Address 120 e 8th st unit 810
Los Angeles CA
90014

Contact matt@teleidoscope.com
310.774.1123

www.teleidoscope.com

Teleidoscope

Visual tracking software for resource constrained devices

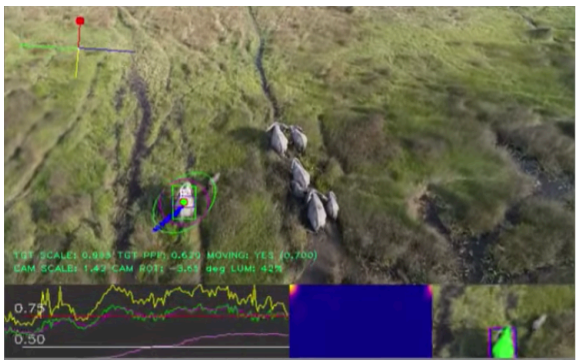
Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

Visual Target Tracking

Visual target tracking refers to following a target that is designated by a bounding box on a single frame. In autonomous and semi-autonomous systems, this is often used for applications such as “Smart Track”, “Follow Me”, and Automatic Object Tracking.

Teleidoscope has developed its own proprietary visual target tracker called RAD which stands for Recoverable Adaptive Discriminative Appearance Model Tracking. Teleidoscope’s RAD Tracker can track any patch of an image, whether it’s an elephant, an aircraft, a cloud, or a patch of mud on the ground.

Below are two demonstrations of Teleidoscope’s Visual Target Tracker paired with it’s single point designator which is discussed below.



RGB – https://youtu.be/X0o_XXqOmeQ

LWIR - https://youtu.be/EGATZIX-XVI

RAD Initialization Options

RAD’s internal model requires a bounding box to initialize, but this bounding box can be designated using the following modalities:

- explicit bounding box
- single point
- detection (i.e bounding box with ancillary data).

Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

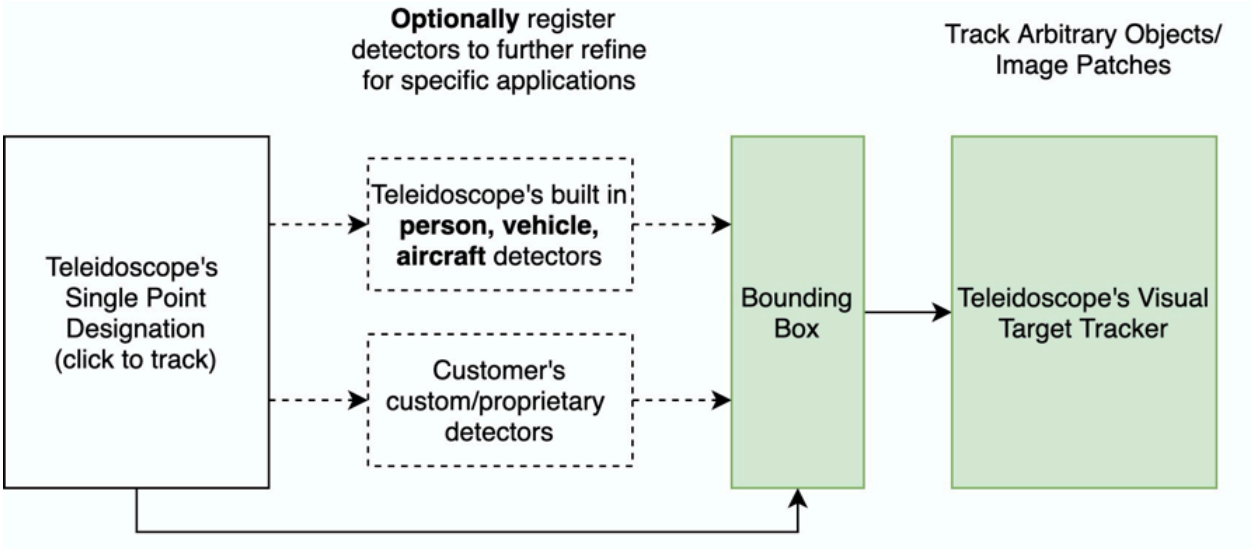
Explicit Bounding Box Initialization

Designating a portion of the image via an explicit bounding box allows the operator to track very specific regions of the image. This can be useful in cases where the object is very difficult to see, or if the operator is looking to track a specific piece of an object e.g: tracking a window of a building rather than the entire building.

