# Supplementary information

**Embedding high-resolution touch across robotic hands enables adaptive human-like grasping**

Zihang Zhao[1,2 †], Wanlin Li[2 †], Yuyang Li[1,2 †], Tengyu Liu[2 †], Boren Li[2], Meng Wang[2], Kai Du[1], Hangxin Liu[2 *], Yixin Zhu[1,3 *], Qining Wang[4], Kaspar Althoefer[5 *], Song-Chun Zhu[1,2]

† equal contributors  * corresponding authors

[1] Institute for Artificial Intelligence, Peking University, Beijing 100871, China

[2] Beijing Institute for General Artificial Intelligence (BIGAI), Beijing 100080, China

[3] PKU-WUHAN Institute for Artificial Intelligence, Wuhan 430075, China

[4] College of Engineering, Peking University, Beijing 100871, China

[5] School of Engineering and Materials Science, Queen Mary University of London, London, E1 4NS, UK

# S1 Details of GelSight-inspired image formation model

## S1.1 Near-field camera model and calibration

The near-field camera model consists of two main components: camera geometry and radiometry. The geometry employs a perspective projection model [51], suitable for the compact GelSight-inspired sensors where elastomer deformation is significant relative to the camera's working distance. A 3D surface point, ${}^{\{C\}}\widetilde{\mathbf{X}}$, is projected to a 2D image point, ${}^{\{P\}}\mathbf{x}$, using the equation

$$
{}^{\{P\}}\mathbf{x} = \frac{1}{{}^{\{C\}}Z}\mathbf{K}{}^{\{C\}}\widetilde{\mathbf{X}}, \tag{S1}
$$

where $\{C\}$ and $\{P\}$ are the camera and pixel coordinate frames, respectively. The camera intrinsics, $\mathbf{K}$, determine the unit viewer direction, $\bar{\mathbf{v}}$, at each pixel. The world coordinate frame, $\{W\}$, aligns its XY plane with the elastomer support plane and is related to $\{C\}$ through a 3D rigid transformation, serving as camera extrinsics. The camera is calibrated prior to elastomer attachment using Zhang *et al.* [52] to correct lens distortions and obtain intrinsics and extrinsics. Given a uniform and known elastomer thickness, the contact plane equation is straightforwardly derived.

Assuming vignetting effects are negligible due to the central location of the perceptible region in the camera's field of view, the camera radiometry focuses solely on photometric response. Fixed white balance gain is set for the color cameras, and each color channel, $\{F_c\}_{c=R,G,B}$, has its own monotonic photometric response correlating image irradiance, $\{I_c\}_{c=R,G,B}$, to measured intensity, $\{M_c\}_{c=R,G,B}$, as $M_c = F_c(I_c)$. This response is calibrated using $\{G_c = F_c^{-1}\}_{c=R,G,B}$, following Mitsunaga *et al.* [53], where $G_c$ is modeled as a third-order polynomial. Surface radiance, $\{L_c^{i,j}\}_{c=R,G,B}$, is then derived from measured intensity as $L_c^{i,j} = G_c\left(M_c^{i,j}\right)$, where $(i,j)$ denotes the pixel location.

Given that all compact GelSight-inspired sensors utilize the same camera type, a single calibration suffices for all, with the exception of an additional calibration image required to estimate the extrinsics for each individual sensor.

### S1.2 Ground-truth elastomer surface geometry acquisition

To construct the sensor model, acquiring ground-truth elastomer surface geometry is essential. Direct measurement using high-accuracy devices being impractical, an indirect approach is adopted. The experimental setup employs an XYZ 3-axis linear trimming stage, as depicted in Figure S1A. The sensor is affixed at the stage's bottom, allowing lateral adjustments in the X and Y directions. Assuming ideal manufacturing, the stage's XY plane is parallel to the sensor contact plane. A 3D-printed cube probe is mounted upside down at the top, perpendicular to the sensor contact plane. The stage coordinate frame, $\{S\}$, aligns its XY plane with the mount's flat base and its Z-axis perpendicular to the sensor contact plane. The ${}^{\{S\}}Z = 0$ is set by lowering the cube probe until just before contact. The cube probe's position is adjusted to appear near the perceptible region's center. The 3D transformation from $\{S\}$ to $\{C\}$ is calibrated using a perspective-n-point problem [54], involving camera intrinsics and 3D corner points of the cube probe. Calibration performance is assessed by reprojection error, as shown in Figure S1A.

