

## RoboCupSSL\_Detection

Essa é a lib que armazena as seguintes estruturas:

- **SSL\_DetectionBall**
- **SSL\_DetectionFrame**
- **SSL\_DetectionRobot**

### SSL\_DetectionBall

Essa estrutura é responsável por armazenar os dados gerados pela detecção da bola em campo.

Ela possui os seguintes métodos disponíveis para uso:

- **new()**
  - Cria uma nova instância do SSL\_DetectionBall
- **default\_instance()**
  - Cria uma nova instância padrão do SSL\_DetectionBall (estática)
- **clear\_confidence(&mut self)**
  - Reseta a confiança
- **has\_confidence(&self)**
  - Verifica se há confiança
- **set\_confidence(&mut self, float)**
  - Seta a confiança
- **get\_confidence(&self)**
  - Obtém a confiança
- **clear\_area(&mut self)**
  - ~~~verificar depois~~~
- **has\_area(&self)**
  - ~~~verificar depois~~~
- **set\_area(&mut self, unsigned int)**
  - ~~~verificar depois~~~
- **get\_area(&self)**
  - ~~~verificar depois~~~
- **clear\_x(&mut self)**
  - Reseta a posição X
- **has\_x(&self)**
  - Verifica se há a posição X
- **set\_x(&mut self, float)**
  - Seta a posição X
- **get\_x(&self)**
  - Obtém a posição X
- **clear\_y(&mut self)**
  - Reseta a posição Y
- **has\_y(&self)**
  - Verifica se há a posição Y
- **set\_y(&mut self, float)**
  - Seta a posição Y

- **get\_y(&self)**
  - Obtém a posição Y
- **has\_z(&self)**
  - Verifica se há a posição Z
- **clear\_z(&mut self)**
  - Reseta a posição Z
- **set\_z(&mut self, float)**
  - Seta a posição Z
- **get\_z(&self)**
  - Obtém a posição Z
- **clear\_pixel\_x(&mut self)**
  - Reseta o pixel X (RAW x)
- **has\_pixel\_x(&self)**
  - Verifica se há o pixel X (RAW x)
- **set\_pixel\_x(&mut self, float)**
  - Seta o pixel X (RAW x)
- **get\_pixel\_x(&self)**
  - Obtém o pixel X (RAW x)
- **clear\_pixel\_y(&self)**
  - Reseta o pixel Y (RAW y)
- **has\_pixel\_y(&self)**
  - Verifica se há o pixel Y (RAW y)
- **set\_pixel\_y(&mut self, float)**
  - Seta o pixel Y (RAW y)
- **get\_pixel\_y(&self)**
  - Obtém o pixel Y (RAW y)

## SSL\_DetectionFrame

Estrutura responsável por armazenar dados coletados no frame atual, como o ID da camera, número do frame, tempo de captura, latência de processamento, entre outros.

Essa estrutura possui os seguintes módulos disponíveis para uso:

- **new()**
  - Cria uma nova instância do SSL\_DetectionFrame
- **default\_instance()**
  - Cria uma nova instância padrão do SSL\_DetectionFrame (estática)
- **clear\_frame\_number(&mut self)**
  - Reseta o numero do frame
- **has\_frame\_number(&self)**
  - Checa se a estrutura possui um numero para o frame
- **set\_frame\_number(&self)**
  - Seta o numero do frame
- **frame\_number(&self)**
  - Retorna o numero do frame atual

- **clear\_t\_capture(&mut self)**
  - Reseta o tempo de captura do frame
- **has\_t\_capture(&self)**
  - Verifica se existe tempo de captura para o frame
- **set\_t\_capture(&mut self, float)**
  - Seta o tempo de captura para o frame
- **t\_capture(&self)**
  - Retorna o tempo de captura do frame
- **clear\_t\_sent(&mut self)**
  - Zera a medição de latência entre a network (conexão com o socket)
- **has\_t\_sent(&self)**
  - Verifica se houve medição tempo de latência da network
- **set\_t\_sent(&self)**
  - Seta o tempo de medição de latência da network
- **t\_sent(&self)**
  - Pega o tempo de medição de latência da network
- **clear\_camera\_id(&mut self)**
  - Reseta o ID da câmera
- **has\_camera\_id(&self)**
  - Verifica se a câmera possui um ID
- **set\_camera\_id(&mut self, unsigned int)**
  - Seta o ID da câmera
- **camera\_id(&self)**
  - Obtém o ID da câmera
- **clear\_balls(&mut self)**
  - Reseta (apaga) as bolas presentes no campo
- **set\_balls(&mut self, SSL\_DetectionBall)**
  - Seta uma bola SSL\_DetectionBall no campo
- **mut\_balls(&mut self)**
  - ~~~verificar depois~~~
- **take\_balls(&mut self)**
  - ~~~verificar depois~~~
- **get\_balls(&self)**
  - Obtém as bolas no frame
- **clear\_robots\_yellow(&mut self)**
  - Reseta os robôs do time amarelo
- **set\_robots\_yellow(&mut self, SSL\_DetectionRobot)**
  - Seta os robôs do time amarelo com o SSL\_DetectionRobot
- **mut\_robots\_yellow(&mut self)**
  - ~~~verificar depois~~~
- **take\_robots\_yellow(&mut self)**
  - ~~~verificar depois~~~
- **get\_robots\_yellow(&self)**
  - Obtém os robôs do time amarelo
- **clear\_robots\_blue(&mut self)**
  - Reseta os robôs do time azul

