Computer Vision
Surgical Applications

00970222

Final Project Report

Noam Carmon
Raz Biton
Shahar Hillel

1 Phase 1: Synthetic Data Generation

1.1 Data Generation Pipeline

The synthetic pipeline generates labeled surgical tool images through scene setup, randomized lighting, tool placement, keypoint extraction, 2D projection with post-projection corrections, and annotation. The pipeline is:

Scene Setup and Object Loading

Scenes begin in a clean Blender environment, where surgical tools are imported as 3D mesh objects. A point light is randomized in position and intensity, and HDRI environment lighting is applied with random rotation. Material properties (e.g., roughness, metallic, IOR) are varied to simulate realistic reflections and appearances.

3D Keypoint Generation (Ground Truth)

Keypoints are computed per tool using tailored logic based on 3D mesh geometry; the method varies by tool category. **Despite minor keypoint drift, auto-labeling preserves tool structure and pose well**, so we prefer it over manual labeling.

Keypoint Computation for Needle Holder

Tool	**Keypoint(s)**	**Computation Method (Steps)**
Needle Holder	top_left, top_right	1. Select top 15% of vertices by Z-axis. 2. Pick leftmost (min X) and rightmost (max X) among them.
Needle Holder	bottom_left, bottom_right	1. Select bottom 15% of vertices by Z-axis. 2. Pick leftmost and rightmost by X-axis.
Needle Holder	middle_left, middle_right	1. Filter mid-range Z values (30–70%). 2. Choose vertex near the centroid of left/right X-percentile subsets.
Needle Holder	joint_center	1. Average the two top tips and the two middle points. 2. move 30% from the tip-center toward the middle-center, then choose the closest vertex.

Keypoint Computation for Tweezers

Tool	**Keypoint(s)**	**Computation Method (Steps)**
Tweezers	bottom_tip	1. Find the vertex with lowest Z value. 2. Assign as bottom_tip.
Tweezers	top_left, top_right	1. Split vertices into left/right arms using median X. 2. In each arm, select top 20% Z vertices, pick extreme X.
Tweezers	mid_left, mid_right	1. Find median Z in each arm. 2. Select vertex closest to it.

Keypoint Computation for Fallback Tools

Tool	**Keypoint(s)**	**Computation Method (Steps)**
Fallback (Generic)	top_left, top_right	1. Filter top 20% Z vertices. 2. Pick min X (left) and max X (right).
Fallback (Generic)	bottom_left, bottom_right	1. Filter bottom 20% Z vertices. 2. Pick min X (left) and max X (right).

2D Projection

We project the 3D keypoints to 2D using the camera intrinsics and camera poses sampled around the tools. We fix left/right labels right after projection and again at the end.
Corrections: Clamp any keypoint outside the object's bounding box to the edge; if a point is inside the box but off the mask, move it to a plausible location (needle-holder points along side/center lines; tweezers arm points along their arms), then snap off-mask points to the nearest foreground pixel inside the box (skip this snap for the tweezers tip). For the tweezers `bottom_tip`, only ensure it lies inside the box (clamp if needed); do not snap it onto an arm.

View-Dependent Labeling

To maintain consistent left/right annotation across views, projected X-values are compared, and labels swapped if needed.

Rendering and Annotation

We compute each object's bounding box from instance segmentation, composite the RGBA render over a photo background with a soft surgical spotlight, save the background-composited RGB image and a visualization image, and write one JSON record per image with the saved image name and, for each object, its id/category, 2D keypoints, and bounding box.

Final Dataset Statistics

Category	**Items**	**Images / item**	**Total images**
Single objects	25	10	250
Tool pairs	150	10	1500
Total	**175**		**1750**

1.2 Implemented Variations

Instrument positioning and orientation

We place the camera around the tools but prefer a top-down view.

Lighting conditions (intensity, color, direction)

Each render randomizes a point light's position (direction) and intensity. We also use an HDRI environment with random rotation and vary the world/background brightness each try.

Background variation

Rendered frames are composited over randomly chosen photographic backgrounds.

Creative variation 1: Surgical-room spotlight

We add a soft, radial "OR" spotlight centered on the tools to emulate operating-room lighting and emphasize instrument visibility.

Creative variation 2: Multi-tool images

For every needle-holder and tweezers pair, we generate a joint top-down scene to reflect common two-instrument frames and introduce realistic occlusions.

1.3 Domain Gap Analysis

Aspect	**Synthetic**	**Real**	**Domain Gap**
Tool Usage Context	Tools are alone and not used by hands.	Tools are held by people and touch tissue.	No hands or actions shown.
Background Semantics	Backgrounds are generic, not medical.	Surgical room environment.	Missing medical setting cues.
Texture and Surface Realism	Surfaces look clean and perfect.	There are reflections, scratches, stains, fluids.	Too perfect compared to real life.
Scene Complexity and Occlusion	Simple scenes with few overlaps.	Crowded scenes with hands and tools overlapping.	Less clutter makes training easier than it should be.

1.4 Discussion of Challenges and Key Findings

Challenges

- Annotation Strategy:
Manual labeling was unscalable; automation required mesh understanding and consistent landmark definitions.
- Ground Truth Modeling:
Bounding boxes lacked precision; keypoints added detail but required careful design.
- Domain Alignment:
Synthetic scenes missed surgical cues like hands, lighting, and tissue.

Key Findings

- Reliable Auto-Annotation:
Ground truth labels were generated using mesh metadata—accurate and scalable.
- Multi-Tool and Multi-Pose Scenes:
Improved realism and contextual diversity.
- Spotlight Lighting:
Enhanced tool visibility and mimicked surgical illumination.
- Pose and Camera Variation:
Supported diverse perspectives and improved generalization.

2 Section 2

3 Section 3