Responses to reviewer comments for IEEE RA-L 21-2276, version 1

Arkadeep Narayan Chaudhury*, Tim Man, Wenzhen Yuan and Christopher G. Atkeson

January 15, 2022

Dear Prof. Popa and the Reviewers,

Thank you for the very thorough review and thoughtful comments on our manuscript. We are enclosing our clarifications (answers in line) to the reviewer questions below with pointers to actual revisions in the manuscript in **bold text** in response to the reviewer's comments. The changes resulting in the paper due to the reviewer comments are in blue. Some questions have also resulted in addition of more material/ results to the paper website. They can be accessed here: `https://arkadeepnc.github.io/projects/collocated_vision_touch/index.html`. Please let us know if you have any further questions on this modified version of the manuscript.

Thanks and best regards,
The authors of RA-L 21-2276

1 Associate Editor Comments

Technical:

1. Introduction / Paragraph 2 / bullets 3 and 4: Gross object pose errors are reported at less than 1 cm in translation and 2° where the fine localization error reports improvements to the pose estimate to an uncertainty of ±1.5 mm and ±0.5°. Reformat the error descriptions for easier comparison. Personally, I would choose the 2nd description format using uncertainty of ± distance/angle. And adjust numbers accordingly if need be. I would interpret the first as an uncertainty of ±10 mm and ±2°.
   Thanks for pointing it out. **Changes have been made in introduction / Paragraph 2 / bullets 3 and 4.**

*All authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh PA. Email: arkadeepnc@cmu.edu

2. The paper lacks good experimental setup description. What is the ground truth used to arrive at quantification of the improvements in localizing objects using this method? Ideally, such a benchmark would measure improved localization externally to the system under test. For example, a motion capture system can track 3D position to with sub-millimeter precision and multiple targets can be used to resolve 6DOF pose.

   As described in the work, we use the pose of the object calculated with respect to the robot as the ground truth and we repeat our localization experiments to measure the uncertainty in the pose estimation. This protocol is common in the literature (see e.g. [7,8]) where the authors demonstrate the repeatability of the pose estimation algorithm as a measure of performance. **We have modified parts of Section IV-4 to answer this question. Additionally, we have added section IV-5 where we demonstrate the performance of our localization against random ground truth pose perturbations.**

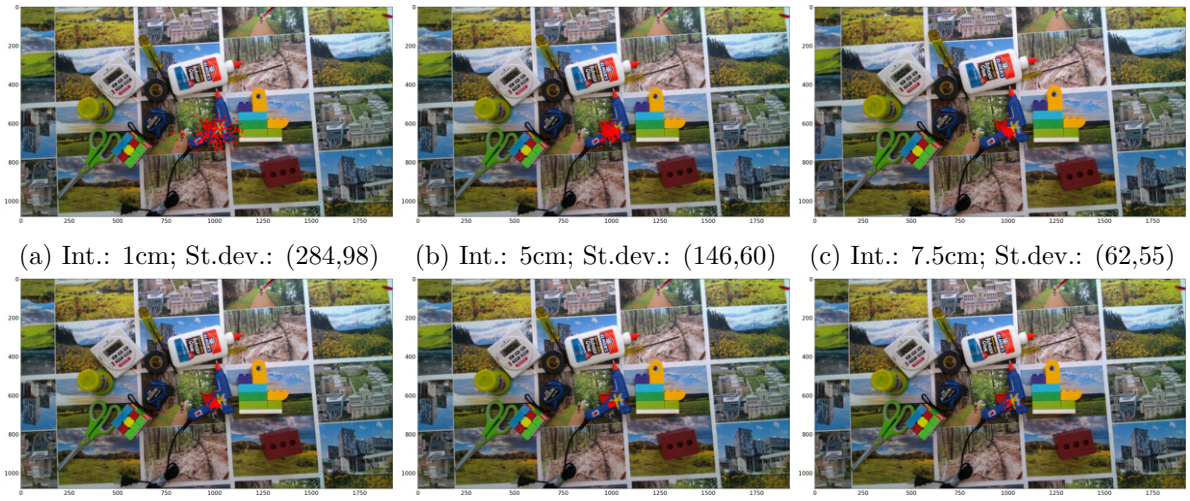
3. Present the experimental data for each object tested in table form and report on performance statistics over a multiple number of trials number of trials (for example, 30 trials per object each with a different start pose).

