

Foundation Vision Models for Robotics: Understanding, Comparing, and Using Them

2026-10-19

Contents

1. Start with the robot's task, not the model name	2
2. Core models: what they do and how they work	2
3. Comparisons that matter when choosing a model	6
4. Four representative robotics workflows	7
5. Deployment: turning useful models into a reliable system	7
6. Frequent questions and the final takeaway	8
Official references	9

[Course home](#) · [简体中文](#) · [PDF](#)

Item	Details
Topic	Vision Foundation Models
Revision	2026-10-19
License	CC BY-NC-SA 4.0 · Xiong Zeng and contributors

1. Start with the robot's task, not the model name

Scope: conceptual mechanisms, inputs and outputs, usage workflows, model combinations, and limitations. Camera-calibration derivations, state-estimation filters, network training, code, and lab exercises are intentionally out of scope.

Reading map: language or visual prompt → **find and segment the target** → **recover distance or shape** → (only when needed) **estimate a rigid object's 6D pose** → **robot policy or controller**. These are different capabilities, often supplied by different tools; no robot pipeline needs to run every model.

1.1 Five different kinds of visual output

Question from the robot	Vision task	Typical output	What it does not establish
"Where is the red screwdriver?"	Detection / open-vocabulary localization	Boxes, category, candidate instances	Precise contour or metric distance
"Which pixels belong to it?"	Instance segmentation / video tracking	Mask and identities across frames	Full 3D orientation
"How far is it from the camera?"	Depth / stereo matching	Depth or disparity map	The object's complete unseen surface
"What does its shape look like?"	3D reconstruction	Mesh, texture, spatial layout	Metrology-grade CAD, mass or inertia
"Where is it, and how is it rotated?"	6D pose estimation	3D translation and 3D rotation	Contact force, stable grasp or slip

Key distinctions: a mask is not depth; depth is not a complete object model; a mesh is not the object's current pose; pose is not contact. Cameras primarily observe visible surfaces. Finger forces, occluded contacts, and material properties generally require additional sensing or physics-based inference.

1.2 Ask four questions before selecting a model

1. **What geometric precision does the task require?** Object identity may only need boxes or masks; orientation-sensitive grasping and alignment often need explicit geometry.
2. **Which sensors are available?** A monocular camera, a functioning RGB-D camera, and a synchronized, calibrated stereo pair enable different approaches.
3. **Which object priors are available?** CAD, physical dimensions, and reference images change what a 6D pose estimator can do.
4. **When must the result be ready?** Offline reconstruction can be slower; closed-loop robotics cares about the freshness of observations, throughput, latency, and recovery after failure.

2. Core models: what they do and how they work

2.1 SAM 2: prompt a target, then propagate its mask with video memory

In one sentence: SAM 2 addresses "I have indicated this object; continue telling me which pixels belong to it." It is a **promptable image/video segmenter**, not a depth or 6D pose model. [Official repository](#)

Core idea. On an initial frame, positive/negative points, a bounding box, or an existing mask specify an object. The model extracts image features and predicts its segmentation. On later frames, **streaming memory** preserves information about earlier appearances and predictions, allowing the mask to propagate through motion, deformation, and some occlusions. A user can add corrective prompts later; memory does not guarantee that the instance identity will always remain correct.

Inputs and outputs. Inputs are RGB images or video plus prompts such as points, boxes, or masks. Outputs include per-frame instance masks and an inference state or object IDs for handling video targets. Pixel membership alone says nothing

about metric camera-to-object distance.

Practical workflow. (1) Indicate the screwdriver on the first frame; (2) propagate the mask over subsequent frames; (3) check identity when hands occlude it, similar tools cross, or it leaves the view; (4) supply a new prompt or invoke global redetection if tracking drifts.

Where it helps and fails. SAM 2 is useful for known-target video segmentation, pseudo-label generation, and supplying object regions to downstream pose estimators. If the command is solely “find a red-handled tool,” a separate text-grounded localizer is usually needed. Heavy occlusion, motion blur, and visually similar objects can cause drift or ID switches. SAM 2.1 denotes updated checkpoints and implementation, not a new metric-depth capability.

