Московский Государственный Университет им. М. В. Ломоносова
Механико-математический факультет
Кафедра вычислительной математики

Медников Дмитрий
группа 508

Дипломная работа

Распознавание объектов на полутновых изображениях на основе "активного сенсора". Алгоритмы и описание процесса.

Научный руководитель
д. ф.-м. н.
Кумсков М. И.

Москва, 2009

В дипломной работе развивается один из подходов к распознаванию объектов на полутоновых изображениях – подход использующий активный сенсор.

Проводится унификация определений с предыдущими публикациями на эту тему и их строгая математическая фиксация. Делается обзор существующих подходов к распознавания, на основании которого проводится классификация развиваемого подхода и указываются его преимущества.

Приводятся теоретические оценки вычислительной предлагаемой двухэтапной схемы распознавания. Описывается конкретная реализация системы использующей активный сенсор, и приводятся результаты численных экспериментов на ее основе.

Приводится процесс построения объектного классификатора, использующего активный сенсор. Приведены требования к информационной системе автоматизирующей данный процесс. Произведена разбивка классификатора на отдельные модули.

Оглавление

Оглавление 3

Введение. 5

Глава 1. 8

Обзор литературы. 8

Общая постановка задачи. 14

Основные определения. 15

Классификация и преимущества предлагаемого подхода к распознаванию. 17

Выводы. 18

Глава 2. 19

Описание РИАС подхода. 19

Теоретические оценки. 24

Реализация РИАС подхода. 26

I. Блок построения описания 26

II.Блок кодирования описания. 30

III.Модуль сравнения ОТ 31

IV.Быстрый поиск похожих ОТ в Хранилище шаблонов. 31

Результаты работы 32

Выводы. 34

Глава 3. 35

Процесс построения классификатора. 35

Краткое описание этапов. 35

Роли. 36

Артефакты. 37

Задачи. 38

Описание архитектуры предлагаемого классификатора. 44

Первичное распознавание. 44

Тонкое распознавание....45
Описание модулей системы....47
Конфигурация классификатора....50
Выводы....51
Заключение....52
Список литературы....54
Приложение....56
Эскизы системных сценариев....56
Библиотекарь....56
Архитектор....58
Тестировщик....59
Математик....59
Системные сценарии использования заказчика в случае наличия GUI....60

Введение.

Задача распознавания объектов на изображениях является одной из саамы актуальных среди задач распознавания. Существует множество предметных областей, где важно уметь качественно решать такие задачи – распознавание текстов, сравнение отпечатков пальцев, охранные системы, системы автоматического выписывания штрафов за превышение скорости и многие другие.

Целью данной работы является развитие метода идентификации целевого объекта на полутоновых изображениях со сложным фоном[21,22] и построение процесса создания классификатора и верификации качества распознавания (UML – модель процесса).

В работе решались следующие задачи:

1) Анализ существующих подходов распознавания изображений, управляемых моделями целевых объектов.
2) Развитие метода распознавания объектов (изображений) на основе Активного Сенсора, в плане унификации обозначений, этапов и формулировок базовых терминов метода.
3) Создание словаря терминов распознавания на основе использования Активного Сенсора (РИАС).
4) Определение основных артефактов, задач и соответствующих ролей для процесса построения и верификации объектного классификатора (программного модуля) (ПиВОК).
5) Построение UML представления процесса ПиВОК.
6) Использование системы моделирования (бизнес) процессов IBM Rational Method Composer для автоматизированного создания веб сайта с регламентом работ процесса ПиВОК.
7) Введение унифицированных обозначений терминов и понятий для представления формул в описании процесса ПиВОК.
8) Описание основных этапов РИАС и определение архитектуры системы распознавания, описание ее основных компонент.
9) Создание прототипа для тренировочной предметной области и проведение вычислительных экспериментов.

В результате были получены следующие результаты:

1) Создан словарь терминов РИАС и процесса ПиВОК, унифицированный с прошлыми публикациями по этой теме, введены символьные обозначения для формального описания процесса распознавания.
2) Построена UML модель процесса ПиВОК, описаны его основные этапы и артефакты, введено разбиение на задачи и роли исполнителей.
3) При помощи IBM Rational Method Composer построен сайт регламентов работ по процессу ПиВОК.
4) Описана базовая архитектура системы автоматизирующей процесс ПиВОК и введено описание ее основных компонент.
5) Описана архитектура распознающего программного модуля (РПМ) основанного на методе РИАС, дано описание унифицированного интерфейса его компонент.
6) На основе, описанной в работе, архитектуры создан прототип РПМ для ограниченной тренировочной предметной области, как результат работ проведенных по описанному процессу ПиВОК.