To the right is an example of a case where explicit bounding box initialization is useful. There is a bike rider which is very difficult to see, and would not be picked up by a detector, or Teleidoscope's Single Point Designator.

Explicit Box – https://youtu.be/RmBJBI1xhOs

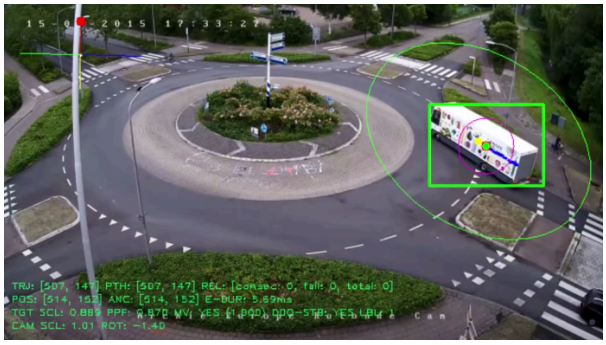
The pipeline components involved for explicit bounding box initialization can be seen below and are highlighted in green.



Detection Based Initialization

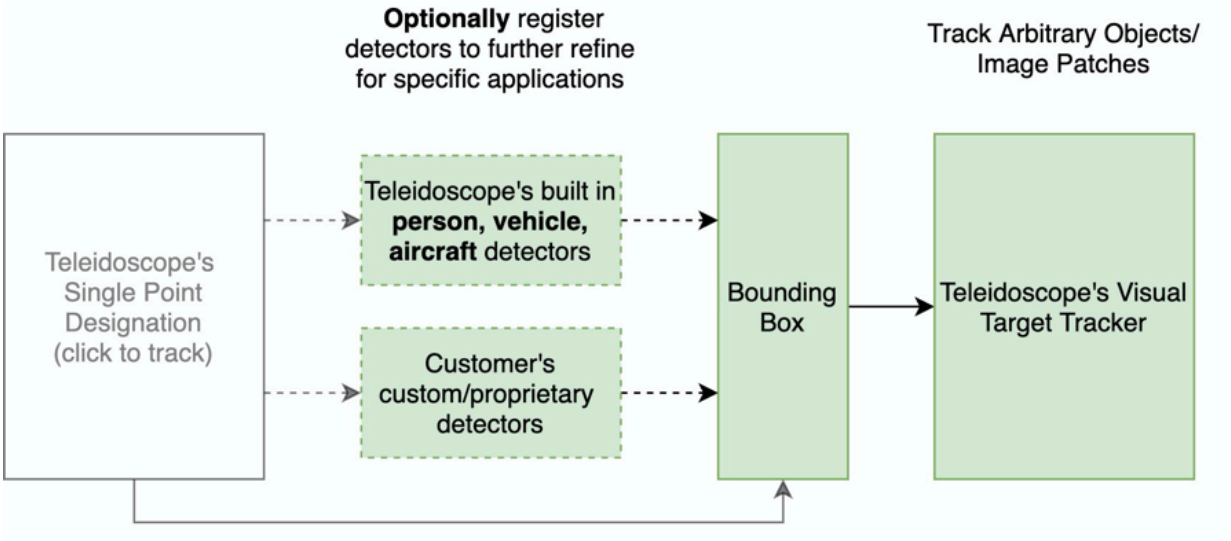
Detection Based Initialization is useful in fully autonomous systems. This is often referred to as automatic detection tracking, and does not require user input, but is limited to tracking specific types of objects. This is ideal for long-term surveillance of an area. An example use case is automatically tracking and counting of vehicles or pedestrian traffic.

