Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук

Н.С.Байгарова, Ю.А.Бухштаб, Н.Н.Евтеева, Д.А.Корягин

Некоторые подходы к организации содержательного поиска изображений и видеоинформации

Москва
2002

В работе рассматриваются способы решения проблемы содержательного описания изображения на различных уровнях абстракции. Технология доступа к коллекциям изображений и видеофильмов по визуальному содержанию реализуется на базе сопоставления им набора визуальных примитивов и определением количественной оценки близости изображений по значениям примитивов.

The paper discusses various questions connected with content-based visual information description on differen levels of abstraction. The access technology for image and video collections is developed on the base of corresponding visual primitives. Once the distance between the template image and each target image is computed, target images can be ranked in order of their similarity with the template image.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант N 01-01-00267.

1. Введение

Предлагаемые подходы ориентированы на решение проблемы обеспечения доступа к современным электронным коллекциям изображений и видеоматериалов с использованием различных средств - как текстовых описаний, так и характеристик визуального содержания, простейших типа цветовой гаммы, и более сложных, связанных с распознаванием образов, наиболее интересных для предметной области.

До недавнегонего времени традиционным считался поиск визуальной информации, опирающийся на индексирование текстовых описаний, ассоциированных с изображением или фильмом. Однако поиск по названию, авторам, теме, словам описания содержания и по другой текстовой информации, ассоциированной с изображениями коллекции, представляется недостаточным. Неоднозначность соответствия между визуальным содержанием и текстовым описанием снижает показатели точности и полноты поиска.

2. Визуальные примитивы и механизм поиска по образцу

Для организации электронных библиотек, связанных с визуальными данными, требуются методы создания и использования поисковых образов, отражающих визуальное содержание изображений. Методы распознавания образов и понимания сцены в настоящее время из-за отсутствия эффективных универсальных алгоритмов применяются в узких предметных областях. Современная универсальная технология доступа к коллекциям изображений связана с сопоставлением изображению набора визуальных примитивов (характеристик цвета, формы, текстуры, а для видео еще и параметров движения сцены и объектов) и определением количественной оценки близости изображений по значениям примитивов [7, 14, 15, 16].

Визуальные примитивы - это характеристики изображения, которые автоматически вычисляются по оцифрованным визуальным данным, позволяют эффективно индексировать их и обрабатывать запросы с использованием визуальных свойств изображения. Поисковый образ изображения, сгенерированный из визуальных примитивов, невелик по размеру в сравнении с самим изображением и удобен для организации поиска. Вычисление подобия изображений заменяет принятую в традиционных СУБД операцию установления соответствия запросу. Хотя запросом в такой системе может быть описание набора примитивов, более удобен запросный механизм поиска по образцу, когда система отыскивает

изображения, визуально похожие на предоставленный образец. Система анализирует образец аналогично тому, как это делается при составлении поисковых образов изображений базы. Вычисление подобия изображения-образца изображениям коллекции осуществляется на основании сравнения значений отдельных визуальных примитивов, при этом система определяет меру их отличия, а затем сортирует изображения базы в соответствии с близостью к образцу по всем параметрам, с учетом указываемой в запросе степени важности каждого параметра. Поиск на таком уровне абстракции не предполагает идентификацию объектов. Скажем, если в качестве образца взято изображение собаки, то система будет искать изображения, похожие на образец по цветовой гамме, композиции, наличию определенных форм и т.п., но нет никакой гарантии, что среди них окажется изображение именно этого животного. Тем не менее, метод поиска по образцу на основании визуальных примитивов представляется на сегодняшний день достаточно эффективным и универсальным средством доступа к коллекциям оцифрованных изображений.

3. Методы анализа изображений

Различными группами исследователей уже накоплен определенный опыт реализации алгоритмов, позволяющих автоматически описывать изображения в терминах простых вычислимых визуальных свойств, а также определять меру их отличия. Авторами был подготовлен обзор этих алгоритмов [8].

Наши текущие исследования в этой области направлены на дальнейшее развитие методов вычисления и сравнения визуальных примитивов. Реализован метод количественной оценки близости статичных изображений по их цветовым гистограммам. Решена задача пространственного сегментирования изображения. Разработан и реализован алгоритм, осуществляющий вычисление параметров форм для выделенных объектов картинки и сравнение форм по их параметрам. Проводятся работы и имеются результаты, которые позволят выполнять локальное индексирование, отражающее распределение на изображении цветовых множеств. С целью вычисления измерений текстур исследуются возможности использования метода функций Габора и характеристик матрицы взаиморасположения оттенков серого цвета. (Методы обработки текстур изложены в [10, 20, 22, 26].)

3.1. Цветовые гистограммы

Метод цветовых гистограмм – наиболее популярный из методов, использующих цветовые характеристики для индексирования изображений. Возможно также использование таких показателей, как средний или основной цвета, а также множества цветов; эти характеристики имеет смысл использовать для локального индексирования областей изображения [11, 19, 20, 21].

Идея метода цветовых гистограмм для индексирования и сравнения изображений сводится к следующему. Все множество цветов разбивается на набор непересекающихся, полностью покрывающих его подмножеств Vi, 0<=i<N. Будем называть такое разбиение множества цветов базовой палитрой. Для изображения формируется гистограмма, отражающая долю каждого подмножества цветов в общей цветовой гамме изображения - массив H[i]=N[i]/ Σ N[i], где N[i] - число точек с цветом из множества Vi. Для сравнения гистограмм вводится понятие расстояния между ними. Известны различные способы построения и сравнения цветовых гистограмм [1, 2, 8, 19, 20], отличающиеся между собой изначальной цветовой схемой (RGB, CMY , HSV, grayscale и т. д.), размерностью гистограммы и определением расстояния между гистограммами.

В данной работе реализовано несколько модификаций метода, использующих разные способы квантования множества цветов и вычисления расстояния между гистограммами. Используются две базовые палитры и, следовательно, два метода построения гистограммы.