В дальнейшем планируется реализация системы автоматизирующей описанный в работе процесс (на данный момент готова только ее архитектура), добавление в нее большого количества версий алгоритмов и проведение тестовых расчетов на большом количестве предметных областей для доказательства эффективности предложенной модели процесса ПиВОК.

В первой главе данной работы проводится обзор существующих подходов к распознаванию, дается постановка задачи распознаванию объектов на полутоновых изображениях и проводится классификация развиваемого в работе подхода к распознаванию с использованием активного сенсора.

Во второй главе дается описание РИАС подхода к распознаванию, приводится реализация классификатора, использующего такой подход, и даны результаты его работы.

В третьей главе дипломной работы дается описание процесса построения РИАС классификатора, описана модульная структура такого классификатора.

В приложениях приведены требования к системе автоматизирующей процесс, описанный в третьей главе, дан глоссарий используемых при работе с РИАС подходом терминов и определений.

Результаты работы опубликованы:

1) Mednikov D.I., Milovidov A., Sergunin S.Yu., Kumskov M.I. **“**Object identification on grayscale images with complicated background using the active sensor model”, РОАИ – 2008, труды конференции.

2)D. I. Mednikov, A. Milovidov, S. Yu. Sergunin, M. I. Kumskov “Identification of stable description elements using an active sensor”, Pattern Recognition and Image Analysis, Volume 19, Number 1 / Март 2009 г.

Глава 1.

Обзор литературы.

Рассмотрим существующие подходы к распознаванию объектов на изображениях, для каких предметных областей решается эта задача и каким образом. Преимущество будем отдавать методам, использующим в своей работе модели, как наиболее перспективным, с нашей точки зрения, и наиболее близким к развиваемому нами подходу.

В работе View-Based Recognition [1] вводится метод распознавания трехмерных объектов только по набору их 2D проекций, без использования 3D модели, который лег в основу данной работы. Считается, что объект представляет собой твердое тело с некоторыми особенностями (в наших терминах их можно назвать особыми точками). На проекциях расположение этих особенностей может несколько отличаться от положения на объекте (так моделируется разное падение света при съемке). В работе показано вычислительное преимущество рассмотрения набора этих особенностей, не как трехмерного облака, а как набора проекций этого такого облака и вычислено какое число проекций необходимо брать для такой же точности распознавания как и в трехметном случае, в зависимости от того насколько особенности на проекциях отличиаются от особенностей на объекте.

Так же для предложенного в данной работе подхода важна статья Computational Limitations of Model-Based Recognition[2], где показано, что Model-Based Recognition (является ли сцена объединением как-то повернутых и отмасштабированных объектов из базы данных) является NP-полной задачей. То есть она экспоненциальная относительно количества объектов в базе. При этом объясняется, что сложность можно снизить использую более сложные структуры, чем просто наборы точек с характеристиками.

Дальше мы рассмотрим к каким задачам применим основанный на моделях подход к распознаванию, и какие конкретно реализации этого подхода существуют(какие бывают модели, как они используются в процессе распознавания и т.д.)

Для начала рассмотрим универсальные подходы к распознаванию, не предназначенные для какой-то конкретной предметной области.

В статье A Novel Algorithm for Automatic 3D Model-based Free-form Object Recognition[3] используются трехмерные модели объектов. Объект представляет из себя mesh(трехмерная сетка, многогранник сложной формы). Выбираются все пары вершин, лежащих не слишком далеко и не слишком близко друг от друга, и для каждой пары строится тензор. Вокруг середины отрезка, соединяющего пару вершин, строится

координатная сетка некоторого размера. В качестве координат тензора используются значения площади кусочка объекта попавшего в соответствующую ячейку. Для тензоров строится 4D хэш таблица (координаты точки, вокруг которой строили тензор, и номер отрезка, в который попадает угол между нормалями пары вершин, по которым тензор построен, шаг бискретизации 5 градусов) для ускорения поиска. Распознавание происходит в два этапа. Выбирается гипотеза наиболее похожая на тензор тестового объекта, проверка производится совмещением мэшей гипотезы и тестового объекта и сравнения их перекрытия.

