Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий
Кафедра прикладной математики
наименование кафедры

Курсовой проект защищен с оценкой_____
Руководитель
проекта(работы) _____ М.Г. Казаков
(подпись) (и.о. Фамилия)
“_____” _____ 2018г.
(дата)

Система оценки повторяемости извлечения дескрипторов

тема проекта(работы)

Пояснительная записка
к курсовому проекту
по дисциплине Интеллектуальные технологии обработки изображений
(наименование дисциплины)

КП 09.04.04.20.000 ПЗ
обозначение документа

Студент группы 8ПИ-61 А.Г. Шевелёва
и.о., фамилия

Руководитель
проекта(работы) старший преподаватель М.Г.Казаков
должность, учетное звание и.о., фамилия

БАРНАУЛ 2018

ЗАДАНИЕ

Учебная дисциплина - «Интеллектуальные технологии обработки изображений»

Разделы работы и сроки выполнения:

1) Развёрнутая постановка задачи, изучение необходимой учебной и научно технической литературы (февраль 2017);
2) Разработка структуры данных и алгоритма решения задачи (октябрь – декабрь 2017);
3) Написание текста программы (январь – февраль 2018);
4) Тестирование и отладка программного продукта (март 2018);
5) Оформление отчёта о проделанной работе (апрель 2018);

Дата выдачи задания — 15.02.2017 г.

Руководитель работы старший преподаватель_____ М.Г.Казаков
(подпись)

					КП 09.04.04.20.000 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разработал		Шевелёва А.Г.			*Система оценки повторяемости извлечения дескрипторов*	*Лит.*	*Лист*	*Листов*
Пров.		Казаков М.Г.				*У*	*2*	*21*
						АлтГТУ гр. 8ПИ-61		
Н. Контролер		Казаков М.Г.						
Утв.		Кантор С.А.						

Аннотация

В отчете описывается проектирование и реализация программного обеспечения системы оценки повторяемости извлечения дескрипторов.

К отчёту прилагается диск с проектом и данным отчётом. Данное приложение реализовывалось на языке C++ с использованием среды разработки: Microsoft Visual Studio и сторонней библиотеки OpenCV.

Содержание

Введение

Уже несколько десятилетий ученые из разных стран занимаются разработкой алгоритмов, позволяющих научить компьютер видеть так же, как видит сам человек. Если для людей получать необходимую информацию посредством зрительного канала является чем-то простым и само собой разумеющимся, то обучить компьютер подобным вещам является и по сей не выполнимой задачей.

Множество IT корпораций работают над её решением, но это требует большого вложения человеческого труда, финансовых затрат и вычислительных мощностей. Но существуют методы, которые позволяют, хоть и при использовании в узконаправленных задачах, получить желаемый результат при меньших затратах. Они основаны не на структуре человеческого аппарата анализа и интерпретации изображений, а непосредственно на особенностях самого изображения. Одни из таких методов основаны на нахождении особых точек и их численного описания - десрипторов, на которые люди даже не обращают внимания. Основываясь только на наборе таких данных цифрового изображения можно с достаточно высокой точностью позволить компьютеру работать с визуальными образами подобно человеку. Вопрос эффективности таких алгоритмов ставится наиболее остро.

1 Постановка задачи

Необходимо разработать программное обеспечение для персонального компьютера, которое должно определять качество работы дескрипторов при применении различных преобразований на изображении.

На вход программе пользователь должен подавать набор изображений, одно из которых является эталонным, а остальные являются модификациями данногоизображения.

2 Обзор существующих методологий расчета дескрипторов

2.1 Дескриптор SIFT

Для формирования дескриптора SIFT (Scale Invariant Feature Transform) сначала вычисляются значения магнитуды и ориентации градиента в каждом пикселе, принадлежащем окрестности особой точки размером 16x16 пикселей. Магнитуды градиентов при этом учитываются с весами, пропорциональными значению функции плотности нормального распределения с математическим ожиданием в рассматриваемой особой точке и стандартным отклонением, равным половине ширины окрестности (веса Гауссова распределения используются для того, чтобы уменьшить влияние на итоговый дескриптор градиентов, вычисленных в пикселях, находящихся дальше от особой точки).

