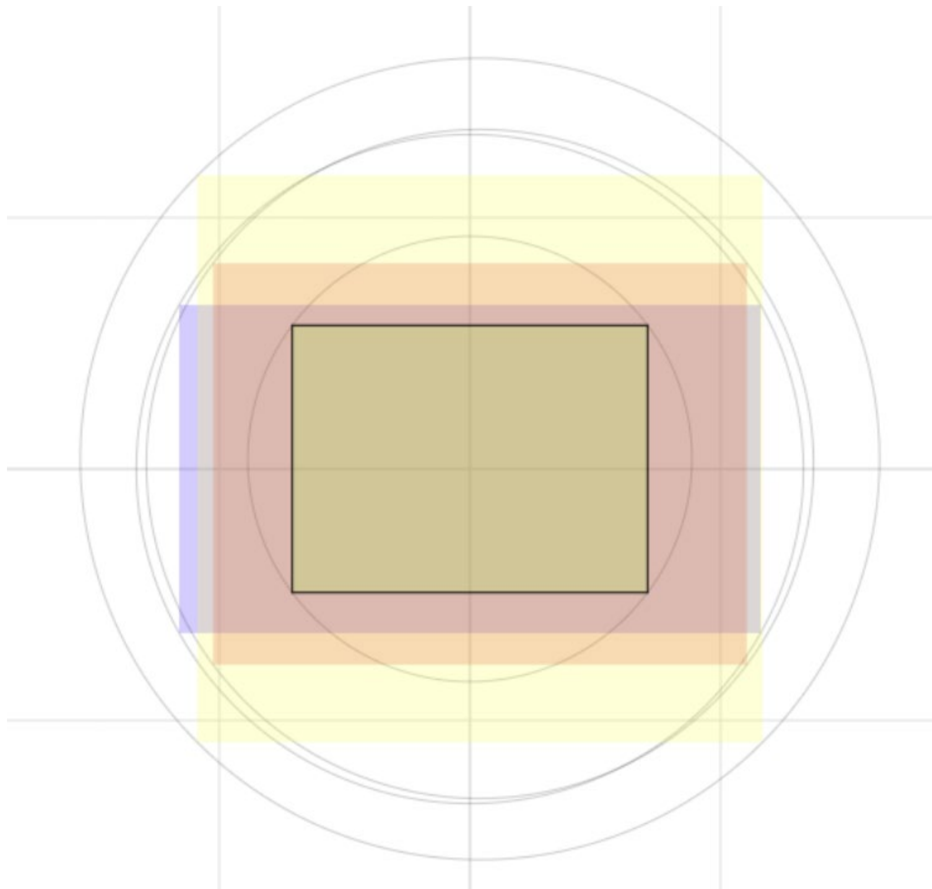


FOVisor User Guide

JavaScript code by Jim Foley

Documentation (every word) by Gemini 3.5 Flash, based on source code and prompts for audience, use-case, and style

FOVisor is a field-of-view (FOV) calculation and visualization tool designed to assist engineers in planning multi-sensor computer vision arrays, such as combined RGB and depth sensor configurations.

In a multi-sensor array, two distinct spatial challenges dictate final coverage:

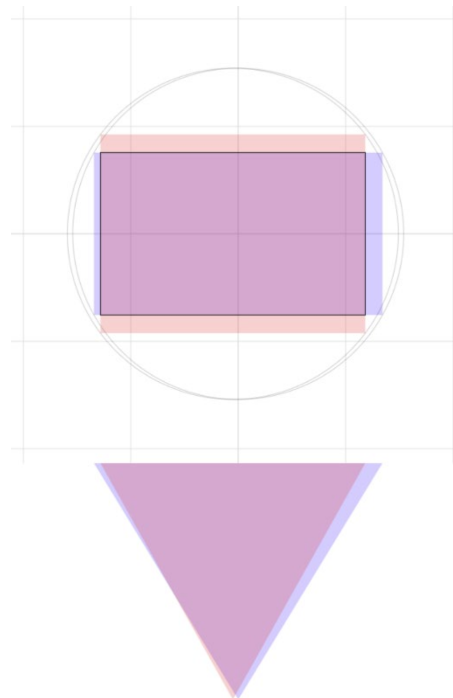
- **Physical Offsets:** Because individual sensors occupy different physical positions on a device, their projected fields of view exhibit baseline spatial displacement.
- **Varied Aspect Ratios:** Different sensor models frequently utilize conflicting frame proportions (e.g., matching a 16:9 widescreen sensor with a 4:3 depth sensor).

This application simulates how these disparate footprints project into space simultaneously, calculating the precise geometric overlap, area metrics, and final intersection boundaries over variable target distances.

Contents

1. Core Visualization Models

FOVisor uses a dual-canvas system to display sensor coverage from two distinct perspectives simultaneously.