   Unfortunately, given the page limits, we will not be able to fit a table in the paper, however, we present this table on the accompanying paper website. We repeated the localization of the 6 objects above, for 9 times each across 3 different portions of the robot workspace and found that using tactile sensing the localization uncertainties were brought down to of ± 1.5mm in translation and 0.5° in rotation from about 2° in rotation, and ± 1 cm in translation, at a distance of about 30 cm from the camera while using only vision to localize them. **We have also included a new section (Sec IV-5) where we present more quantitative localization experiments with random ground truth pose perturbations. We have a table of quantitative results of our experiments available on the accompanying website here.**

4. Include experimental data leading to the conclusion that "using optic flow from hand mounted cameras had to be integrated across about 10cm of camera travel to provide useful heading estimates"

   For our robot setup, where a maximum of 1m downward travel was possible, averaging the trajectory over 10 cm was empirically chosen between averaging over 1cm, 5 cm, 7.5cm, 10cm, 15cm and 20cm, as it provided the maximum number of corrections possible during the robot's motion towards the goal while yielding reasonable trajectory error estimates due to a larger averaging window. **We have added this explanation at the end of section III-B-1 to answer this question.** We provide more evidence below in figure 1.

   Unfortunately, these figures cannot be included in the paper due to space constraints but are already explained on the supplementary video and on the paper website.



(a) Int.: 1cm; St.dev.: (284,98) (b) Int.: 5cm; St.dev.: (146,60) (c) Int.: 7.5cm; St.dev.: (62,55)

(d) Int.: 10cm; St.dev.: (19,27) (e) Int.: 15cm; St.dev.: (21,20) (f) Int.: 20cm; St.dev.: (17,15)

Figure 1: In this experiment, we move the robot vertically down by 65 cm to a goal location slightly below the yellow cross mark on the handle of the glue gun. There are no errors in the goal location being tracked for this case. This experiment is repeated 10 times. The red dots are the predictions of potential point of contact (calculated as the instantaneous POE). We note that the predictions are centered about the actual point of contact – a point slightly below the yellow cross mark on the glue gun. We report 6 cases where we predict the potential point of contact by looking at various intervals of the trajectory. We report the interval lengths (in cm) and the standard deviation in predicting the point of contact (in pixels) as the labels of the figures. We note that as we increase the length of the interval, the standard deviation of the prediction decreases (as seen through "clumping" of the predicted potential points of contact), but the number of possible predictions decreases (as seen through fewer number of red dots with increasing interval sizes). This leads us to conclude that the averages across larger temporal (and spatial) windows produce smoother and more stable error signals (or correction signals in the case of visual servoing). For our use case, averaging across 10 cm intervals provided us with "enough" number of correction signals while having reasonably low variance.

Conclusion/Next steps:

This seems to be a proof-of-concept implementation of this method and lacks a clear understanding of the intended application space. If the intention is to apply this method to pick and place applications, how difficult and what will be the steps needed to implement this with an end-effector and attached to the arm. Things will get crowded with two camera systems and the rather bulky tactile sensors attached to even a simple parallel gripper. It would have been nice to see and implementation. Is this planned for future work?
Yes, this work was indeed a first step to see how a hand mounted collocated tactile sensor and camera setup can be used for visual servoing and contact localization in a robot's workspace. We did not focus on a particular robot task that we wanted to solve (e.g. pick and place as you have pointed out) so we did not emphasize on designing an accompanying gripper setup along with the presented sensor setup that can be maneuvered in the robot's workspace. We did notice that things are getting crowded as is given the size of the sensors. We are currently working on integrating smaller high quality cellphone cameras to bring down the overall footprint of the sensor setup so that it can be efficiently collocated with a standard gripper. **We have modified the final paragraph of section V to reflect this.**

Editorial:

1. Format figure test using a different font (smaller) to help decipher body text from figure text.
   Thanks for suggesting this. **Changes have been made in all the figure titles on the manuscript.**

2. UR5E $->$ Universal Robots UR5E
   Thanks. **The changed text now reads as** Universal Robots UR5E manipulator.

2 Reviews of Reviewer 2, Reviewer ID 171465

General questions

1. I want to emphasize that according to the authors the paper does not focus on global (gross) object localization and assumes a bounding box of the object is already available. However, it wasn't clear how that in practice affected the computations and the difficulty of the approach. How much did vision pose estimations had to improve the original priors?
   Thanks for the great questions. For some more details, please also see the answer to this question as well (rev. 6, question item 3).
   For the first question: the computational load for generating the initial pose estimates are

not very high. We purposefully trivialize it by using a black background so that identifying the object and segmenting the edges could be done using simple template matching, color thresholding and Canny edge detection. **We have added a few lines to the preamble of section IV to explain this.**