В каждом квадрате размером 4x4 пикселя вычисляется гистограмма ориентированных градиентов путем добавления взвешенного значения магнитуды градиента к одному из 8 бинов гистограммы. Чтобы уменьшить различные "граничные" эффекты, связанные с отнесением похожих градиентов к разным квадратам (что может возникнуть вследствие небольшого сдвига расположения особой точки) используется билинейная интерполяция: значение магнитуды каждого градиента добавляется не только в гистограмму, соответствующую квадрату, к которому данный пиксель относится, но и к гистограммам, соответствующим соседним квадратам. При этом значение магнитуды добавляется с весом, пропорциональным расстоянию от пикселя, в котором вычислен данный градиент, до центра соответствующего квадрата. Все вычисленные гистограммы объединяются в один вектор, размером, равным 128 = 8 (число бинов) x 4 x 4 (число квадратов).

Полученный дескриптор преобразуется, чтобы уменьшить возможные эффекты от изменения освещенности. Изменение контраста изображения (значение интенсивности каждого пикселя умножается на некоторую константу) приводит к такому же изменению в значениях магнитуд градиентов.

Поэтому очевидно, что данный эффект может быть нивелирован путем нормализации дескриптора таким образом, чтобы его длина стала равна единице. Изменения яркости изображения (к значению интенсивности каждого пикселя прибавляется некоторая константа) не влияют на значения магнитуд градиентов. Таким образом, SIFT-дескриптор является инвариантным по отношению к аффинным изменениям освещенности. Однако могут возникать и нелинейные изменения в освещенности вследствие, например, различной ориентации источника света по отношению к поверхностям трехмерного объекта. Данные эффекты могут вызвать большое изменение в отношении магнитуд некоторых градиентов (при этом оказывают незначительное влияние на ориентацию вектора градиента). Чтобы избежать этого, используют отсечение по некоторому порогу (по результатам экспериментов показано, что оптимальным является значение 0.2), которое применяют к компонентам нормализованного дескриптора. После применения порога дескриптор вновь нормализуется. Таким образом, уменьшается значение больших магнитуд градиентов и увеличивается значение распределения ориентаций данных градиентов в окрестности особой точки.

2.2 Дескриптор PCA-SIFT

Дескриптор PCA-SIFT по существу является модификацией SIFT. На начальном этапе аналогично вычисляются значения магнитуды и ориентации градиента. Только для каждой особой точки рассматривается окрестность размером 41 x 41 пиксель с центром в точке, которая является особой. По факту строится карта градиентов вдоль вертикального и горизонтального направлений. Как следствие, получается вектор, содержащий 2 x 39 x 39 = 3042 элементов. Далее выполняется построение SIFT дескриптора согласно схеме, описанной в предыдущем разделе. Для результирующего набора SIFT дескрипторов осуществляется снижение размерности векторов до 32 элементов посредством анализа главных компонент (Principal Component Analysis, PCA).

2.3 Дескриптор SURF

Система **CVS Авто** разработана Дескриптор SURF (Speeded up Robust Features) относится к числу тех дескрипторов, которые одновременно выполняют поиск особых точек и строят их описание, инвариантное к изменению масштаба и вращения. Кроме того, сам поиск ключевых точек обладает инвариантностью в том смысле, что повернутый объект сцены имеет тот же набор особых точек, что и образец.

Определение особых точек на изображении выполняется на основании матрицы Гессе (FAST-Hessian detector). Использование Гессиана обеспечивает инвариантность относительно преобразования типа "поворот", но не инвариантность относительно изменения масштаба. Поэтому SURF применяет фильтры разного масштаба для вычисления Гессиана. Предположим, что исходное изображение задается матрицей интенсивностей I, текущий рассматриваемый пиксель обозначим через X=(x,y), а sigma – масштаб фильтра. Тогда матрица Гессе имеет вид:

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix}$$

Рисунок 1 – Матрица Гессе

где $L_{xx}(X,\sigma), L_{xy}(X,\sigma), L_{yy}(X,\sigma)$ – свертки аппроксимации второй производной Гауссова ядра с изображением I. Детерминант матрицы Гессе достигает экстремума в точках максимального изменения градиента яркости. Поэтому SURF пробегается фильтром с Гауссовым ядром по всему изображению и находит точки, в которых достигается максимальное значение детерминанта матрицы Гессе. Отметим, что такой проход выделяет как темные пятна на белом фоне, так и светлые пятна на темном фоне.

Далее для каждой найденной особой точки вычисляется ориентация – преобладающее направление перепада яркости. Понятие ориентации близко к понятию направления градиента, но для определения ориентации особой точки применяется фильтр Хаара.

