

Using Collocated Vision and Tactile Sensors for Visual Servoing and Localization

Arkadeep Narayan Chaudhury , Timothy Man, Wenzhen Yuan , *Member, IEEE*, and Christopher G. Atkeson

Abstract—Coordinating proximity and tactile imaging by collocating cameras with tactile sensors can 1) provide useful information before contact such as object pose estimates and visually servo a robot to a target with reduced occlusion and higher resolution compared to head-mounted or external depth cameras, 2) simplify the contact point and pose estimation problems and help tactile sensing avoid erroneous matches when a surface does not have significant texture or has repetitive texture with many possible matches, and 3) use tactile imaging to further refine contact point and object pose estimation. We demonstrate our results with objects that have more surface texture than most objects in standard manipulation datasets. We learn that optic flow needs to be integrated over a substantial amount of camera travel to be useful in predicting movement direction. Most importantly, we also learn that state of the art vision algorithms do not do a good job localizing tactile images on object models, unless a reasonable prior can be provided from collocated cameras.

Index Terms—Force and tactile sensing, localization, sensor fusion.

I. INTRODUCTION

THIS work is motivated by FingerVision [1] where the same camera was used for tactile sensing and to view nearby objects through a transparent elastomer. Although FingerVision convinced us of the importance of proximity imaging, it did not provide high resolution images of contact surface texture, and the transparent elastomer blurred proximity imaging, attracted dust, and got scratched and worn so the view of nearby objects was often not as good as we would like. Separating tactile and proximity imaging enables us to get the high resolution of GelSight [2] tactile sensors that produce images of the surface texture (tactile imaging), and better proximity imaging with rigid lenses that don't attract dust as much, are easier to clean and don't scratch or get worn as easily. This paper explores an alternative to using the same camera for tactile and proximity imaging, where a tactile sensor is collocated with a camera for proximity sensing (Fig. 1). Since the tactile sensor we use, a GelSight variant, is also based on a camera, we are actually collocating multiple cameras to provide tactile and proximity imaging.

Cameras that move with a robot hand can have less occlusion and more resolution since they are closer to manipulated objects.

Manuscript received September 9, 2021; accepted January 4, 2022. Date of publication February 1, 2022; date of current version February 10, 2022. This letter was recommended for publication by Associate Editor J. Falco and Editor D. Popa upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation through Award 1717066 and the Toyota Research Institute. (*Corresponding author: Arkadeep Chaudhury.*)

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: arkadeepnc@cmu.edu; tman2@andrew.cmu.edu; yuanwz@cmu.edu; cga@cmu.edu).

Digital Object Identifier 10.1109/LRA.2022.3146565

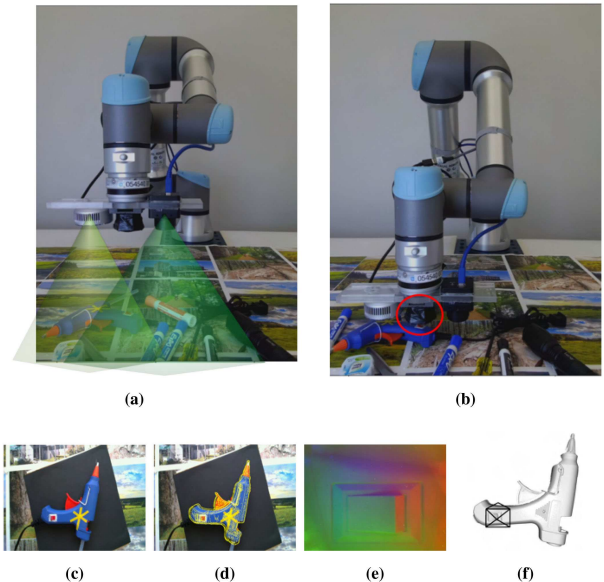


Fig. 1. We demonstrate an approach to integrate sensors with different fields of view to visually servo the robot arm to a predetermined contact point and estimate the pose of a fixed object relative to the sensors at contact. Fig. 1(a) and (b) show our sensor platform. We use 2 cameras with 70° and 100° fields of view, which we collocate with a modified GelSight sensor in the middle (Fig. 1(b)). The cameras are used to visually servo the robot (Fig. 1(c)) and generate a preliminary pose estimate (Fig. 1(d)) while the robot is moving towards the target. At contact, the GelSight data is observed (Fig. 1(e)) and the preliminary pose is then refined to generate the object pose at contact. Fig. 1(f) shows the camera pose superimposed on the mesh model of the object.

Direct measurement of the direction or bearing to an object and its pose relative to the hand can be used to guide the hand to a particular contact location and center the hand with respect to an object. External and head-mounted cameras are often occluded by the robot itself as well as manipulated objects, and need to use stereo, multi-view, or other forms of depth measurement to locate the hand relative to the approach axis and object, which involves subtracting two noisy estimates (typically large numbers) to estimate a smaller quantity, which is usually less accurate than directly measuring the smaller quantity. We have found depth measurements from stereo or time of flight (TOF) cameras usually have low spatial resolution, so getting the camera close to the hand is useful to improve depth resolution.