Upon calibrating the stage, the cube probe can be substituted with any 3D-printed object of known geometry. A depth map is generated using the calibrated camera's visibility constraint. Only pixels with depth within the elastomer thickness range are preserved in the contact area, while the contact-free area is filled using the contact plane's depth map. Thus, by mounting various objects and adjusting the calibrated stage, per-pixel elastomer surface geometry is obtained as ground truth.

## S1.3 Near-field lighting model and calibration

The near-field lighting model comprises lighting geometry and radiometry. In the compact GelSight-inspired sensors, the geometry involves multiple LEDs illuminating the surface from various positions. Given their small size relative to working distance, LEDs are modeled as point sources. Their positions are denoted as $\{^{\{C\}}\widetilde{\mathbf{S}}_k\}_{k\in[1,N]}$, where $N$ is the LED count for a sensor. For each $k^{th}$ LED, the unit light direction, $\bar{\mathbf{s}}_k$, and light-surface distance, $\|\mathbf{s}_k\|$, are determined. A 3D-printed calibration object with two cylindrical sundials on a square mount's diagonal is used for light position estimation. The object is carefully pressed onto the elastomer using the calibrated XYZ stage. Individual LEDs are activated sequentially, and their positions are estimated using shadow cues [55] and triangulation.

Lighting radiometry accounts for the spatial distribution of emitted energy reaching the surface. It is influenced by the LED's radiation pattern and its distance to the surface point. For the chosen LEDs with symmetrical radiation patterns, the radiometry is characterized by

$$\kappa_k^{i,j} = \eta_k \left( \left( -\bar{\mathbf{s}}_k^{i,j} \right)^\top \bar{\mathbf{p}}_k \right)^{\mu_k} / \|\mathbf{s}_k^{i,j}\|^2, \tag{S2}$$

where $\bar{\mathbf{p}}_k$ is the principal light direction for the $k^{\text{th}}$ LED. Parameters $\eta_k$, $\mu_k$, and $\bar{\mathbf{p}}_k$ are calibrated jointly with reflectance parameters.

## S1.4 Reflectance model and calibration

The reflectance model for our compact GelSight-inspired sensors is based on the elastomer material's reflectance property. Although the elastomer is theoretically near-Lambertian, its appearance in camera images is flatter, which we attribute to surface roughness. To address this, we adopt the generalized Lambertian model [56]. The surface radiance under specific lighting conditions is modeled as

$$\begin{cases} L_c^{i,j}\left(\theta_r^{i,j},\theta_i^{i,j},\phi_r^{i,j}-\phi_i^{i,j};\rho_c,\sigma\right)=\kappa^{i,j}\frac{\rho_c}{\pi}\max\left[0,\cos\theta_i^{i,j}\right]f_r^{i,j} \\ f_r^{i,j}=A+B\max\left[0,\cos\left(\phi_r^{i,j}-\phi_i^{i,j}\right)\right]\sin\alpha^{i,j}\tan\beta^{i,j} \\ A=1.0-0.5\frac{\sigma^2}{\sigma^2+0.33} \\ B=0.45\frac{\sigma^2}{\sigma^2+0.09} \end{cases}, \tag{S3}$$

where $\left(\theta_i^{i,j},\phi_i^{i,j}\right)$ and $\left(\theta_r^{i,j},\phi_r^{i,j}\right)$ are the light and viewer directions in a local coordinate frame. Additional parameters include $\alpha^{i,j}$, $\beta^{i,j}$, $\sigma$, and $\{\rho_c\}_{c=R,G,B}$.