2.2 SAM 3 / SAM 3.1: discover instances by concept, then segment and track them

In one sentence: SAM 3 can discover image or video instances from **short text concepts or visual exemplars**, output their masks, and track them over time. It does not require a person to point to the single target first. [Official paper](#) · [Official repository](#)

Core idea. SAM 3 combines an image-level **detector** for presence and localization of a prompted concept with a **memory-based video tracker**. The components share visual representations. A concept can be a short noun phrase, such as “red-handled screwdriver,” or an image exemplar. The system can return **multiple instances matching the concept**, not necessarily the one a robot should pick. A downstream task selector must resolve duplicates and maintain the selected object’s identity.

The substantive distinction from SAM 2. SAM 2 is most naturally used when the **instance has already been specified** and its mask must be propagated. SAM 3 adds **concept-driven discovery, segmentation, and tracking**. SAM 3.1 further improves the computational organization of **multi-object video tracking** through object multiplexing. Neither SAM 3 nor 3.1 turns a 2D mask into a metric 6D pose. [Official SAM 3.1 update](#)

Inputs and outputs. Inputs are images/video and a text concept, exemplar image, or interactive visual prompt. Outputs include candidate detections, masks, and tracked identities. A long instruction with complex relations may need an upstream VLM or agent to extract concise localizable concepts, followed by spatial and task-specific instance selection.

Practical workflow. (1) Convert “pick up the red-handled screwdriver on the table” to a concise concept; (2) obtain all matching masks; (3) choose an instance from task and spatial constraints; (4) track it in video; (5) rerun concept-based detection if its identity becomes uncertain.

Where it helps and fails. Useful for language-conditioned grasping, open-category target discovery, and semantic selection. Ambiguous phrases, tiny objects, severe occlusion, and near-duplicate items remain difficult. Outputs do not contain CAD, reliable metric depth, a canonical object frame, or contact state. Published video FPS depends on hardware, resolution, and object count, and should not be substituted for actual robot closed-loop frequency.

2.3 FoundationPose: initialize, refine, and track the 6D pose of a rigid object

In one sentence: SAM tells us **which pixels belong to the screwdriver**. FoundationPose estimates **where that rigid screwdriver is and how it is oriented relative to the camera**, provided suitable object information is available. [Official paper](#) and [code](#)

What is 6D pose? A 3D translation plus a 3D rotation (often represented as an $SE(3)$ transform) is meaningful only relative to an object representation with a **defined scale, origin, and axes**. For an elongated tool, the tip, handle, and model origin are distinct: a small error at the model center can still become a consequential error at the tip.

Two operating conditions, neither fully unconditioned:

Setup	What must be prepared	Correct interpretation
Model-based	Known CAD/mesh, physical scale, object-frame convention	Use the known shape to explain current images; often the sensible choice for repeatedly manipulated tools.
Model-free	A small set of reference views and an appropriate object representation derived from them	No pre-existing CAD is required, but reference information is ; the released workflow may include construction of a neural object field, so this is not “single unknown image in, precision pose out.”

Core idea. The system first generates plausible pose **hypotheses** from the observed object region and an object prior. A learned **scorer** ranks their agreement with the observation, and a **refiner** improves promising candidates. On subsequent frames, **tracking** starts near the previous pose rather than searching the entire pose space again. This is generally more economical, but local tracking can perpetuate an earlier error. The unified formulation supports object priors obtained from CAD or reference-view reconstruction.

Inputs and outputs. A representative workflow needs RGB-D observations, camera intrinsics, an object mask, and a CAD model or reference-view-based object representation. The output is a rigid 6D pose relative to a defined object coordinate frame; video tracking produces updated poses. To control in the robot base frame, the system must still combine the result with separately calibrated camera transforms and robot configuration at image-capture time; the calibration derivation belongs elsewhere.