Front View (Visual FOV)

The upper canvas displays the physical footprint of each enabled sensor as a shaded rectangle projected onto a flat vertical plane at the specified target depth.

- Multiple sensors stack transparently on top of each other.
- Overlapping regions display blended colors.
- When two or more sensors intersect, a solid black boundary outlines the exact shared coverage area.

Top-Down View (Frustum Projection)

The lower canvas displays a horizontal slice of the fields of view looking down from above.

- Each sensor projects outward as an inverted triangle (frustum) originating from its physical position on the device apex down to the target distance.
- This view isolates horizontal sensor offsets (X-axis spacing) and demonstrates how spatial divergence becomes less significant as target distance increases.

2. Interface and Control Mapping

The left-hand control panel configures global environment variables and individual sensor parameters.

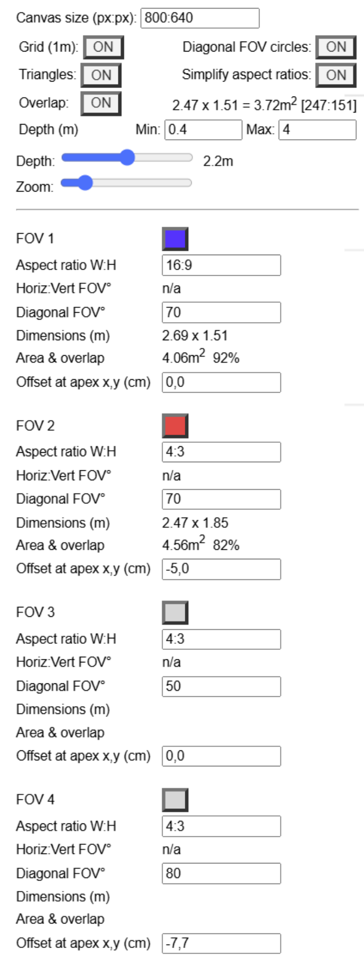
Global Environment Controls

Control Element	Input Type	Description
Canvas size	Text String (W:H)	Adjusts the resolution of both drawing surfaces. Default is 800:640.
Grid	Toggle Button (ON/OFF)	Toggles a background reference grid. Each grid square represents exactly 1x1 meter.
Diagonal FOV circles	Toggle Button (ON/OFF)	Draws a bounding circle circumscribing the corners of each sensor's rectangular footprint.
Triangles	Toggle Button (ON/OFF)	Toggles the visibility of the top-down frustum visualization canvas.
Simplify aspect ratios	Toggle Button (ON/OFF)	Employs an internal greatest common divisor (GCD) calculation to display normalized integer aspect ratios for the calculated overlap zone.
Overlap	Toggle Button (ON/OFF)	Enforces the drawing of the black intersection boundary and toggles the global intersection metrics display.
Depth (m) Min/Max	Numeric Inputs	Sets the hard boundaries for the distance slider.
Depth Slider	Range Slider	Controls the current target distance (d) in meters. Adjusting this updates all calculations in real time.
Zoom	Range Slider	Modifies the pixel density (pxPerMeter). Use this to scale the graphics up or down to maximize the use of the canvas area.

Sensor Configuration Panels (FOV 1 through 4)

Each of the four available sensor profiles contains independent properties:

