single search space. This is done based on weights that scale all the values in the images where the sum of the two weights equals one in order to preserve a spectrum of scores between 0 and 1. The exact value of the weights is determined experimentally by maximising the total score for the correct position in multiple test positions.

3.4.1 Trajectory Generation

Some templates generate higher scores as can be seen in Figure 9. When during the challenge the camera is supposed to see the object, the maximum score from the search space is retrieved. This score is based on a labelled template that contained the orientation of the object relative to the camera. If this initial score is not high enough to be sure that in this position the object is actually visible, a new position for the camera must be generated. Similar to how the scores in Figure 9 were generated, all templates for an object are validated under different settings. These different settings include various angles from which the object is visible and different lighting conditions. The camera is moved with one of the objects in view, while analysing the images the maximum scores that the templates generate on a positive detection are recorded. This results in a set, per object, that contains the maximum score each template has generated.

The goal of active vision is to generate a motion path for the camera from which this camera can obtain a view that will generate a high score with one of the templates. From the initial view of the object a score will be calculated based on the template matching technique before, this score is based on a template which is labelled with a perspective. As long as the score from the current perspective is not higher then a certain threshold, a next perspective can be picked from a sub set of the templates with their maximum scores as determined previously. The subset contains all perspectives that can be reached from the perspective that the camera is currently believed to be in, by a single 45 degree shift in orientation around the location at which the object is supposed to be at.

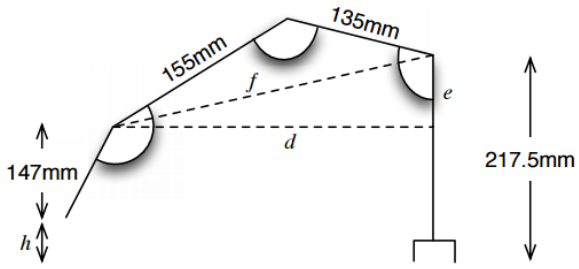


Figure 12: A schematic overview of the Kuka youBot robotic arm and the angles which are required to calculate the final joint positions. The image itself is created by SwarmLab [4].

In order to test if the use of depth information positively influences this process, the set of maximum scores for the templates is generated for both the colour search spaces alone, as well as for the fused search spaces that are composed of the colour and depth images. The entire process of active vision, as used in this system, will be considered more successful if the fused images require less moves of the camera before a perspective is reached that has a score above a threshold. Less moves of the camera translate into less time required for the object to be recognised. The initial perspective is one of the object being somewhere in the field of view of the camera.

3.5 Inverse Kinematics

When a new perspective is chosen the right angles for all the joints of the robotic arm must be calculated. As the youbot arm only has 5 joints, of which a maximum of 3 in the same plane, only a closed loop solution using geometric algebra is required. The measurements and angles as described in Figure 12 are used. This simple inverse kinematics node generated the joints based by starting with joint 1 which rotates around the Z axis after which the remaining joints can be easely calculated.

4 Results

As active vision can be divided into two section: object recognition and the selecting of the next perspective, the results are also presented as such starting with object recognition.

4.1 Fusion of Search Spaces

When the search spaces are created from both the colour and the depth images these search spaces need to be fused. After the depth search space has been

Figure 13: The fused search space consisting of both the depth and colour search space.

transformed to the coordinate system of the colour image, they can be combined resulting in a new search space as can be seen in FiguresearchspaceFused.

As can be seen in the original search spaces from Figures 7(c) and 6(c) both these search spaces contain a region in which they detect the object. By combining these regions a higher score, as shown in the next section, is present in the search space at the position where the object is placed relative to the camera.

4.2 Object Recognition

All the objects were tested using all their templates resulting in a total of 7 objects with between 5 and 8 templates which are rotated at intervals of 45 degrees. Although the set of objects varies across both colour and dimensions, their results are very similar. Therefore the data of one object, the bolt object, will be explained in detail while the data of the remaining objects is presented in a table in the appendix.