1) Разбиение RGB-цветов по яркости.

В базовой палитре Vi (0<=i<N) определяется как множество цветов C : C ∈ Vi ⇔ i/N*Imax<=I(C)< (i+1)/N*Imax, где I(C) – интенсивность цвета C, нормализованная так, что 0<=I(C)<Imax. Интенсивность вычисляется по классической формуле:

I(C)=0.3*R(C)+0.59*G(C)+0.11*B(C),

где R, G и B – красная, зеленая и синяя компоненты цвета C. Imax=256; 0<=I(C)<256. В частности, для черно-белых полутоновых изображений на N подмножеств разбивается исходное множество оттенков. Значение N выбиралось практически произвольно, сейчас установлено N=16.

Для сравнения гистограмм вводится понятие расстояния между ними - сумма модулей разности соответствующих элементов гистограмм. Некоторое усовершенствование метода достигается при вычислении расстояния на основании поэлементного сравнения гистограмм с учетом соседних элементов. Для каждого элемента гистограммы первого изображения вычисляется не одна, а три разности:

R1(i)=|H1[i]-H2[i-1]|

R2(i)=|H1 [i]-H2[i] |

R3(i)=|H1[i]-H2[i+1]|(для i=0 и i=N вместо невычислимых разностей подставляются заведомо большие значения), итоговое же расстояние равно:

$$S=\sum_{i=0}^{N-1} \min_{1<=k<=3} (Rk(i)),$$

Этот способ не годится для произвольной базовой палитры, т. к. предполагает строгую упорядоченность множества цветов, как в случае с разбиением по яркости. Заметим, что так определенное S не является расстоянием в математическом смысле из-за несимметричности (нельзя гарантировать, что S(H1,H2)= S(H2,H1)). Основное преимущество алгоритма состоит в том, что он слабо чувствителен к изменению освещенности, что ощутимо улучшает результаты его применения на широком классе изображений.

Этот метод построения гистограмм наиболее эффективен для черно-белых полутоновых изображений. Для цветных RGB-изображений лучшие результаты дает другой способ.

2) Разбиение RGB-цветов по прямоугольным параллелепипедам.

Цветовое RGB-пространство рассматривается как трехмерный куб, каждая ось которого соответствует одному из трех основных цветов (красному, зеленому или синему), деления на осях пронумерованы от 0 до 255 (большее значение соответствует большей интенсивности цвета). При таком рассмотрении любой цвет RGB-изображения может быть представлен точкой куба. Для построения цветовой гистограммы каждая сторона делится на n (n=4) равных интервалов, соответственно RGB-куб делится на N (N=64) прямоугольных параллелепипедов. Vi – множество цветов, все компоненты которых попадают в определенные интервалы. Гистограмма изображения отражает распределение точек RGB-пространства, соответствующих цветам пикселов изображения, по параллелепипедам.

Выбор размерности гистограммы определялся из следующих соображений. При n=2 (N=8) считались бы одинаковыми, например, {126,128,126} и {0, 255, 0}, что, естественно, недопустимо. Установка n=8 (N=512) приводит к тому, что базовая палитра становится более строгой, чем 8-битная. Такая точность не только автоматически дает некорректную обработку 256-цветных изображений, но и на остальных изображениях приводит к неестественным результатам. Очевидно, что при росте n ситуация только ухудшается. Поэтому было установлено n=4.

В качестве расстояния между гистограммами используется покомпонентная сумма модулей разности между ними. Несмотря на предельную простоту подхода, он показывает довольно стабильные

результаты. Распознаются схожие по цветовой гамме серии картинок, если они имеются в базе.

Более точное сравнение изображений достигается с помощью техники квадродеревьев, когда методы вычисления и сравнения цветовых гистограмм применяются не ко всему изображению, а к его четверти (одной шестнадцатой и т. д.). Сейчас программа позволяет работать не только с полными изображениями, но и с их разбиением на четверти. Для реализации этой возможности, при построении гистограммы автоматически считаются и гистограммы всех четырех квадрантов изображения. Сравнение изображений основывается на расстоянии, определенном как Евклидово в пространстве расстояний между гистограммами их частей - вместо вычисления расстояния между полными гистограммами, рассчитываются расстояния между четвертями, итоговым результатом считается корень из суммы их квадратов. Этот метод дает результат, семантически отличный от других вариантов: изображения, отличные только по взаимному расположению похожих по цвету объектов, считаются различными, а не практически идентичными, как было бы без использования этой техники. Целесообразность ее применения определяется значением для пользователя расположения на картинке-образце определенных цветовых областей.

3.2. Объекты изображения

Пространственное сегментирование изображения может осуществляться автоматически, когда выделяются области с некими общими свойствами - одинаковыми или сильно схожими значениями того или иного примитива. Полученные в результате области характеризуются расположением на изображении и размерами. Кроме того, они связываются со значениями примитивов – характеристиками формы, цвета, текстуры.

Контур - граница объекта - представляет собой замкнутую последовательность точек (xs, ys), где 1<=s<=N. Удобно считать, что (xs+N, ys+N) = (xs, ys). Задача выявления контуров связана с локализацией на изображении резких перепадов яркости цвета или изменений параметров, характеризующих текстуру.

Определение границ объектов изображения выполняется нами по следующей схеме: цветное изображение переводится в черно-белое полутоновое и сглаживается, осуществляется пространственное дифференцирование - вычисляется градиент функции интенсивности в каждой точке изображения и, наконец, подавляются значения меньше установленного порога. За основу взят метод Собеля [23], использующий

для вычисления градиента первого порядка функции интенсивности специальные ядра, известные как «операторы Собеля»:

-1	0	1
-2	0	2
-1	0	1

X-оператор Собеля

-1	-2	-1
0	0	0
1	2	1

Y-оператор Собеля

Ядра применяются к каждому пикселу изображения: он помещается в центр ядра, и значения интенсивности в соседних точках умножаются на соответствующие коэффициенты ядра, после чего полученные значения суммируются. X- оператор Собеля, примененный к 3x3 матрице исходного изображения, дает величину горизонтальной составляющей градиента интенсивности в центральной точке этой матрицы, а Y-оператор Собеля дает величину вертикальной составляющей градиента. Коэффициенты ядра выбраны так, чтобы при его применении одновременно выполнялось сглаживание в одном направлении и вычисление пространственной производной – в другом.

