Contents

[User story: Video source selection UI [Pending] 2](#_Toc450556396)

[User Story: Supported file types 2](#_Toc450556397)

[User Story: User-friendly video file input [Pending] 2](#_Toc450556398)

[User Story: User-friendly camera feed input [Pending] 3](#_Toc450556399)

[User Story: Object selector 3](#_Toc450556400)

[User Story: Object tracking routine hub 3](#_Toc450556401)

[User Story: Main routine UI [Pending] 4](#_Toc450556402)

[User Story: Object trajectory prognosis [Pending] 4](#_Toc450556403)

[User Story: Tracked object highlight 5](#_Toc450556404)

[User Story: Tracking process logging 5](#_Toc450556405)

[User Story: Command arguments 5](#_Toc450556406)

[User Story: Debug mode 6](#_Toc450556407)

[User Story: Template algorithms [Pending] 6](#_Toc450556408)

[User Story: Basic training video generator 7](#_Toc450556409)

[User Story: Advanced training video generator 7](#_Toc450556410)

# User story: Video source selection UI [Pending]

As a user  
I want select between several video source types  
So that I am able to use video sources situationally

Summary:

I want to be able to select between a camera feed and video file.

Acceptance Criteria:

Has options for Video file and Camera feed, that are mutually exclusive and cannot be selected both at once. They cannot be switched at runtime without exiting the tracking process. Upon selection, the user has an option to go back to source type selection.

# User Story: Supported file types

As a user  
I want to use some file types  
So that I am able to use this software

Summary:

I want to be able to use AVI-file format.

Acceptance Criteria:

Supports AVI-file format.

# User Story: User-friendly video file input [Pending]

As a user  
I want to select the input video in a user-friendly way  
So that the flow of the application does not hinge on command prompt arguments

Summary:

When selecting a video file I want to be able to select it in a user-friendly manner, e.g. via an open file dialog window.

Acceptance Criteria:

Supports mouse input and folder navigation and possibly drag-and-drop. Upon selection, the user has an option to go back to source file selection or (if implemented as an open-file-dialog, has an option to summon the dialog again).

# User Story: User-friendly camera feed input [Pending]

As a user  
I want to select the camera feed in a user-friendly way  
So that the flow of the application does not hinge on command prompt arguments

Summary:

If I have several cameras, when selecting a camera feed I want to be able to select it in a user-friendly manner, e.g. via dialog window with miniatures of camera feeds.

Acceptance Criteria:

Dialog shows miniatures of camera feeds (still or animated) and supports mouse input for selection of feeds. Only one feed can be selected at once. Upon selection the user has an option to go back to feed selection (or summon the dialog).

# User Story: Object selector

As a user   
I want to be able to select the tracked object  
So that I have control over the tracking process

Summary:

I can specify the ROI either on an active feed or on a still frame. The resulting part of an image will be analyzed for features of the object later used in tracking.

Acceptance Criteria:

* Supports mouse input;
* The ROI is visible;
* Works both for paused and running video;
* Features are processed only after the selection;
* If applied to an active feed, only the last frame in the selector activity period will be processed.

# User Story: Object tracking routine hub

As a user  
I want the object to be tracked  
So that so that the main purpose of the software will be served

Summary:

The object position is acquired with the algorithm that based on object features detects the presence of the object in the frame and outputs its coordinates for things like logging, highlighting. All the information transfer is being performed by one routine hub.

Acceptance Criteria:

* Capable of taking frames for analysis from different sources – video file or camera feed;
* Performs intermediary data transfer between algorithms – frames sent to algorithms, masks to mask handlers, coordinates to trajectory handlers and logging;
* Performs object highlight drawing – highlights object by given coordinates;
* Performs prognosis – given previous object coordinates predicts next object position/area;

# User Story: Main routine UI [Pending]

As a user  
I want to have a set of defined controls for the tracking process  
So that I can utilize them to have more control over the tracking process

Summary:

After the tracking routine has started, the user is shown a screen that contains controls for pausing the routine, showing debug output and stopping the routine entirely. This includes the rework of Object selector, as the existing OpenCV based implementation is no longer valid.