In Figure 14 the maximum scores that are awarded to each template over a single rotation over the Z-axis is visualised. The scores that are visualised are the maximum scores that were found while trying to match all the templates from the bolt object. These found templates always matched the right object. The object here was placed in similar lighting conditions as in which the template images were created: a standard office building with windows on a sunny day. The obtained scores always match the right perspective at which the object was placed relative to the camera. As can be seen, the resulting scores from the search space that are created by fusing the colour and depth information are always higher then the scores obtained by only the colour search space, this also applies to all the other objects used in the test set as can be seen in Table 1.

In the second comparison, the lighting conditions were changed and the room was dimmed using curtains resulting in the only source of light being a desk lamp. With these lower illuminance levels, both the score for the colour and the score for the fused colour and depth search spaces are lower than when the templates are compared in similar lighting conditions, as can be seen in Figure 15. This figure also shows that the difference between the maximum scores obtained for both the different search spaces is increased in comparison to Figure 14. The scores for the fused colour and depth search space, although

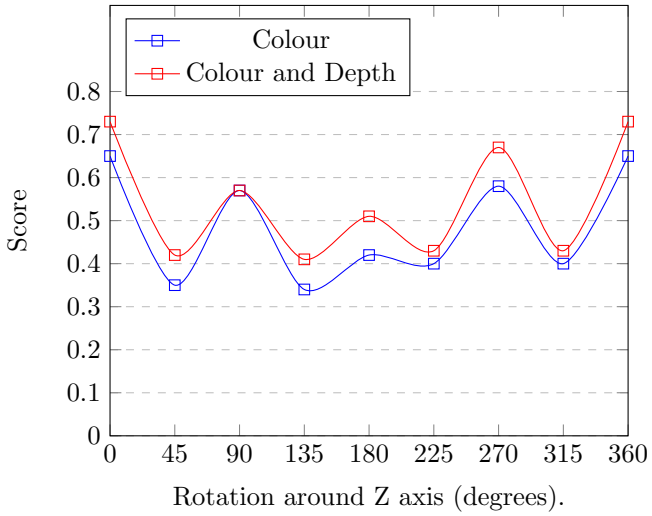


Figure 14: Comparison of the highest scores of the matching templates between the colour and the colour and depth search spaces. The test area consisted of similar lighting conditions as when the templates where created.

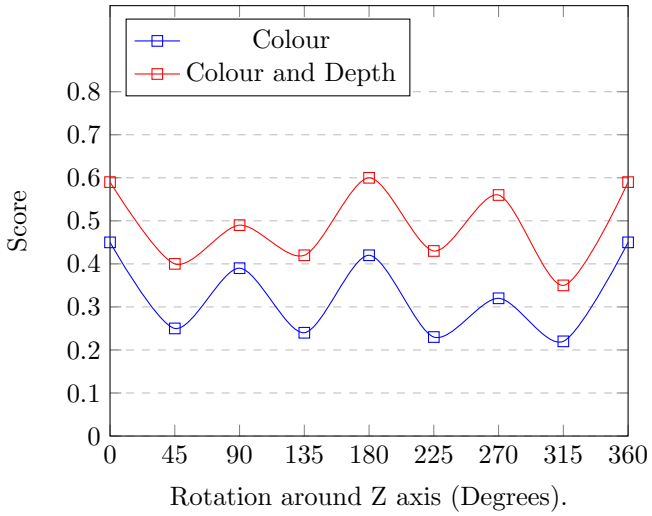


Figure 15: Comparison of the highest scores of the matching templates between the colour and the colour and depth search spaces. The test area consisted of dimmed lighting conditions compared to when the templates were created.