Practical workflow. (1) Ensure the model or reference views match the actual object’s dimensions and frame; (2) obtain an object mask and usable RGB-D; (3) **initialize** candidate poses; (4) **score and refine** the best candidates; (5) **track** in nearby frames; (6) **reinitialize** after loss or large pose discontinuities instead of indefinitely consuming a stale estimate.

Where it helps and fails. Appropriate for rigid tool grasping, tool orientation, and geometrically constrained alignment. Long occlusion, reflective or transparent materials, narrow rod-like shapes, missing depth, and inaccurate CAD can cause failure. **Symmetric geometry** may make some rotations visually indistinguishable. FoundationPose estimates pose, not grasp force, contact stability, or finger slip. Strongly deformable objects may not admit one useful rigid pose at all.

2.4 FoundationStereo: use true stereo correspondence to recover distance

In one sentence: FoundationStereo matches the same scene locations in left and right cameras, predicts a dense **disparity map**, and uses **calibrated stereo geometry** to recover metric depth. [Official repository](#)

Core idea. For rectified left and right images, the model finds where a pixel in one view is located in the other. The horizontal offset is the **disparity**. With the same cameras and baseline, closer points generally have larger disparity. The geometric conversion is summarized by $Z = f \times B / d$, for focal length f , camera baseline B , and disparity d . Learned features and context support robust stereo correspondences across different visual domains; the **metric scale still comes from the calibrated focal length and baseline**.

Inputs and outputs. A synchronized and calibrated **rectified stereo pair** is required. The primary output is disparity, from which depth and a 3D point cloud can be calculated. The model does not perform text-grounded segmentation, identify a screwdriver’s canonical frame, or directly solve its 6D pose.

Practical workflow. (1) Confirm that the cameras form a real stereo pair with known geometry; (2) rectify their images and verify overlap; (3) estimate disparity; (4) convert it to depth; (5) optionally use a mask to isolate visible object surfaces; (6) add object-conditioned pose estimation only when orientation in an object frame is genuinely needed.

What changes in Fast-FoundationStereo? Its goal is to reduce stereo inference cost while retaining much of the original model’s cross-domain accuracy. The reported approach combines knowledge distillation, architecture search under latency budgets, and structured pruning. The core output remains **stereo disparity**, not a different vision task. [Official repository](#)

Where it helps and fails. Valuable when a calibrated stereo rig exists and its depth quality can benefit from learned matching. A working RGB-D camera already providing sufficient depth does not automatically need an extra stereo network. Low texture, reflective or transparent surfaces, repetitive patterns, very small far-range disparity, and imperfect camera calibration can create errors. A head RGB-D camera and a moving wrist RGB camera **are not automatically a standard rectified stereo pair**. Reported inference speed must be rechecked with the actual GPU, resolution, and pipeline.

2.5 SAM 3D Objects: reconstruct a plausible object-shape prior from one image

In one sentence: SAM 3D Objects addresses “what might this object’s **complete shape, appearance, and spatial layout** be?” It is not a metrology-grade CAD generator with guaranteed dimensions. [Official repository](#)

Core idea. A single image reveals only some surfaces; hidden geometry is not directly measured. SAM 3D Objects uses learned shape and appearance priors to generate plausible complete 3D geometry, texture, and scene layout for a selected object, including in cluttered or occluded scenes. **Plausibility is not evidence that every invisible geometric detail is correct.**

Inputs and outputs. A typical input is one RGB image with a target region or selection. Outputs include a candidate 3D shape, appearance, and related spatial information. Distinguish five different requirements: **renderable mesh, metric object dimensions, canonical object frame, collision geometry, and mass/inertia**. A single generated mesh does not automatically satisfy all of them.

Practical workflow. (1) Select the image region representing the target; (2) obtain a candidate mesh and texture; (3) inspect scale, topology, surface normals, symmetry, and coordinate conventions; (4) validate critical grasp or contact surfaces using RGB-D, known dimensions, or additional views; (5) use only the checked geometry as a prior in simulation or pose estimation.