For calibration, a 3D-printed sphere with a 3 mm diameter is pressed at $Q$ different locations onto the elastomer surface using a calibrated XYZ stage. At each press, $N$ images are captured, each with a single LED lit. The calibration aims to solve the following constrained nonlinear fitting problem:

$$\begin{aligned} &\min_{\eta_k,\mu_k,\bar{\mathbf{p}}_k,\sigma,\rho_c}\sum_{q=1}^{Q}\sum_{k=1}^{N}\sum_{c=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W}\left[G_c\left(M_{c,k,q}^{i,j*}\right)-\kappa_k^{i,j}\frac{\rho_c}{\pi}\cos\theta_i^{i,j}f_r^{i,j}\right]^2+\lambda\sum_{k=1}^{N}\left(\eta_k-1\right)^2 \\ &\text{s.t.}\quad \begin{aligned} & i,j\in\mathcal{I}_k\cap\mathcal{S} \\ & \|\bar{\mathbf{p}}_k\|=1 \\ & \mu_k\geq 0 \\ & \sigma\geq 0 \end{aligned} \end{aligned}, \tag{S4}$$

where $H$ and $W$ are the height and width of the image sensor in pixels. $\mathcal{I}_k$ is the non-shadow area of the $k^{th}$ image, and $\mathcal{S}$ is the perceptible region of GelSight-inspired within the camera's field of view. The asterisk denotes measurement throughout the paper. The first term in the equation is for least-square fitting, while the second term is for regularization. Due to the problem's high nonlinearity, careful initialization is required. $\bar{\mathbf{p}}_k$ is initialized to point towards the center of the perceptible region, $\mu_k$

is initialized to 1.2, $\eta_k$ and $\rho_c$ are initialized to 1, and $\sigma$ is initialized to 0.2. The problem is solved using the Levenburg-Marquardt algorithm. Calibration performance is evaluated using a test image not seen during fitting, with the error converging at 4 presses, as shown in Figure S1B. This implies that only 4 N calibration images are needed for accurate parameter estimation. Since the sensors share the same type of LEDs and elastomer material, calibration is required only once.

Finally, with all LEDs turned on, the overall surface radiance, $\widehat{L}_c^{i,j}$, is given by

$$\widehat{L}_c^{i,j} = \sum_{k=1}^{N} L_{c,k}^{i,j}. \tag{S5}$$

## S1.5 Cast shadow model

In addition to the pixel-wise shading model, global effects like inter-reflection and cast shadow also influence pixel intensities in tactile images. Inter-reflection is negligible due to the black low-albedo matte material of the sensor's cover. For cast shadows, given the known calibrated point light positions and surface geometry, we compute the shadow for each light using the canonical hidden point removal operator [57] to mask the shadow-casting areas.

## S1.6 Soft elastomer deformation model

The contact-free elastomer is assumed to have a 3D geometry parallel to the reference plane. Given the camera extrinsics and elastomer thickness, the contact plane equation in $\{C\}$ is determined. When an object contacts the elastomer, its depth map is computed using camera visibility constraints, object mesh, and 6D pose. Only pixels within the elastomer thickness range are preserved in the contact area. The non-contact area is filled with the depth map of the contact-free elastomer. Due to the material's softness, the elastomer deforms into a smoothed shape of the contacting object. We apply a simple soft body simulation approximation, smoothing the depth map boundaries between contact and non-contact areas using pyramid Gaussian kernels [58]. Finally, pixel values in non-contact and non-shadow areas are replaced by their counterparts in a background image captured without elastomer deformation.

### S1.7 Sensor simulator evaluation

We evaluate the sensor simulator by comparing its outputs to real sensor data, using the setup shown in Figure S1A. We 3D print 20 objects from the tactile shape dataset [59], excluding the *Cone* to prevent sensor damage. For each object, we collect multiple sensor outputs under varying conditions, resulting in a dataset of 140 real images.

We employ three metrics for comparison: mean absolute error (L1), structural index similarity (SSIM), and peak signal-to-noise ratio (PSNR). Qualitative and quantitative comparisons are presented in Figure S1C and Table S1, respectively. The simulator excels in structural accuracy (SSIM) due to our geometrical calibration method and also accurately replicates image intensity (L1 and PSNR) through lighting radiometry and reflectance calibration. The simulated and real cast shadow regions align well, confirming the effectiveness of our light position calibration.