Величина градиента определяется как квадратный корень из суммы квадратов значений горизонтальной и вертикальной составляющих градиента.

В результате образуется массив чисел, характеризующих изменения яркости в различных точках изображения. Затем выполняется операция сравнения с порогом и определяется положение элементов изображения с наиболее сильными перепадами яркости. Выбор порога является одним из ключевых вопросов выделения перепадов. В нашей реализации он отличается от оригинального метода Собеля. В качестве основного порога берется средняя для изображения величина градиента - Smid. Для достаточно большого изображения с малым числом точек, обладающих сильным перепадом яркости, данной пороговой величины недостаточно, т.к. оказывается весьма сильным влияние шума. Для ликвидации этой проблемы для каждой точки изображения считается величина Slocal, равная средней величине градиента в области 3x3 вокруг анализируемой точки. Пороговое условие выглядит так:

(G(i, j)>=Smid) AND (Slocal >= Smid)

В результате обработки получается бинарная матрица, где единицам соответствуют точки со значительным перепадом яркости, нулям – все остальные. В качестве дополнительной меры в борьбе с шумом и

ликвидации возможных разрывов в контурах применяются морфологические операции.

Следующий этап – сегментация изображения. Целью сегментации является выделение на изображении контуров объектов. В бинарной матрице единицами представлены точки, принадлежащие искусственно утолщенным на предыдущем этапе границам объектов. Для выделения границы одного объекта в матрице по определенному алгоритму ищется элемент, равный единице, не отнесенный ранее ни к какому другому объекту; далее считается, что все соседние элементы, равные единице, также принадлежат этому объекту; и т. д. Для выделения точек внешнего контура используется обход полученного объекта по внешней его стороне, начиная с нижней левой точки объекта и заканчивая ею же. Обход точек ведется последовательно против часовой стрелки. В результате получаем массив точек, образующий замкнутый контур объекта. Из него равномерно выбирается 128 точек (xs, ys), где 1<=s<=128, которые используются для вычисления предназначенных для индексирования характеристик формы. (Небольшие объекты исключаются из рассмотрения.)

3.3. Характеристики формы

Существует практика использования формы объектов для индексирования изображений с целью их дальнейшего сравнения [2, 8, 26].

Для точек выделенного контура объекта в данной работе вводятся две функции.

1) Функция расстояния от точек контура до центра фигуры:

$$R(s)^2 = (x_c - x_s)^2 + (y_c - y_s)^2$$

где (xc=$\sum$ xs /N, yc=$\sum$ ys/N) - центр масс контура

2) Углы поворота:

$$Y(s) = \arccos((a^2 + b^2 - c^2)/\ 2ab))$$

$$a^2(s) = (x_{s-1} - x_s)^2 + (y_{s-1} - y_s)^2$$

$$b^2(s) = (x_{s+1} - x_s)^2 + (y_{s+1} - y_s)^2 \qquad c^2(s) = (x_{s-1} - x_{s+1})^2 + (y_{s-1} - y_{s+1})^2$$

Для обеспечения инвариантности относительно поворотов и масштаба, выполняется нормирование величин, а в качестве начальной точки контура берется та, расстояние до центра от которой наименьшее, соответственно упорядочиваются элементы векторов.

Предлагается способ сравнения форм объектов на основании вычисления общего расстояния между парами соотвествующих векторов.

Определение объекта изображения на основании близких значений интенсивности соседних точек позволяет довольно точно характеризовать выделенную область изображения с помощью такого показателя, как средний цвет точек области. Таким образом, для выделенных объектов могут быть определены и включены в индекс такие характеристики, как расположение, размеры, измерения формы, средний цвет.

4. Методы анализа видеоданных

4.1. Временное сегментирование видеофильма

В связи с большим объемом видео-файлов для организации эффективного поиска данных с удовлетворительными показателями полноты и точности, а также для обеспечения быстрого предоставления пользователю релевантной информации имеет смысл индексировать каждый фильм не как единое целое, а как последовательность логически самостоятельных частей — видеофрагментов [15]. Задача сводится к определению границ видеофрагментов, они могут быть связаны с точками монтажа, изменением положения снимающей камеры и т.п. Формально задачу можно поставить так: на вход подается упорядоченный набор кадров, необходимо выделить из них последовательность номеров, каждый из которых соответствует началу нового фрагмента.

Временное сегментирование может выполняться путем автоматического анализа изображения, соответствующие приемы известны [3, 6, 12, 25]. Достаточно эффективны для выделения кадров, на которых происходит значительное изменение видеоизображения, методы, основывающиеся на вычислении низкоуровневых характеристик изображения.

Предлагается алгоритм, основанный на сравнении цветовых гистограмм соседних кадров. Система вычисляет цветовую гистограмму очередного кадра и сравнивает с предыдущей. Построение и сравнение гистограмм осуществляются идентично работе со статичными изображениями. Есть дополнительная возможность “выравнивания” гистограмм, применение которой имеет смысл только в случае разбиения множества цветов по яркости. Целью ее является приведение гистограммы к виду, при котором верно:

$$\sum_{i=0}^{i=N/2-1} H[i] = \sum_{i=N/2}^{i=N-1} H[i]$$

После построения гистограммы определяется ее медиана, т. е. такое k, что

$$\sum_{i=0}^{i=k} H[i] = \sum_{i=k+1}^{i=N-1} H[i]$$

После этого все точки гистограммы пересчитываются, исходя из соотношения:
H'[i]=(H[k/(N/2)*i]+ H[k/(N/2)*i+1])/2, при 0<=i<N/2 и аналогичного выражения для N/2<=i<N. Оригинальная гистограмма заменяется на H'. Потребность такой обработки объясняется тем, что во многих реальных образцах видео (особенно черно-белых и/или плохого качества) часто встречаются практически идентичные соседние кадры, отличающиеся только по яркости среднего освещения. Семантически они должны попадать в один фрагмент, но их первоначальные гистограммы могут существенно различаться, что и приводит к необходимости выравнивания. Прием не дает полного решения проблемы, но, по крайней мере, существенно улучшает результаты на широком классе изображений.