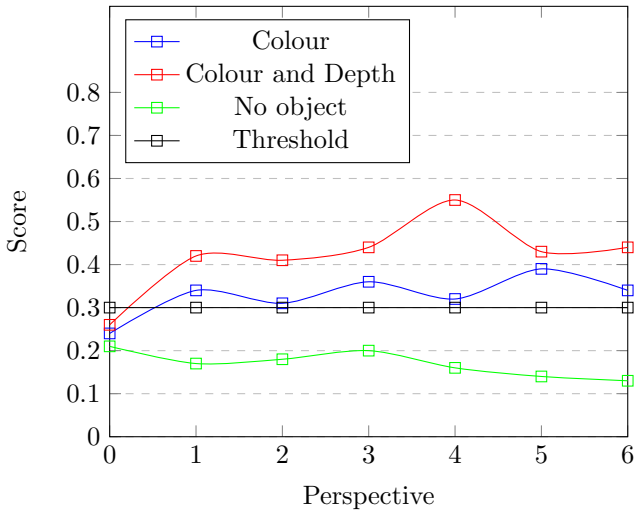


Figure 16: An example of the perspective generation and the resulting scores with the bolt object.

lower, are not as much affected as the maximum values from the colour search space.

4.2.1 Perspective Selection

As previously mentioned, the second section is to move to a new perspective that has the highest probability of providing a high score. Again the results from the bolt object will be visualised as the results from the remaining objects in the test set provide similar results.

In Figure 16 an example of the behaviour of the changes in perspective is presented. The red line indicates the scores of the fused colour and depth search space and the blue line the scores obtained from just the colour search space. In green is the score that would have been obtained when there is no object in the view of the camera, also know as false positive (F/P in table 2). Finally, the black line is the threshold that could be used to distinguish between the situation when no object is observed and when an object is observed. The Figure shows similar results to those in Figure 14 where again the scores that are obtained from the fused search space are higher than those of the colour search space. Despite the scores from the colour search space being lower, these values are still higher then those of the values obtained from an empty scene where no object is present. The chosen threshold is between the scores of these possible false positives and the scores of the colour search space. This same trend continues in the other objects in the testing set as can be seen in Table 2.

5 Conclusion

Template Matching

From the Figures 14 and 15 and from the Table 1 can be seen that the template matching algorithm used is already performing at such a level that the need for the depth information is initially not present. This is due to the nature of the algorithm and its use of the correlation coefficient instead of directly comparing the values of the template with the values in the colour image. Due to these already high scores the addition of depth information is not resulting in scores that are significantly higher. Only when the template matching is conducted with a very low illuminance level, the difference of the additional depth information becomes more apparent. The score increases significantly after the initial perspective, as in the initial perspective the object is not necessarily in the centre of the field of view. After the first change of perspective the object is placed in the centre.

Perspective Planning

Because the template matching algorithm is already functioning so well with just the colour images, no difference can be made in the planning aspect of active vision as can be seen in Figure 16 and Table 2. In all the obtained results, there is always a threshold that can be set that separates the values of the colour image search space from those of the false positives. Although no difference can be made between the two search spaces, the addition of depth information does add a safety margin that further separates the scores of the perspectives from the scores of the false positives.

To answer the research question:

> *How much does RGB-D data impact the performance of active vision while trying to recognise a known object compared to RGB data?*

the answer is that the additional depth information added to the search space does not result in a reduction in required amount of perspectives. For both methods two perspectives are sufficient.

6 Discussion

Contrary to the initial thoughts of the author, the results of template matching on only the colour image deem to be enough to drive the planning of the different perspectives. This might be different however, if a different test set of objects is used as the test set provided in this specific RoCKIn challenge is limited to a total of 7 objects (including the bolt object). With a broader range of objects different results might be obtained from which a difference in perspective planning becomes apparent. With the current amount of templates the pose of the object can be estimated up to a precision of 45 degrees as templates are

available at this interval, whether or not this is precise enough to perform a manipulation action such as picking up the object remains to be tested. With the use of more templates at a lower interval, the time that would be needed to process a single colour and depth image combination would take too long to consider the application to be real-time. It is however possible that with a more optimised implementation the amount of templates can be increased without the time that is required to process these images to increase as much.