Acceptance Criteria:

* Has a pause control;
* Has a routine stopping control;
* Has a debug output control;
* Has a debug output screen;
* Supports Object selection;

# User Story: Object trajectory prognosis [Pending]

As a user  
I want to have a prognosis of the object position   
So that I could predict object movements and use that to my advantage

Summary:

An algorithm based on previous positions of the object (and possibly optical flow) carries out the prognosis of the object position. This feature should be also added to the tracking hub.

Acceptance Criteria:

* Performs position prognosis based on previous states of the object;
* Provides possible position (or area);

# User Story: Tracked object highlight

As a user  
I want to see the object highlighted  
So that I can see the results of tracking

Summary:

The object after being tracked is being highlighted in an explicit way, e.g. a colored box, contour, cropping or arrows.

Acceptance Criteria:

Highlights the given ROI with a rectangle or with contour/contour fill highlight;

# User Story: Tracking process logging

As a user   
I want to have the coordinates of the object and specifics of the process output into a file   
So that I can see the results of tracking

Summary:

The position of the object is output into a file in an explicit way.

Acceptance Criteria:

Writes object coordinates and initial settings to a file from settings, also possible exceptional situations should be logged as well.

# User Story: Command arguments

As a developer  
I want the application to support command prompt arguments  
So that I select the input video after compilation

Summary:

In order to select video sources at runtime, I need a way to dynamically send parameters to the program, in this occasion this will be performed via command arguments.

Acceptance criteria:

Allows settings:

Source type – video file and camera feed, source location – video source file path in case of video file, zero-based feed number in case of camera feed, log location – file location, start with debug flag (or show debug at all).

# User Story: Debug mode

As a developer  
I want to have a debug mode  
So that I can tell wtf is going on in the program

Summary:

In order to see the state of image-like variables in runtime I need a uniform way to show them, as well as show multiple possible ROIs on the frame to be able to tweak the detection algorithm better. For this I need a debug mode.

Acceptance criteria:

* Supports registration of images for debug display;
* Supports registration of variables for debug display;
* Shows up as a separate named window (oversized collected image could be downscaled or navigated with scrollbars);

# User Story: Template algorithms [Pending]

As a developer  
I want to have detection algorithms to be implementing the same interface  
So that more of them could be added in future iterations

Summary:

Several detection algorithms that implement an interface allow for better scalability of the program.

Acceptance Criteria:

* Interface contains methods that take arguments common for all algorithms and return a common type;
* Algorithms implement this interface;

# User Story: Basic training video generator

As a developer  
I want a basic training videos generator (random motion on static background)  
So that I can perform smoke tests without additional effort

Summary:

A standalone program that generates a basic training video based on input images and/or videos. The result contains a simulation of a random motion of an object and a file containing object positions.

Acceptance Criteria:

This standalone program takes command arguments that can set:

Background source type – image file, video file or camera feed, background source – video or image source file path or feed number, sprite source – image file path, output video file path, output log file path, and debug flag. The program would then overlay the sprite in a random start location and then move it by a random interval with a random deviation from its original orientation within the frame bounding box, log object center coordinates and repeat. If a video file was selected as background source, when it runs out, the process will stop and the program will finish, otherwise the program will exit by a designated keystroke.

# User Story: Advanced training video generator

As a developer  
I want an advanced training videos generator (random motion on dynamic background)  
So that I can perform smoke tests without additional effort

Summary:

A standalone program that generates a basic training video based on input images and/or videos. The result contains a simulation of a random motion of an object on a randomly moving background.

Acceptance Criteria:

Being a direct upgrade to the previous program, this program needs to have an option of generating a dynamic background, when an image was selected as a background. This will be achieved by applying the same motion principle that is moving the sprite around the frame, with an only exception being that the bounding box would be smaller than the source background.