Результатом сравнения гистограмм последовательных кадров является массив R[i] чисел от 0 до 2, где i-ый компонент – расстояние между гистограммами i-ого и (i+1)-ого кадров. Опираясь на эти данные, фильм разбивается на фрагменты. Граница фрагментов считается обнаруженной, если разница гистограмм рассматриваемых соседних кадров выше некоторого абсолютного порога и одновременно в определенное число раз превышает среднее значение разницы гистограмм соседних кадров, посчитанное по кадрам от начала выделяемого фрагмента (относительный порог).

Попытка ограничиться абсолютным порогом L не привела к успеху. Значение порога, дающее желаемые результаты, существенно зависит от качества записи фильма, динамики его связных фрагментов и средней освещенности, а эти характеристики могут быть различны в разных фрагментах одного и того же фильма (т. е. для их вычисления потребуются уже готовые результаты временной сегментации), а кроме того, определение, например, качества записи - трудная задача, и зависимость от нее неизбежно внесет дополнительную погрешность в итоговый результат. Аналогично, не удалось решить задачу и введением относительного порога, т. е. такого M, что при R[i]>M*ΣR[j]/(i-1-k) (сумма берется от k до i-1, k-й кадр – первый в текущем фрагменте), (i+1)-й кадр считается началом следующего фрагмента. Проблема заключается в возможности практически полного совпадения первых нескольких гистограмм фрагмента и минимального отличия от них следующей - условие порога будет выполнено, и программа выдаст лишний фрагмент. Наиболее эффективной оказалась комбинация этих подходов, т. е. для принятия решения о том, что

(i+1)-й кадр – начало нового фрагмента требуется выполнение обоих условий:

$R[i]>M*\Sigma R[j]/(i-1-k)$

$R[i]>L$

Результаты сегментирования, разумеется, сильно зависят от выбора параметров. Они установлены эмпирически для достижения приемлемых результатов с точки зрения минимизации числа ошибок, связанных с обнаружением ложной границы и пропуском действительной. (При уменьшении вероятности одной из ошибок, неизбежно повышается вероятность другой.) Текущие значения L=0.15 и M=3 подобраны так, чтобы ложные обнаружения встречались примерно на порядок чаще, чем пропуск переходов. Вызвано это тем, что на “двойном” видеофрагменте невозможно корректно вычислить оптический поток, что является одной из главных целей временного сегментирования, в то время как два фрагмента вместо одного дают лишь некоторое увеличение требуемых для дальнейшей обработки ресурсов.

Для тестирования программы использовались видеофильмы, взятые из различных источников, разного качества. Все фильмы разбиты на несколько непересекающихся групп, результаты внутри которых были более или менее одинаковы. Приводим усредненные результаты, отражающие процент ошибок, допускаемых различными методами временной сегментации на различных типах видеофрагментов.

Ложные обнаружения

Метод	(1)	(2)	(3)	(4)	(5)
Мультфильмы	10%	10%	31%	18%	0%
Цветное видео	33%	11%	33%	33%	27%
Черно-белое видео	23%	-	25%	-	12%

Пропущенные границы фрагментов

Метод	(1)	(2)	(3)	(4)	(5)
Мультфильмы	0%	0%	0%	0%	0%
Цветное видео	0%	0%	0%	0%	0%
Черно-белое видео	3%	-	2%	-	10%

Методы:

1 - Палитра разбивается по интенсивности.

2 - Палитра разбивается на RGB-параллелепипеды.

3 – Квадродерево & палитра разбивается по интенсивности.
4 – Квадродерево & палитра разбивается на RGB-параллелепипеды.
5 - Палитра разбивается по интенсивности, применяется усложненная формула расстояния и выравнивание гистограмм.
(Процент ошибок – отношение числа ошибок к сумме ошибок и верно обнаруженных переходов, умноженное на 100%.)

4.2. Индексирование видеофрагментов

После того как видеопоток разбивается на фрагменты, из них выделяются для исследования ключевые стоп-кадры. Стратегия извлечения представительных стоп-кадров из каждого выделенного фрагмента может быть, например, такой [1]: если фрагмент короче секунды, берется один центральный кадр, для более длинных фрагментов берется по одному в секунду. Для каждого выделенного кадра вычисляются с целью индексирования визуальные примитивы: цветовые гистограммы, характеристики формы и цвета объектов изображения, измерения текстуры; для этого применяются те же методы, что и для анализа статичных изображений. Кроме того, представляется важным индексировать фрагмент также характеристиками движения камеры/сцены и движения объектов, определяемыми на основании совокупности кадров видеофрагмента [3, 6].

4.3. Вычисление оптического потока

Для индексирования видеоданных по движению применяется метод оптического потока. Он основан на том, что для видеофрагмента, содержащего некоторые объекты в движении, можно вычислить направление и величину скорости движения в каждой точке видеокадра. Известны разные алгоритмы вычисления оптического потока [9].

В данной работе реализован дифференциальный метод расчёта оптического потока, который предполагает вычисление пространственно-временных производных интенсивности. Для того чтобы повысить точность вычислений, все кадры видеофрагмента предварительно сглаживаются с помощью фильтра Гаусса. (Предварительно осуществляется выделение кадров из видеофрагмента, цветные изображения преобразуются в черно-белые полутоновые.)