For the second question: The vision prior started with just the information of which part is facing up. This provided good initial priors about the pitch and the roll of the object with respect to the robot (as they were lying on a flat table). **In section III-C, we describe how the vision based localization pipeline takes this information and generates pose estimates in the frame of the GelSight.**

2. Another important question that I couldn't resolve, how are goal points for contact selected? If manually, it could be easy to bias results toward good object regions where tactile has an easier time to localize the object. It is really important to clarify how these goal points are selected to ensure the validity of the results.

   Yes, the location of contact is important in guaranteeing that our localization pipeline will succeed. In section IV-4, we manually choose good contact points to demonstrate the accuracy of the pipeline. **To address this comment, we have added section IV-5 where we report additional experiments on recovering poses with randomly selected contacts.**

3. One of the main limitations that I see is that the system does not seem capable to perform any complex manipulation. There is only one contact point available (the tactile sensor) which is surrounded by bulky cameras. Therefore, is this a viable implementation for actual robotic manipulation tasks?

   Yes, you are correct. We answer the same question here (section 1) while addressing the comments from the associate editor. In short, our goal for this work was to demonstrate a method to visually servo and localize objects in a robot's workspace using a suite of vision based sensors of different capabilities. Using an improved version of this sensor suite along with a standard robot gripper is future work. **We have edited the final paragraph of section V to address this comment.**

More detailed questions

1. In related work, it would be important to compare your approach to tactile localization with the different methods explained to provide better context.

   Thanks for pointing this out. **We have edited the end of section I of the manuscript to address your comments.** .

2. In methods "We use a robust algorithm to detect the POE" it is unclear what this algorithm is.

Unfortunately due to space constraints, we could not include the algorithm to the manuscript. We are describing it below for your reference, and it can be found in the accompanying paper website here (link). The main idea behind the algorithm is to robustly identify the minima of the optical flow surface (as shown in fig. 3a in the draft). To do this, we break the image into small overlapping tiles, calculate the optical flows for these tiles and locally fit paraboloids to the square of the magnitude of the optical flow at each of the tiles. We then use the fit paraboloids per tile to vote for the tile which best approximates the minima of the optical flow of the entire image. Dividing the image into tiles and voting across the tiles makes the algorithm robust to variability in the image due to objects coming in or out of the field of view of the camera.

3. The paper would benefit from a more clear explanation of why using robot kinematics + accurate measuring of the robot mount and camera poses wouldn't be better than visual servoing and flow estimation.
   Thanks for pointing it out. **We have edited the visual servo section (III-B-2) in response.**

4. In the subsection "Visual servoing to a goal:" There is not enough information to fully understand what is the approach followed to extend POE. Are you just computing the difference between the goal pose and the point the robot is heading towards? Or using a more advanced control method?
   We have re-written section III-B-2 of the manuscript hoping to make it clearer.

5. In 3.C it is not clear how you take into account this: "we put aside the gross object localization and recognition problems in order to focus on fine localization, so we assume a vision system has already located the object, created a bounding box, and recognized the object by creating or selecting an appropriate 3D model that we want to register the actual object to." Does that mean that you already have a good approximation of the object orientation and more or less of its pose? If so, how good is that approximation?
   We answer this question here (Reviewer 2 – General questions item 1). Please also see (reviewer 6 question item 3) for some additional details. **We have added a few lines to the beginning of section IV to address this.**

6. I did not understand that "The x and y directions along the image plane are initialized using the camera intrinsics and the initial value of z and θ." Do you also take into account the pixel coordinates of the goal point? Also, how is the goal point selected, it wasn't clear in the paper?
   We have edited section III-C to address your comment. Here is a more detailed answer from the accompanying webpage: For localization we do not need the pixel coordinate or world coordinate of the goal point. As the robot approaches the object the cameras generate a stream of data. We segment out the object from an instance (one image or image aligned

depth frame) of the sensor data, use the centroid of the segmented object edge pixels (obtained by a Canny edge detector) and a rough estimate of the projection depth to convert the centroid of the edge pixels to a world point using the camera projection matrix. This is our starting estimate of the object center in the robot's frame. Next, we use the prior that the object is lying on a flat plane to set initial estimates of roll and pitch angles (essentially setting them to 0) and look at the direction of the largest spread of the edge pixels to determine the yaw. **We address the selection of the goal point here (item 2).**

