TDT4195 Visual Computing Fundamentals

Project Fall 2013

TDT4195 - Image Processing and Graphics - Autumn 2013

1 Project Conditions

- You can work in groups of 2 people.
- Notify Igor or Mehdi about the group-composition before the 15th of November.
- Deliver your code along with the report on its-learning before the deadline.
- There will likely be an evaluation presentation of your system. There you should show your working system and explain the used techniques. Where are the limitations? What turned out to be more difficult than thought? Which was easier than thought?

2 Project-Description

In this semester your task for the TDT4195-project is to create a program that is able to segment, locate and count simple shaped objects of different color. You can assume that the objects are recorded with a simple camera setup showing objects in front of a relatively homogeneous background. There might be dark-light gradient present due to varying lighting. Example images are shown in figure 1. The obtained information (position, numbers) is then used to create and visualize a 3D graphical-scene. Below we describe the minimum requirements that your program should be able to solve, but you are free to vary or extend the basic task as long as it has a similar difficulty level.

2.1 Minimal Requirements

- Image Processing: Segment the different colored objects in the image (here skittles/smarties). Mark their center position and automatically count the numbers of objects for each color (object-type).
- Your program should be able to find and count the different colored skittles in the images shown in figure 1. You can assume that the objects do not overlap.
- Graphics: Create a 3D scene where you automatically place a 3D-model at all locations that correspond to the measured center positions obtained in the image processing part. Choose for each object-type a different 3D-model (e.g. 3 different colors would need 3 3D-models). Depending on the object-count display the number of appearances on top of each 3D-model and let the number rotate.
- Write a report on 1-2 pages describing the key-points of your solution. Shortly comment on how your solution can be improved (When does it fail? What could be done to make it more robust.)

2.2 Example Images (standard task)

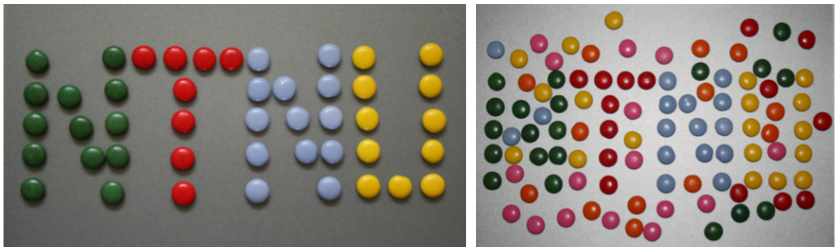

Figure 1: *Example images for the standard task*

The test-images can be downloaded from its-learning!.

2.3 Possible Extensions (Voluntary)

Beyond the above suggested basic task you might want to solve more challenging tasks. Here a few suggestions:

- Make your basic program work on images showing objects that overlap and touch quite often (left image in figure 2).
- Use your own images of different shaped objects (rectangles, triangles, disk-shape objects, ...) and identify the locations and different objects.
- Use objects that allow to estimate an orientation (i.e. rectangle, domino-stone) and use this information during your automated 3D-model creation.
- Instead of counting the objects count the dots that appear on the domino-stones in images like the one shown in figure 2.

Note: If you work on an extension-task or have changed the original task this counts as a full substitute for the basic task.

2.4 Example Images (extended difficulty)

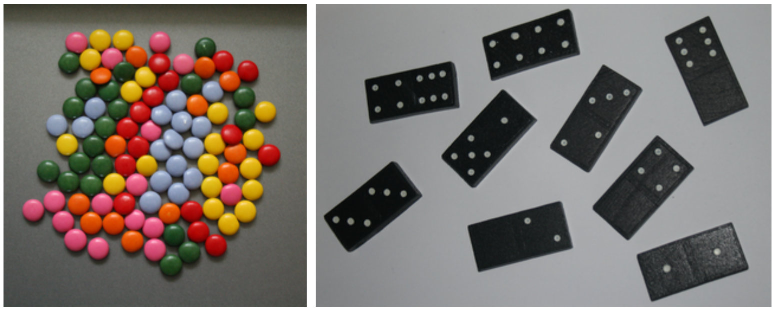

Figure 2: *Example Images for an extended task*

Does your program still work on images like the one shown on the left side of figure 2?