The today's assignment is about real-time multimodal template matching: the goal is to manually mark a certain fixed size area in the image, to learn this area and to detect it in following images if the camera pose does not change significantly. For this homework you are allowed to use OpenCV. For simplicity we suggest that you work on the gray value image instead of the color image.

1. Build an image pyramid of the gray value input image with 4 image levels. For each level you have to downscale the image by a factor of 2. For that you first have to smooth the input image with a Gaussian kernel and then downscale it. If you use OpenCV you can directly use cvPyrDown(). You also have to build an image pyramid with the depth images.
2. Compute the Sobel Filter on each image of the gray value pyramid in x and y direction (if you use opencv you can use cvSobel()). Compute the normal estimation on each image of the depth image pyramid (use the functions we attached on the website).
3. Learn your template: when you click in your image (mark your selection with a rectangle) you have to memorize the relative location (to the selection center) and the gradient/normal values of a certain number n of features (let n=300: 150 gradient features and 150 normal features). IMPORTANT: make sure that your normal features are well distributed and that your gradient features only come from gradients with high magnitude. You also have to take into account that you have to memorize a template on each pyramid level (so the template size becomes smaller by the factor of 2 on each pyramid level). Also use less features on the higher pyramid levels (because you have smaller templates).
4. The matching score is the sum of the sum of dot products of normalized gradients and the sum of dot products of normalized 3D normals. For the better understanding of the general matching schema, please read the following (easy) paper: http://ias.in.tum.de/people/steger/publications/2002/ISPRS-Comm-III-Steger.pdf
   Note: for matching you always have to take the relative location of the feature to the object center into account.
5. Matching strategy: first match your template on the highest pyramid level (smallest image). You have to do it in a dense way (so each image pixel could be the template center). If you have an image location where the matching score is higher than a threshold, propagate this hypothesis down to the next pyramid level. Search only in the local neighborhood of this propagated hypothesis for a possible match (don't forget to multiply the x and y coordinate of the location by a factor of 2 when you propagate it to the next level). If a matching exceeds a certain threshold at the lowest pyramid level we take it as a true match. Display this match in the image. Note, the thresholds are usually different for the different pyramid levels.
6. Tip: First implement the matching for one modality (e.g. 2D gradients). If that works, add the other modality. Use r=7 in the surface normal method.