- **set\_robots\_blue(&mut self, SSL\_DetectionRobot)**
  - Seta os robôs do time azul com o SSL\_DetectionRobot
- **mut\_robots\_blue(&mut self)**
  - ~~~verificar depois~~~
- **take\_robots\_blue(&mut self)**
  - ~~~verificar depois~~~
- **get\_robots\_blue(&self)**
  - Obtém os robôs do time azul

## SSL\_DetectionRobot

Estrutura responsável por armazenar dados coletados referentes aos robôs presentes em campo.

Essa estrutura possui os seguintes módulos disponíveis para uso:

- **new()**
  - Cria uma nova instância do SSL\_DetectionRobot
- **default\_instance()**
  - Cria uma nova instância padrão do SSL\_DetectionRobot (estática)
- **clear\_confidence(&mut self)**
  - Zera a confiança do robô
- **has\_confidence(&self)**
  - Checa se o robô possui confiança
- **set\_confidence(&mut self, float)**
  - Seta a confiança do robô
- **confidence(&self)**
  - Retorna a confiança do robô
- **clear\_robot\_id(&mut self)**
  - Clear no ID do robô
- **has\_robot\_id(&self)**
  - Retorna se o robô possui um ID especificado (true, false)
- **set\_robot\_id(&mut self, unsigned int)**
  - Seta o ID do robô
- **robot\_id(&self)**
  - Retorna o ID do robô
- **clear\_x(&mut self)**
  - Clear na coordenada X do robô
- **has\_x(&self)**
  - Verifica se o robô possui coordenada X
- **set\_x(&mut self, float)**
  - Seta a coordenada X do robô
- **x(&self)**
  - Retorna a coordenada X do robô
- **clear\_y(&mut self)**
  - Clear na coordenada Y do robô

- **has\_y(&self)**
  - Verifica se o robô possui coordenada Y
- **set\_y(&mut self, float)**
  - Seta a coordenada Y do robô
- **y(&self)**
  - Retorna a coordenada Y do robô
- **clear\_orientation(&mut self)**
  - Zera a orientação (ângulo) do robô
- **has\_orientation(&self)**
  - Verifica se o robô possui orientação (ângulo)
- **set\_orientation(&mut self, float)**
  - Seta a orientação (ângulo) do robô
- **get\_orientation(&self)**
  - Obtém a orientação (ângulo) do robô
- **clear\_pixel\_x(&mut self)**
  - Zera o pixel X do robô (RAW x)
- **has\_pixel\_x(&self)**
  - Verifica se o robô possui o pixel X (RAW x)
- **set\_pixel\_x(&mut self, float)**
  - Seta o pixel X do robô (RAW x)
- **pixel\_x(&self)**
  - Obtém o pixel X do robô (RAW x)
- **clear\_pixel\_y(&mut self)**
  - Reseta o pixel Y do robô (RAW y)
- **has\_pixel\_y(&self)**
  - Verifica se o robô possui pixel Y (RAW y)
- **set\_pixel\_y(&mut self, float)**
  - Seta o pixel Y do robô (RAW y)
- **pixel\_y(&self)**
  - Obtém o pixel Y do robô (RAW y)
- **clear\_height(&mut self)**
  - 'Limpa' a altura do robô
- **has\_height(&mut self, float)**
  - Checa se o robô possui altura
- **set\_height(&mut self, float)**
  - Seta a altura do robô
- **height(&self)**
  - Retorna a altura do robô (verificar)

## RoboCupSSL\_Geometry

Essa é a lib que armazena as seguintes estruturas:

- **SSL\_GeometryData**
- **SSL\_GeometryCameraCalibration**
- **SSL\_GeometryFieldSize**

### SSL\_GeometryData

Essa estrutura é a responsável por armazenar os dados referentes a geometria do campo (largura do campo, do gol, campos de linha, etc.)