7 Future Possibilities

As mentioned in the conclusion, as of now the poses of the objects can be estimated at a 45 degree interval due to the amount of templates that are available. It might be possible to increase this interval, not by adding more templates, but by comparing the scores of the different templates from neighbouring perspectives. As the camera moves from one perspective to a second perspective, the score obtained from the template that corresponds to the first perspective will decrease, while the score of the template that matches the second perspective will increase. By using these scores, an intermediate position can be estimated as this position can be defined as the crossing of the two scores of the two perspectives. This comparison can possibly be made more precise by instead of directly comparing the scores, feeding the scores into a machine learning algorithm with labelled positions. By using machine learning the scores of the different templates are used as features to calculate the position. As the scores vary at each perspective, not limited to the 45 degree interval, the machine learning algorithm might provide a rough estimate of the pose of the object not limited to the 45 degree intervals.

References

[1] Sébastien Negrijn et al. *UvA@Work - RoCKIn@Work 2014 - Toulouse, France*. Team Description Paper, Intelligent Robotics Lab, Universiteit van Amsterdam, The Netherlands. Nov. 2014.

[2] Valerie Scholten et al. *UvA@Work, Team Description Paper RoCKIn Camp 2015 - Peccioli, Italy*. Team Description Paper, Intelligent Robotics Lab, Universiteit van Amsterdam, The Netherlands. Nov. 2015.

[3] GerhardK. Kraetzschmar et al. "RoboCup@Work: Competing for the Factory of the Future". English. In: *RoboCup 2014: Robot World Cup XVIII*. Ed. by Reinaldo A. C. Bianchi et al. Vol. 8992. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 171–182. ISBN: 978-3-319-18614-6. DOI: 10.1007/978-3-319-18615-3_14. URL: http://dx.doi.org/10.1007/978-3-319-18615-3_14.

[4] Sjriek Alers et al. "How to Win RoboCup@ Work". In: *RoboCup 2013: Robot World Cup XVII*. Springer, 2014, pp. 147–158.

[5] Arne Hitzmann et al. *WF Wolves @Work Team Description - RoboCup 2013*. 2013.

[6] Jan Smisek, Michal Jancosek, and Tomas Pajdla. "3D with Kinect". English. In: *Consumer Depth Cameras for Computer Vision*. Ed. by Andrea Fossati et al. Advances in Computer Vision and Pattern Recognition. Springer London, 2013, pp. 3–25. ISBN: 978-1-4471-4639-1. DOI: 10.1007/978-1-4471-4640-7_1. URL: http://dx.doi.org/10.1007/978-1-4471-4640-7_1.

[7] Krystof Litomisky. "Consumer rgb-d cameras and their applications". In: *Rapport technique, University of California* (2012), p. 20.

[8] Sean Ryan Fanello et al. "Learning to be a depth camera for close-range human capture and interaction". In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 86.

[9] Ross Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[10] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. "Isolated 3D object recognition through next view planning". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 30.1 (2000), pp. 67–76.

[11] Sergi Foix et al. "Information-gain view planning for free-form object reconstruction with a 3d tof camera". In: *Advanced Concepts for Intelligent Vision Systems*. Springer. 2012, pp. 36–47.

[12] Peter Henry et al. "RGB-D Mapping: Using depth cameras for dense 3D modeling of indoor environments". In: *In the 12th International Symposium on Experimental Robotics (ISER*. 2010.

[13] Peter Henry et al. "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments". In: *The International Journal of Robotics Research* 31.5 (2012), pp. 647–663.

[14] Jesus Suarez and Robin R Murphy. "Hand gesture recognition with depth images: A review". In: *RO-MAN, 2012 IEEE*. IEEE. 2012, pp. 411–417.

[15] David G Lowe. "Object recognition from local scale-invariant features". In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[16] Youngmin Park, Vincent Lepetit, and Woontack Woo. "Texture-less object tracking with online training using an RGB-D camera". In: *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*. IEEE. 2011, pp. 121–126.

[17] R. Bischoff, U. Huggenberger, and E. Prassler. "KUKA youBot - a mobile manipulator for research and education". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. May 2011, pp. 1–4. DOI: 10.1109/ICRA.2011.5980575.

[18] Valerie Scholten et al. *Intelligent House Builder*. Project Description, Zoeken Sturen Bewegen, Universiteit van Amsterdam, The Netherlands. June 2014.