На основании имеющейся информации выполняется построение дескрипторов для каждой особой точки:

- Вокруг точки строится квадратная окрестность размером 20s, где s – масштаб, на котором получено максимальное значение детерминанта матрицы Гесса.
- Полученная квадратная область разбивается на блоки, в результате область будет разбита на 4 x 4 региона.
- Для каждого блока вычисляются более простые признаки. Как следствие, получается вектор, содержащий 4 компоненты: 2 – это суммарный градиент по квадранту, 2 – сумма модулей точечных градиентов.
- Дескриптор формируется в результате склеивания взвешенных описаний градиента для 16 квадрантов вокруг особой точки. Элементы дескриптора взвешиваются на коэффициенты Гауссова ядра. Веса необходимы для большей устойчивости к шумам в удаленных точках.
- Дополнительно к дескриптору заносится след матрицы Гессе. Эти компоненты необходимы, чтобы различать темные и светлые пятна. Для светлых точек на темном фоне след отрицателен, для темных точек на светлом фоне – положителен.

SURF используется для поиска объектов. Тем не менее, дескриптор никак не использует информацию об объектах. SURF рассматривает изображение как единое целое и выделяет особенности всего изображения, поэтому он плохо работает с объектами простой формы.

2.4 Дескриптор BRIEF

Цель создания BRIEF-дескриптора (Binary Robust Independent Elementary Features) состояла в том, чтобы обеспечить распознавание одинаковых участков изображения, которые были сняты с разных точек зрения. При этом ставилась задача максимально уменьшить количество выполняемых вычислений. Алгоритм распознавания сводится к построению случайного леса (randomize classification trees) или наивного Байесовского

классификатора на некотором тренировочном множестве изображений и последующей классификации участков тестовых изображений. В упрощенном варианте может использоваться метод ближайшего соседа для поиска наиболее похожего патча в тренировочной выборке. Небольшое количество операций обеспечивается за счет представления вектора признаков в виде бинарной строки, а как следствие, использования в качестве меры сходства расстоянии Хэмминга.

Схема построения векторов признаков достаточно простая. Изображение разбивается на патчи (отдельные перекрывающиеся участки). Допустим патч P имеет размеры S x S пикселей. Из патча выбирается некоторым образом множество пар пикселей $\{(X,Y), \forall X,Y \text{ в окрестности}\}$ для которых строится набор бинарных тестов:

$$\tau(P,X,Y) = \begin{cases} 1, I(X < I(Y)) \\ 0, \text{иначе} \end{cases}$$

Рисунок 2 – Бинарные есты

где I(X) – интенсивность пикселя X. Для каждого патча выбирается множество, содержащее n_d пар точек, которые однозначно определяют набор бинарных тестов. Далее на основании этих тестов строится бинарная строка:

$$f_{n_d}(P) = \sum_{1 \le i \le n_d} 2_{i-1} \tau(P, X_i, Y_i$$

Рисунок 3 – Бинарная строка

2.5 Дескрипторы ORB

ORB представлен в 2011г. В его основе лежит комбинация таких алгоритмов как детектор FAST (Features from Accelerated Segment Test) и дескриптор BRIEF (Binary Robust Independent Elementary Features) с некоторыми улучшениями.

Детектор FAST.

Для поиска угловых точек поочерёдно рассматриваются окрестности по

16 пикселей вокруг каждого пикселя p.

Точка p считается подозрительной на особую, если существует N пикселей (в данной работе N=9) в её окружности длиной 16 пикселей, если все N ярче $IP + t$ или темнее $IP - t$, где IP – яркость точки p, t –пороговая величина. При выполнении этого условия далее исследуется значения яркости на окружности под номерами 1, 5, 9, 13 (рис. 4). Если для трех пикселей из четырех выполняется условие $Ii < IP - t$ или $Ii > IP + t$, $i = 1 \ldots 4$, тогда p считается особой точкой.

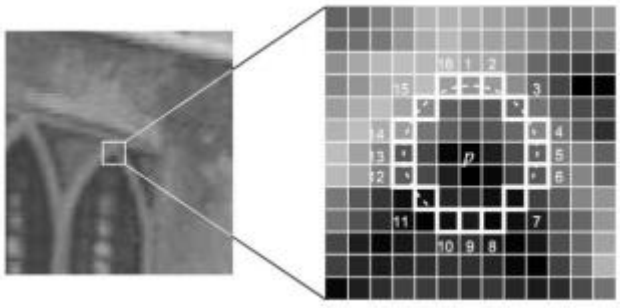

Рисунок 4 – Рассматриваемая окрестность точки p

Выбор только 4 пикселей на окружности позволяет быстро отсеять не подходящие точки, но в некоторых случаях возможно определение разных особенностей в одной окружности. В алгоритме ORB максимальное количество особых точек по умолчанию не более 500, если их больше, то к ним применяется детектор углов Харриса, для исключения наименее значимых.