We divide the problem of contact pose estimation into two parts — The initial phase before contact, when cameras can be used for vision-based servoing to a contact point target as well as estimating a prior for the tactile sensor contact point and object pose estimation, and the contact phase which refines the prior pose estimates. In this paper we assume that 1) the object is fixed,

even during contact, 2) the object is a single rigid body with no articulations, and 3) we have a prior (potentially imperfect) 3-D model of the object (potentially provided by our vision of the current object) so we can express the pose of the object with respect to this model. For this paper 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 3-D model that we want to register the actual object to (e.g. [3]). Our experimental pipeline involves selecting a position goal and then visually servoing to that goal, recording color and depth data from the vision sensors, generating and maintaining pose estimates of the object, and using the estimates along with tactile information received at contact to localize the contact point on the object. Through this work we show that:

- The optic flow, as observed by the hand mounted cameras, can be used to predict the heading direction of the hand using image-based techniques rather than 3-D geometry. In our setup frame-to-frame optic flow was dominated by small changes in orientation of the hand and thus the cameras. Optic flow had to be integrated across about 10 cm of hand travel to be useful.
- A few “mid-course” corrections can correct almost all the error in trajectories.
- Pose errors, when measured only with the cameras are about ± 1.5 cm and $\pm 2^\circ$ in translation and rotation respectively about a vertical axis.
- Given these priors, tactile estimation based on a GelSight sensor further improved the pose estimates to an uncertainty of ± 1.5 mm and $\pm 0.5^\circ$ in translation and rotation respectively in cases distinct tactile signals were available.
- Collocated vision is particularly useful when an object does not have distinctive tactile surface texture, or has repetitive surface texture. We show that using tactile sensing collocated with vision can help disambiguate tactile signals when used for localization.

We provide additional details of our work, tables of results and reference implementations of our algorithms described in the paper here: https://arkadeepnc.github.io/projects/collocated_vision_touch/index.html

II. RELATED WORK

In this section we provide a survey of related work on visual servoing and pose estimation using hand-mounted cameras and tactile sensing. **Recent work** on addressing these issues has used hand-mounted cameras to demonstrate superior performance in classical manipulation tasks such as grasping and bin picking [4]. With the availability of a visual perspective complementary to external (or head mounted) cameras, researchers have diversified the moving cameras to serve as tactile devices ([1], [2]) and have implemented delicate manipulation behaviors (see e.g. [5], [6]). Recent research has also developed tactile sensors and algorithms for estimating contact pose and inferring object from contacts [7], [8], tracking object motion by fusing externally mounted cameras and tactile sensors [9], and transferring information between external cameras and hand-mounted cameras (see e.g. [10]). A closely related work by [11] discusses integration of a visual and tactile measurement through a Bayesian filter.

Vision-based localization and contact prediction: Camera-on-hand or more generally camera-on-mobile-agent arrangements have been investigated by several researchers to pursue diverse goals such as visual servoing to a workspace goal (e.g. [12]), collision avoidance systems on miniature aerial vehicles (e.g. [13]), and how flying insects, birds, and rapidly moving animals perceive motion [14]. Literature on quantitative analysis of looming¹ (e.g. [15], [16]) is of particular interest to us as we try to identify an area in the image space corresponding to the direction of heading of the robot at any particular moment. Recent research on optical expansion (e.g. [17]) is focussed on supervised learning, instead of hand crafted functions (e.g. [15], [16]) to compute dense scene flow from optic flow to identify relative motion of objects and the agents, and has been demonstrated to exhibit state of the art performance in identifying objects heading towards the agent. In the current work we build upon research on optic flow for scene understanding to identify an area in the robot’s visual field corresponding to the physical point in the workspace where the robot is currently headed.

Tactile localization and contact estimation: Vision sensors can have a wide “field of view” and are good for making large scale models. A tactile sensor has a much smaller measurement area (or field of view) and can potentially capture minute details of the surface it interacts with. Early research on tactile sensing leveraged this capability, even with a seemingly low resolution tactile sensor (Weiss Robotics DSA9205), to demonstrate object recognition using image feature descriptors [18] and, recognize and localize an articulated object through a sequence of touches (see e.g. [19]). With the introduction of camera-based higher resolution tactile sensors, most notably the GelSight (see [2]) and its derivative GelSlim [20], investigations on tactile object recognition and localization have made significant progress in tactile sensing driven perception. [21] described tactile localization using the GelSight sensor using conventional feature based image alignment. [22] integrated the GelSlim with a gripper and interfaced it with a model based controller to successfully perform re-grasps of a cable. More recently, [23] demonstrated tactile localization and shape reconstruction using a GelSlim sensor, where the authors trained neural networks to generate height maps with ground truth data obtained from robot experiments with the sensor and known objects. The trained network was then used to generate height maps of the surface of an object and the height maps were registered to reconstruct the object surface. This work was extended by [24] where the authors used a renderer to generate and cache a large number of possible tactile signals of objects from a data set touching a tactile sensor (GelSlim in this case) in different orientations. An actual tactile signal corresponding to a particular object at a particular pose was then compared with the cache to retrieve candidate object and pose pairs and the network demonstrated in [23] was then used to generate a height map which was registered with the candidate object at the candidate pose to localize contact. [25] introduced a new time of flight based tactile sensor, the soft-bubble, and demonstrate object identification using learned embeddings and object localization by registering the tactile signal (obtained as a point cloud by their sensor) with the retrieved geometric model of the recognized object. In the current work we build upon literature on contact localization using high resolution

¹Looming or visual looming is defined as the phenomena of an object getting bigger in the visual field as the relative distance between the observer and the object decreases.