How does it relate to FoundationPose? SAM 3D Objects suggests what an object looks like; FoundationPose estimates where an object with a suitable prior is now. They can conceptually be combined, but scale, coordinate frame, and mesh validity must first be checked. This is **not** an officially guaranteed plug-and-play precision-CAD interface. Rough geometry for a novel object is a more appropriate expectation than millimeter-accurate assembly geometry.

2.6 Depth Anything 3: general geometric priors from one or multiple views

In one sentence: Depth Anything 3 (DA3) is a model family for **depth and spatial geometry from one or multiple images**. A depth output must still be distinguished from **reliable metric depth**. [Official repository](#)

Core idea. The models learn scene geometry from monocular or multiview observations. Depending on the configuration, they can predict depth, consistent multiview geometry, or camera-related quantities. **Different checkpoints are designed for different tasks:** the DA3 Main series covers multiple input configurations; DA3Mono targets relative monocular depth; DA3Metric targets metric monocular depth. The capabilities of one checkpoint should not be silently attributed to every model in the family.

How to use it. When only an RGB image is available, DA3 can offer an approximate front-to-back geometric prior. With multiple views, geometric consistency becomes a central consideration. For centimeter- or millimeter-sensitive manipulation, validate metric scale, cross-frame stability, and viewpoint dependence. Even a metric-trained model does not necessarily outperform a well-calibrated depth sensor or a physical stereo rig on the relevant task.

2.7 SAM 3D Body: human 3D shape, not physical contact force

In one sentence: SAM 3D Body estimates a person’s **3D body mesh and pose** from images, providing human-side geometry for motion understanding or imitation-learning pipelines. [Official repository](#)

Use and limitations. The model processes an image containing a person and predicts human geometry and pose. Turning demonstrations into robot trajectories also requires **temporal consistency, camera motion handling, scale, human-to-robot retargeting, and hand-object contact reasoning**. A human mesh is not a robot joint command, a grasp-pressure measurement, or a guarantee of object tracking under occlusion.

2.8 Grounding DINO: a modular text-to-box detector

In one sentence: Grounding DINO detects candidate **bounding boxes** from open-vocabulary text and is useful when language grounding should remain a separately replaceable component. [Official repository](#)

How to combine it. “Red screwdriver” → Grounding DINO boxes → SAM 2 segmentation and video mask propagation. Compared with SAM 3’s integrated concept-based segmentation, this is more modular, but it must manage box-to-mask association, multiple similar objects, and added inference time. **If a box is all the task needs, stop there rather than constructing a complete 3D pipeline.**

3. Comparisons that matter when choosing a model

3.1 Five commonly confused pairs

Confusable choices	How to decide
SAM 2 vs SAM 3	Known instance; continue its mask → SAM 2. Discover targets from language/exemplars and track them → SAM 3.
SAM 3 vs Grounding DINO	Integrated concept detection, masks and tracking → SAM 3. Text-conditioned boxes or a modular detector → Grounding DINO.
FoundationStereo vs DA3	Calibrated, synchronized left/right cameras and correspondence-based depth → FoundationStereo. Monocular or arbitrary-view images and learned geometric priors → appropriate DA3 variant.
FoundationPose vs SAM 3D Objects	Have an object prior; need its present rigid 6D pose → FoundationPose. Need a plausible object-shape prior first → SAM 3D Objects.
SAM 3D Body vs Objects	Human pose/body mesh → Body. General-object shapes and appearance → Objects. Neither senses contact force.