В работе Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization[4] строится система распознающая объекты на сценах и указывающая их расположение и ориентацию (по три координаты на каждую характеристику, итого 6 мерное пространство). Объект описывается набором своих изображений в насыщенных, ярких цветах (для лучшей сегментации в цветовом пространстве HSV) в разных ориентациях и положениях на черном фоне. Тестовые изображения поступают со стерео камеры, то есть парами. На изображения выделяются сегменты одного цвета, описываемые своим положением в пространстве, окружающим прямоугольником и площадью сегмента. К полученному набору векторов применяется метод главных компонент. Ориентация и положение на сцене определяются за счет стерео эффекта. Распознавание происходит в два этапа: первичное распознавание на основе векторов сегментов и подтверждение на основе простых 3D моделей объектов, строящихся за счет стерео эффекта.

В статье Probabilistic Model-based Object Recognition using Local Zernike Moments[5] вычисляется вероятность нахождения того или иного объекта на сцене, а не утверждается конкретный факт его наличия или отсутствия. Как и в данной работе, для работы используются особые точки (чем то выделяющиеся участки изображения) – пересечения контуров, выделенных при помощи метода Харриса. Окрестности особых точек описываются моментами Зерника (Zernike moments), но поскольку они инвариантны только относительно поворотов, но не изменений масштаба и яркости, то применяется теория пространства масштабов (scale space theory), которая, как утверждается, более эффективна, чем пирамида изображений, и яркость пикселей окрестностей особых точек нормализуется. В процессе распознавания используется метод вероятностного голосования (probabilistic voting) – считается вероятность того, что объект находится на сцене и из всех таких вероятностей ищется максимальная. Для подсчета вероятностей используются две меры: мера стабильности (насколько момент Зерника в особой точке

отличается от моментов в соседних точках) и мера похожести (эвклидово расстояние между двумя моментами Зерника особой точки на модели и на входной сцене). Пары похожих особых точек ищутся при помощи алгоритма нахождения ближайшего соседа. Вероятность того, что объект изображен на сцене, считается как сумма некоторых функций от выше названных двух мер.

В работе Weakly Supervised Learning of Part-Based Spatial Models for Visual Object[6] Recognition в качестве модели объекта рассматривается набор шаблонов и k-fans граф (все вершины соединены с k реперными и не соединены между собой), каждому ребру которого соответствует распределение вероятностей взаимного расположения соединяемых им шаблонов. Шаблон представляет из себя матрицу n*n, где для каждого элемента и любого l известна вероятность, что в нем будет контур при выделении контуров в l направлениях. Шаблоны выбираются автоматически для заданного набора тестовых изображений содержащих целевой объект и набора не содержащих его изображений. Из большого числа различных возможных шаблонов выбираются те, что присутствуют на изображениях, содержащих объект, и не присутствуют на не содержащих. По ним строится k fan граф вероятностей их взаимного расположения. При распознавании на тестовом изображении ищется наиболее вероятная модель и наиболее вероятное расположение ее частей.

В статье Model based selection and classification of local features for recognition using Gabor filters[7] рассматривается метод описания окрестностей особых точек на основе фильтров Габора. На первом этапе окрестности особых точек описываются набором функций от габоровских фильтров, которые при помощи интегрирования делаются независимыми от поворотов и изменений масштаба, для набора разных частот (числовой вектор получается). Часть особых точек после выделения отбрасывается, если их признаковый вектор сильно отличается от среднего признакового вектора для этого типа объектов по всем эталонным изображениям с объектами этого типа. После отбрасывания оставшиеся особые точки описываются векторами габоровских фильтров при параметрах поворота и масштабирования, дающих максимальные по модулю функции Габора в этих точках, для разных частот. Сравнение особых точек происходит в метрике Махаланобиса.

Далее рассмотрим работы, описывающие специализированные методы распознавания для конкретных предметных областей.Начнем с одной из самых популярных групп методов – методов распознавания лиц.

В работе Component-based Face Recognition with 3D Morphable Models[8] строятся morfable 3D модели лиц по трем фотографиям. Они используются во время обучения.

Собственно распознавание производится по 14 зонам лица. Распознавание происходит при помощи настраиваемых на этапе обучения SVM (в процессе обучениям им даются как части лиц так и не имеющие к лицам отношения объекты).

