SVO2-SAM3 Analyzer

Reference Guide

Comprehensive documentation for the SVO2-SAM3 video analysis pipeline

Version 1.0 | January 2026

Table of Contents

1. Overview

SVO2-SAM3 Analyzer is an end-to-end processing pipeline for analyzing video data from Stereolabs ZED 2i stereo cameras using SAM 3 (Segment Anything Model 3) for AI-powered object detection and segmentation.

The system provides a complete workflow from raw SVO2 video files to structured output in industry-standard formats including KITTI, COCO, JSON, and CSV.

2. Processing Pipeline

The system processes data through 5 sequential stages:

Stage	Input	Output	Description
1. Extraction	SVO2 file	Images, depth, point clouds	Extracts frames from ZED recordings
2. Segmentation	Extracted frames	2D detections, masks	Runs SAM 3 object detection
3. Reconstruction	2D detections + depth	3D bounding boxes	Projects detections to 3D space
4. Tracking	3D detections	Object tracks	Links objects across frames
5. Export	All results	KITTI/COCO/JSON/CSV	Generates output files

3. API Endpoints

3.1 File Management (/api/files)

Endpoint	Method	Description	Expected Result
/browse	GET	List SVO2 files	JSON array of file paths
/metadata/{path}	GET	Get SVO2 file details	Frame count, resolution, duration
/validate	POST	Check file integrity	Validation status and errors

3.2 Job Management (/api/jobs)

Endpoint	Method	Description	Expected Result
/create	POST	Create new job	Job ID and initial status
/	GET	List all jobs	Paginated job list with status
/{job_id}	GET	Get job details	Full job info with progress
/{job_id}/start	POST	Begin processing	Status changes to processing
/{job_id}/pause	POST	Pause job	Status changes to paused
/{job_id}/resume	POST	Resume paused job	Continues from pause point
/{job_id}/cancel	POST	Cancel job	Status changes to cancelled
/{job_id}/results	GET	Get processing results	Detections, tracks, statistics
/{job_id}	DELETE	Delete job and data	Job removed from database

3.3 Configuration (/api/config)

Endpoint	Method	Description	Expected Result
/object-classes	GET	List detection classes	Preset + custom classes
/object-classes	POST	Add custom class	New class added
/presets	GET	List config presets	Available templates
/presets	POST	Save preset	New preset created
/model-info	GET	SAM3 model details	Model variant, VRAM info
/system	GET	System configuration	Current settings

3.4 Export (/api/export)

Endpoint	Method	Description	Expected Result
/{job_id}	POST	Trigger export	Export task started
/{job_id}/status	GET	Check export status	Progress and completion
/{job_id}/kitti	GET	Download KITTI format	ZIP file
/{job_id}/coco	GET	Download COCO format	JSON file
/{job_id}/json	GET	Download JSON format	Full results JSON
/{job_id}/csv	GET	Download CSV summary	Statistics spreadsheet
/{job_id}/{format}	DELETE	Remove export files	Files deleted

3.5 Health Check

Endpoint	Method	Description	Expected Result
/health	GET	System health check	Status of DB, Redis, GPU
/	GET	API info	Version and basic info

4. Job Status Values

Status	Description
pending	Job created, waiting to start
extracting	Stage 1: Extracting frames from SVO2
segmenting	Stage 2: Running SAM 3 detection
reconstructing	Stage 3: Building 3D bounding boxes
tracking	Stage 4: Linking objects across frames
exporting	Stage 5: Generating output files
completed	All stages finished successfully
paused	Job paused by user
cancelled	Job cancelled by user
failed	Job failed with error

5. Export Formats

5.1 KITTI Format (ZIP)

Standard autonomous driving dataset structure:

Directory	Contents	Format
image_2/	Left camera RGB images	PNG
image_3/	Right camera RGB images	PNG
depth/	Depth maps	16-bit PNG
velodyne/	Point clouds	BIN
label_2/	3D annotations	TXT
oxts/	IMU/GPS data	TXT
calib/	Camera calibration	TXT

5.2 COCO Format (JSON)

Standard computer vision annotation format containing image metadata, 2D bounding boxes, segmentation masks (RLE encoded), and category information.

5.3 JSON Format

Full processing results including complete detection data, 3D bounding boxes, track assignments, and confidence scores.

5.4 CSV Format

Summary statistics spreadsheet with frame IDs, timestamps, detection counts, track counts, per-class counts, and processing times.