Our physics-based sensor model and calibration techniques enable the simulator to generate accurate sensor outputs at scale. This eliminates the need for extensive data acquisition for each sensor on F-TAC Hand. The calibrated parameters can be reused as the hardware components are shared, simplifying the calibration process for all sensors on F-TAC Hand.

# S2 Details of DPS

## S2.1 Training details

We implemented the deep PS network using PyTorch and employed additive white Gaussian noise (standard deviation 0.01) for data augmentation. The network was trained using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a batch size of 128, and an initial learning rate of 0.01. The learning rate was halved every five epochs. The training was conducted on a single NVIDIA RTX 3090 GPU and took approximately six hours to converge.

## S2.2 Evaluator

We evaluate the trained deep PS network using an unobserved dataset of four shapes with 46,656 samples. The mean angular error (MAE) serves as the evaluation metric.

# S3 Differentiable force closure estimator

Determining force-closure grasps under kinematic constraints is computationally expensive in our context. To mitigate this, we introduce a quick, differentiable force closure estimator, facilitating efficient grasp generation. Computational time benchmarks can be found in Supplementary Information S6.

For a grasp with a set of $n$ contact points $\{x_i \in \mathbb{R}^3, i = 1, \ldots, n\}$, it is in force closure if for any external wrench $\omega$, there exists a combination of contact forces $\{f_i \in \mathbb{R}^3\}$ that can resist $\omega$. Specifically, the $i$-th contact force $f_i$ should lie within the friction cone at the $i$-th contact point $x_i$. Formally, a grasp is in force closure if it satisfies the following constraints:

$$GG' \succeq \epsilon I_{6\times 6}, \tag{S6a}$$

$$Gf = 0, \tag{S6b}$$

$$f_i^T c_i > \frac{1}{\sqrt{\mu^2+1}}|f_i|, \tag{S6c}$$

$$x_i \in S(O), \tag{S6d}$$

where $S(O)$ is the object surface, $c_i$ the friction cone axis at $x_i$, $\mu$ the friction coefficient, $f = [f_1^T f_2^T ... f_n^T]^T \in \mathbb{R}^{3n}$ the unknown variable of contact forces, and

$$G = \begin{bmatrix} I_{3\times 3} & I_{3\times 3} & \ldots & I_{3\times 3} \\ \lfloor x_1 \rfloor_\times & \lfloor x_2 \rfloor_\times & \ldots & \lfloor x_n \rfloor_\times \end{bmatrix}, \tag{S7}$$

$$\lfloor x_i \rfloor_\times = \begin{bmatrix} 0 & -x_i^{(3)} & x_i^{(2)} \\ x_i^{(3)} & 0 & -x_i^{(1)} \\ -x_i^{(2)} & x_i^{(1)} & 0 \end{bmatrix}. \tag{S8}$$

Here, the form of $\lfloor x_i \rfloor_\times$ ensures $\lfloor x_i \rfloor_\times f_i = x_i \times f_i$. In Equation (S6a), $\epsilon$ is a small constant and $A \succeq B$ indicates $A - B$ is positive semi-definite. Equation (S6a) asserts that $G$ is full-rank; Equation (S6b) asserts that contact forces balance each other; Equation (S6c) asserts that $f_i$ stays within its friction cone; and Equation (S6d) asserts that contact points lie on the object surface.

To satisfy these constraints, one must solve for $\{f_i\}$ that meet Equations (S6b) and (S6c), a time-consuming process. To expedite this, we simplify these equations into:

$$Gf = G(f^n + f^t) = 0, \tag{S9a}$$

$$G\frac{f^n}{\|f^n\|_2} = -\frac{Gf^t}{\|f^n\|_2}, \tag{S9b}$$

$$Gc = -\frac{Gf^t}{\|f^n\|_2}, \tag{S9c}$$

where $f^n$ and $f^t$ represent the normal and tangential components of $f$, and $c = [c_1^T c_2^T \dots c_n^T]^T$ represents friction cone axes. We approximate $Gf$ with $Gc$, which consists of object surface normals at each $x_i$. This simplifies Equation (S6) into:

$$GG' \succeq \epsilon I_{6\times6}, \tag{S10a}$$

$$\|Gc\|_2 < \delta, \tag{S10b}$$

$$x_i \in S(O), \tag{S10c}$$

where $\delta$ is the maximum allowed error due to our relaxation. Using Equation (S10), solving for $f$ becomes unnecessary. The constraints for $x_i$ turn quadratic, significantly accelerating force-closure verification. The residual $\|Gc\|_2$ accounts for discrepancies between contact forces and friction cone axes.