Статья Three-Dimensional Model Based Face Recognition[9] описывает системы распознающую по фотографии одну из трехмерных моделей лиц, хранящихся в ее базе данных. Утверждается, что изменения в освещении, угле зрения (ось головы должна быть вертикальной), выражении лица не мешают распознаванию. Используется двух ступенчатый подход – гипотеза строится по соотнесению трех точек (внутреннего края глаза, внешнего края глаза, кончик носа), подтверждение происходит с применением гибридного ICP – системы пытается итерационно соотнести фотографию и 3D модель, минимизируя расстояние по некоторой метрике.

В статье Face Recognition with 3D Model-Based Synthesis[10] лица описываются трехмерными моделями, при этом особенность метода такова, что модель строится по одному (фронтальному) изображению. На изображении вручную отмечаются особые точки соответствующие точкам на 3D модели после чего на основе фронтального изображения делается попытка по яркости и геометрическим размерам восстановить координату глубины. На ее основании модель корректируется. Текстура на 3D модель не натягивается, а просто для каждого треугольника между особыми точками (ОТ больше 100 штук) его яркость определяется на основании яркости в его вершинах, чье положение на изображении лица точно известно и задано вручную. После этого из модели строят варианты моделей этого же лица, но при другом освещении (3D модель освещают разными источниками света, используя метод Фонга) и другом выражении лица (на модели выделено 18 «мышц» и их в разных комбинациях «сокращают»). Оригинальную модель и ее синтезированные вариации рассматривают как длинные вектора (конкатенация всех рядов пикселей отрендеренных моделей) в некотором аффинном пространстве. Соответственно при получении на вход лица на распознавание у него строится набор моделей и определяется расстояние между подпространством этого набора и подпространствами наборов моделей из базы лиц и ищется минимум этого расстояния.

В статье A Model Based Method for Automatic Facial Expression Recognition[11] рассматривается метод, распознающий не просто лица, а их выражения (одна из 7ми основных эмоций). Лицо описывается вектором формы – Principle Component Analysis примененный к вектору из координат особых точек лица - и вектором текстуры значения цвета в некоторых точках внутри области ограниченной ОТ из предыдущего вектора, к этому вектору тоже применяется PCA. Моделью называется вектор получаемый

объединением этих двух. По такой модели можно восстановить исходное лицо с некоторой точностью (не 100% из-за PCA). Поэтому есть возможность подбирать модель и параметры позы для наилучшего соотнесения изображения лица к модели. Делается это итерационно при помощи линейной регрессии. Итак, модель представляет собой вектор. Распознавание эмоций по этим векторам делается при помощи нейронной сети. Используя такие модели, можно не только распознавать эмоции, но и синтетически строить изображения лиц с заданными эмоциями по данным их изображениям с нейтральным выражением лица. В обучающей выборке считаются разности векторов между эмоцией I лица A и его нейтральным выражением. Потом для каждой эмоции в полученных для нее векторах выделяются характерные компоненты, получается «характерный» вектор смещения для данной эмоции, он то и добавляет к модели целевого лица для получения целевой эмоции.

В работе Video-Based Face Recognition Using Adaptive Hidden Markov Models[12] распознаются видео последовательности, содержащие лица. Для каждого лица в базе хранится видео последовательность, содержащая это лицо. Каждый кадр описывается вектором размера равного количеству пикселей умноженному на три. К полученному набору векторов применяется метод выделения главных компонент для уменьшения размерности векторного пространства, его содержащего. Каждое лицо в базе описывается N звенной Марковской цепью, где каждый элемент цепи соответствует одному из N классов, на которые разбиваются собственные вектора этого лица. Ключевой особенностью предложенного подхода является адаптируемость Марковских цепей в процессе распознавания – после распознавания тестовой последовательности ее вектора для обновления модели распознанного на ней лица.

В конце рассмотрим, для каких еще предметных областей применяются методы распознавания, основанные на моделях.

В статье Combining Model-based and Discriminative Classifiers : Application to Handwritten Character Recognition[13] распознаются рукописные буквы. Система состоит из двух частей – модельного классификатора и нейронной сети для определения модели, когда классификатор относит символ к двум моделям с почти одинаковой вероятностью. Модель – это набор картинок эталонных символов написанных разным почерком. При помощи кластеризации считается вероятность принадлежности нового символа к каждой из моделей, при этом, чтобы обеспечить устойчивое распознавание часто путаемых моделей (типа 1 и 7) для каждой такой пары строится нейронная сеть их друг от друга различающая. Если две наиболее вероятных модели, к которым относится символ,

принадлежат множеству пытаемых пар, то окончательное решение о принадлежности символа принимает нейронная сеть.