Ela possui os seguintes métodos disponíveis para uso:

- **new()**
  - Cria uma nova instância do SSL\_GeometryData
- **default\_instance()**
  - Cria uma nova instância padrão do SSL\_GeometryData (estática)
- **clear\_field(&mut self)**
  - Zera todos os dados do campo
- **has\_field(&self)**
  - Verifica se há um campo
- **set\_field(&mut self, SSL\_GeometryFieldSize)**
  - Seta um novo campo
- **mut\_field(&mut self)**
  - ~~~ verificar depois ~~~
- **take\_field(&mut self)**
  - Retorna o campo
- **get\_field(&self)**
  - Obtém o ponteiro para o campo
- **clear\_calib(&mut self)**
  - Reseta a calibração da câmera
- **set\_calib(&mut self, RepeatedField<SSL\_GeometryCameraCalibration>)**
  - Seta a calibração da câmera
- **mut\_calib(&mut self)**
  - ~~~ verificar depois ~~~
- **take\_calib(&mut self)**
  - Retorna a calibração
- **get\_calib(&self)**
  - Obtém a calibração da câmera

### SSL\_GeometryCameraCalibration

Essa estrutura é a responsável por armazenar os dados referentes a calibração da câmera, como sua distorção, foco, etc.

Ela possui os seguintes métodos disponíveis para uso:

- **new()**
  - Cria uma nova instância do SSL\_GeometryCameraCalibration

- **default\_instance()**
  - Cria uma nova instância padrão do SSL\_GeometryCameraCalibration (estática)
- **clear\_camera\_id(&mut self)**
  - Reseta o ID da câmera
- **has\_camera\_id(&self)**
  - Retorna um booleano indicando se a câmera possui um ID ou não
- **set\_camera\_id(&mut self, int)**
  - Seta um ID para a câmera
- **get\_camera\_id(&self)**
  - Retorna o ID da câmera
- **clear\_focal\_length(&mut self)**
  - Reseta o foco da câmera
- **has\_focal\_length(&self)**
  - Retorna um booleano indicando se a câmera possui foco
- **set\_focal\_length(&mut self, float)**
  - Seta o foco da câmera
- **get\_focal\_length(&self)**
  - Obtém o foco da câmera
- **clear\_principal\_point\_x(&mut self)**
  - Reseta o ponto principal x da câmera
- **has\_principal\_point\_x(&self)**
  - Verifica se a câmera possui um ponto x principal.
- **set\_principal\_point\_x(&mut self, float)**
  - Seta o ponto x principal da câmera
- **get\_principal\_point\_x(&self)**
  - Obtém o ponto x principal da câmera
- **clear\_principal\_point\_y(&mut self)**
  - Reseta o ponto principal y da câmera
- **has\_principal\_point\_y(&self)**
  - Verifica se a câmera possui um ponto y principal
- **set\_principal\_point\_y(&mut self, float)**
  - Seta o ponto y principal da câmera
- **get\_principal\_point\_y(&self)**
  - Obtém o ponto y principal da câmera
- **clear\_distortion(&mut self)**
  - Reseta a distorção da câmera
- **has\_distortion(&self)**
  - Verifica se a câmera possui distorção
- **set\_distortion(&mut self, float)**
  - Seta a distorção da câmera
- **get\_distortion(&self)**
  - Obtém a distorção da câmera
- **clear\_q0(&mut self)**

- ~~~verificar dps~~~
- **has\_q0(&self)**
  - ~~~verificar dps~~~
- **set\_q0(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_q0(&self)**
  - ~~~verificar dps~~~
- **clear\_q1(&mut self)**
  - ~~~verificar dps~~~
- **has\_q1(&self)**
  - ~~~verificar dps~~~
- **set\_q1(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_q1(&self)**
  - ~~~verificar dps~~~
- **clear\_q2(&mut self)**
  - ~~~verificar dps~~~
- **has\_q2(&self)**
  - ~~~verificar dps~~~
- **set\_q2(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_q2(&self)**
  - ~~~verificar dps~~~
- **clear\_q3(&mut self)**
  - ~~~verificar dps~~~
- **has\_q3(&self)**
  - ~~~verificar dps~~~
- **set\_q3(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_q3(&self)**
  - ~~~verificar dps~~~
- **clear\_tx(&mut self)**
  - ~~~verificar dps~~~
- **has\_tx(&self)**
  - ~~~verificar dps~~~
- **set\_tx(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_tx(&self)**
  - ~~~verificar dps~~~
- **clear\_ty(&mut self)**
  - ~~~verificar dps~~~
- **has\_ty(&self)**
  - ~~~verificar dps~~~
- **set\_ty(&mut self, float)**
  - ~~~verificar dps~~~