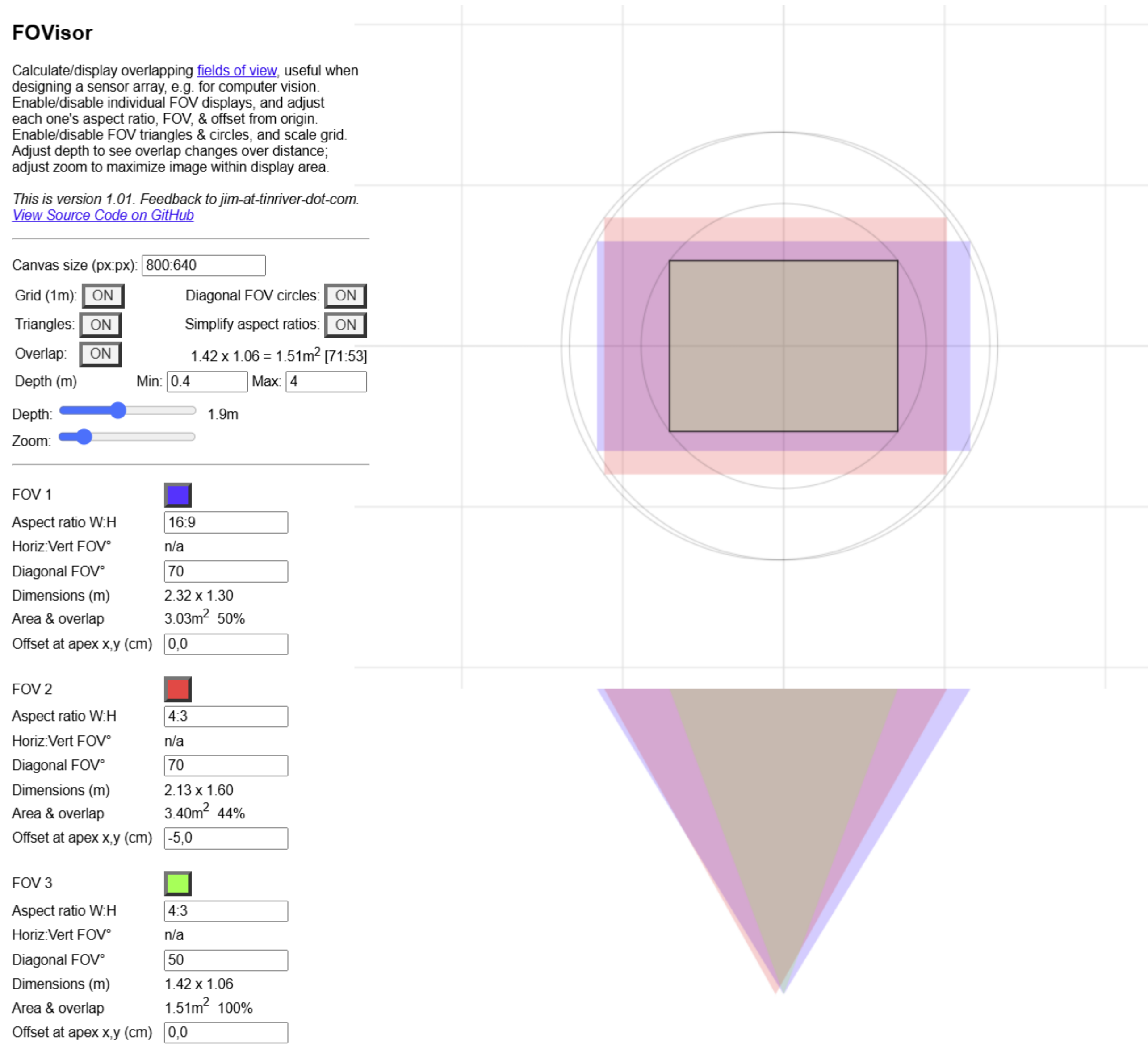
- **Activation Button:** Toggles the individual sensor on or off. When active, the button matches the sensor's assigned display color (Sensor 1: Blue, Sensor 2: Red, Sensor 3: Green, Sensor 4: Yellow).
- **Aspect ratio W:H:** Sets the sensor's frame proportions (e.g., 4:3 or 16:9).
- **Horiz:Vert FOV°:** (Reserved for future expansion).
- **Diagonal FOV°:** Accepts the sensor's actual diagonal field of view specification in degrees.
- **Offset at apex x,y (cm):** Defines the physical distance of the sensor from the array origin point $(0,0)$ in centimeters.
    - x dictates horizontal displacement.
    - y dictates vertical displacement.



3. Operational Workflows

Configuring a Custom Array

1. Identify the physical spacing of your sensors on your mounting bracket.
2. For each sensor, click the corresponding activation button to turn it **ON**.
3. Enter the target manufacturer specifications: **Aspect ratio**, **Diagonal FOV** (in degrees), and the physical displacement (**Offset**) relative to your center point.



Analyzing Intersection Metrics

Once your array is populated, use the **Depth** slider to simulate performance environments:

1. Observe the **Area & overlap** fields within the individual sensor tables.
2. Read the real-time calculated area (m^2) for each footprint alongside the percentage value indicating how much of that specific sensor's view falls within the shared overlap zone.
3. Review the global **Overlap** indicator to read the absolute dimensions and area of the shared intersection space.

4. Engineering Appendix: Mathematical Model

The application translates angular manufacturer specifications into real-world spatial dimensions using explicit trigonometric conversions.

For any given sensor, the width (W) and height (H) of the visual footprint at distance (d) are derived from the diagonal field of view (α, converted to radians) and the defined aspect width (A_W) and aspect height (A_H):

$$\text{Width} = 2d \cdot \tan\left(\frac{\alpha}{2}\right) \cdot \cos\left(\arctan\left(\frac{A_H}{A_W}\right)\right)$$

$$\text{Height} = 2d \cdot \tan\left(\frac{\alpha}{2}\right) \cdot \sin\left(\arctan\left(\frac{A_H}{A_W}\right)\right)$$

The application determines total area ($A = W \cdot H$) and applies physical offsets to establish bounding coordinate arrays (loX, hiX, loY, hiY), isolating the highest minimum boundaries and lowest maximum boundaries to define the final intersection matrix.