В работе Evaluation of Model-Based Interactive Flower Recognition[14] описывается система, распознающая цветы. Важно отметить, что как и в предлагаемом в данной работе методе, в процессе распознавания активно участвует человек и меняет параметры модели для наилучшего распознавания. Цветок описывается кривой «розы» и цветовыми моментами внутри этой кривой. Человек меняет параметры «розы» для распознаваемого цветка, система выдает наиболее похожие цветки в базе. Человек уточняет параметры розы, после чего процесс повторяется. Объясняется, что такая модель объединяет преимущество людской точности распознавания и компьютерной быстроты работы с большими объемами информации. По сути, имеется обычная систему «гипотеза - подтверждение», только подтверждением занимается человек, а не машина.

В статье Human Model and Motion Based 3D Action Recognition in Multiple View Scenarios[15] описывается система, распознающая движения людей снимаемых на несколько камер сразу. Кадры с N камер сегментируются алгоритмом Стауффера-Гимсона – движущиеся объекты отделяются от неподвижных. Вокруг движущихся объектов строятся ограничивающие прямоугольники. Для уменьшения вычислений, попавшие в них силуэты объектов, превращаются в наборы вокселей, при этом изолированные воксели выкидываются. В любой момент движущиеся люди описываются натянутыми на них 4 элипсойдами (ноги и руки). Движение в любой точке сцены описывается «энергией» (как долго в этой точке находился движущийся объект) и «историей» (как давно через эту точку прошел движущийся объект). Наборы значений истории и энергии описываются 5 моментами (метод Дона-Ло) элипсойды частей тела описываются тем сколько внутри них движущихся вокселей, таким образом, каждый движущийся объект на сцене (1 кадр с каждой из N камер) описывается 14 мерным вектором. В получившемся векторном пространстве выделяются главные компоненты, и проводится Байселовская классификация.

В работе Appearance based Recognition Methodology for Recognising Fingerspelling Alphabets[16] распознаются знаки, показываемые руками (язык глухонемых) на полутоновых изображениях. Для описания изображений применяется метод главных компонент (PCA), но перед этим изображения либо обрабатываются винеровским фильтром, либо к ним добавляются результаты применения собелевского фильтра, либо изображения загрубляется с 256 до трех градаций серого. В статье проведено сравнение результатов распознавания для всех трех предложенных обработок.

В статье A hidden Markov Model Based Framework fro Recognition of Humans from Gait[17] Sequences распознается походка человека. Каждая походка в базе задается черно-белой (двухцветной) видео последовательностью. Моделирование происходит при помощи Марковских цепей. Походка естественным образом разбивается на N фаз, поскольку они повторяются, в каждой видео последовательности можно выделить наиболее характерные кадры для каждой фазы, они то и составляют элементы марковской цепи.

В работе Hierarchical Model-Based Activity Recognition With Automatic Low-Level State Discovery[18] описывается система, распознающая на видео последовательностях сложные действия людей, например, купил человек вещь в магазине или своровал ее. Действие рассматривается как последовательность простых поддействий. Используется трех уровневая иерархическая динамическая байесовская сеть. Первый уровень это уровень высокоуровневых действий, их мы пытаемся распознать, они маркируются в тестовой информации. Второй уровень - это уровень низкоуровневых действий, они в тестовой выборке не маркируются. Третий уровень – уровень фаз низкоуровневых действий. Каждое действие на любом из трех уровней представляет из себя дискретный стохастический процесс X(t) и булеву функцию E(t), определяющую в каждый момент времени, закончился ли процесс X(t) или нет. На действия накладываются ограничения вида: k-ая фаза действия X (X(k)) не может начаться, пока не закончится l-ая фаза действия Y(Y(l)). В результате получаем граф, вершинами которого являются фазы действий и их E, а ребра определяют вероятность перехода от одной фазы к другой. В нем высокоуровневые события зависят от низкоуровневых и наоборот (это важно), при этом вероятность перехода от одного события к другому зависит от того, завершилось первое или нет. На стадии обучения в качестве событий нижнего уровня берутся кластеры векторов наблюдений (центры движущихся областей и их скорости на каждом кадре). По ним и известным высокоуровневым действиям строятся средне уровневые действия как некоторое объединение низкоуровневых.

Общая постановка задачи.