Для инвариантности к масштабированию применяется алгоритм на пирамиде Гаусса. Октавами ci которой является изначальное изображение $c0$ сжатое с линейным шагом.

Введение параметра угловой ориентации позволяет добиться устойчивости детектирования при вращении объекта. Он основан на направлениях градиента яркости относительно центра точки, направление с наибольшей интенсивностью назначается ориентацией особой точки θ.

Дескриптор направленный BRIEF.

Данный дескриптор представляется в виде вектора длиной 256, состоящего из результатов бинарных тестов вокруг особой точки. В

окрестности 31 × 31 пиксель сравниваются средние значения яркостей между x и y, где x, y – области 5 × 5 пикселей:

$$\tau(I;x,y) \coloneqq \begin{cases} 1 : I_x < I_y \\ 0 : I_x \geq I_y \end{cases};\ I\ - \textit{средняя яркость выбранной области.}$$

Рисунок 5 – Средняя яркость

Для достижения инвариантности к вращению область вычисления дескриптора ориентируется по ориентации особой точки θ.

Все n = 256 наборов xi и yi формируют матрицу S размерностью 2 × n. Далее S с помощью матрицы поворота R_θ ориентируется в соответствии с углом θ:

$S_\theta = R_\theta\, S$. А сам вектор дескриптора записывается как:

$$g_n(I,\theta) \coloneqq f_n(I) | (x_i, y_i) \in S_\theta,$$
$$\textit{где}\ \ f_n(I) \coloneqq \sum_{1 \leq i \leq n} 2^{i-1} \tau(I; x_i, y_i).$$

Рисунок 6 – Вектор дескриптора

3 Решение задачи

3.1 Применяемые в решении задачи модели

Для решения задачи была задействована библиотека OpenCV.

OpenCV - библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab и других языков.

Из данной библиотеки были взяты методы для загрузки изображений, нахождения и описания точек интереса и дескрипторов методом ORB, а также метод сопоставления дескрипторов.

3.2 Алгоритм решения поставленной задачи

Алгоритм разбит на несколько этапов:

- загружаем изображения средством Opencv;
  - производится загрузка оригинального изображения;
  - загружается по 2 изображения на каждую модификацию;
- производим поиск точек интереса для изображений;
- производим расчет дескрипторов по найденным точкам для каждого из изображений;
- находим пары похожих дескрипторов между оригинальным изображением изображениями с модификациями;
- Для каждой пары изображений каждой из модификаций производим сравнение соответствий дескрипторов с оригинальным изображением:
  - количество совпавших дескрипторов;
  - количество несвязанных дескрипторов;
  - средняя разница расстояний среди соотнесенных дескрипторов для обоих изображений.

На вход программе было подано 7 изображений:

- оригинальное;
- изображение осветленное на 26% и на 100%;
- изображение повернутое на -15° и 15°;
- изображение уменьшенное в 2 и 3 раза.

```
Осветление изображения
Количество дескрипторов: 200
Количество дескрипторов несопоставленных в обоих модификациях: 29
Количество дескрипторов расстояние для которых совпало в обоих модификациях: 1
Количество несопоставленных дескрипторов для 'Осветление на 26' :44
Количество сопоставленных дескрипторов для 'Осветление на 26' :105
Количество несопоставленных дескрипторов для 'Осветление на 100' :60
Количество сопоставленных дескрипторов для 'Осветление на 100' :21
Стистика
Средняя разница дистанций между сопоставленными дескрипторами обоих модификаций: 16.168

Поворот изображения
Количество дескрипторов: 200
Количество дескрипторов несопоставленных в обоих модификациях: 33
Количество дескрипторов расстояние для которых совпало в обоих модификациях: 5
Количество несопоставленных дескрипторов для 'Поворот 15 градусов' :68
Количество сопоставленных дескрипторов для 'Поворот 15 градусов' :55
Количество несопоставленных дескрипторов для 'Поворот -15 градусов' :64
Количество сопоставленных дескрипторов для 'Поворот -15 градусов' :51
Стистика
Средняя разница дистанций между сопоставленными дескрипторами обоих модификаций: 10.2772

Уменьшение размера изображения
Количество дескрипторов: 200
Количество дескрипторов несопоставленных в обоих модификациях: 69
Количество дескрипторов расстояние для которых совпало в обоих модификациях: 1
Количество несопоставленных дескрипторов для 'Уменьшение в 2 раза' :87
Количество сопоставленных дескрипторов для 'Уменьшение в 2 раза' :51
Количество несопоставленных дескрипторов для 'Уменьшение в 3 раза' :116
Количество сопоставленных дескрипторов для 'Уменьшение в 3 раза' :16
Стистика
Средняя разница дистанций между сопоставленными дескрипторами обоих модификаций: 21.3636
```

Рисунок 7 - Результат работы алгоритма

3.3 Классовая структура

ImageOPtions — класс для хранения данных о каждом загруженном изображении, его модификации, точек и дескрипторов.

