|  |  |  |
| --- | --- | --- |
| inout\_rect | | |
| const std::vector<cv::KeyPoint>& keypoints = {  (10,10), (21.5,10), (50,10),  (10,29.3), (31.1,55.7), (50,68.2),  (10,70), (41.12,70), (50,70),  (100.1,75.58)  }  cv::Point2f topleft = (10,10)  cv::Point2f bottomright = (50,70) | std::vector<cv::KeyPoint>& in = {  (31.1,55.7)  }  std::vector<cv::KeyPoint>& out = {  (10,10), (21.5,10), (50,10),  (10,29.3), (50,68.2), (10,70),  (41.12,70), (50,70), (100.1,75.58)  } | OK |
| getAverage | | |
| cv::Mat arr = {  00000000  00000000  00000000  }  cv::Mat arr = {  11111111  11111111  11111111  }  cv::Mat arr = {  11100010 (226)  01110100 (116)  00111011 (59)  }  int descriptorLength = 8  cv::Mat arr = {  11011011 01101101 (219, 109)  10110110 11011011 (182, 219)  01101101 10110110 (109, 182)  }  int descriptorLength = 16 | 00000000  11111111  01110010  11111111 11111111 | OK  OK  ОК(исправлено)  ОК(косячно выводятся unsigned в отладчике) |
| in1d | | |
| const std::vector<int>& a =  {1, 2, 3, 4, 5}  const std::vector<int>& b =  {1, 1, 3, 2}  const std::vector<int>& b =  {5}  const std::vector<int>& b =  {10} | True true true false false  False false false false true  False false false false false | OK |
| argSortInt | | |
| const std::vector<int>& list =  {6, 1, 3, 2, 8, 0}  const std::vector<int>& list =  {1, 2, 3, 4, 5}  const std::vector<int>& list =  {7, 7, 7, 7, 7}  const std::vector<int>& list =  {7, 6, 5, 4, 3, 2} | 6 2 4 3 1 5  1 2 3 4 5 6  1 2 3 4 5 6  6 5 4 3 2 1 | OK |
| Count | | |
| unsigned char a = 255  unsigned char a = 0  unsigned char a = 1  unsigned char a = 128  unsigned char a = 170 | 8  0  1  1  4 | OK |
| calcDist | | |
| cv::Mat avg\_object = {01010101} (85)  cv::Mat avg\_background = {10101010} (170)  cv::Mat avg\_object = {00000000}  cv::Mat avg\_background = {11111111}  cv::Mat avg\_object = {01010101}  cv::Mat avg\_background = {01010101} | 8  8  0 | ОК |
| findPoints | | |
| std::vector<std::vector<int> > pairs =  {  {1, 2}  {0, 2}  {1, 3, 4}  {2, 4}  {0}  }  int idx = 1  std::vector<std::vector<int> > pairs =  {  {0, 2}  {3}  {2, 0}  {}  {}  }  int idx = 1  std::vector<std::vector<int> > pairs =  {  {1}  {2}  {0}  {}  {}  }  int idx = 0  std::vector<std::vector<int> > pairs =  {  {2}  {2, 0}  {0, 1}  {4}  {3}  }  int idx = 2;3 | 0 2 1 3 4  3  2 1 0  0 1 2; 4 3 | OK |
| std::vector<int> calcHist | | |
| std::vector<cv::Point2f> votes =  {  (0, -0.1)  (0.78, 2.01)  (-1, 0.1)  (0.05, 0.3)  (0, 0)  }  int threshold =2; 1;0 | 0 1 2 3 4 5; 1 4 5; -- | ОК(исправлено) |
| getBestPoints\_ | | |
| std::vector<cv::KeyPoint> &selected\_keypoints = (response) {0.1, 0, 1, 0.3, 0.5, 0.7, 0.1} | 1, 0.7, 0.5, 0.3, 0.1 | ОК(исправлено) |
| dropPoint | | |
| int idx = 1  std::vector<bool> \*isActive =  {true true true}  int backgroundPointsNum = 2  cv::Mat &featuresDatabase =  {  11111111 (255)  11111110 (254)  00000001 (1)  11111111 (255)  10000000 (128)  }  cv::Mat &selectedFeatures =  {  00000001  11111111  10000000  } | cv::Mat &featuresDatabase =  {  11111111  11111110  00000001  00000000  10000000  }  cv::Mat &selectedFeatures =  {  00000001  00000000  10000000  } | OK |
| fillModelParams | | |