Задача распознавания объектов на полутоновых изображениях ставится следующим образом. Пусть дан набор полутоновых изображений проекций $Pr_j(O_i)$ объектов O_i, набор полутоновых изображений эталонных фонов B_k и полутоновое изображение тестовой сцены Sc со сложным фоном. Требуется найти какие объекты, в каких проекциях находятся на сцене, и в какой части сцены они находятся, или же сообщить, что известных

объектов на сцене нет, то есть построить функцию $F(Sc, O_i) = \{e_i, pos_i, orient_i, scale_i\}$, где $e_i = \begin{cases} 0, \text{ если объект } O_i \text{отсутствует на сцене } Sc \\ 1, \text{ если объект } O_i \text{присутствует на сцене } Sc \end{cases}$, $pos_i = (x_i, y_i)$ – координаты объекта O_i на сцене Sc (определено если $e_i = 1$), $orient_i = (\rho_i, \theta_i, \varphi_i)$ - ориентация проекции $Pr(O_i)$ присутствующей на сцене Sc относительно эталонной системы координат объекта O_i, $scale_i$ - масштаб проекции $Pr(O_i)$ присутствующей на сцене Sc, относительно эталонного масштаба объекта O_i.

Программный модуль решающий такую задачу распознавания мы назовем *классификатором*. Задача построения и верификации классификатора определенной предметной области ставится следующим образом. Пусть дан набор полутоновых изображений проекций $Pr_j(O_i)$ объектов O_i и фонов B_k, характерных для определенной предметной области, а так же набор тестовых сцен Sc_l, содержащих проекции O_i (возможно отличные от $Pr_j(O_i)$) на B_k для верификации классификатора, требуется построить классификатор $F(Sc_l, O_i)$, показывающий заданные критерии качества распознавания PC на тестовых сценах Sc_l ($PC(F, Sc_l) = 1, \forall l$) . Под *критериями качества* PC могут пониматься любые численные ограничения на качество распознавания – процент ошибок первого и второго рода, время работы и т.д.

Основные определения.

Введем основные объекты, с которыми мы будем иметь дело при описании РИАС подхода к распознаванию.

Назовем *объектом* O_i твердый трехмерный предмет определенной текстуры с привязанной к нему эталонной сферической системой координат и эталонным масштабом.

Проекцией объекта $Pr(O_i)$ мы будем называть полутоновое изображение объекта на однородном фоне определенного цвета в определенном масштабе и расположенного под определенным углом к эталонной системе координат объекта.

Силуэт объекта $Sil(Pr(O_i))$ – замкнутый контур, охватывающий минимальную область содержащую объект O_i на его проекции $Pr(O_i)$.

Будем называть *особой точкой* (ОТ) $SP(Im)$ некоторый характерный участок изображения. ОТ $SP(Im)$ описывается вектором (x, y, m), где (x, y) координаты SP на Im, m - числовой маркер, описывающий класс этой точки. Полагается, что в *r*-окрестности особой точки не может быть других особых точек, если они есть то вместо этой группы ОТ ставится одна *конгломерирующая ОТ*. Конкретные реализации могут быть пересечениями и \ или изломами контуров объектов O_i на изображении Im, центрами

геометрических фигур, которые можно выделить на этих контурах, или центрами цветовых сегментов изображения.

Облаками особых точек мы будем называть подмножества множества всех особых точек данного изображения.

Будем называть *описанием изображения* $D(Im)$ множество особых точек $SP_i(Im)$ такое, что по нему можно определить «похожесть» изображения на другой изображение, используя только функции сравнения облаков ОТ, но не участков изображений.

Конфигурация облака особых точек $Conf(O_i(Im))$ – вектор определенной размерности, описывающий взаимное расположение $O_i(Im)$, находящихся на изображении Im, принадлежащих этому облаку.

Похожесть двух *конфигураций* особых точек $Sim(Conf_i, Conf_j)$ расстояние между векторами этих конфигураций в некоторой метрике, например, Евклидовой.

Рейтингом одной ОТ относительно другой $Rank(O_i, O_j)$ будем называть число, описывающее похожесть облаков ОТ, попадающих к круговые окрестности $Reg(O_i)$ этих O_i, друг на друга. Чем он больше, тем более похожи окрестности.

В качестве *модели* $M(Pr(O))$ проекции Pr объекта O рассматривается пара $(\{SP_i(\mathrm{Pr}(O))\}, Sil(\mathrm{Pr}(O)))$, то есть набор ее особых точек и силуэт объекта на этой проекции.