DescriptorOperations — класс позволяющий производить вычисления совпадений дескрипторов и получения значений их расстояний

DescriptorCompare — структура, содержащая в себе информацию об общем наименовании модификации, о изменении относительно оригинального изображения для двух изображений подвергнутых данной

модификации, а также параметры подсчитанные при вычислении соответствий дескрипторов.

Заключение

В результате работы была получена программа, удовлетворяющая поставленным требованиям.

В дальнейшем планируется улучшение программы: будет добавлен пользовательский интерфейс, будет добавлено полноценное меню, появится возможность расчета модификаций для различного количества изображений.

Список использованных источников

1) Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. П75 Приемы объектно-ориентированного проектирования. – СПб: Питер, 2001. – 368 с.: ил (Серия «Библиотека программиста»).

2) Szeliski R. Computer Vision: Algorithms and Applications. - Washington: Microsoft Research, 2010. - 812 c.

3) Л. Шапиро, Дж. Стокман. Computer Vision. — М.: Бином. Лаборатория знаний, 2006. — 752 с.

4) Дэвид Форсайт, Жан Понс. Computer Vision: A Modern Approach. — М.: «Вильямс», 2004. — 928 с.

5) А.А. Лукьяница ,А.Г. Шишкин. Цифровая обработка видеоизображений. — М.: «Ай-Эс-Эс Пресс», 2009. — 518 с.

6) Желтов С.Ю. и др. Обработка и анализ изображений в задачах машинного зрения. — М.: Физматкнига, 2010. — 672 с.

ПРИЛОЖЕНИЕ А Руководство пользователя

Для запуска приложения запустите файл ConsoleApplication5.exe и в папку C:\Users\ks\Desktop положить оригинальное изображение и по 2 изображения на каждую из модификаций.

Откроется консоль программы, представленная на рисунке 8.

```
Осветление изображения
Количество дескрипторов: 200
Количество дескрипторов несопоставленных в обоих модификациях: 29
Количество дескрипторов расстояние для которых совпало в обоих модификациях: 1
Количество несопоставленных дескрипторов для 'Осветление на 26' :44
Количество сопоставленных дескрипторов для 'Осветление на 26' :105
Количество несопоставленных дескрипторов для 'Осветление на 100' :60
Количество сопоставленных дескрипторов для 'Осветление на 100' :21
Стистика
Средняя разница дистанций между сопоставленными дескрипторами обоих модификаций: 16.168

Поворот изображения
Количество дескрипторов: 200
Количество дескрипторов несопоставленных в обоих модификациях: 33
Количество дескрипторов расстояние для которых совпало в обоих модификациях: 5
Количество несопоставленных дескрипторов для 'Поворот 15 градусов' :68
Количество сопоставленных дескрипторов для 'Поворот 15 градусов' :55
Количество несопоставленных дескрипторов для 'Поворот -15 градусов' :64
Количество сопоставленных дескрипторов для 'Поворот -15 градусов' :51
Стистика
Средняя разница дистанций между сопоставленными дескрипторами обоих модификаций: 10.2772

Уменьшение размера изображения
Количество дескрипторов: 200
Количество дескрипторов несопоставленных в обоих модификациях: 69
Количество дескрипторов расстояние для которых совпало в обоих модификациях: 1
Количество несопоставленных дескрипторов для 'Уменьшение в 2 раза' :87
Количество сопоставленных дескрипторов для 'Уменьшение в 2 раза' :51
Количество несопоставленных дескрипторов для 'Уменьшение в 3 раза' :116
Количество сопоставленных дескрипторов для 'Уменьшение в 3 раза' :16
Стистика
Средняя разница дистанций между сопоставленными дескрипторами обоих модификаций: 21.3636

Для продолжения нажмите любую клавишу . . .
```

Рисунок 8 - Консоль программы

ПРИЛОЖЕНИЕ Б Руководство системного программиста

Для запуска программы необходимо как минимум 1024 Мб. RAM; Windows XP/Vista/7/8/10; 100 Мб. на жёстком диске.

ПРИЛОЖЕНИЕ В Исходный код программы

Программа, исходный код, а также отчёт представлены на диске, прилагаемом к отчету.