Ядро 2-мерного фильтра Гаусса представляет собой квадратную матрицу нечётного порядка, значения элементов которой соответствуют нормальному распределению. Значения интенсивности в точках изображения пересчитываются следующим образом: $\bar{I}_{kl} = \sum_{i,j=-n}^{n} G_{i+n+1,\,j+n+1} I_{k+i,l+j}$,

где $G \in \mathrm{R}^{(2n+1)\times(2n+1)}$ — ядро фильтра Гаусса. В данной работе использовалась матрица третьего порядка $G = \frac{1}{25}\begin{pmatrix} 2 & 3 & 2 \\ 3 & 5 & 3 \\ 2 & 3 & 2 \end{pmatrix}$. Авторы работы [9] предлагают применять, кроме пространственного, также временное сглаживание, а именно использовать для расчета интенсивности в точке кадра значения в близких точках соседних кадров. К сожалению, эта техника применима только в том случае, если скорость движения объектов на видеофрагменте не превышает одного-двух пикселов на кадр, а это условие далеко не всегда выполняется для реальных видеофрагментов.

Дифференциальная техника вычисления скорости в каждой точке опирается на простое правило: при движении объекта интенсивность составляющих его точек не изменяется:

$$I(\mathbf{x}+\mathbf{v}dt, t+dt) = I(\mathbf{x},t), \text{ или } \frac{dI(\mathbf{x},t)}{dt} = 0,$$

где $\mathbf{v} = (u,v)^t$ — скорость точки $\mathbf{x} = (x,y)^t$, $I(\mathbf{x},t)$ — интенсивность в точке $\mathbf{x}$ в момент времени t.

Это правило дает одно линейное уравнение для двухкомпонентного вектора скорости: $\nabla I(\mathbf{x},t)\cdot\mathbf{v} + I_t(\mathbf{x},t) = 0$,

где $\nabla I(\mathbf{x},t) = \left(\frac{\partial I(\mathbf{x},t)}{\partial x}, \frac{\partial I(\mathbf{x},t)}{\partial y}\right)^t$, $\nabla I(\mathbf{x},t)\cdot\mathbf{v}$ — скалярное произведение векторов.

Дополнительные ограничения могут быть получены различными способами. Например, реализованный в настоящей работе метод минимума градиента, идея которого принадлежит исследователям из Массачусетского технологического института [13], предполагает гладкое изменение значения скорости от точки к точке: $\|\nabla u(\mathbf{x})\|^2 + \|\nabla v(\mathbf{x})\|^2 = 0$, или $\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 = 0$.

Итак, для определения вектора скорости необходимо минимизировать интеграл

$$\iint_D \left[(\nabla I(\mathbf{x},t)\cdot\mathbf{v} + I_t(\mathbf{x},t))^2 + \alpha^2\left(\|\nabla u(\mathbf{x})\|^2 + \|\nabla v(\mathbf{x})\|^2\right)\right] d\mathbf{x}, \quad (1)$$

где D — множество точек изображения, а α — параметр, определяющий вес слагаемого, отвечающего за гладкость оптического потока. В настоящей реализации используется значение $\alpha = 100$, которое в процессе тестирования показалось оптимальным.

В работе [13] показано, что условие минимума интеграла (1) эквивалентно следующей системе равенств:

$$\begin{cases}\left(\alpha^2 + I_x^2 + I_y^2\right)\left(u - \bar{u}\right) = -I_x\left(I_x\bar{u} + I_y\bar{v} + I_t\right), \\ \left(\alpha^2 + I_x^2 + I_y^2\right)\left(v - \bar{v}\right) = -I_y\left(I_x\bar{u} + I_y\bar{v} + I_t\right).\end{cases} \quad (2)$$

где $\bar{w}(j,i,k) = \frac{1}{6}\left(w(j,i-1,k) + w(j+1,i,k) + w(j,i+1,k) + w(j-1,i,k)\right) +$
$+ \frac{1}{12}\left(w(j+1,i-1,k) + w(j+1,i+1,k) + w(j-1,i+1,k) + w(j-1,i-1,k)\right)$ — локальное среднее для *w(j,i,k)*, а символ *w* обозначает *u* либо *v*.

Авторы метода предлагают решать полученную систему линейных уравнений итерационным методом, например методом Гаусса — Зейделя. Тогда

$$\begin{cases} u_{n+1}(j,i) = \bar{u}_n(j,i) - I_x(j,i)\dfrac{\left(I_x(j,i)\bar{u}_n(j,i) + I_y(j,i)\bar{v}_n(j,i) + I_t(j,i)\right)}{\left(\alpha^2 + I_x^2(j,i) + I_y^2(j,i)\right)}, \\ v_{n+1}(j,i) = \bar{v}_n(j,i) - I_y(j,i)\dfrac{\left(I_x(j,i)\bar{u}_n(j,i) + I_y(j,i)\bar{v}_n(j,i) + I_t(j,i)\right)}{\left(\alpha^2 + I_x^2(j,i) + I_y^2(j,i)\right)}, \end{cases} \quad (3)$$

где $\mathbf{v}_k(j,i) = \left(u_k(j,i), v_k(j,i)\right)$ — значение вектора скорости в соответствующей точке кадра на *k*-й итерации; $\mathbf{v}_0 = 0$. Итерация связана с временным шагом. Итеративное вычисление оптического потока выполняется на основании всех кадров фрагмента для точного определения скоростей. В отличие от оригинального метода [13], для вычисления производных интенсивности по времени и по каждой из координат, вместо двухточечной схемы, применялась центральная 5-точечная разностная схема с коэффициентами $\frac{1}{12}(-1, 8, 0, -8, 1)$.

.

Полученный оптический поток используется затем для определения более высокоуровневых характеристик, связанных с движением, которые предназначены для индексирования и поиска видеоданных.