6. Configuration Parameters

6.1 SAM 3 Model Settings

Parameter	Default	Description
model_variant	sam3_hiera_large	Model size (tiny/small/base/large)
confidence_threshold	0.5	Minimum detection confidence
iou_threshold	0.7	NMS IoU threshold
batch_size	4	Frames per batch

6.2 Model VRAM Requirements

Variant	VRAM Required
sam3_hiera_tiny	4 GB
sam3_hiera_small	8 GB
sam3_hiera_base	12 GB
sam3_hiera_large	16 GB

6.3 Extraction Settings

Parameter	Options	Description
depth_mode	NEURAL, ULTRA, QUALITY, PERFORMANCE	Depth estimation quality
frame_skip	Integer (0+)	Skip N frames between extractions
start_frame	Integer	First frame to process
end_frame	Integer	Last frame to process

6.4 Tracking Settings (ByteTrack)

Parameter	Default	Description
track_thresh	0.5	High confidence detection threshold
match_thresh	0.8	Track-detection matching threshold
track_buffer	30	Frames to keep lost tracks

7. Detection Output Formats

7.1 2D Detection

```
{"bbox": [x1, y1, x2, y2], "confidence": 0.95, "class_id": 1, "class_name": "person",
"mask_path": "path/to/mask.png"}
```

7.2 3D Bounding Box

```
{"center": [x, y, z], "dimensions": [length, width, height], "rotation_y": 0.5,
"confidence": 0.92}
```

7.3 Track

```
{"track_id": 1, "class_name": "car", "start_frame": 10, "end_frame": 150, "trajectory":
[[x, y, z], ...]}
```

8. CLI Commands

Command	Description	Expected Result
uvicorn backend.app.main:app --host 0.0.0.0 --port 8000	Start Backend	API at localhost:8000
celery -A worker.celery_app worker --loglevel=info	Start Worker	Worker processes tasks
cd frontend && npm run dev	Start Frontend	UI at localhost:5173
alembic upgrade head	Run Migrations	Database schema updated

Command	Description	Expected Result
python scripts/download_sam3.py	Download Model	Weights saved to models/
python scripts/verify_gpu.py	Verify GPU	CUDA info displayed

9. Environment Variables

Variable	Description	Example
POSTGRES_HOST	Database host	localhost
POSTGRES_PORT	Database port	5432
POSTGRES_DB	Database name	svo2_analyzer
REDIS_HOST	Redis host	localhost
REDIS_PORT	Redis port	6379
DATA_ROOT	Base data directory	/data
SVO2_DIRECTORY	Input SVO2 files	/data/svo2
OUTPUT_DIRECTORY	Processing output	/data/output
SAM3_MODEL_VARIANT	Model to use	sam3_hiera_large
LOG_LEVEL	Logging verbosity	INFO

10. Error Codes

Code	Description	Resolution
FILE_NOT_FOUND	SVO2 file doesn't exist	Check file path
INVALID_SVO2	Corrupted or unsupported file	Re-export from ZED software
GPU_OUT_OF_MEMORY	Insufficient VRAM	Use smaller model or reduce batch
ZED_SDK_ERROR	ZED SDK issue	Verify SDK installation
TASK_TIMEOUT	Processing exceeded limit	Split into smaller jobs
DATABASE_ERROR	Database connection failed	Check PostgreSQL status
REDIS_ERROR	Message broker unavailable	Check Redis status

11. Typical Processing Results

Expected results for a 1000-frame SVO2 recording:

Metric	Typical Value
Extraction time	2-5 minutes

Metric	Typical Value
Segmentation time	10-30 minutes (GPU dependent)
Reconstruction time	1-3 minutes
Tracking time	< 1 minute
Export time	1-2 minutes
Output size (KITTI ZIP)	500 MB - 2 GB
Detections per frame	0-50 (scene dependent)
Tracks generated	10-500 (scene dependent)

12. Docker Services

Service	Port	Description
postgres	5432	PostgreSQL database
redis	6379	Message broker
redis-commander	8081	Redis web UI (debug)
pgadmin	5050	Database admin UI (debug)

Start services:

```
docker-compose up -d
```

Start with debug tools:

```
docker-compose --profile debug up -d
```

13. Frontend Pages

Page	Route	Description
Home	/	Feature overview and quick start
Jobs	/jobs	Job list and management
Job Detail	/jobs/:id	Job progress and results
Settings	/settings	System configuration