Stauffer Grimson Background Segmentation Method Documentation

Department of Informatics and Telecommunications

Nikiforos Pittaras, M1422

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 options Class Reference

The options class receives, modifies by user input and stores CLI arguments.

Public Member Functions

- void info ()

  print current settings to stdout
- void resize (cv::Mat &frame)

  resize the input frame
- void peek (cv::Mat &frame)

  set the imageSize options by peeking at the frame
- bool goodDim (int v)

  checks whether input dimension is valid
- bool goodIter (int v)

  checks whether input iteration is valid
- bool argparse (int argc, char ∗∗argv)

  parse command line arguments

Static Public Member Functions

- static void usage ()

  print accepted usage
- static void usage_controls ()

  usage_controls Display control interface of main.

Public Attributes

- vector< string > methodOpts

  Container to pass on options to the method that will run.
- pair< int, int > imageSize

  Specified image size, std::pair.
- bool resizeImage

  Should resize image or not.
- bool useVideo

  Should get input from a video file.
- bool doUpdate

  Should print updated parameters.
- int strelOpenDim

  Opening by reconstruction strel dimension.
- int strelCloseDim

  Closing by reconstruction strel dimension.
- int numErode

  How many times to erode in opening by reconstruction.
- int numDilate

  How many times to dilate in closing by reconstruction.
- string method

  Method to use.
- string videoFile

  Video path.

Static Public Attributes

- static const int **MAXITER** = 25

2.1.1 Detailed Description

The options class receives, modifies by user input and stores CLI arguments.

method: the background estimation method to use imageSize: the size in pixels to resize the image numErode: how many times to apply erosion numDilate: how many times to apply dilation strelSize: structuring element dimension

The documentation for this class was generated from the following file:

- main.cpp

2.2 pixel Class Reference

A class to house the SG estimation for a single pixel.

`#include <sgb.h>`

Public Member Functions

- bool init (int num, double lr, double thresh)

  Initialize the class with parameters passed on from the SGB class.
- void normalize ()

  normalize the gaussian weights to unit length
- void setPixelValue (float px)

  setPixelValue assign the pixel intensity
- bool processValue ()

  processValue processes an incoming pixel value.
- void initializeGaussian (double *mu, double *sigma, double *weight)

  initializeGaussian initializes a gaussian distribution.
- void calculateWSigma ()

  calculateWSigma Calculate the weight/stdev metric for sorting gaussians.
- void partitionGaussians ()

  partitionGaussians Specify which gaussians will form the background
- ~pixel ()

  Destructor.
- void setMetaparameters (double lr, double thr)

  setMetaparameters Parameter setter
- double gaussian (int idx)

  gaussian function evaluator
- void updateGaussian (int index)

  updateGaussian Recompute gaussian parameters
- void printGaussians (string msg)

  printGaussians Debugging print function.
- void printGaussiansWS ()

  printGaussiansWS Debugging print function.

Public Attributes

- float pixelValue

  The current pixel value.
- int maxGaussians

  Max number of gaussians.
- int numGaussians

  Current number of gaussians.
- vector< pair< double, int > > wsigma

  Vector where to store the w/sigma quantities, per gaussian.
- double * mu

  Array of gaussian means.
- double * sigma

  Array of gaussian standard deviations.
- double * weights

  Array of gaussian weights.
- double learningRate

  The learning rate.
- double thresh

  Threshold value.
- vector< int > backgroundIdx

Vector of indexes of the gaussians that form the background.

- const float initialSigma = 10

  Initial large standard deviation value for new gaussians.

- const bool verbose = false

  For debug printing.

- bool penalizeWeight

  Posterior addition to penalize the weight of new gaussians.

2.2.1 Detailed Description

A class to house the SG estimation for a single pixel.

This class contains the gaussian mixture model (means, standard deviations, number of gaussians) for a pixel.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 pixel::~pixel ( ) [inline]

Destructor.

2.2.3 Member Function Documentation

2.2.3.1 void pixel::calculateWSigma ( ) [inline]

calculateWSigma Calculate the weight/stdev metric for sorting gaussians.

This method uses std::vector and a lamda comparator to std::sort to sort the gaussians by the w/sigma metric.