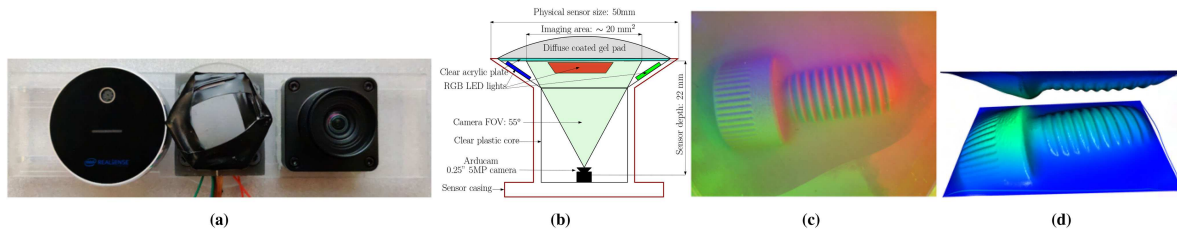


Fig. 2. Fig. 2(a) shows our sensor platform which is attached to the robot manipulator. The sensors, from left to right, are the Intel RealSense L515 LiDAR camera, our modified version of the GelSight and an RGB camera. The RGB camera and the RealSense are at a distance of 6 cm each from the camera to the left and right respectively. Fig. 2(b) shows the schematic of our modified version of the GelSight, Fig. 2(c) is an instance of the raw data collected by our GelSight when pressed against an 18 mm long 6 mm diameter bolt with 1 mm pitch. Through the image, we note the physical scope of the sensor is almost entirely covered by the 18 mm long bolt. We process the raw GelSight data to yield metric depth and normal maps. The depth of the reconstructed surface is rendered as a shaded point cloud in 2 views in Fig. 2(d).

tactile sensors, visual localization and hand mounted cameras to demonstrate a suite of collocated vision based sensors that can be used to visually servo a robot to touch and localize objects in the workspace.

III. METHODS

In this section we describe our sensor platform (A) and algorithms to estimate where the robot hand will go (B1), visually servo the robot to a target contact point (B2), estimate the pose of the target object (C), and combine vision and tactile information to estimate both the contact point and refine the object pose estimate (D).

A. Sensor Platform

In this work we collocate cameras with a camera-based tactile sensor by putting the cameras and the tactile sensor in close physical proximity while operating them independently. The sensor platform consists of a GelSight tactile sensor in the middle (co-incident with the robot wrist's axis) (Figs. 1 and 2), with a camera on either side. We modify the GelSight as described by [26] in the physical sensor form factor introduced by [2]. We use a LIDAR-based RGBD sensor (Intel RealSense L515) with a 70° field of view to provide depth (Fig. 1(a) and 2(a) left) and color images of the workspace, and a USB camera (a Sony IMX 291 sensor) with a wider 100° field of view lens (Figs. 1(a) and 2(a) right). We chose these cameras partly because they were of comparable size to our GelSight sensor, thus ruling out most other popular RGBD cameras. Also, the USB camera yielded high resolution images which were better than the color channel of our 3-D sensor and helped us localize small objects. Further details on the sensors can be found in Fig. 2.

B. Visual Servoing With Hand Mounted Cameras

In this section we describe a method to estimate the robot hand's heading direction in the workspace and then we describe how to servo to a goal using that estimate.

1) *Optical Point of Expansion From in Hand Cameras*: We process the scene to identify the location of a 3-D point corresponding to the heading direction of the robot in the image space. To achieve this, we calculate the optical flow between the consecutive frames obtained by the hand mounted cameras and identify the region in the image from which the optic flow seems to be emerging (i.e., we look for a portion of the scene which has zero translation) as the camera moves towards the scene. Assuming that the world scene is relatively flat (object depth

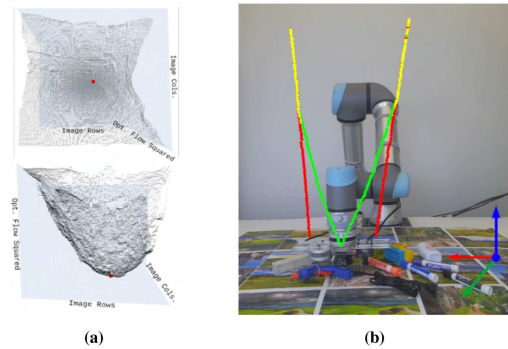


Fig. 3. Fig. 3(a) shows the surface of the squared magnitude of the optical flow between a consecutive frame pair in 2 views. We note that this surface assumes a parabolic shape. The red dot is the minima of the optical flow surface as identified by our algorithm. Fig. 3(b) demonstrates the usage of our algorithm to correct trajectory errors using both cameras as shown in Fig. 2(a). The X, Y and Z axes are marked in red, green and blue in Fig. 3(b) on the bottom right of the figure.

$\ll$ projection depth), the square of the magnitude of the optic flow at each pixel is roughly distributed as a parabolic surface (see Fig. 3(a)). We calculate the motion field (per pixel optic flow magnitude and direction) between 2 consecutive frames using the OpenCV implementation [27] of the Lucas-Kanade dense optical flow, which solves for per pixel motions (along horizontal and vertical directions) over the full image. We also tested the Farenbäck optical flow [28] and the Brox optical flow [29], and found that the dense Lucas-Kanade optical flow performs slightly better in computation speed and produces smoother optic flow fields. The optical point of expansion (POE) is obtained as the minima of the surface representing the square of the magnitude of the optic flow. We use a robust algorithm to detect the POE².

The heading direction estimated by the optical flow between consecutive frames was too noisy to yield meaningful heading estimates. To address this, we looked at the trajectory correction estimates for each of the on hand cameras and found their prediction to be very closely correlated (almost equally incorrect or equally correct), which led us to rule out camera noise and incorrect robot kinematics including the camera mounts as the cause of the noise. We concluded that the errors were being caused by small unmeasured rotations of the robot wrists (play or backlash). Numerical modeling showed that the expected amount of play in orientation led to optic flow values comparable

²Please see the paper webpage for more details.

to the “noisy” shifts in the POE. We also noted that the noise in the predicted trajectory errors was centered about zero. We averaged POE shifts over a robot travel of at least about 10 cm to get usable POE estimates. For our robot setup, where a maximum of 1 m downward travel was possible, we observed empirically that about 10 cm intervals provided useful trajectory corrections and provided the opportunity for several corrections as the robot moved to the target