[19] Sebastien Negrijn et al. *Basic order picking met behulp van de KUKA YouBot*. Project Report, Universiteit van Amsterdam. Science Park 904 1098 XH Amsterdam, July 2013.

[20] Zhengyou Zhang. "A flexible new technique for camera calibration". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.11 (2000), pp. 1330–1334.

Appendices

Template Scores

Object	Type	T1	T2	T3	T4	T5	T6	T7	T8	T9
Aidtray_A	Colour	0.337	**0.511**	0.405	0.373	0.295	0.497	0.427	0.429	0.430
Aidtray_A	Fused	0.564	0.662	0.511	0.586	0.525	**0.676**	0.494	0.614	0.542
Aidtray_B	Colour	0.438	0.505	0.376	**0.592**	0.399	0.428	0.423	0.589	0.460
Aidtray_B	Fused	**0.606**	0.536	0.431	0.596	0.494	0.550	0.483	0.640	0.521
Axis	Colour	**0.657**	0.594	0.271	0.648	0.457	0.360	0.378	0.429	0.424
Axis	Fused	**0.725**	0.611	0.481	0.662	0.461	0.586	0.440	0.585	0.497
Bearing	Colour	0.492	0.606	0.498	0.639	0.465	0.713	0.354	**0.715**	0.548
Bearing	Fused	0.684	0.692	0.577	0.737	0.594	**0.737**	0.551	0.735	0.600
Gearbox_A	Colour	0.429	0.541	0.462	0.491	0.311	**0.573**	0.407	0.445	0.411
Gearbox_A	Fused	0.579	**0.620**	0.516	0.602	0.455	0.619	0.549	0.575	0.521
Gearbox_B	Colour	**0.606**	0.592	0.331	0.411	0.331	0.372	0.257	0.477	0.353
Gearbox_B	Fused	**0.689**	0.593	0.395	0.592	0.406	0.591	0.412	0.619	0.492

Table 1: All the templates of all the objects were scored under different perspectives using both the colour and the fused colour and depth search space. All their resulting scores are presented here.

Perspective Generation

Object	Type	P1	P2	P3	P4	P5	P6
Aidtray_A	Colour	0.248	**0.468**	0.308	0.496	0.260	0.403
Aidtray_A	Fused	0.415	**0.533**	0.420	0.502	0.435	0.524
Aidtray_A	F/P	0.107	0.098	0.111	0.106	**0.121**	0.120
Aidtray_B	Colour	0.434	0.496	0.314	**0.552**	0.403	0.488
Aidtray_B	Fused	0.421	0.533	0.466	**0.564**	0.452	0.551
Aidtray_B	F/P	**0.129**	0.079	0.087	0.098	0.099	0.093
Axis	Colour	0.378	**0.547**	0.406	0.479	0.417	0.380
Axis	Fused	0.535	0.595	0.558	**0.640**	0.503	0.593
Axis	F/P	0.130	0.067	0.087	0.088	0.082	**0.097**
Bearing	Colour	0.360	0.449	0.337	**0.545**	0.384	0.511
Bearing	Fused	0.413	0.547	0.452	0.569	0.457	**0.573**
Bearing	F/P	0.144	0.092	0.134	0.119	0.122	**0.126**
Gearbox_A	Colour	0.353	0.394	0.439	0.449	0.276	**0.494**
Gearbox_A	Fused	0.394	0.581	0.497	0.546	0.459	**0.590**
Gearbox_A	F/P	0.102	0.094	0.092	0.109	0.093	**0.135**
Gearbox_B	Colour	0.410	0.518	0.442	0.430	0.468	**0.557**
Gearbox_B	Fused	0.479	**0.582**	0.520	0.528	0.490	0.573
Gearbox_B	F/P	0.119	0.127	0.108	0.089	**0.135**	0.133

Table 2: Test runs were done with the different objects which resulted in a set of perspectives per object. The type of data is either colour search space score(colour), the fused search space score of colour and depth(fused) or the false positive score of the scene without the object present(F/P). The different perspectives (P1, ..., P2) are perspectives that were chosen based on the previous maximum found scores of their corresponding templates.