4.4. Выделение движущихся объектов

Применяемый для вычисления оптического потока метод позволяет, выполнив лишь одну итерацию, правильно определить нормальную составляющую вектора скорости на границе объекта изображения. Затем, по мере увеличения числа итераций, значение на границе приближается к реальному значению скорости. С другой стороны, согласно формуле (3) в точках изображения с нулевым градиентом интенсивности скорость будет определяться как среднее значение скоростей соседних точек. Следовательно, с увеличением числа итераций размер и форма объектов будет искажаться. Кроме того, ненулевые значения вектора скорости

получат все точки изображения, принадлежавшие объекту хотя бы на одном кадре последовательности.

Значит, чем больше число итераций, тем более правильно определяются величины скоростей, но в то же время неточно воспроизводится форма объекта. Поэтому для выделения движущихся объектов осуществляется вычисление потока отдельно по последним пяти кадрам без учёта предыдущих, с последующей подстановкой в точках с ненулевыми скоростями результатов итеративно вычисленного по всем кадрам оптического потока, как более адекватных. В полученном оптическом потоке представлены как форма движущихся объектов (если таковые присутствуют на видеофрагменте), так и достаточно точные значения скоростей:

Информация об оптическом потоке используется для пространственного сегментирования изображения: группу расположенных близко друг от друга точек, движущихся с примерно одинаковыми скоростями (или хотя бы приблизительно однонаправленными), можно считать движущимся объектом. В индекс можно включить информацию о расположении, размерах и некоторых других параметрах таких областей.

Выделение областей происходит по схеме: две точки считаются принадлежащими одному объекту, если они отстоят друг от друга не более чем на 3 пиксела и направления скоростей в них отличаются не более чем на 45°; не принимаются во внимание области, размер которых пренебрежимо мал ($< 0.1\%$) либо слишком велик ($> 30\%$) относительно размера кадра (в последнем случае движение области учитывается при определении глобальных характеристик движения сцены/камеры).

Для выделенных областей рассматриваются минимальные охватывающие их прямоугольники. Помимо их размеров и расположения, определяется тип движения, по той же схеме, что и для картинки в целом (см. следующий раздел). Для этого вычисляются средние значения скорости в четырёх квадрантах этого прямоугольника.

Затем вычисляется средний модуль скорости по всем точкам, принадлежащим объекту, для обеспечения возможности поиска видеофрагментов с требуемой интенсивностью движения объекта. Кроме того, вычислив оптический поток не только для последних кадров, но также для первых и для некоторых промежуточных, в случае поступательного движения можно определить характеристики траектории движения объектов. В качестве дальнейшего этапа исследований рассматривается задача определения происходящих с объектами событий.

После обработки объектов исследуются глобальные характеристики движения, для чего вычисляются средние значения вектора скорости в квадрантах изображения и средний модуль по всем точкам с ненулевыми скоростями (интенсивность движения сцены). Предварительно значения скорости в точках, принадлежащих найденным объектам, обнуляются — таким образом удаётся избежать влияния движения объектов на вычисление параметров движения сцены.

4.5. Характеристики движения

После вычисления оптического потока в каждой точки видеофрагмента и его общих характеристик (например, средней интенсивности), возникает задача привести эти сложные данные к более простой и пригодной для индексирования форме. В работе [1] излагается возможный способ решения проблемы.

В данной работе предлагаются новые характеристики, вычисляемые исходя из средних скоростей четвертей видеофрагмента. В случае обнаружения движущихся объектов, характеристики движения вычисляются отдельно для каждой прямоугольной области, содержащей объект, а также для всего изображения. Таким образом, предлагаемые алгоритмы могут применяться для вычисления как глобальных, так и локальных характеристик видео.

Создана многоуровневая классификация видеофрагментов по типу движения:

1) Идентификатор схемы движения

На основании средних скоростей в квадрантах анализируемого изображения выбирается наиболее близкая схема движения, определяющая для каждого квадранта одно из 8 основных направлений движения (с точностью до 45 градусов) или же отсутствие существенного движения. При определении схемы учитываются не только направления средних скоростей, но и соотношение их модулей. Так как в разных фрагментах интенсивность может сильно различаться, порог, значения ниже которого считаются нулевыми, не может быть установлен изначально. Сейчас принят следующий алгоритм его вычисления. Значения скоростей разбиты на интервалы экспоненциально растущего размера. Определяется интервал, в который попадает наибольшее значение модуля скорости, нулевыми же считаются все значения из остальных интервалов. В результате скорости каждого квадранта переводятся в целые числа от 0 до 8, что для четырех квадрантов дает 9*9*9*9=6561 комбинаций. Если назвать эту комбинацию идентификатором схемы, то и получится первый вид классификации: схожими будут считаться фрагменты с одинаковыми идентификаторами их

схемы движения. Несмотря на простоту, данная методика соответствует семантике многих реальных запросов. В частности, распознаваемы характерные функции камеры (приближение, удаление, сдвиг). Например, постепенному переходу к крупному плану будет соответствовать движение к центру во всех четвертях.

2) Доминирующее направление

Идея – разбиение всего множества фильмов на два класса: с выраженным общим направлением и без него. С человеческой точки зрения, в первый тип попадают фрагменты с крупным планом и движущимся центральным объектом, эпизоды снятые движущейся камерой или с движением фона. Во второй – все остальные фрагменты. Естественно, для фрагментов с доминирующим движением целесообразно хранить не только сам факт его наличия, но и направление, а также число квадрантов с этим направлением (мощность доминанты).

Вычисление производится по идентификатору схемы. Предварительно создается массив D[i] (1<=i<=8), такой, что для каждого i D[i] равно числу квадрантов с направлением i. Определение доминирующего направления после этого сводится к нахождению такого i, что D[i]>=3 или D[i]=2 и $\forall j \neq i \Rightarrow D[j]<=1$. Если удовлетворяющего этим условиям i не найдено, то доминирующего направления нет.

3) Мощность схемы определяется как количество квадрантов с не близкими к нулю скоростями.

