МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

Кафедра автоматизированных систем и технологий

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

Дисциплина Технологии компьютерного зрения

# Лабораторная №6.

# Оценка параметров моделей.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. Что такое фазовое пространство для преобразования Хафа?

Фазовое пространство для преобразования Хафа - многомерный массив, где каждое измерение соответствует определённому параметру модели. Каждая ячейка фазового пространства соответствует одной возможной модели.

1. В общем случае, чему равна размерность фазового пространства Хафа? Чему она равна при поиске прямых? При поиске окружностей?

Размерность фазового пространства Хафа в общем случае равна 2.

При поиске прямых размерность фазового пространства равна 2 (угол наклона нормали, расстояние до начала координат).

При поиске окружностей размерность фазового пространства равна 3 (центр окружности (её координаты) и радиус).

1. В чём различие между преобразованиями Хафа и Радона?

Основное различие между преобразованиями Хафа и Радона в том, что метод Радона вместо того, чтобы перебирать все точки и искать подходящие модели, он перебирает все возможные модели и определяет их суммарный вес (которые определяют точки). Метод эффективен, когда много входных точек, но мало возможных конфигураций моделей

Вывод в консоль.

##### **Задание 1**

Создайте программу, использующую преобразование Хафа (функция **cv2.HoughLinesP()**) для поиска прямых на изображении. Предварительно размойте изображение (**cv2.blur()** с окном 3х3) и найдите на нём границы фильтром Кэнни (**cv2.Canny()**, нижний порог 50, верхний порог 150, apertureSize 3).

Программа должна позволять регулировать с клавиатуры следующие параметры:

* threshold - порог регистрации прямой (среднее значение для данного задания - 60)
* minLineLength - минимальная длина прямой (среднее значение для данного задания - 100)
* maxLineGap - максимальный допустимый пропуск в прямой (среднее значение для данного задания - 100)

Используйте пары клавиш Q/A, W/S, E/D, и выводите текущее значение параметров в консоль.

Программа должна отметить найденные прямые на изображении.



##### **Задание 2**

Создайте программу, использующую преобразование Хафа (функция **cv2.HoughCircles()**) для поиска кругов на изображении. Предварительно размойте изображение (**cv2.blur()** с окном 3х3). Используйте метод поиска **cv2.HOUGH\_GRADIENT**.

Программа должна позволять регулировать с клавиатуры следующие параметры:

* param2 - порог регистрации круга (среднее значение для данного задания - 50)
* minDist - минимальное расстояние между центрами (среднее значение для данного задания - 10% от ширины изображения)
* minRadius - минимальный радиус круга (среднее значение для данного задания - 4% от ширины изображения)

param1 - параметр встроенного фильтра Кэнни - примите равным 50.  
 maxRadius - максимальный радиус круга - не ограничен (0).  
dp - отношение размера изображения к размеру аккумулятора - примите равным 1.

Используйте пары клавиш Q/A, W/S, E/D, и выводите текущее значение параметров в консоль.

Программа должна отметить найденные круги на изображении.



##### **Задание 3**

Создайте программу, использующую обобщённое преобразование Хафа без инвариантности по повороту и масштабу - метод Балларда (функция **cv2.createGeneralizedHoughBallard()**) для поиска указанного образца (**sil[1-5].png**) на изображении (**sils.png**). Предварительно размойте образец и сцену (**cv2.blur()** с окном 3х3).

Задайте следующие параметры метода:

* MinDist = 50 - минимальное расстояние между центрами
* dp = 2 - отношение размера изображения к размеру аккумулятора
* levels = 360
* VotesThreshold = 50 - порог обнаружения

Программа должна найти указанный образец, и нарисовать вокруг каждого экземпляра окружность.



##### **Задание 4**

Создайте программу, использующую обобщённое преобразование Хафа с инвариантностью по повороту и масштабу - метод Гуил (функция **cv2.createGeneralizedHoughGuil()**) для поиска указанного образца (**sil[1-5].png**) на изображении (**sils.png**). Предварительно размойте образец и сцену (**cv2.blur()** с окном 3х3).

Задайте следующие параметры метода:

* MinDist = 50 - минимальное расстояние между центрами
* dp = 2 - отношение размера изображения к размеру аккумулятора
* levels = 360
* maxBufferSize = 1000
* Xi = 45 - шаг детализации образца в градусах
* minAngle = 0 - минимальный угол поворота
* maxAngle = 360 - максимальный угол поворота
* angleEpsilon = 5 - насколько близкие ориентации будут считаться идентичными
* angleStep = 5 - с каким шагом угла поворота искать ориентацию
* minScale = 1.0 - минимальный приемлемый масштаб
* maxScale = 1.01 - максимальный приемлемый масштаб
* scaleStep = 0.1 - шаг при поиске масштаба

Рекомендуется указать следующие пороговые значения (подбираются индивидуально для задачи).

* angleThresh = 50
* scaleThresh = 50
* posThresh = 150

Программа должна найти заданный образец, и нарисовать вокруг каждого экземпляра окружность. Радиус окружности должен быть построен с учётом коэффициента масштаба. Внутри окружности следует изобразить радиус, обозначающий ориентацию образца (вверх для неповёрнутого).



##### **Дополнительное задание**

Модифицируйте программу из задания 4 следующим образом.

* Программа должна поочерёдно загружать и искать все образцы, и отмечать их на одном изображении.
* Каждый образец должен отмечаться своим цветом.
* В ходе расчётов программа должна выводить имя текущего обрабатываемого образца.