Важнейшим определением для этой работы будет ***окно сенсора*** $W(Im, x, y, s)$ – прямоугольник фиксированного размера, заданный рассматриваемым изображением Im, координатами положения своего центра на этом изображении (x, y) и масштабом попадающего в него кучка изображения Im. Система «смотрит» на изображение только через окно сенсора, в котором происходит описание ОТ их окрестностями и сравнение облаков ОТ, ОТ вне его при этом не учитываются. В каждый момент времени система работает только с частью изображения расположенной внутри окна сенсора, и «забывает», что происходит вне окна (вернее она помнит только, насколько похожа внешняя часть изображения на тот или иной объект, но не помнит данный об особых точках этой области). Окно сенсора может перемещаться по изображению в процессе распознавания, и динамически изменять уровень «зума», т.е. отображать разные по площади участки изображения в зависимости масштаба, в котором в текущий момент происходит работа.

Подход к распознаванию объектов, развиваемый в этой работе, мы будем называть Распознаванием с Использованием Активного Сенсора (*РИАС*).

Процесс построения и верификации объектного классификатора реализующего методику РИАС, которому посвящена существенная часть этой работы, мы будем называть процессом Построения и Верификации Объектного Классификатора (ПиВОК).

Классификация и преимущества предлагаемого подхода к распознаванию.

Проведем классификацию РИАС подхода к распознаванию на основе проведенного обзора существующих методов распознавания и введенных определений. Процесс распознавания двухэтапный – сначала происходит направляемое человеком оператором обучение системы, на характерных объектах и фонах предметной области, затем автоматический процесс верификации настроенной системы на тестовых сценах этой предметной области. (Похожую структуры мы видели в статье Evaluation of Model-Based Interactive Flower Recognition [14]) Процесс РИАС является универсальным, то есть его можно настроить под любую предметную область, не меняя самого процесса, только настройкой его компонент. (Среди рассмотренных нами подходов к распознаванию к универсальным можно отнести работы [3,4,5,6,7]) Для распознаванию используются двумерные проекции объектов, а не их трехмерные модели[3,8,10] В методе используются только локальные особенности изображения, называемые, как сказано выше, особыми точками (как и в [5,7]), при этом алгоритм их получения не важен и является одним из параметров, подстраивающихся под конкретную предметную область, особые точки должны лишь удовлетворять определенным критериям – их число на изображении должно быть небольшим и они должны быть практически инварианты относительно поворотов изображенных объектов, изменения масштаба и освещенности сцены (отличными свойствами инвариантности удовлетворяют, например, SIFT особенности[19], их главным недостатком для предложенного метода является только их относительно большое число на изображении).

Теперь остановимся на особенностях РИАС подхода, отличающих его от ранее рассмотренных методов распознавания. Основной особенностью метода РИАС является активность распознаваемых шаблонов – в каждый момент времени наиболее вероятный шаблон руководит собственным подтверждением, двигая окно сенсора по своим реперным особым точкам (по сути, при помощи окна активного сенсора и активного шаблона моделируется поведение человеческого глаза при распознавании объектов [20]), таким образом, алгоритм подтверждения модели не зашит, а является индивидуальным для каждой модели (обычно это не так[3,4 и другие]). Из-за этой особенности у оператора

появляется возможность подстраивать не только параметры выделения и поиска особых точек, но и саму схему распознавания индивидуально для каждой проекции объекта, изменяя набор реперных особых точек. То есть появляется дополнительный мощный инструмент настройки, позволяющий улучшить качество распознавания. Другой особенностью, позволяющей улучшить качество распознавания, является отделение объектов от фона при помощи использования силуэтов на этапе подтверждения моделей, это позволяет убрать влияние особых точек фона на распознавание. Еще одно преимущество в вычислительной сложности за счет двухуровневого процесса распознавания «поиск-подтверждение»[4, например] будет доказано в дальнейшем.

Выводы.

Мы рассмотрели несколько основополагающих работ, на основе которых, по сути, были разработаны подходы к распознаванию, используемые в данной работе. Так же было показано разнообразие задач сводящихся к распознаванию объектов на изображениях и подходов к решению таких задач. На основании проведенного обзора была сформулирована постановка конкретной задачи распознавания, которую мы будем решать (как мы видели, формулировки могут быть очень разнообразны),проведена классификация предлагаемого подхода по методам решения этой задачи и обозначены преимущества РИАС подхода над рассмотренными выше.

Глава 2.

Описание РИАС подхода.

Теперь подробно опишем общую схему работы РИАС системы. В работе распознающей системы такого типа можно разделить на две основные задачи: наполнение базы объектов, во время которого строятся шаблоны, и собственно распознавание (тоже двухуровневое)[23].