3.2 A task-driven decision path

1. **Only need object semantics and location?** Text grounding: SAM 3 or Grounding DINO. Manually specified object: SAM 2. Do not add depth by default.
2. **Need a mask over time?** Track with SAM 2 or SAM 3, and specify what triggers global reacquisition.
3. **Need metric distance?** Prefer functioning RGB-D. If a calibrated stereo pair exists, use FoundationStereo or Fast-FoundationStereo. With only monocular input, consider the relevant DA3 checkpoint while validating scale.
4. **Need complete object shape?** Prefer trusted CAD when available; otherwise consider SAM 3D Objects and validate physical dimensions and important surfaces.
5. **Need a rigid object’s actual orientation?** Given RGB-D, an object mask, and CAD/reference views, consider FoundationPose. A mask or an unchecked generated mesh alone does not satisfy these prerequisites.
6. **Need contact-rich manipulation?** Add robot proprioception, execution feedback, and force/tactile sensing when available. Vision describes only part of the state.

Minimal sufficient model principle: needing grasp orientation does not imply a need for SAM 3D; already having reliable RGB-D does not imply a need for FoundationStereo; and a VLA that consumes images directly does not make explicit pose estimation useless. Choose by the required output and the available inputs.

4. Four representative robotics workflows

4.1 Tool grasping and orientation: a known target with a 6D pose

Task: Pick up a screwdriver specified by language and orient its tip toward a desired direction.

Recommended path: task language → SAM 3 discovers candidates and selects the instance → mask + head-mounted RGB-D + the corresponding CAD/reference views → FoundationPose initializes a 6D pose → subsequent tracking and geometric feedback → robot controller. If the target and mask are already known, SAM 3 can be skipped; if pose tracking is reliable, fully rerunning the segmenter each frame may be unnecessary.

Why these components? SAM 3 resolves **which object**; depth describes **where visible surfaces are**; FoundationPose resolves **which way the rigid object faces**; the controller executes actions. Vision alone does not detect tip force. Slips or drops should trigger target verification and relocalization.

4.2 A stereo robot: estimating depth without an RGB-D camera

Task: Obtain dense depth over a workbench from two calibrated, synchronized cameras.

Recommended path: rectify the left/right pair → FoundationStereo (or assess Fast-FoundationStereo when latency matters) → convert disparity to depth → select the visible object surface with a mask. Add an object-specific pose estimator only if a canonical object orientation is required.

Decision point: stereo has a physical baseline as its scale source; monocular depth draws much more heavily on learned priors. Both can fail, but for different reasons. A moving wrist camera and a head camera do not automatically form a standard stereo input.

4.3 An unfamiliar object: obtain a shape hypothesis before planning

Task: Manipulate a tool for which no CAD model exists.

Recommended path: select the target with SAM 3 or SAM 2 → generate a candidate shape with SAM 3D Objects → use known dimensions, RGB-D, or additional views to check scale and geometry → choose either mask/point-cloud-based manipulation or construction of a suitable representation for later pose estimation.

Decision point: slots, back faces, and contact edges invisible to the camera can be hallucinated incorrectly. Precision insertion, force-fit assembly, and contact-rich rotation require further verification. For rough obstacle geometry, visualization, or object-level search, high-accuracy CAD may be unnecessary.

4.4 Human demonstration and VLA: explicit 6D pose is optional

Task: Interpret human-action video or execute manipulation using a vision-language-action policy.

Recommended path: SAM 3D Body can supply a body mesh or pose for analyzing the demonstrator. Transferring that motion to a robot additionally requires trajectory consistency, kinematic correspondence, and contact information. A VLA may generate actions directly from camera images, task language, and robot proprioception without computing an explicit FoundationPose output for every object.

When explicit geometry still matters: tip orientation, collision constraints, interpretable execution checks, and recovery after an object is dropped. SAM, depth, and 6D pose can serve as independent observation or safety-monitor interfaces. Explicit geometry and direct visual policies are **complementary approaches**, not an exclusive choice.