2) *Correcting Trajectory Errors Using Pixel Space Errors:* In the previous section we described a method to identify the image coordinates of the point in the workspace to which the robot is headed. In this section we address the problem of correcting trajectory errors using those predictions. This can be useful when the robot’s trajectory needs correction and the only information about the updated goal is available in pixel space – possibly from a hand mounted RGB camera which detected a movement of the target. To reach a pre-planned goal in the robot’s workspace, the robot, almost always, needs to move towards that goal, hence the image of the goal point and the observed POE should be very close. Errors in the pixel space between these two, indicates an error in the trajectory being followed by the robot, which we aim to correct. To do this, we note that as the cameras are registered to the robot, we can compute the Jacobian which correlates the pixel space error to the task space error and we can invert the Jacobian (up to an arbitrary scale in the projection direction) to obtain trajectory space corrections from pixel space errors. In Fig. 3(b), on the two sides of the robot, we show 2 trajectories where the robot travels about 1 m (vertically) from the start to the final contact position, and each of the trajectories need a correction of 10 cm errors in the X and Y directions in the robot workspace. We show the initial portions of the trajectories in yellow, the planned portions in red, and the corrected portions in green. The final accuracy achieved was within 5 mm of the target.

C. Pose Estimation Through Vision

In this section we discuss visual localization of an object with known 3-D geometry. At least in the early stages of approach, the hand-mounted cameras can see all or large portions of the target object, and standard image based registration methods ([30], [31]) can be used to estimate the object’s pose relative to the robot hand. This is quite different from the situation with tactile sensing where only a small part of the object is visualized. As is common in the whole-object-visible camera-based pose estimation literature (see e.g. [30], [31] and [32]), we decompose pose estimation into two parts – coarse pose estimation by aligning the centroids and edge moments, and finer alignment using a distance-based cost applied densely.

As a prerequisite for this part we need the edge pixels of the object in the image and for this we use either the depth edges on the object if available or else use the image gradient edges of the object – we use a Canny edge finder for this purpose. Let us denote this binary edge image as $\mathbf{I}_S$. Next, we formulate an optimization problem to identify an object pose that produces the most similar edge distribution to $\mathbf{I}_S$.

To do this, we generate an initial guess for θ (the angle of rotation about the camera projection axis) for the object from the principal components of the camera image and if an aligned depth map is available, we use its mean to initialize the z component (distance between the camera and object along the camera projection axis), or else we initialize the z depth arbitrarily.

The x and y directions along the image plane are initialized by back-projecting the centroid of the edge pixels using the camera intrinsics and the initial value of z . With the initial guess $\omega = [x, y, z, 0, 0, \theta]$, we use a differentiable renderer (we use a modified version of the DIRT renderer from [33]) to render the mesh model of the object, extract the corresponding edge image $\mathbf{I}_R(\omega)$ and solve the following minimization to obtain a rough pose estimate from the camera image. For each $\mathbf{I}_R$, we identify the image edge pixels $\mathbf{p}_R^i$ and $\mathbf{p}_S$ for $\mathbf{I}_R$ and $\mathbf{I}_S$ respectively and minimize the following sparse edge matching cost $\mathbb{E}_{cs}$ in (1)

$$\mathbb{E}_{cs}(\omega) = \gamma [\|\bar{\mathbf{p}}_R(\omega) - \bar{\mathbf{p}}_S\|_2] + (1 - \gamma) \langle \mathcal{V}(\mathbf{p}_R(\omega) - \bar{\mathbf{p}}_R) \cdot \mathcal{V}(\mathbf{p}_S - \bar{\mathbf{p}}_S) \rangle \quad (1)$$

where, $\mathcal{V}(\mathbf{p})$ is the direction of largest variance of the mean centered point set $\mathbf{p} \in \mathbb{R}^2$, given by the eigenvector corresponding to the maximum eigenvalue, $\langle \cdot, \cdot \rangle$ is the dot product between vectors and γ is a weighting factor. Minimizing (1) aligns the centroid and approximately recovers the angle of rotation along the camera projection axis.

The expression for $\mathbb{E}_{cs}$ does not admit automatic gradient calculation due to the non-differentiable selection of pixel indices to obtain $\mathbf{p}_R$ from $\mathbf{I}_R$, therefore, we obtain a finite difference gradient using central differences. We minimize $\mathbb{E}_{cs} \forall \theta_i$ and obtain the candidate pose parameters $\hat{\omega} = \{X_{cs}, Y_{cs}, Z_{cs}, \theta_{cs}\}$ in the camera coordinate frame, corresponding to the minimum E_{cs} . From Fig. 4(b) we note that minimization of $\mathbb{E}_{cs}$ is not expected to solve for the projection depth, as it only recovers the orientation of the camera and not the projection depth. In the next part, we solve the projection depth by minimizing a modified version of the dense differentiable cost using the directional chamfer matching energy as discussed in [30] and [31]

$$\mathbb{E}_{cm}(\mathbf{I}_S, \mathbf{I}_R^c) = \sum_{\xi \mathbf{p}_S^i \in \xi \mathbf{I}_S} \left[\min_{\xi \mathbf{p}_R^j \in \xi \mathbf{I}_R(\zeta)} \|\xi \mathbf{p}_S^i - \xi \mathbf{p}_R^j(\zeta)\| \right] \quad (2)$$