2.2.3.2 double pixel::gaussian (int *idx*) [inline]

gaussian function evaluator

Parameters

idx	Which gaussian we want to compute.

Returns

The computed value.

2.2.3.3 bool pixel::init (int *num,* double *lr,* double *thresh*) [inline]

Initialize the class with parameters passed on from the SGB class.

Parameters

num	number of gaussians
lr	learning rate
thresh	threshold

Returns

successfull initialization

2.2.3.4 void pixel::initializeGaussian (double * *mu,* double * *sigma,* double * *weight*) [inline]

initializeGaussian initializes a gaussian distribution.

Parameters

mu	pointer to the mu value of the gaussian
sigma	pointer to the stdev value of the gaussian
weight	pointer to the weight value of the gaussian

Mu is set to the current pixel value, sigma to a high, const value (default 10), and the weight to a low value of 1/numGaussians^2.

2.2.3.5 void pixel::partitionGaussians ( ) [inline]

partitionGaussians Specify which gaussians will form the background

A running sum of the gaussian weights is computed. The gaussians are considered by decreasing w/sigma value. When the threshold value is reached, the partitioning ends, and icluded gaussians form the background.

2.2.3.6 void pixel::printGaussians (string *msg*) [inline]

printGaussians Debugging print function.

Parameters

msg	

2.2.3.7 bool pixel::processValue ( ) [inline]

processValue processes an incoming pixel value.

Returns

true if it belongs in the background, else false.

The intensity is checked with the gaussian match criterion and the distributions update their parameters if they match the input or not, and update their weights as well. If a match does not occur, a new distribution is added to the collection. The wsigma computation and partitioning follows, and the function returns true if the pixel matched a background distribution.

2.2.3.8 void pixel::setMetaparameters (double *lr,* double *thr*) [inline]

setMetaparameters Parameter setter

Parameters

lr	learning rate
thr	threshold

2.2.3.9 void pixel::setPixelValue (float *px*) [inline]

setPixelValue assign the pixel intensity

Parameters

px	the pixel intensity value

2.2.3.10 void pixel::updateGaussian (int *index*) [inline]

updateGaussian Recompute gaussian parameters

Parameters

index	The index of the gaussian wwe want to modify.

This method uses the current pixel field value to alter the mean and stdev values.

The documentation for this class was generated from the following file:

- sgb.h

2.3 SGb Class Reference

The Stauffer-Grimson class.

`#include <sgb.h>`

Public Member Functions

- SGb (int cols, int rows)
  SGb Constructor.
- ~SGb ()
  Destructor.
- bool initialize (vector< string > opts, cv::Mat &frame)
  initialize Initialize the class,
- cv::Mat **process** (cv::Mat inputFrame)
- void info ()
  info Display current parmeters.
- void decreaseLR ()
  decreaseLR Decrease learning rate
- void increaseLR ()
  increaseLR Increase learning rate
- void increaseThreshold ()
  increaseThreshold Increase the threshold value
- void decreaseThreshold ()
  decreaseThreshold Decrease the threshold value

Static Public Member Functions

- static void usage ()
  usage Display method parameters
- static void usage_controls ()
  usage_controls Display user interface infrmation

Public Attributes

- pixel * pixels
  Array of all pixels in the image.
- cv::Mat outputFrame
  Output frame mat container.
- int rows
  Image rows.
- int cols
  Image columns.
- int numGaussians
  Gaussians per pixel.
- double learningRate
  The learning rate per pixel.
- double threshold
  The threshold per pixel.

Static Public Attributes

- static const string name ="SG"
  Method name.

2.3.1 Detailed Description

The Stauffer-Grimson class.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 SGb::SGb (int *cols*, int *rows*) [inline]

SGb Constructor.

Parameters

cols	Image columns
rows	Image rows

2.3.2.2 SGb::~SGb ( ) [inline]

Destructor.

2.3.3 Member Function Documentation

2.3.3.1 bool SGb::initialize (vector< string > *opts*, cv::Mat & *frame*) [inline]

initialize Initialize the class,

Parameters

opts	Options string vector
frame	Initial frame

Returns

Successful initialization

The documentation for this class was generated from the following file:

- sgb.h

Index