4) Эквивалентность схем с точностью до поворотов

Поддерживается разбиение на набор классов, каждый из которых образован поворотом схемы с базовым идентификатором вокруг своей оси на 0 (базовая схема), $\pi/2$, π и $3\pi/2$. Семантически они соответствуют некому целостному движению, показанному с разных сторон. Полностью сохраняются все типичные видеоэффекты. Реализация, организованная с использованием библиотек классов, обеспечивает возможность настройки и расширения классификации.

Предложенная классификация обеспечит разносторонний поиск видеофрагментов. При таком подходе запрос сможет задавать для искомого видеофрагмента частично или полностью определенную схему движения, наличие некоторого доминирующего направления, количество квадрантов с ненулевыми скоростями, а также степень интенсивности движения. Понятие эквивалентности схем с точностью до поворотов позволит определять в запросе относительную схему движения.

5. Распознавание лица

Пользователю электронной библиотеки изображений должна быть предоставлена возможность строить запросы с использованием различных визуальных средств - в терминах не только визуальных примитивов, но и высокоуровневых объектов. Для этого в поисковом образе должен отражаться факт присутствия на изображении объектов наиболее интересных классов, а также размеры и расположение на кадре этих объектов. Задача нахождения на изображении объектов в настоящее время не ставится глобально. Как правило, речь идет об объектах определенного класса, особенно интересных для рассматриваемой предметной области. В рамках данной работы решается задача локализации фронтального вида лица человека на неподвижных изображениях / стоп-кадрах с помощью нейронной сети. Использование большого количества положительных и отрицательных примеров для обучения классифицирующего механизма позволяет автоматически получить достаточно точную модель объекта [17]. Примеры систем распознавания лица, использующих контролируемое обучение – [18, 24].

Разрабатываемая система применяется к черно-белым полутоновым изображениям.

Та часть системы, которая непосредственно определяет наличие или отсутствие лица на картинке, применяется к небольшой области изображения, размеры которой выбираются так, чтобы в этой области можно было бы свободно поместить неискаженное лицо, и в то же время чтобы вычисления не занимали много времени (20 на 20 пикселов). На выходе выдается число, близкое к "1", если эта часть картинки содержит лицо, и к "-1" в противном случае.

Чтобы определить наличие лица на изображении, описанный фильтр применяется к каждому его участку. Для определения лиц, размеры которых превосходят размеры входного изображения для фильтра, картинку уменьшают в размерах и снова к каждому участку полученного изображения применяют фильтр и так далее, пока размеры картинки не уменьшатся до размеров входного изображения фильтра. Механизм дает возможность находить лица разного размера.

Перед непосредственным применением фильтра его входная картинка проходит предварительную обработку, которая позволяет сети работать с изображениями независимо от их средней интенсивности и от влияния источника света на изображение.

Результатом выполнения этих шагов является множество областей, где обнаружены лица. Последний этап состоит в том, чтобы отбросить те области, где ошибочно обнаружены лица, и объединить те, в которых обнаружено одно и то же лицо несколько раз, учитывая процесс многократного масштабирования.

Таким образом, систему можно разделить на 2 подсистемы: 1) нейросетевой фильтр, включающий предварительную обработку, который для каждой области исходного изображения размером 20 на 20 пикселов выдает ответ, подтверждающий или опровергающий факт наличия лица, и 2) арбитр, который отбрасывает ошибочно обнаруженные лица. На данный момент реализована первая подсистема, а также механизм обучения сети.

Нейронная сеть данной системы состоит из трех слоев – входного и двух уровней нейронов. Вектор примитивов $\bar{x}$, поступающий на вход нейронной сети, является значением функции интенсивности пикселов изображения размером 20 на 20 пикселов и вычисляется по формуле $\bar{x} = 2\bar{I}/255 - 1$, где $\bar{I}$ - вектор интенсивностей пикселов картинки, $\bar{x} \in [-1;1]^{400}$. Внутренний слой состоит из 26 узлов, чувствительных к определенным областям входного изображения. Области выбраны для облегчения нахождения характерных черт лица. Внешний уровень состоит из одного узла, выходное значение которого интерпретируется как наличие или отсутствие лица. Используются биполярные функции активации нейронов.

Обучение сети на примерах позволяет автоматически настроить ее параметры. На вход системы подается набор изображений с известным выходом (1/-1). Система настраивается так, чтобы данное изображение отображать в заданное число. Процесс обучения основан на градиентном методе, минимизирующем суммарную квадратичную ошибку $E = E(\bar{w},\bar{u}) = \sum_{q=1}^{Q} ||t^{(q)} - z^{(q)}||^2$, где Q - количество примеров, $z^{(q)}$ - полученное выходное значение нейронной сети, $t^{(q)}$ - желаемое выходное значение. Обучающее множество состоит из двух подмножеств картинок. Одно содержит картинки с лицами, другое - картинки без лиц. Первое подмножество генерируется из реальных изображений, лица на которых различаются размерами, расположением, наклоном, интенсивностью освещения. На каждой картинке глаза помечаются вручную. Эти метки используются для приведения каждого лица к одному масштабу, расположению и наклону. Нормализация выполняется путем поворота, масштабирования и сдвига исходного изображения так, чтобы метки глаз заняли предопределенное место в окне 20 на 20 пикселов. В обучающий набор включаются как сами преобразованные изображения, так и изображения, полученные из них путем поворота на угол 5 градусов по часовой и против часовой стрелки, также включаются зеркальные отображения и повернутые зеркальные отображения на тот же угол. Это делает систему распознавания инвариантной к небольшим наклонам головы. Обучающий набор проходит предварительную обработку, для

устранения влияния источника света на изображение. Набор отрицательных примеров генерируется случайным образом.

Чтобы определить момент окончания обучения, используется проверочное множество (25% от общего количества обучающих примеров). Несколько итераций система обучается на 75% обучающего множества, затем вычисляется суммарная квадратичная ошибка на проверочном множестве. Такая процедура выполняется до тех пор, пока ошибка проверочного множества не начнет расти - тогда обучение заканчивается. Такая процедура позволяет избежать настройки системы исключительно на обучающее множество.

