Rob Hess의 SIFT c library의 소스코드의 함수와 주석

KITECH 양광웅 작성

_sift_features: Finds SIFT features in an image using user-specified parameter values. All detected features are stored in the array pointed to by ₩a feat.

create_init_img: Converts an image to 8-bit grayscale and Gaussian-smooths it. The image is optionally doubled in size prior to smoothing.
  convert_to_gray32: Converts an image to 32-bit grayscale
build_gauss_pyr: Builds Gaussian scale space pyramid from an image
  downsample: Downsamples an image to a quarter of its size (half in each dimension) using nearest-neighbor interpolation
build_dog_pyr: Builds a difference of Gaussians scale space pyramid by subtracting adjacent intervals of a Gaussian pyramid
scale_space_extrema: Detects features at extrema in DoG scale space. Bad features are discarded based on contrast and ratio of principal curvatures.
  pixval32f: A function to get a pixel value from a 32-bit floating-point image.
  is_extremum: Determines whether a pixel is a scale-space extremum by comparing it to it's 3x3x3 pixel neighborhood.
  interp_extremum: Interpolates a scale-space extremum's location and scale to subpixel accuracy to form an image feature.
  Rejects features with low contrast. Based on Section 4 of Lowe's paper.
    interp_step: Performs one step of extremum interpolation. Based on Eqn. (3) in Lowe's paper.
      deriv_3D: Computes the partial derivatives in x, y, and scale of a pixel in the DoG scale space pyramid.
      hessian_3D: Computes the 3D Hessian matrix for a pixel in the DoG scale space pyramid.
    interp_contr: Calculates interpolated pixel contrast. Based on Eqn. (3) in Lowe's paper.
      deriv_3D: Computes the partial derivatives in x, y, and scale of a pixel in the DoG scale space pyramid.
    new_feature: Allocates and initializes a new feature
    feat_detection_data:: returns a feature's detection data
  feat_detection_data: returns a feature's detection data
  is_too_edge_like: Determines whether a feature is too edge like to be stable by computing the ratio of principal curvatures at that feature.
  Based on Section 4.1 of Lowe's paper.
calc_feature_scales: Calculates characteristic scale for each feature in an array.
  feat_detection_data: returns a feature's detection data

adjust_for_img_dbl: Halves feature coordinates and scale in case the input image was doubled prior to scale space construction.

calc_feature_oris: Computes a canonical orientation for each image feature in an array. Based on Section 5 of Lowe's paper.

This function adds features to the array when there is more than one dominant orientation at a given feature location.

  feat_detection_data

  ori_hist: Computes a gradient orientation histogram at a specified pixel.

  smooth_ori_hist: Gaussian smooths an orientation histogram.

  dominant_ori: Finds the magnitude of the dominant orientation in a histogram

  add_good_ori_features: Adds features to an array for every orientation in a histogram greater than a specified threshold.

    interp_hist_peak: Interpolates a histogram peak from left, center, and right values

    clone_feature: Makes a deep copy of a feature

      new_feature: Allocates and initializes a new feature

      feat_detection_data

compute_descriptors: Computes feature descriptors for features in an array. Based on Section 6 of Lowe's paper.

  feat_detection_data: returns a feature's detection data

  descr_hist: Computes the 2D array of orientation histograms that form the feature descriptor. Based on Section 6.1 of Lowe's paper.

    calc_grad_mag_ori: Calculates the gradient magnitude and orientation at a given pixel.

    interp_hist_entry: Interpolates an entry into the array of orientation histograms that form the feature descriptor.

  hist_to_descr: Converts the 2D array of orientation histograms into a feature's descriptor vector.

    normalize_descr: Normalizes a feature's descriptor vector to unitl length