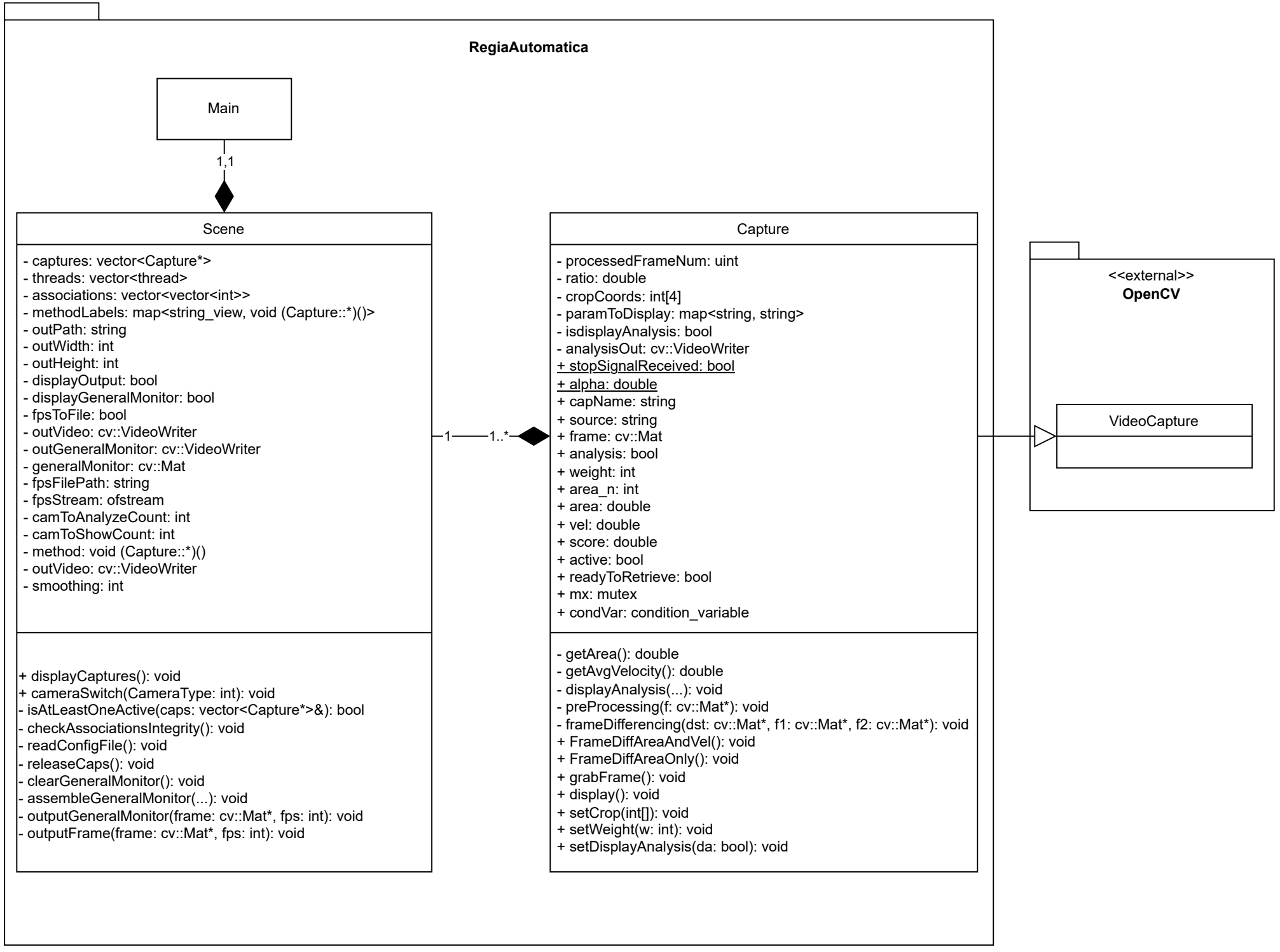
RegiaAutomatica
Main
1,1
Scene
- captures: vector<Capture*>
- threads: vector<thread>
- associations: vector<vector<int>>
- methodLabels: map<string_view, void (Capture::*)()>
- outPath: string
- outWidth: int
- outHeight: int
- displayOutput: bool
- displayGeneralMonitor: bool
- fpsToFile: bool
- outVideo: cv::VideoWriter
- outGeneralMonitor: cv::VideoWriter
- generalMonitor: cv::Mat
- fpsFilePath: string
- fpsStream: ofstream
- camToAnalyzeCount: int
- camToShowCount: int
- method: void (Capture::*)()
- outVideo: cv::VideoWriter
- smoothing: int
+ displayCaptures(): void
+ cameraSwitch(CameraType: int): void
- isAtLeastOneActive(caps: vector<Capture*>&): bool
- checkAssociationsIntegrity(): void
- readConfigFile(): void
- releaseCaps(): void
- clearGeneralMonitor(): void
- assembleGeneralMonitor(...): void
- outputGeneralMonitor(frame: cv::Mat*, fps: int): void
- outputFrame(frame: cv::Mat*, fps: int): void
1
1..*
Capture
- processedFrameNum: uint
- ratio: double
- cropCoords: int[4]
- paramToDisplay: map<string, string>
- isdisplayAnalysis: bool
- analysisOut: cv::VideoWriter
+ stopSignalReceived: bool
+ alpha: double
+ capName: string
+ source: string
+ frame: cv::Mat
+ analysis: bool
+ weight: int
+ area_n: int
+ area: double
+ vel: double
+ score: double
+ active: bool
+ readyToRetrieve: bool
+ mx: mutex
+ condVar: condition_variable
- getArea(): double
- getAvgVelocity(): double
- displayAnalysis(...): void
- preProcessing(f: cv::Mat*): void
- frameDifferencing(dst: cv::Mat*, f1: cv::Mat*, f2: cv::Mat*): void
+ FrameDiffAreaAndVel(): void
+ FrameDiffAreaOnly(): void
+ grabFrame(): void
+ display(): void
+ setCrop(int[]): void
+ setWeight(w: int): void
+ setDisplayAnalysis(da: bool): void
<<external>>
OpenCV
VideoCapture