Other solutions lock customers into using the provided detectors, but with Teleidoscope's Visual Pipeline it is easy for customers to pair their own detectors.



Detector Init – https://youtu.be/m0ioI3O5I7A

This decoupling allows a customer to continue to refine their own machine learning detectors while reaping the benefits of Teleidoscope's Visual Target Tracker. The customer is able to use Teleidoscope's built in detectors if they wish, but are not locked in.

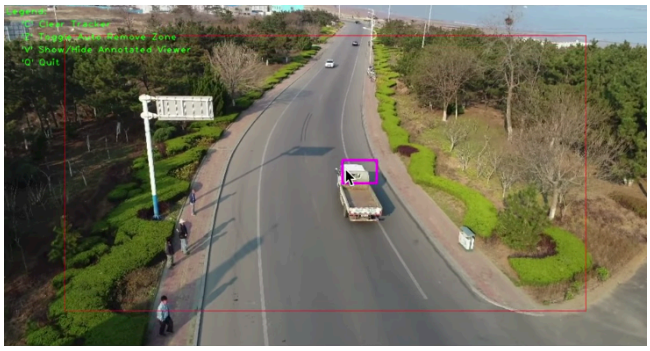


Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

Single Point Initialization

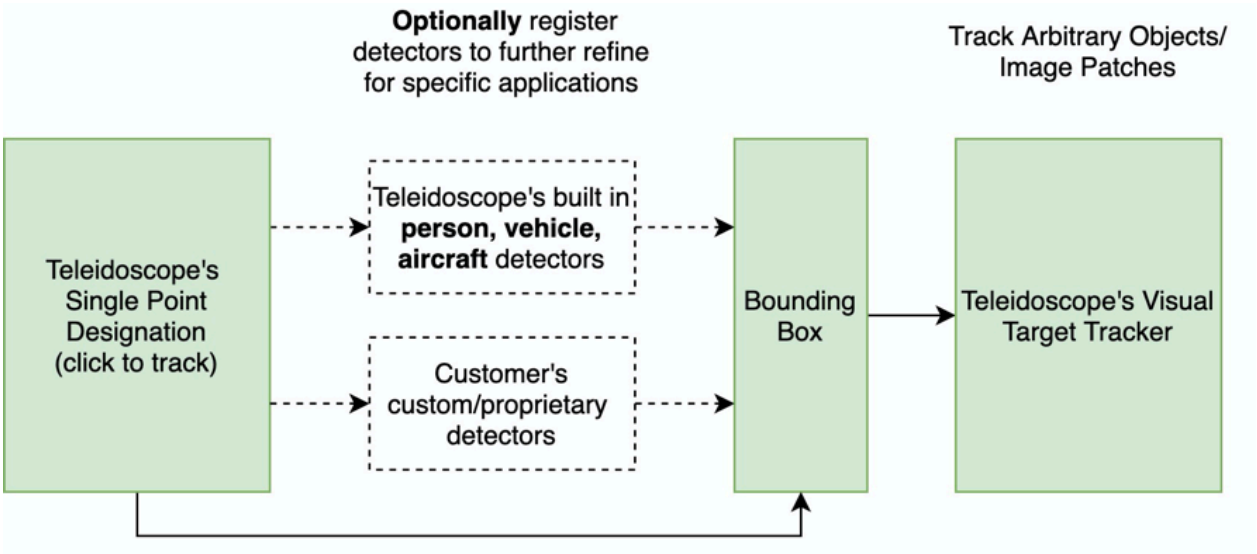
Single point initialization makes it easy for an operator to select objects and image patches in live video. This can optionally be paired with a detector for specific use cases where the operator will always be tracking a particular object like a person, or a vehicle.

To the right is an example of a case where Single Point Initialization is useful. This example is not paired with a detector, and is only using the single point designator to specific the bounding box.