We further rewrite Equation (S10) as soft constraints for gradient-based optimization:

$$FC(x, O) = \lambda_0(GG' - \epsilon I_{6\times 6}) + \|Gc\|_2 + w \sum_{x_i \in x} \max(d_O^{\mathrm{SDF}}(x_i), 0), \tag{S11}$$

where $\lambda_0(\cdot)$ gives the smallest eigenvalue, and $d_O^{\mathrm{SDF}}(x)$ is the SDF from point $x$ to the object $O$, consistent with its definition in Equation (4). By minimizing $FC(x, O)$, we can find the contact points $x = \{x_i\}$ that make contact with the object while providing force-closure support on it, satisfying the constraints in Equation (S10).

# S4 Grasp classification

To showcase the diversity of grasps generated by our system, we synthesize a total of 3,450 grasps, comprising 150 grasps with 2-5 contact points for each of the 23 objects in our study. These grasps are then categorized into 19 distinct types based on the grasp taxonomy proposed by Feix *et al.* [23]. Due to the motion limitations of F-TAC Hand, we make some simplifications and merge similar grasp types that share the same opposition type and thumb position. For example, we combine *Prismatic 2-Finger*, *Prismatic 3-Finger*, and *Prismatic 4-Finger* into a single category, as they differ only in the number of fingers involved. The resulting 19 types are detailed in Table S2.

To ensure high-quality annotations, we enlist human annotators to label each grasp. Instead of providing a single image, we present them with an interactive HTML file that contains various grasp-related details, such as the F-TAC Hand configuration, object mesh, and contact areas. This interactive approach allows annotators to view the grasp from multiple perspectives and adjust the visibility of different elements, providing a more comprehensive understanding. Examples of this interactive process are visualized in Figure S2.

Given the subtle differences among the 19 grasp types, we adopt a two-step annotation process. Initially, annotators are asked to categorize the grasp into one of the three broad types (*Power*, *Precision*, *Intermediate*) based on the definitions in Feix's taxonomy [23]. They then proceed to identify the specific type from the 19 available options, considering factors such as the number of contact areas and the visual similarity to examples provided in Feix *et al.* [23].

## S5 2D cluster map for generated grasps

The 2D map illustrated in Extended Data Fig. 4 shows the human-like characteristics of the grasp synthesis algorithm. To plot this map, we adopt contact map [38, 60] and compute it over the hand surface to represent hand-object contact across different object geometries. The hand contact map $\Omega$, defined over the hand surface $\mathcal{S}(H)$, is computed as the distance from the hand $H$ to the object $O$:

$$\Omega = \log(\epsilon_1 + \min(\mathbf{D}(H, O), \epsilon_2)), \tag{S12}$$

where

$$\mathbf{D}(x_h, O) = \min_{x_o \in \mathcal{S}(O)} \|x_h - x_o\|_2. \tag{S13}$$

The distance function $\mathbf{D}$ measures the Euclidean distance (in meters) from any point on the hand surface $x_h \in \mathcal{S}(H)$ to the object surface $\mathcal{S}(O)$. To improve sensitivity in regions of close contact for better classification, we apply the log function to curve the distance. A small $\epsilon_1 > 0$ is used to maintain the appropriate value range for the log function. As grasping only concerns nearby areas of the object, we introduce $\epsilon_2 > 0$ to truncate the distance, mitigating the influence of distant areas not in contact.

In practice, the contact map is approximated with a dense point cloud with 2,170 points enveloping the hand surface. The parameters $\epsilon_1 = 0.0001, \epsilon_2 = 0.05$ are empirically chosen. Given its high dimensionality, this map is subsequently condensed to 6-dimensional space with PCA after normalization, and visualized on a 2D surface with t-SNE. For clarity, Extended Data Fig. 4 only consists of 1000 grasps, sampled using Furthest Point Sampling (FPS) on the Euclidean distance of the contact maps, from all the generated grasps. The decision boundary is also determined by a SVC with the RBF kernel using the sample points for *Power* grasps and *Precision grasps*, achieving an 81.54 % classification accuracy.