7. At which point of a trajectory do you do object pose estimation from vision? Do you combine pose estimates from multiple times?
   We maintain a coarse pose estimate using equation (1) in the manuscript and refine it using equation (2) at a point where the object appears the largest in the image. This was about 25 cm away from the object for the larger objects and 10 cm away from the objects for the smaller ones. **We have edited the end of Section III-C to address this.**

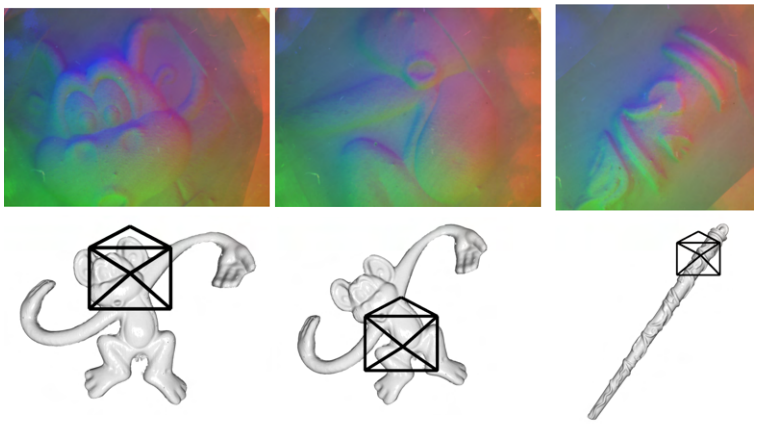
8. In 3.D I am not sure I understood this "We note here that this is not a simulation of the GelSight sensor through our renderer – the GelSight should not be able to". Does it mean that the rendered N_s and D_s will contain more information than the ones provided by the actual tactile sensor?
   The rendered GelSight basically simulates a camera which can "see" the object's surface normals and depth with respect to itself. This is not a true simulation of the sensor as the actual GelSight can only sense objects touching it i.e. $\sim$ 25mm away from the camera. We had to relax that for our gradient descent steps to work. We also discuss some issues related to this here (item 3). **We have edited section III-D in response to this comment.**

9. The results section would benefit from some introduction that explains what are subsections 1-4. Are these "potential problems" that your solution tackled? Or different cases that you evaluated?
   These are different cases we evaluated. Thanks for pointing it out. **We have re-written the beginning of the results section (section IV) to address this.**

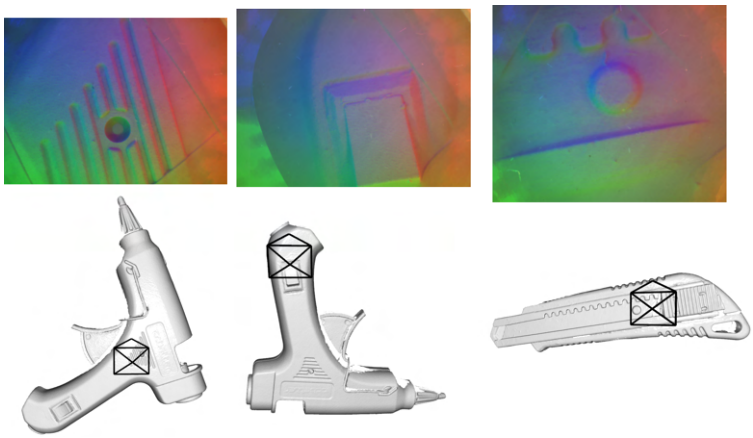
10. In results "For the objects described in figs. 1, 4 and 5, we could use the strategy [...] " it is unclear if this is actually the strategy that you end up using. Maybe the "could" is just a typo?
    Thanks for pointing out the typo. **We have corrected it**. Echoing it below: "... we used this strategy to transfer the pose estimates obtained ..."

11. Typo: "can then refined" $\rightarrow$ can then be refined
    Thanks for pointing out the typo. **Echoing the corrected version below**: " ... can then be refined with the tactile ..."

12. In results, the paper would benefit from having a table where each object is shown and the



(a) Tactile maps and corresponding matches for small flat objects



(b) Tactile maps and corresponding matches for bigger objects

Figure 2: Tactile maps obtained from the tactile images vs. the actual model at convergence of our pipeline

errors obtain for vision, tactile, vision+tactile and other methods tried. Otherwise, it is hard to assess the difficulty of localizing such objects. I would also include some images of the tactile maps obtained from the tactile images vs. the actual model.