In (2), the outer sum $\sum_{\xi \mathbf{p}_S^i \in \xi \mathbf{I}_S} (\cdot)$ implements the edge awareness by binning the edges according to their orientation (as quantized by $\xi \in [-\pi, \pi)$), and implicitly assigning edge pixel correspondences. We observe that the inner minimization problem $\min_{\xi \mathbf{p}_R^j \in \xi \mathbf{I}_R(\zeta)} \|\xi \mathbf{p}_S^i - \xi \mathbf{p}_R^j(\zeta)\|$ for each $\xi \mathbb{E}_{cm}$ can be solved by the Euclidean distance transform. As the $\xi \mathbb{E}_{cm}$ cost is cumulative over the $\xi \mathbf{I}_S$ image, the cost boils down to the pixel-wise sum of absolute differences between the Euclidean distance transforms (EDT) of images $\xi \mathbf{I}_S$ and $\xi \mathbf{I}_R(\zeta)$. So using the definition of Euclidean distance transform from [34] in (2), we simplify our dense edge matching energy as

$$\mathbb{E}_{cm}(\zeta) = \sum_{\xi} \left[\sum_{\mathbf{g}} [|\text{EDT}(\xi \mathbf{I}_S) - \text{EDT}(\xi \mathbf{I}_R(\zeta))|] \right] \quad (3)$$

We minimize this function with gradient descent to obtain a coarse pose estimate $\hat{\zeta}$ from the camera image and transfer the pose estimate to the GelSight camera frame as $\hat{\zeta}_{GS}$. In contrast to [30], [31], we implemented modified and differentiable versions of the matching costs and thus, our gradient steps are about 80% faster than the reference implementations of [30], [31] for the same size of the image. In practice, to keep the computational cost low, we maintain a coarse pose estimate using (1) throughout the major part of the trajectory and switch to (3) at a point beyond which the objects are de-focussed. For



Fig. 4. Our pipeline to estimate pose through vision and touch. Fig. 4(a) is the color channel of the RGB-D sensor stream captured at a particular point in the trajectory. Fig. 4(b) is the pose estimate obtained after solving (1), which is refined by solving (3) and we obtain a camera pose estimate $\hat{\zeta}$ (shown in Fig. 4(c)) which is mostly correct in the camera projection depth and camera yaw. We transfer $\hat{\zeta}$ to the reference frame of the GelSight and obtain Fig. 4(d) showing the relative pose of the object and the GelSight. This corresponds to the pose $\hat{\zeta}_{GS}$. This estimate is further refined by minimizing (4) using the GelSight data (Fig. 4(e)) obtained at contact, and we obtain the final pose shown in Fig. 4(f).

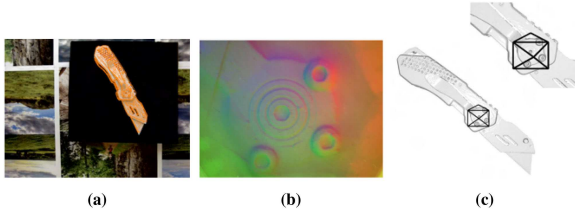


Fig. 5. Fig. 5(a) is the pose estimated using the color channel image. This pose estimate was initialized with coarse pose estimates obtained using depth images. Fig. 5(b) is the GelSight data obtained at contact and Fig. 5(c) is the camera pose at contact. Figs. 1 and 4 contain similar results.

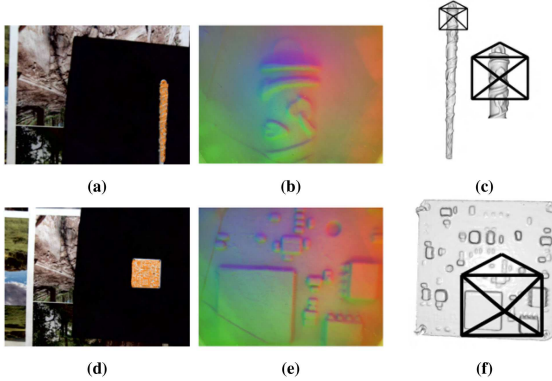


Fig. 6. Fig. 6(a) to (c) are the steps in localizing a slender metallic object, Fig. 6(d) to (f) are the steps in localizing a miniature circuit board. L-R we generate a coarse pose estimate with just the color image, receive the GelSight image on contact and finally obtain the camera pose on contact. The object in Fig. 6(a) is 10 cm long and 8.5 mm in diameter at the thickest part. The circuit board in Fig. 6(d) is 38 mm $\times$ 38 mm square.

objects in Figs. 1, 4 and 5 this distance was around 15 cm and for objects in Fig. 6 it was 10 cm.

D. Precise Contact Pose Estimation Through Touch

We found that our vision-based localization had errors on the order of a centimeter, due to our algorithms, imperfect camera calibration, local minimums in matching, and sometimes a lack of visual features to match or track. In addition, as the hand comes near the object, the depth and image sensor measurements become unusable – the LiDAR based sensor provides reliable depth estimates at distances greater than 25 cm, and the camera image measurements were usable at distances greater than 12–15 cm to the object, after which the items of interest often go out of view and the image becomes too blurry to extract high

quality edges needed by our 3-D pose estimation algorithms. We find that using the tactile image on contact can improve contact point and pose estimation. In Fig. 2 we noted that we could generate metrically correct depth and normal maps of the deformed GelSight surface at contact. In this section we use that information, along with the pose estimated from the previous section to localize contact, given that we know the geometry of the object. To achieve this, we use multi-scale dense depth and normal map alignment to obtain the pose of object with respect to the tactile sensor.

We capture the tactile image in our GelSight’s native camera resolution of (640 $\times$ 480) pixels and obtain depth and normal maps. We then decompose each of the normal and depth maps into 4 lower pyramid levels. We denote these 5 normal and depth maps of the source image as N_S and D_S respectively. Next, we render the object (using [33]) through the GelSight’s viewport using the pose estimated in the previous section and obtain normal and depth maps corresponding to the frame sizes of N_S and D_S respectively.