# S6 Quantitative analysis of grasp generation algorithm

Our proposed algorithm efficiently estimates force closure errors using Equation (S11), addressing the most time-consuming aspect of traditional grasp generation methods [61–65]. We validate its efficiency by running experiments on an Intel Xeon CPU (2.90 GHz) coupled with a single NVIDIA RTX 3090 GPU. Various combinations of contact points and parallel instances are tested, and the results are shown in Figure S3A. Notably, even the most complex scenario involving 64 contact points across 64 parallel instances takes under 2 ms per test.

For gradient-based methods, navigating a complex, non-convex energy landscape without getting stuck in local minima is a significant challenge. Our method addresses this issue effectively through MALA. We demonstrate this by running 512 grasp generation instances for grasping a ball on the same hardware setup. The results, presented in Figure S3B, reveal that the entire process takes just 754.61 s and yields 287 successful grasps, achieving a 56 % success rate. This efficiency in generating a diverse set of grasps is a direct result of our approach.

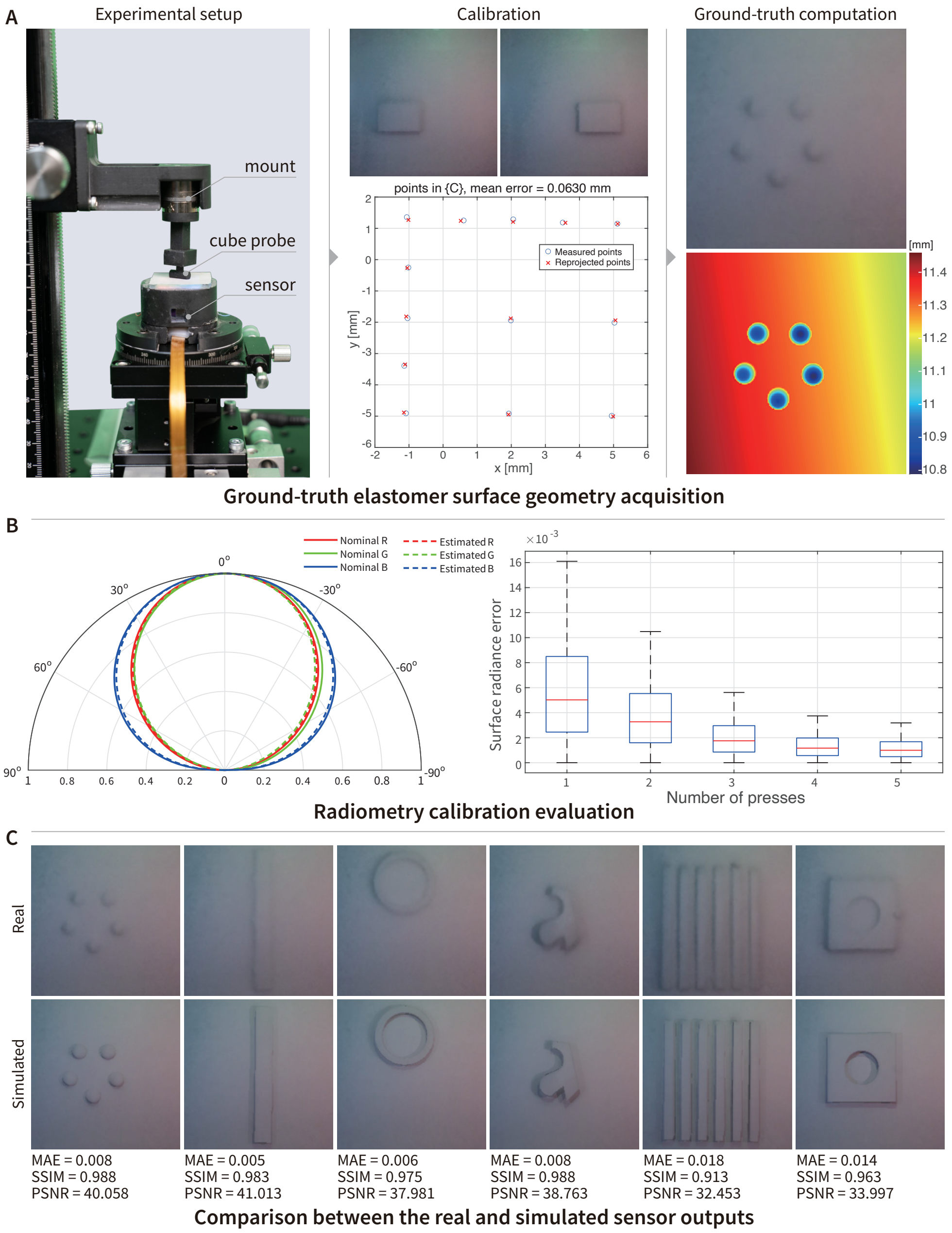


**Fig. S1 Sensor calibration and simulation.** **(A)** Setup for obtaining ground-truth elastomer surface geometry. **(B)** Calibrated light radial attenuation pattern versus nominal values, evaluated by surface radiance error across different press counts. **(C)** Real versus simulated sensor outputs with similarity scores indicated.

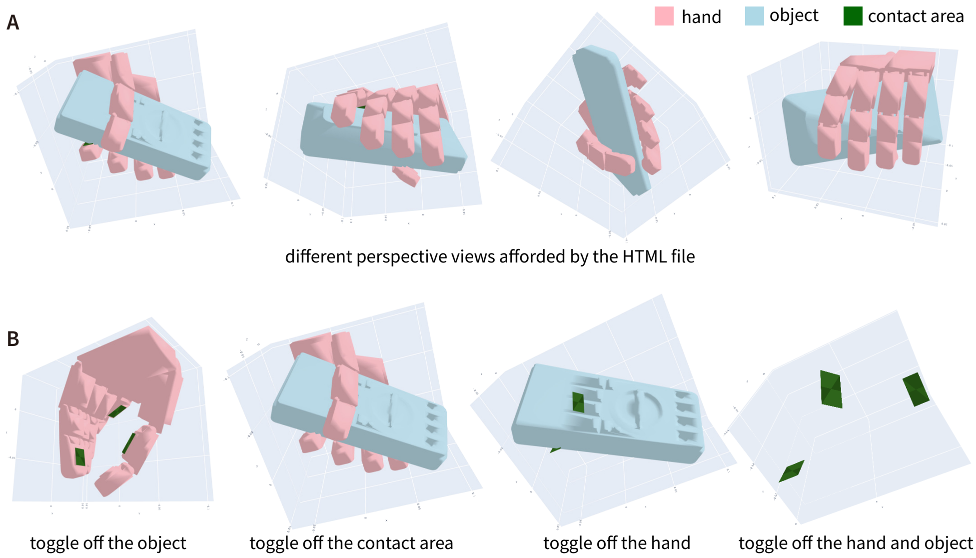


**Fig. S2 Flexible interaction for high-quality annotation:** Screenshots from an HTML file displaying all grasp-related details. The F-TAC Hand is shown in pink, the object (multimeter) in blue, and contact areas in green. **(A)** Annotators can rotate the view for different perspectives. **(B)** Visibility of grasp elements can be toggled for clearer visualization. The annotation for this example is *Extension Type*.