- **get\_ty(&self)**
  - ~~~verificar dps~~~
- **clear\_tz(&mut self)**
  - ~~~verificar dps~~~
- **has\_tz(&self)**
  - ~~~verificar dps~~~
- **set\_tz(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_tz(&self)**
  - ~~~verificar dps~~~
- **clear\_derived\_camera\_world\_tx(&mut self)**
  - ~~~verificar dps~~~
- **has\_derived\_camera\_world\_tx(&self)**
  - ~~~verificar dps~~~
- **set\_derived\_camera\_world\_tx(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_derived\_camera\_world\_tx(&self)**
  - ~~~verificar dps~~~
- **clear\_derived\_camera\_world\_ty(&mut self)**
  - ~~~verificar dps~~~
- **has\_derived\_camera\_world\_ty(&self)**
  - ~~~verificar dps~~~
- **set\_derived\_camera\_world\_ty(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_derived\_camera\_world\_ty(&self)**
  - ~~~verificar dps~~~
- **clear\_derived\_camera\_world\_tz(&mut self)**
  - ~~~verificar dps~~~
- **has\_derived\_camera\_world\_tz(&self)**
  - ~~~verificar dps~~~
- **set\_derived\_camera\_world\_tz(&mut self, float)**
  - ~~~verificar dps~~~
- **get\_derived\_camera\_world\_tz(&self)**
  - ~~~verificar dps~~~

## SSL\_GeometryFieldSize

Essa estrutura é a responsável por armazenar os dados referentes ao tamanho do campo, como sua largura, seu comprimento, sua profundidade, linhas de campo, arcos de campo, etc.

Ela possui os seguintes métodos disponíveis para uso:

- **new()**
  - Cria uma nova instância do SSL\_GeometryFieldSize
- **default\_instance()**

- Cria uma nova instância do `SSL_GeometryFieldSize` (estática)
- **clear\_field\_length(&mut self)**
  - Reseta a largura do campo
- **set\_field\_length(&self, int)**
  - Seta a largura do campo
- **get\_field\_length(&self)**
  - Obtém a largura do campo
- **has\_field\_length(&self)**
  - Verifica se o campo possui largura
- **clear\_field\_width(&mut self)**
  - Reseta o comprimento do campo
- **set\_field\_width(&self, int)**
  - Seta o comprimento do campo
- **get\_field\_width(&self)**
  - Obtém o comprimento do campo
- **has\_field\_width(&self)**
  - Verifica se o campo possui comprimento
- **clear\_goal\_width(&mut self)**
  - Reseta o comprimento do campo de gol
- **set\_goal\_width(&self, int)**
  - Seta o comprimento do campo de gol
- **get\_goal\_width(&self)**
  - Obtém o comprimento do campo de gol
- **has\_goal\_width(&self)**
  - Verifica se o campo de gol possui comprimento
- **clear\_goal\_depth(&mut self)**
  - Reseta a profundidade do campo de gol
- **set\_goal\_depth(&self, int)**
  - Seta a profundidade do do campo de gol
- **get\_goal\_depth(&self)**
  - Obtém a profundidade do do campo de gol
- **has\_goal\_depth(&self)**
  - Verifica se o campo de gol possui profundidade
- **clear\_boundary\_width(&mut self)**
  - Reseta o comprimento da fronteira
- **set\_boundary\_width(&self, int)**
  - Seta o comprimento da fronteira
- **get\_boundary\_width(&self)**
  - Obtém o comprimento da fronteira
- **has\_boundary\_width(&self)**
  - Verifica se a fronteira possui comprimento
- **clear\_field\_lines(&mut self)**
  - Reseta os campos de linha
- **set\_field\_lines(&self, RepeatedField<SSL\_FieldLineSegment>)**
  - Seta os campos de linha

- **mut\_field\_lines(&mut self)**
  - ~~~ verificar dps ~~~
- **take\_field\_lines(&mut self)**
  - Retorna o ponteiro para as linhas de campo
- **get\_field\_lines(&self)**
  - Obtém as linhas de campo
- **clear\_field\_arcs(&mut self)**
  - Reseta os campos de arco
- **set\_field\_arcs(&self, RepeatedField<SSL\_FieldCircularArc>)**
  - Seta os campos de arco
- **mut\_field\_arcs(&mut self)**
  - ~~~ verificar dps ~~~
- **take\_field\_arcs(&mut self)**
  - Retorna o ponteiro para as linhas de arco
- **get\_field\_arcs(&self)**
  - Obtém as linhas de arco

**Links úteis:**

<https://github.com/RoboCup-SSL/ssl-vision>

<https://github.com/RoboCup-SSL/ssl-vision/blob/master/src/client/main.cpp>

[https://roboime.github.io/roboime-next/roboime\\_next\\_protocol/index.html](https://roboime.github.io/roboime-next/roboime_next_protocol/index.html)

**Obs.:** Verificar e catalogar os outros métodos disponíveis, verificar e confirmar sua utilização em código.