We note here that this is not an exact simulation of the GelSight sensor through our renderer – the ideal GelSight should only measure objects touching it i.e. ~ 25 mm from the camera. Anything beyond or closer than that is either not touching the sensor or interfering with it. We relax this requirement, and also neglect the effects of the soft body contact between the gel and the object to get a better basin of convergence in the optimization problem. We denote these sets of rendered depth maps by $N_R(\hat{\zeta})$ and $D_R(\hat{\zeta})$. Our alignment cost function $\mathbb{E}_{gs}(\zeta)$ for the GelSight data is

$$\mathbb{E}_{gs}(\zeta) = \sum_{i=1}^5 [|D_S^i - D_R^i(\zeta)| + [1 - \langle N_S^i \cdot N_R^i(\zeta) \rangle]] \quad (4)$$

where $\langle N_S^i \cdot N_R^i \rangle$ denotes the pixel-wise dot products of the normal maps. We solve this with $\hat{\zeta}_{GS}$ as the starting guess using gradient descent. Fig. 4 describes the steps discussed above.

IV. RESULTS

In this section we present the results of localizing contacts using the sensor setup and the algorithms discussed in this work through 6 scenarios. For the experiments presented below, we use a black background to simplify the object segmentation and edge detection. Except for the glue gun, all the other objects had reflective parts and were painted matte white to remove specularities.

1) *Localization With Color and Depth Edges:* Our RGBD sensor captures the depth of the scene reliably from 30 cm

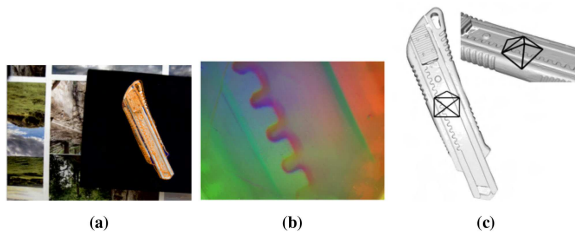


Fig. 7. Collocated vision can help us disambiguate between repeated tactile features. In this experiment we touch the middle of the set of teeth and could localize the contact using our pipeline.

away and can produce aligned depth and color images. As the depth images are unchanged by changes in lighting and texture, extracting edges from depth images and maintaining coarse pose estimates as the robot moves closer to the object is a viable strategy to generate initial pose estimates which are later refined to estimate the pose at contact. For the objects described in Figs. 1, 4 and 5, when the depth sensor output degraded (at 20 cm) we transferred the pose estimates obtained (and maintained) using the depth image, to the color image. refined the pose estimates from the final color image using the tactile image to obtain the pose at contact.

2) *Localizing Small, Flat and Thin Objects:* Depth cameras don't capture much in situations where an object has little depth variation. The limited field of view of most depth cameras means they lose sight of where the GelSight sensor will land as the hand approaches an object. For these reasons we collocated a high resolution large field of view RGB camera with the GelSight sensor, so the system would also work well with small thinner objects with less depth variation. In Fig. 6 we demonstrate the localization of a slender metallic pin and a 4 cm square circuit board.

3) *Disambiguating Tactile Measurements With Vision:* With collocated vision, we can disambiguate between repetitive surface features, which would not have been possible with just tactile sensing. To show this, we set up an experiment (Fig. 7) where the robot approaches the box cutter from above and touches around the middle of the slider. The tactile image recorded is shown in Fig. 7(b). The pose estimates obtained from the camera when transferred to the GelSight yield useful initial guesses which can then be refined with the tactile data to localize the contact.

4) *Estimating Object Pose With Vision And Touch:* In this section we present our results on localizing objects with vision and touch, and evaluate the performance of our localization pipeline. We focus on the case where the contact point has a unique arrangement of useful tactile features. This is the best case for a combined approach. In practice, if the tactile sensor contacts a featureless area, we can detect this situation and ignore the output of the tactile sensor. Most work in this area specializes for a set of objects used in a training set for a learning approach. We did not find work that combined a camera with a high resolution tactile imaging sensor for localizing objects upon contact, so there isn't an obvious method to compare to. This along with our choice of smaller and highly textured objects preempts easy comparison with recent conventional and learned localization approaches (e.g. [23], [24], [35]–[37] etc.). Instead, we present overall results of accuracy experiments below.

For each of the 6 objects used in this work (Figs. 1(f), 4, 5, 6(c), 6(f), 7), we fix the object to the robot table. Assuming that

there is zero error in position control of our robot (we use a Universal Robots UR5E manipulator fixed to a vention.io table of recommended design for the same robot), we register the object with respect to the robot base and treat this pose as our ground truth. Next, we move the robot vertically 1 m above the object and move the robot down to touch the object at a chosen point that will yield good tactile information, and localize the object with respect to the robot. We repeat this 3 times for the same object positions and repeat this experiment for 2 more positions of the object with respect to the robot – i.e. localizing each object 9 times with respect to the robot. Fixing the objects is a restrictive assumption in the context of localizing objects especially with touch, however, to ensure repeatability of the experiments reported in the section, we had to fix the objects to a rigid base. Following [30], [31] we report the repeatability of our pose estimation pipeline as the measure of its performance.

Using tactile sensing the localization errors were brought down to ± 1.5 mm in translation and $\pm 0.5^\circ$ in rotation from about 1.5 cm in translation and 2° in rotation using only vision³. However, for cases where the tactile features were not unique, e.g. the box cutter teeth (Fig. 7) and the metallic object (Fig. 6(c)), the tactile sensing actually increased the localization errors in the horizontal directions. The order of these errors were equivalent to the scale of the repeated features – 5 mm for the experiment described in Fig. 7(c) (the box-cutter teeth are about 3 mm wide placed in intervals of 5 mm) and about 2 mm for the experiment described in Fig. 6(c) (the embossed features are very similar at intervals of 3.5 mm). This observation is consistent with the fact that the final gradient descent step (4) to refine the camera based pose estimates will converge to the wrong local minima if the tactile measurements are not distinctive enough. However, for objects with rich tactile features, using tactile sensing assisted with vision provides better localization than exclusively using either as we show in Section IV. We repeated a subset of the experiments reported above where we first corrected the robot trajectory using the procedure described in Section III-B2 – and observed similar localization performance.