Для обучения системы было подготовлено 500 изображений с лицами, из которых было сгенерировано 3000 примеров лиц. Из них 2250 использовалась для обучения и 750 для проверки. Случайным образом было сгенерировано 375 изображений, не содержащих лица, которые использовались для обучения, и 125, которые составили проверочное множество. Предусмотрено повторное обучение системы на примерах, не содержащих лиц, если система ошибочно их выявляет.

Качество обучения проверяется на тестовом наборе задач, не пересекающемся с обучающей выборкой. После проведенного на данный момент обучения система показывает высокие показатели распознавания лиц, однако велик процент ошибочного их обнаружения. Это связано со сложностью подготовки всеобъемлющего множества отрицательных примеров и с тем, что пока не применяется арбитражная система.

Литература

1. Ardizzone, E., La Cascia, M., and Molinelli, D.,
Motion and Color Based Video Indexing and Retrieval,
Proc. Int. Conf. on Pattern Recognition, (ICPR-96), Wien, Austria, Aug. 1996.
http://www.cs.bu.edu/associates/marco/publications.html

2. Ardizzone, E., La Cascia, M., Vito di Gesu, and Valenti, C.,
Content Based Indexing of Image and Video Databases by Global and Shape Features, 1996.
http://www.cs.edu./associates/marco/publications.html

3. Baigarova, N. S. and Bukhshtab, Yu. A.,
Some Principles of Organization for Searching through Video Data,

Programming and Computer Software, Vol. 25, Nu. 3, 1999, pp. 165-170

4. Baigarova, N. S. and Bukhshtab, Yu. A.,
Digital Library of Documentaries "Cinema Chronicle of Russia",
10th DELOS Workshop on Audio-Visual Digital Libraries, Santorini, Greece, June 1999

5. Н.С. Байгарова, Ю.А. Бухштаб
Проект «Кинолетопись России» : представление и поиск видеоинформации
Труды I Всероссийской конференции «Электронные библиотеки», Санкт-Петербург, 18-22 октября 1999 г., стр. 209-215

6. Н.С. Байгарова, Ю.А. Бухштаб, Н.Н. Евтеева
Организация электронной библиотеки видеоматериалов
Препринт Института прикладной математики им. М.В. Келдыша РАН, 2000, N 5

7. Н.С. Байгарова, Ю.А. Бухштаб, А.А. Воробьев, А.А. Горный
Организация управления базами визуальных данных
Препринт Института прикладной математики им. М.В. Келдыша РАН, 2000, N 6

8. Н.С. Байгарова, Ю.А. Бухштаб, А.А. Горный
Методы индексирования и поиска визуальных данных
Препринт Института прикладной математики им. М.В. Келдыша РАН, 2000, N 7

9. Baron, J. L., Fleet, D. J., and Beauchemin, S. S., Performances of optical flow techniques. 1994.

10. Carson, C., Belongie, S., Greenspan, H., and Malik, J., Color- and Texture-Based Image Segmentation Using EM and Its application to Image Querying and Classification. 1997. http://elib.cs.berkeley.edu/papers/

11. Carson, C. and Ogle, V.E., Storage and Retrieval of Feature Data for a Very Large Online Image Collection. 1996.
http://elib.cs.berkeley.edu/papers/

12. Chrictel, M., Stevens, S., Kanade, T., Mauldin, M., Reddy, R., and Wactlar, H.,

Techniques for the Creation and Exploration of Digital Video Libraries, Multimedia Tools and Applications, Boston: Kluwer, 1996, vol. 2.

13. Horn, B.K.P. and B.G.Schunk, *Determining optical flow,* Artificial intelligence, 17,1981.

14. Jain, R. and Gupta, A., *Computer Vision and Visual Information Retrieval,* 1996 http://vision.ucsd.edu/papers/rosenfeld/

15. Jain, R. and Gupta, A., *Visual Information Retrieval,* Communications of the ACM, 1997, vol. 40, no. 5.

16. Jain, R., Pentland, A.P., Petkovic, D., *Workshop Report: NSF – ASPA Workshop on Visual Information Management Systems,* 1995. http://www.virage.com/vim/vimsreport95.html

17. Looney, C.G. ,"Pattern Recognition Using Neural Networks. Theory and Algorithms for Engineers and Scientists". Oxford University Press, 1997.

18. Rowley, H.A., Baluja, S., and Kanade, T., *Neural Network-Based Face Detection,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998. and In Proceedings of International Conference on Computer Vision and Pattern Recognition, pp. 203-208, San Francisco, CA.

19. Smith, J.R. and Shih-Fu Chang. Tools and Techniques for Color Image Retrieval. 1996. http://www.ctr.columbia.edu/~jrsmith/html/pubs/

20. Smith, J.R. and Shih-Fu Chang. Automated Image Retrieval Using Color and Texture. 1995. http://www.ctr.columbia.edu/~jrsmith/html/pubs/

21. Smith, J.R. and Shih-Fu Chang. VisualSEEk: a fully automated content-based image query system. http://www.ctr.columbia.edu/~jrsmith/html/pubs/

22. Smith, J.R. and Shih-Fu Chang. Automated Binary Texture Feature Sets for Image Retrieval. 1996. http://www.ctr.columbia.edu/~jrsmith/html/pubs/

23. Sobel, I., An isotropic image gradient operator. Machine Vision for Three-Dimensional Scenes, pp.376-379. Academic Press, 1990
24. Sung, K-K. and Poggio, T.,
Example-Based Learning for View-based Human Face Detection.
A.I. Memo No. 1521, December 1994.

25. Wactlar, H.D., Kanade, T., Smith, M.A., and Stevens, S.M.,
Intelligent Access to Digital Video: Informedia Project, 1996.
http://computer.org/computer/dli/r50046/r50046.htm

26. Wei-Ying Ma, NETRA: A Toolbox for Navigating Large Image Databases. 1997. http://vivaldi.ece.ucsb.edu/Netra/