5. Deployment: turning useful models into a reliable system

5.1 Six interface checks

Concern	Question to answer	Common mistake
Inputs	RGB, RGB-D, stereo, text, CAD, or reference views?	Treating one RGB frame as FoundationPose's complete input.
Scale	Relative or metric depth? Is a generated mesh physically scaled?	Using unverified mesh size in precision assembly.
Frames	Are masks, depths, cameras, and object frames aligned?	Treating a camera-frame pose as a robot-base pose.
Timing	Is acquisition time distinguished from inference completion time?	Transforming an old image using the latest robot joint state.
Identity	Is it still the same object? What triggers reacquisition?	Keeping a smooth mask that has switched to another object.
Resources	Do memory, resolution, update rate, and licenses fit?	Equating a paper's single-model GPU FPS with robot control frequency.

5.2 Recovery matters more than an attractive single-frame result

Occlusion: short predictions can be useful, but a long-invisible pose must not be labeled a precise fresh observation. **Symmetry:** some rotations are intrinsically ambiguous from the current view. **Thin, transparent, or reflective objects:** both depth and mask quality may degrade. **Abrupt motion or dropping:** local tracking can diverge; redetect, verify the instance, and reinitialize. **Precision contact:** even accurate 6D geometry does not reveal force, friction, or microslip.

The robot should separately track whether the object is visible, the geometry is trustworthy, the observation is fresh, and the target ID is continuous. Slower action, active viewpoint adjustment, or halting contact are downstream system decisions—not automatic guarantees of a foundation model.

5.3 When should the heavy models run?

Semantic search triggered by events (task start or target loss) can run infrequently; **continuous mask or pose tracking** can process incoming camera frames; a **controller** often needs more timely proprioception and state predictions. Throughput, per-model latency, and the age of information when an action takes effect are different quantities. Do not automatically execute SAM 3, FoundationStereo, SAM 3D Objects, and FoundationPose sequentially on every frame.

6. Frequent questions and the final takeaway

Q1: Can SAM 3 replace FoundationPose? No. The former mainly provides semantic candidates and masks; the latter estimates an object-conditioned rigid 6D pose.

Q2: Is model-free FoundationPose truly prior-free? No. It removes the need for pre-existing CAD but still requires reference views and, in the released workflow, preparation of an object representation.

Q3: If reliable RGB-D is available, do I need FoundationStereo? Usually not. Add stereo only when actual sensors and task quality make it beneficial.

Q4: Can a SAM 3D Objects mesh be used immediately in FoundationPose? Not without qualification. Validate physical scale, coordinate convention, geometry, and task precision before treating it as a prior.

Q5: Can a single RGB image give precise absolute depth? Learned monocular models can estimate a prior, and metric-specific checkpoints target scale; neither justifies assuming unverified precision for robot manipulation.

Q6: Does accurate 6D pose mean a successful grasp? No. Pose is geometric information; contact force, grip stability, and slip need other evidence.

Remember this map: SAM finds, segments, and tracks objects; FoundationStereo measures stereo disparity for depth;

DA3 infers geometry from monocular/multiview images; SAM 3D proposes 3D shape; FoundationPose estimates a rigid object's current pose. Start with the smallest necessary toolset, then add sensors, tracking, and recovery as the task demands.

Official references

1. SAM 2 — Segment Anything in Images and Videos (Meta, official repository); paper.
2. SAM 3 — Segment Anything with Concepts (Meta, official paper); official repository with SAM 3.1 updates; SAM 3.1 announcement.
3. FoundationPose (NVIDIA, CVPR 2024, official repository); paper.
4. FoundationStereo (NVIDIA, CVPR 2025, official repository).
5. Fast-FoundationStereo (NVIDIA, CVPR 2026, official repository).
6. SAM 3D Objects (Meta, official repository); SAM 3D Body (Meta, official repository).
7. Depth Anything 3 (ByteDance Seed, official model-family repository).
8. Grounding DINO (IDEA-Research, official repository).

Provenance: adapted from the Foundation of Embodied AI course's Visual Perception and State Estimation notes (CC BY 4.0). Third-party models, code, weights, and materials retain their own licenses. Revised October 2026; always check the specific model version's official interfaces, licenses, and performance conditions.