5) *Effect of Points of Contact on Localization Accuracy:* In this section we present the effect of randomly selected contacts for localization. For this set of experiments, we fix each of the objects and the black background plate used in Section IV to a graduated compound slide capable of in plane translation and rotation. We then moved the robot vertically down to make contact at the same location on the object used for the experiments reported in Section IV to generate a starting pose. Next, we generated 5 random configurations per object in translation and orientation on the plane of the table and moved the robot vertically down to touch the object and attempted to recover the randomly generated pose perturbations we introduced. For each of the objects, as expected, we observed similar errors in localization using only vision as reported in Section IV. For the box cutter (Fig. 4) most of the contacts yielded useful tactile signals so the errors in recovering the perturbations in pose were in the range reported in Section IV – i.e. ~ 8 mm in translation and $\sim 1.5^\circ$ in rotation. This observation was also consistent for the smaller textured objects³ (Fig. 6). However, tactile sensing was not always helpful in localizing the objects – for the glue gun (Fig. 1) and the folding knife (Fig. 5, significant parts of the object were featureless and the tactile signals obtained when touched at these parts were unusable in localizing the objects as

³Please visit the project website to view the metrics, CAD models, etc.

the final gradient descent step ((4) in Section III-D) re-introduced localization errors of about 3-4 cm and 15° by converging to incorrect poses.

6) *Localizing Contact Using Only Vision*: We explored two approaches for predicting contact with only vision – an image-based approach that tracks the POE similar to visual servoing and a 3-D geometric approach that can estimate contact based on a single image captured before the object gets out of focus or gets mostly out of the frame.

In the **first case**, we identify the POE as described in Section III-B1. Based on the known offset between the camera and the GelSight, we can identify the predicted GelSight contact point in any camera image. This approach works best when the hand is moving straight to the contact point (no trajectory corrections or going around obstacles), there are a lot of visual features to track, and the hand starts high enough so there is little offset of the object and the background caused by the different distances to the object and the background. We tested performance with the data from the previous section for objects in Figs. 4, 7 and 1(f). We could predict the location of contact with 1 cm accuracy.

In the **second case**, we took the final usable image in these trajectories and used the procedure described in Section III-C to obtain the pose of the object with respect to the RGB camera. Next, we transformed the pose of the object to the frame of the GelSight camera and simulated moving blindly until the vertical distance between the object and the GelSight is 3 cm (the gel surface is 2.5 cm away from the GelGight camera), and reported the location of contact. We compared these estimates of the point of contact for all the objects we used. We found that for experiments with larger objects (as shown in Figs. 4, 7, 5 and 1(f)) the average error between the estimated true location of contact and the location estimated by blindly moving downwards in the frame of the GelSight camera was 3.5 cm, whereas the same metric for experiments with smaller objects (as shown in Fig. 6) were about 1.5 cm. The experiments with the smaller objects have lower errors because the distance between the GelSight and the object at the point the pose of the object was measured with only vision was much lower (8-10 cm) than the experiments with the larger object where the distance was about 25-30 cm, so the length of the blind descent was smaller for the smaller objects and hence the accumulated error in predicting the point of contact was smaller.

To localize contact with a *single tactile image*, we attempted to match the processed tactile image obtained from the modified GelSight (as an image (Fig. 2(c)) or a point cloud (Fig. 2(d))) to equivalent representations of the object model. We used standard image feature matching (ORB, SIFT) and point cloud feature matching ([35], [36]) techniques but were unable to generate useful matches. This leads us to conclude that in the absence of good initial pose estimates from an external sensor or learned embeddings (e.g. [24]), the portion of the object observed by the GelSight is often not large enough for conventional feature matching algorithms to work.

V. DISCUSSION AND FUTURE WORK

We focused on objects with significant surface tactile texture that our small tactile sensor could image. This led us to not use existing sets of objects for manipulation research (YCB [38], McMaster [39]). This different choice is due to the absence of a data set to benchmark tactile imaging and vision working

together. It is unclear what objects should be in the data set, how approaches trained on such a data set would generalize to novel objects and how much effort is needed to introduce new objects into the data set and pipelines based on them. In our approach, we used an off the shelf 3-D scanner (EinScan-SE) to generate 3-D models of objects and reference poses for our initial estimates (see Section III-C). We do expect that versions of the current state-of-the-art edge based localization pipelines (what we use) to be inherently slower than learned pipelines for generating initial pose estimates (e.g. [24] or [37]), but we believe that our localization pipeline would perform well for objects we have not tested here. Also, in the current literature, we did not find explicit or implicit feature transforms invariant across tactile images (or processed tactile data) and visual images. Although there exists works on learning features (see [10], [23], [24]), using them for explicit correspondences to match tactile data to visual data is relatively unexplored and we consider promising future work.

While integrating visual sensors of different capabilities, we noted that having optical sensors with almost co-incident optical axes would trivially solve some of the issues we faced while localizing small objects and disambiguating possible localizations. Collocating an ensemble of cameras of different capabilities (similar to smart phone multi-camera systems) creating a synthetic vision system is a natural next step of this work. Such a system could create a virtual fovea focused on where the tactile sensor could contact the object and bring down the overall footprint of the sensor setup so that it can be efficiently colocated with a standard gripper. Using a gripper with the ensemble of sensors is also future work. The GelSight data can be inexpensively processed to obtain surface normals of objects with respect to a fixed frame. This is valuable for pose estimation and should be emphasized in future object shape representations and manipulation algorithms. Computing object surface normals by controlling illumination at the scale of a robot workspace is also future work.