Detector Init – https://youtu.be/PSmw7Nh-1lE

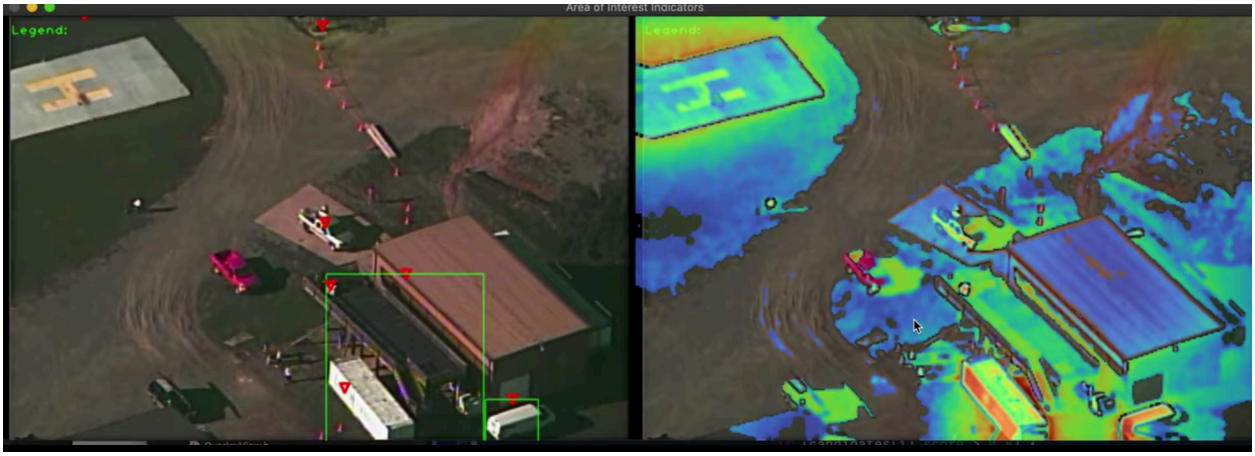
The pipeline components involved for Single Point Initialization can be seen below and are highlighted in green.



Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

Additional PED Solutions

Area Of Interest



Trimmed AOI – https://www.youtube.com/watch?v=MTBJ_uaJXb4

Teleidoscope's Area Of Interest solution provides operators with additional information on the scene, and highlights both small and large areas that may be of interest. This solution can also be used as a component of obstacle avoidance systems.

Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

Modularity and Portability

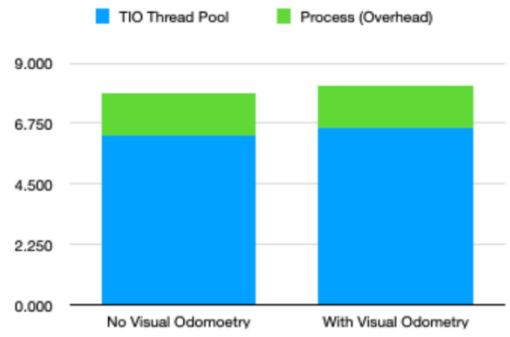
Software Development Kit (SDK)

Teleidoscope's visual target tracking software is available as a C++ SDK, python SDK, and comes with a set of samples and testing utilities.

Resources – CPU vs GPU

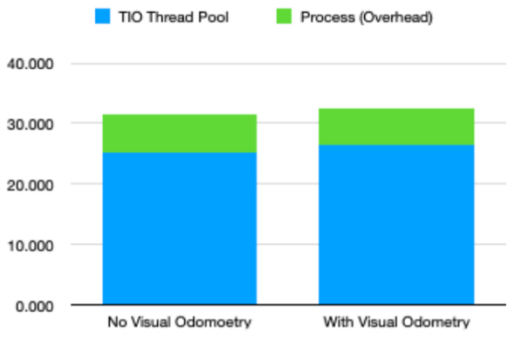
Teleidoscope's visual target tracker was built from the ground up to run on resource constrained devices. This tracker is designed specifically to run on CPU which means it can run a wider range of devices than GPU specific solutions. For UAVs with a GPU, more GPU resources can be dedicated to other tasks.