Thanks for pointing this out. Please also see our response to this question (item 1) from the associate editor. Please also find some images relating tactile maps obtained from the tactile images vs. the actual model at convergence of our pipeline below (figure 2). **Unfortunately, due to space constraints, we will not be able to add all these figures to the manuscript, but they are displayed and described on the paper website.**

13. "Than the the experiments" → typo
Thanks for pointing out the typo. **We have corrected it.**

14. I didn't understand this: "Extending the GelSight's visual capabilities to the scale of the robot workspace is also future work." in the discussion section.
    Thanks for pointing it out. **We have edited the text to clarify this**

15. What do you mean in the conclusion when saying "enables tactile localization to work at all,"? Reference [23] can provide good pose estimates from just tactile so I would be careful to say that without vision tactile localization isn't possible. This reflection should also be considered for results subsection 6.
    We believe that the confusion is arising from a typo in the first sentence of the conclusion section. **To address it we have re-written the conclusion section. Our discussion in section V acknowledges the references [5 and 6] which use tactile signals exclusively to localize contact, and why the methods described in [5 and 6] are out of scope of the current work.**

3 Reviews of reviewer 6, Reviewer ID 174871

The paper describes a framework that integrates a wide field camera, a Lidar camera, and a Gelsight tactile sensor to localize an object first roughly via the vision sensors, and then refine such localization via contact with the tactile sensor. The paper is well structured and easy to follow. However, some of the descriptions are vague, especially regarding the results section.

Major points:

1. Lucas-Kanade generally refers to sparse optical flow computation [1]. The authors mention that they use it for dense optical flow. Are you then using such an algorithm to compute the motion considering all the pixels as the features to track? This point should be clarified.
   Sorry we should have been more clear about this – we are using a pyramidal implementation of the sparse optical flow which is "densified" to generate optical flow for all pixels using nearest neighbor interpolation. This is the function we use link, which first calculates the sparse optical flow across image pyramids using the Shi Tomasi features [2] (`goodFeaturesToTrack`) across 3 pyramidal levels of our image and then interpolates the flow calculated at the tracked features to all the other pixels using nearest neighbor interpolation. We did not describe this in detail as we felt it was out of scope of the presented the work. **To address your question, we have included a pointer to the function** (**same link as above**) **we use in the references** .

2. More details should be provided about the visual servoing: do you use information from both the cameras? How is the correction computed? Are the cameras parallel to each other, or

are they directed towards the gelsight central axis.

We have re-written section III-B-2 of the manuscript hoping to make it clearer. To answer your question in short, the cameras are parallel to each other, but this is not a hard requirement as we can get the camera-to-hand transforms through calibration. Also, we use correction predictions from both of the cameras. **This is now noted in the title of fig 3.**

3. How is the object rendered through the GelSight's viewport? Also, could you better explain the 25 mm limitation for the GelSight measurements?

   We modify a differentiable renderer called `DIRT`[4] to render our images. We note it in section III-C as "we use a modified version of the DIRT renderer from .." in the draft. **We have modified the current text in section III-D to explain this.** and we have added a pointer to the paper in section III-D as well. Thanks for pointing this out.

   In practice, we simulate the GelSight's parameters in `DIRT`, including the camera parameters and the sensor thickness. As the GelSight is physically $\sim$ 25mm thick (distance of the sensing surface from the camera), it cannot sense any contact closer than that (that is physically inside the sensor) or farther than that (object not touching the sensor). If we implement this exact behavior in simulation the gradient descent steps in eq.(4) would fail at it would have no gradient signals from the initial pose (as measured by the ensemble of cameras).

4. In part 4 of the results, I am confused about "the tactile sensing actually increased the localization errors in the directions orthogonal...". Are the ambiguous features detected by the tactile sensor increasing the errors compared to the localization provided by the cameras? This seems an undesired behavior: if information is ambiguous, then wouldn't it be better to keep the original localization from the images?

   We pointed it out to demonstrate a limitation of the tactile localization method we presented in the work. The localization uncertainties increase in the direction of the repeated features (the increased uncertainties are at the scale of the repeated features) but the overall uncertainty of the pose estimation decreases in the direction of contact and in the angular directions. However, even with the increased uncertainty (which is in the order of mm) we are better off considering the tactile measurements as the localization (at contact) errors using only vision is much more (in the order of 2-4 cm). **We have added a line at the end of this section (section IV-.4) to clarify this.**

5. In part 4 of the results, I did not understand the subset of experiments repeated by employing the technique described in III-B.2. What is the difference with the experiments described a few lines above?