VI. CONCLUSION

In this work we show that collocating cameras with tactile sensors on a robot hand in many cases enables tactile localization to work where vision only or tactile sensing only localization may perform badly or fail. We demonstrated this with common objects without substantial pre-computation and without using a known and fixed set of object models (learning). Vision can help disambiguate tactile image matching, especially in cases of limited, repetitive, or otherwise ambiguous tactile features. Tactile sensing almost always improves visual localization as well. We also found that using optic flow from hand mounted cameras had to be integrated across about 10 cm of camera travel to provide useful heading estimates, and could be used to correct trajectories.

REFERENCES

- [1] A. Yamaguchi and C. G. Atkeson, "Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 1045–1051.
- [2] W. Yuan, S. Dong, and E. H. Adelson, "GelSight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017, Art. no. 2762.
- [3] Facebook Research, "Detectron: FAIR's platform for object detection research," *GitHub*. [Online]. Available: <https://github.com/facebookresearch/Detectron>

- [4] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6DoF closed-loop grasping from low-cost demonstrations," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4978–4985, Jul. 2020.
- [5] A. Yamaguchi and C. G. Atkeson, "Implementing tactile behaviors using fingervision," in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot.*, 2017, pp. 241–248.
- [6] W. Yuan, M. A. Srinivasan, and E. H. Adelson, "Estimating object hardness with a gelsight touch sensor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 208–215.
- [7] S. Wang *et al.*, "3-D shape perception from monocular vision, touch, and shape priors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1606–1613.
- [8] E. J. Smith *et al.*, "3-D shape reconstruction from vision and touch," 2020, *arXiv:2007.03778*. [Online]. Available: <https://arxiv.org/pdf/2007.03778.pdf>
- [9] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 4000–4007.
- [10] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba, "Connecting touch and vision via cross-modal prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10601–10610.
- [11] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Localizing the object contact through matching tactile features with visual map," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3903–3908.
- [12] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Trans. Mechatronics*, vol. 5, no. 1, pp. 39–48, Mar. 2000.
- [13] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 1750–1757.
- [14] J. R. Serres and F. Ruffier, "Optic flow-based collision-free strategies: From insects to robots," *Arthropod Struct. Develop.*, vol. 46, no. 5, pp. 703–717, 2017.
- [15] D. Raviv, "A quantitative approach to looming," US Department of Commerce, National Institute of Standards and Technology, 1992. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir4808.pdf>
- [16] D. Raviv and K. Joarder, "The visual looming navigation cue: A unified approach," *Comput. Vis. Image Understanding*, vol. 79, pp. 331–363, 2000.
- [17] G. Yang and D. Ramanan, "Upgrading optical flow to 3-D scene flow through optical expansion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1331–1340.
- [18] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Novel tactile-SIFT descriptor for object shape recognition," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5001–5009, Sep. 2015.
- [19] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Iterative closest labeled point for tactile object shape recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 3137–3142.
- [20] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "GelSlim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1927–1934.
- [21] R. Li *et al.*, "Localization and manipulation of small parts using gelsight tactile sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3988–3993.
- [22] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," 2019, *arXiv:1910.02860*.
- [23] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3811–3817.
- [24] M. Bauza, E. Valls, B. Lim, T. Sechopoulos, and A. Rodriguez, "Tactile object pose estimation from the first touch with geometric contact rendering," *Conf. Robot Learn.*, PMLR, 2021, pp. 1015–1029.
- [25] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake, "Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation," in *Proc. 2nd IEEE Int. Conf. Soft Robot.*, 2019, pp. 597–604.
- [26] M. K. Johnson, F. Cole, A. Raj, and E. H. Adelson, "Microgeometry capture using an elastomeric sensor," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–8, 2011.
- [27] *OpenCV*, "Open source computer vision library," 2021.
- [28] G. Farnéback, "Two-frame motion estimation based on polynomial expansion," in *Proc. Scand. Conf. Image Anal.*, 2003, pp. 363–370.
- [29] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 25–36.
- [30] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *Int. J. Robot. Res.*, vol. 31, no. 8, pp. 951–973, 2012.
- [31] M. Imperoli and A. Pretto, "D²CO: Fast and robust registration of 3-D textureless objects using the directional chamfer distance," in *Proc. Int. Conf. Comput. Vis. Syst.*, 2015, pp. 316–328.
- [32] C. Choi and H. I. Christensen, "3-D textureless object detection and tracking: An edge-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 3877–3884.
- [33] P. Henderson and V. Ferrari, "Learning single-image 3-D reconstruction by generative modelling of shape, pose and shading," *Int. J. Comput. Vis.*, vol. 128, pp. 835–854, 2020.
- [34] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory Comput.*, vol. 8, pp. 415–428, 2012.
- [35] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3-D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3212–3217.
- [36] S. M. Prakhya, B. Liu, W. Lin, V. Jakhetiya, and S. C. Guntuku, "B-SHOT: A binary 3-D feature descriptor for fast keypoint matching on 3-D point clouds," *Auton. Robots*, vol. 41, no. 7, pp. 1501–1520, 2017.
- [37] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *Robot. Sci. Syst.*, 2018.
- [38] B. Calli *et al.*, "Yale-CMU-Berkeley dataset for robotic manipulation research," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 261–268, 2017.
- [39] E. Corona, K. Kundu, and S. Fidler, "Pose estimation for objects with rotational symmetry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7215–7222.