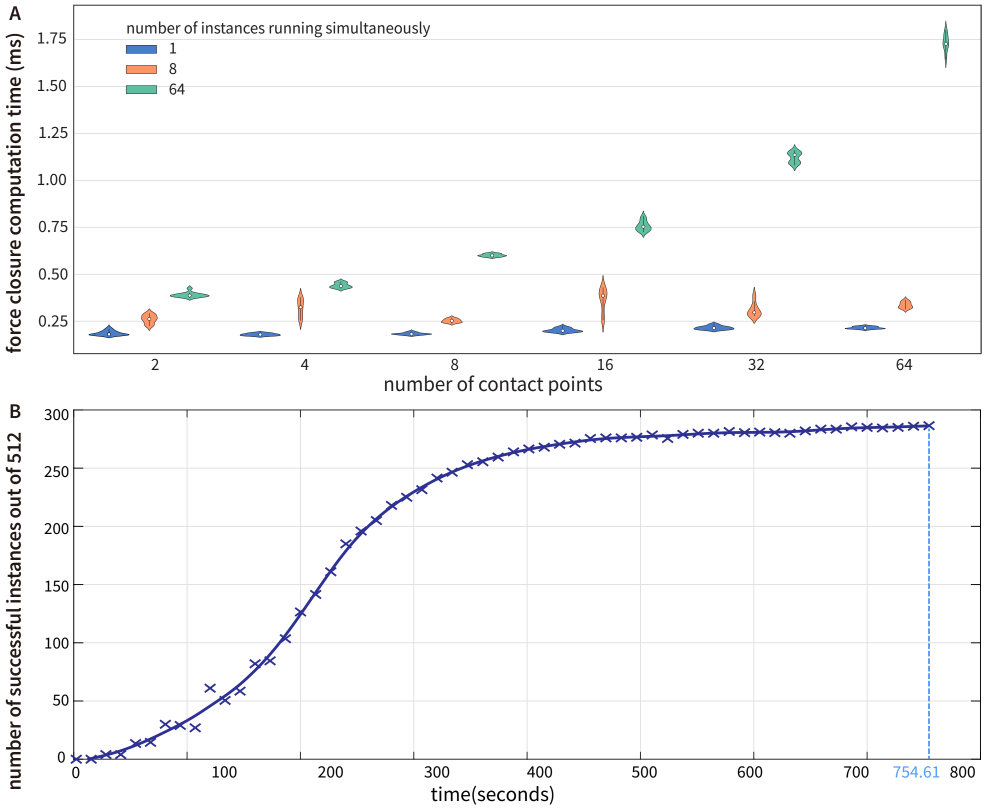


**Fig. S3 Efficient grasp repertoire generation. (A)** Our method rapidly estimates force closure errors using Equation (S11). Even with 64 contact points across 64 parallel instances, each test takes under 2 ms. **(B)** Utilizing MALA, the algorithm navigates the complex energy landscape efficiently. In a test with 512 parallel instances for grasping a ball, the entire process takes just 754.61 s, yielding 287 successful grasps at a 56 % success rate.

**Table S1** Quantitative similarity comparison between the real and the simulated sensor outputs.

| **L1** | **SSIM** | **PSNR** |
|---|---|---|
| $0.009 \pm 0.003$ | $0.976 \pm 0.008$ | $38.808 \pm 1.236$ |

**Table S2** The 19 distinct grasping poses utilized in our classification scheme.

| Grasp Type | Included Types in [23] | Type |
|---|---|---|
| *Adducted Thumb* | *Adducted Thumb, Light Tool* | Power |
| *Fixed Hook* | *Fixed Hook* | Power |
| *Index Finger Extension* | *Index Finger Extension* | Power |
| *Ring* | *Ring* | Power |
| *Power Disk* | *Power Disk* | Power |
| *Extension Type* | *Extension Type* | Power |
| *Palmar* | *Palmar* | Power |
| *Heavy Wrap* | *Large Diameter, Small Diameter, Medium Wrap* | Power |
| *Power Sphere* | *Sphere 3-Finger, Sphere 4-Finger, Power Sphere* | Power |
| *Distal Type* | *Distal Type* | Power |
| *Stick* | *Stick, Ventral* | Intermediate |
| *Lateral* | *Lateral* | Intermediate |
| *Lateral Tripod* | *Lateral Tripod* | Intermediate |
| *Prismatic* | *Prismatic 2-Finger, Prismatic 3-Finger, Prismatic 4-Finger* | Precision |
| *Writing Tripod* | *Writing Tripod* | Precision |
| *Precision Disk* | *Precision Disk* | Precision |
| *Pincer* | *Palmar Pinch, Tip Pinch, Inferior Pincer* | Precision |
| *Precision Sphere* | *Tripod, Quadpod, Precision Sphere* | Precision |
| *Parallel Extension* | *Parallel Extension* | Precision |