| cv::Point2f &topleft = (10,1)  cv::Point2f &bottomright = (24, 21)  float &point\_avg\_dist = 0 std::vector<std::vector<float>\* > squareForm = {}  std::vector<std::vector<float>\* > angles = {} std::vector<cv::Point2f> \*springs = {}  cv::Point2f &center = ()  std::vector<cv::KeyPoint> &selected\_keypoints =  {  10,1 1,32  10,1 8,03  15,5 10  15,5 10,001  21 15  23,8 20,05  }  cv::Point2f &centerToTopLeft  cv::Point2f &centerToTopRight  cv::Point2f &centerToBottomRight  cv::Point2f &centerToBottomLeft | float &point\_avg\_dist = 10,48144652  std::vector<std::vector<float>\* > squareForm =   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 0 | 6,71 | 10,22264 | 10,22349 | 17,4915 | 23,20567 | | 6,71 | 0 | 5,748121 | 1,971 | 12,93796 | 18,22554 | | 10,22264 | 5,748121 | 0 | 0,001 | 7,433034 | 13,03428 | | 10,22349 | 5,748464 | 0,001 | 0 | 7,432362 | 13,03351 | | 17,4915 | 12,93796 | 7,433034 | 7,432362 | 0 | 5,774296 | | 23,20567 | 18,22554 | 13,03428 | 13,03351 | 5,774296 | 0 |   std::vector<std::vector<float>\* > angles =   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 0 | 1,570796 | 1,201518 | 1,201596 | 1,005135 | 1,079531 | | 1,570796 | 0 | 0,132312 | 1,570796 | 0,388151 | 0,656042 | | 1,201518 | 0,132312 | 0 | 1,570796 | 0,69066 | 0,972211 | | 1,201596 | 0,132445 | 1,570796 | 0 | 0,690464 | 0,972118 | | 1,005135 | 0,388151 | 0,69066 | 0,690464 | 0 | 1,272545 | | 1,079531 | 0,656042 | 0,972211 | 0,972118 | 1,272545 | 0 |   std::vector<cv::Point2f> \*springs =  -5,9 -9,4135  -5,9 -2,7035  -0,5 -0,7335  -0,5 -0,7325  5 4,2665  7,8 9,3165  cv::Point2f &center = 16 10,7335  cv::Point2f &centerToTopLeft = (-6, -9.73)  cv::Point2f &centerToTopRight = (-6, 10.27)  cv::Point2f &centerToBottomRight = (8, -9.73)  cv::Point2f &centerToBottomLeft = (8, 10.27) | Что-то странное с atan2. 0 и -0 не одно и то же?для ху одно а для ух надо с минусом?  Сделала как в исходном СМТ – для ху и ух одно и то же |
| recalcCenter\_ | | |
| float scaleEstimate =0.7  float rotationEstimate = 0.9 std::vector<bool> \*isActive =  true, false, true, true, true  std::vector<cv::Point2f> \*springs =  -5,1 -9  -1 -1  -0,5 1,2  10,2 7,8  17,01 10  cv::Point2f &center = 101,1 53,09  cv::Point2f &centerToTopLeft = (-6, -9.73)  cv::Point2f &centerToTopRight = (-6, 10.27)  cv::Point2f &centerToBottomRight = (8, -9.73)  cv::Point2f &centerToBottomLeft = (8, 10.27) | cv::Point2f &topLeft cv::Point2f &topRight  cv::Point2f &bottomRight cv::Point2f &bottomLeft  :   |  |  | | --- | --- | | 109,2289 | 48,06472 | | 98,26232 | 56,76726 | | 115,3207 | 55,74132 | | 104,3541 | 64,44386 |   std::vector<cv::Point2f> \*springs=  -10.50250244 -11,5  -1 -1  -5.902503967 -1.299999237  4.797492981 5.299999237  11.60749817 7,5  cv::Point2f &center = 106.5025024 55.59000015 | OK(без поворота и увеличения и тоже ок) scaleEstimate = 0 ок |
| addPoint | | |