   The only difference is that in the previous part of the experiment, we moved the robot vertically before contact, but in this case, we used trajectory correction (as described in sec III-B-2) before touching the object, so the descent was not necessarily vertical and was

determined using the visual servoing algorithm we described in sec III-B-2. **We have edited the text of IV-4 to clarify this.**

6. Part 6 of the results is not convincing. You are using a specific technique to localize with touch only. This technique seems to not work properly, but it is hard to draw general conclusions from this. Another strategy (using only touch data) could work better, so what is the point that the authors are conveying with this section?
   In Section IV-6 we presented our attempts to localize the contact using only touch and traditional methods. **The original section IV-6 was confusing, so we have removed it and edited the section on localization using vision only (previously IV-5, currently IV-6) to describe our experiments on localizing with touch only.**
   We reported the results of conventional feature matching in the image space (between tactile 'images' and object images) and 3D space (point cloud derived from tactile images and object mesh). We tried all the commonly known 2D and 3D feature matching techniques and reported that they did not work for our case. This was our only conclusion for this part. We tried to provide some justifications for why these did not work in the latter part of the section. The only other strategy for localizing high resolution tactile images that we know of are the works of Bauza et al.([5,6]) which use learned pipelines to do so. We described why that is out of the scope of the current work in the Discussions (Sec. V).

Minor points:

1. At the end of the related work, it feels that this is missing a concluding paragraph to explain how the proposed work improves the state of the art approaches mentioned in the section.
   Thanks for pointing this out. **We have edited the end of section I of the manuscript to address your comments.**

2. You assume that the object is not moving, is this a reasonable assumption once you make contact with it?
   Yes we agree that it is a restrictive assumption especially in the context of localizing using touch. In the results (Section IV-4 and IV-5) we demonstrate the performance of our algorithm through it's repeatability, therefore, fixing the objects were necessary. **We have edited section IV-4 to address this concern.** However, in our initial trials (not reported in this work) with the object not rigidly fixed to the table, we moved the robot slowly to minimize wear and tear on the GelSight sensor, so the motion induced was small, usually less than 5mm. A low velocity at contact was necessary to prolong the life of the tactile sensor surface.

3. Is a black surface placed below each of the objects (see Fig. 4a-c)? Is that needed for the algorithm to work properly?
   The black background for the objects trivializes the object recognition, segmentation and

edge detection problems. We mentioned this at the beginning of the work in the line " For this paper we put aside the gross object localization and recognition problems...". We solve them using standard template matching, color thresholding and Canny edge detection respectively. **We have added a few lines to the beginning of section IV to address this.** Alternatively, we one can use an off the shelf segmentation network (e.g. MaskRCNN[3]) and train on the objects at hand and generate segmentation masks at run time. These masks can then be used on the images to extract the object edges exclusively from the image.

4. Fig. 5 and 6 seem redundant. Removing one figure could help making space for the additional clarifications.
   Thanks for your suggestion. We believe that the figures 5 and 6 convey qualitative results on how our system can be used to localize objects of very different scales, so removing one of them would make it harder to convey the results we obtained. **To make space for the additional content introduced to address the reviewers' questions, we have removed the section IV.6 (Localizing contact using touch only).**

4 References

1. https://docs.opencv.org/4.5.4/d4/dee/tutorial_optical_flow.html
2. Jianbo Shi and Carlo Tomasi. Good features to track. In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on, pages 593–600. IEEE, 1994.
3. https://github.com/facebookresearch/Detectron
4. P. Henderson and V. Ferrari, "Learning single-image 3D reconstruction by generative modelling of shape, pose and shading,"International Journal of Computer Vision, pp. 1–20, 2019
5. M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.
6. M. Bauza, E. Valls, B. Lim, T. Sechopoulos, and A. Rodriguez, "Tactile object pose estimation from the first touch with geometric contact rendering," arXiv preprint arXiv:2012.05205, 2020.
7. M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, andR. Chellappa, "Fast object localization and pose estimation in heavyclutter for robotic bin picking,"The International Journal of RoboticsResearch, vol. 31, no. 8, pp. 951–973, 2012.
8. M. Imperoli and A. Pretto, "D2CO: Fast and Robust Registration of3D Textureless Objects Using the Directional Chamfer Distance," inInternational conference on computer vision systems. Springer, 2015