CPU Utilization



2.2 GHz 6-Core Intel Core i7 (12 Threads)

	TIO Thread Pool	Process (Overhead)
No Visual Odomoetry	6.282	1.604
With Visual Odometry	6.592	1.560



1.2GHz 4-Core ARM Cortex-A53 (4 Threads)

	TIO Thread Pool	Process (Overhead)
No Visual Odomoetry	25.127	6.416
With Visual Odometry	26.367	6.241

*Only 2 threads used by tracker

Camera

Teleidoscope's visual target tracker does not have specific camera requirements and works with standard RGB, and IR frames.

Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

What sets Teleidoscope's Target Tracker Apart

Teleidoscope's target tracking solution is easy to use, interoperable with built in, and custom machine learning detectors, and is camera and platform agnostic.

Though the above are important differentiators, what really sets Teleidoscope's solution apart is its performance, which is due to Teleidoscope's core IP called a Recoverable Adaptive Discriminative Appearance Model(RAD) Tracker. This is the result of ongoing R&D efforts.

RAD Self Diagnostics and Recovery vs State of the Art (SotA)

Current SotA trackers use a binary pass/fail approach when tracking by computing a confidence score and comparing it against a fixed threshold. A tracked object is considered tracked while above the threshold and lost when it falls below the threshold. Avoiding target loss is a top priority for a tracker because once a target is lost the tracker must be reinitialized. To avoid spurious loss from sudden appearance changes, most trackers will lower the threshold to consider an object tracked. This workaround is often seen in the KCF tracker, a popular tracker used by companies like DJI. Overtime this can lead to appearance contamination from the background or partial occlusions of the tracked object.

When this contamination occurs, the tracker may slowly drift without realizing it's no longer tracking the original object (blind drift). Whether or not this can be tolerated depends on what the tracker is being used for. If the tracker is being used to track faces in a photo application, drift likely isn't an issue. However, if the tracker is being used in a critical system (i.e. defense), correctly reporting target loss and ensuring the correct target is being tracked is essential.

This complicates tracking because it forces trackers to choose between aggressively failing or accepting drift. RAD attempts to solve this problem by using a per-target auto calibrated

confidence threshold and adding two additional intermediate states (weak and unstable) that guide tracking behavior:

1. Strongly tracked
2. Weakly tracked
3. Unstable
4. Lost

Strongly Tracked State: Being in the Strongly Tracked state has the similar implications as the tracked state reported by other trackers.

Weakly Tracked State: When in the Weakly tracked state RAD will temporarily disable updates to the appearance model and start estimating two positions.

Unstable State: When in the Unstable state RAD will stop model updates and model estimators then begin the target re-localization and recovery process before considering a target permanently lost.

These extra states act as an early warning indicator allowing applications to respond to situations where target loss is more likely but is still being tracked. An example usage might be within the detect-to-engage sequence of a fire-control system. If the target suddenly becomes weakly tracked but is still trackable, the FCS operator may wish to defer engagement to avoid the scenario where the target is lost immediately after engaging.

Hybrid Model Estimator

Unlike most SotA trackers which only use a single model for estimation, RAD uses at least 3 and up to 5. The first is the RAD Appearance Model, which consists of an N dimensional blob that encodes information about the color, texture and shape of the target.
The second is the RAD target motion model, which models the movement of the target and its surroundings. This is used to compute target trajectories relative the last strong update.

Address 120 e 8th st unit 810 Los Angeles CA 90014
Contact matt@teleidoscope.com 310.774.1123
www.teleidoscope.com

The third is the RAD camera motion model, which estimates the motion of the camera relative to the target. A homography is computed for each frame and used to reconstruct the targets traveled path in a homogeneous coordinate space. RAD stores 3 RAD Appearance Models, the ‘fixed model’ from initialization, the ‘stable model’ from the last update above the strongly tracked upper threshold and the ‘active model’ from each strongly tracked update or recoverably weak update.