| cv::KeyPoint &point = (10,10)  cv::Mat &descriptor = 10000000  std::vector<bool> \*isActive =  {false, true, true}  int backgroundPointsNum = 2  std::vector<cv::Point2f> \*springs =  {  (,) (1,1) (-1,-1) (,)(,)  }  cv::Point2f &center = (100,100)  cv::Mat &featuresDatabase =  {  11111111  11111110  00000001  00000000  10000000  }  cv::Mat &selectedFeatures =  {  00000001  00000000  10000000  } std::vector<std::vector<float>\* > squareForm =  {  (0)(100)(1)  (100)(0)(5)  (1)(5)(0)  }  std::vector<std::vector<float>\* > angles =  {  (0)(-0.1)(1)  (0.1)(0)(0.5)  (1)(0.5)(0)  }  std::vector<std::pair<cv::KeyPoint,int> > &activeKeypoints = {(101, 101)(99, 99)}  std::vector<int> \*weigthsDatabase = (0, 200, 200)  нет места | std::vector<bool> \*isActive =  {true, true, true}  std::vector<cv::Point2f> \*springs =  {  (-90,-90) (1,1) (-1,-1) (,)(,)  }  cv::Mat &featuresDatabase =  {  11111111  11111110  10000001  00000000  10000000  }  cv::Mat &selectedFeatures =  {  10000000  0000000  00000100  }  std::vector<std::vector<float>\* > squareForm =  {  (0)(128)(125)  (128)(0)(5)  (125)(5)(0)  }  std::vector<std::vector<float>\* > angles =  {  (0)(0.79)(1)  (0.79)(0)(0.5)  (1)(0.5)(0)  }  std::vector<std::pair<cv::KeyPoint,int> > &activeKeypoints = {(101, 101, (1))(99, 99,(2))(10,10)}  std::vector< int> \*weigthsDatabase = (1, 200, 200) | OK(исправлено)  ок |
| deletePoints | | |
| int &backgroundPointsNum = 1  cv::Mat &featuresDatabase =  {  11111111  10000000  }  cv::Mat &selectedFeatures =  {  10000000  }  std::vector<int> \*weigthsDatabase = {0} std::vector<std::pair<cv::KeyPoint,int> > &activeKeypoints = {}  std::vector<bool> \*isActive,= {}  bool &hasResult | cv::Mat &featuresDatabase =  {  11111111  00000000  }  cv::Mat &selectedFeatures =  {  00000000  }  std::vector<int> \*weigthsDatabase = {-1}  bool &hasResult = true; | OK |
| getActivePoints | | |
| cv::Mat &activeFeaturesDB  cv::Mat &activeSelectedFeatures  cv::Mat &backgrdDb =  {  11111111  }  cv::Mat &featuresDatabase =  {  11111111  00000001  00000000  }  std::vector<int> &featuresIdxs={0}  std::vector<int> &selectedfeaturesIdxs = {}  std::vector<bool> \*isActive =  {true, false}  int &backgroundPointsNum=1 | cv::Mat &activeFeaturesDB=  {  11111111  00000001  }  cv::Mat &activeSelectedFeatures=  {  00000001  }    std::vector<int> &featuresIdxs={0, 1}  std::vector<int> &selectedfeaturesIdxs = {0} | OK |
| fillDB | | |
| cv::Mat &selectedFeatures  {  00000001  10000000  }  std::vector<int> \*selectedClasses  cv::Mat &featuresDatabase  cv::Mat &backgroundDatabase  std::vector<int> &classesDatabase cv::Mat &background\_features  {  11111111  }  std::vector<cv::KeyPoint> &background\_keypoints = {(0,0)} | std::vector<int> \*selectedClasses = {1,2}  cv::Mat &featuresDatabase  {  11111111  00000001  10000000  }  cv::Mat &backgroundDatabase  {  11111111  }  std::vector<int> &classesDatabase = {0,1,2} | OK |
| estimate | | |
| std::vector<std::pair<cv::KeyPoint, int> >& keypointsIN =  {  (10, 10)1  (5, 5)2  (1, 1)0  }  float& scaleEstimate = 0  float& medRot = 0 std::vector<std::pair<cv::KeyPoint, int> >& keypoints = {}  cv::Point2f &center= (100,100)  int thrOutlier = 5 std::vector<std::vector<float>\* > squareForm =  {  0 1 5  1 0 7  5 7 0  }  std::vector<std::vector<float>\* > angles =  {  0 0.1 -0.1  0.1 0 1  -0.1 1 0  }  bool estimateScale = true  bool estimateRotation = true  std::vector<cv::Point2f> &votes = {}  std::vector<cv::Point2f> \*springs =  {  (1,1) (2,2) (-1,-1)  } |  | Похоже на правду(просто прогнала в отладке) |