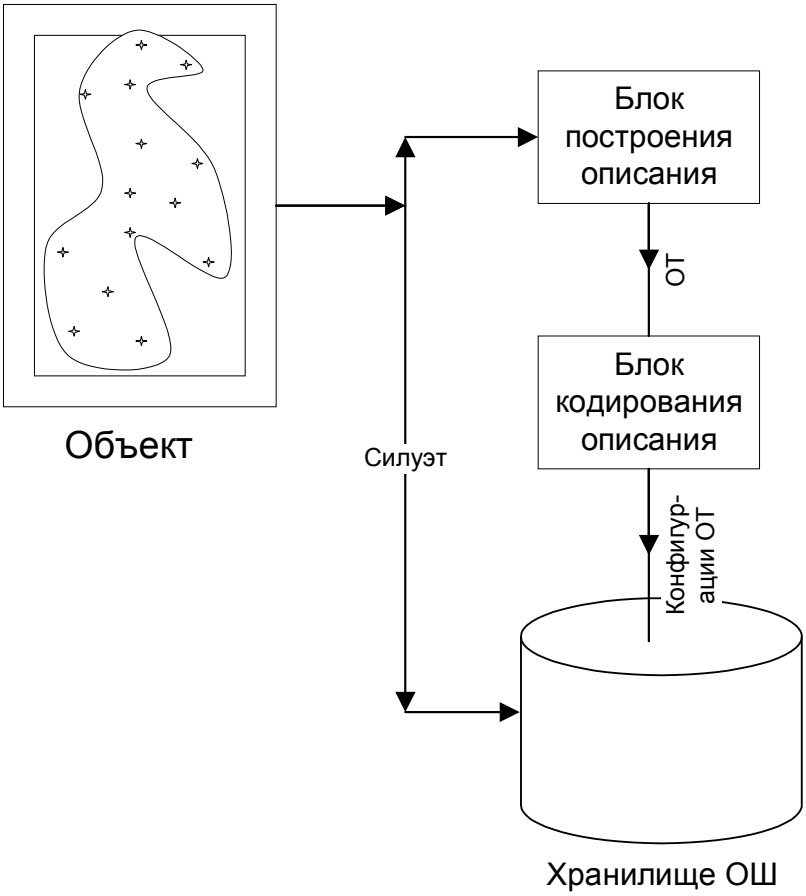

Рисунок 2-1.Схема наполнения базы ОШ

Этап наполнения базы проходит при активном участии человека оператора, то есть является полуавтоматическим. В систему заносятся изображения объектов O_i. В **Блоке построения описания** на каждом изображении *Im* для каждого масштаба (количество масштабов используемых в работе является параметром системы) этого изображения при помощи того или иного алгоритма определяется положение особых точек $\{SP_i(Im)\}$, далее им присваиваются некоторые характеристики участка изображения, составляющего их ближайшую окрестность (например средняя яркость этого участка, или количество ветвей контуров объектов пересекающихся в особой точке), то есть каждая особая точка характеризуется своими координатами и вектором характеристик своей окрестности.

Некоторые из них выделяются оператором как реперные, они должны располагаться в самых характерных участках объекта на изображении. После того как ОТ всех объектов выделены (они составляют множество $\{SP_i\}$), в том же блоке проводится их маркировка на основе кластер анализа векторов характеристик, соответствующих особым точкам (в качестве маркера берется номер кластера i в множестве $\{KL_i\}$). Таким образом, многомерное пространство векторов характеристик сводится к одномерному пространству маркеров. Полученные маркированные особые точки поступают в **Блок кодирования описания**, где описываются векторами своих конфигураций $K(SP_i)$, то есть происходит обратный переход от одномерного пространства маркеров в многомерное пространство векторов конфигураций, но теперь описания ОТ зависят только от других особых точек, а не от соответствующих им участков изображения. Так получаются необходимые нам описания изображений $D(Im_i)$, абстрагированные от самих изображений, то есть двумерная матрица пикселей изображения переводится в плоский граф своих особых точек, таким образом существенно уменьшается вычислительная сложность при дальнейшем распознавании. Полученные векторы записываются в **Хранилище объектных моделей** (базу данных), параллельно туда же для каждого объекта записывается его силуэт Sil. В хранилище вектора конфигураций особых точек индексируются для дальнейшего быстрого поиска похожих по конфигурациям ОТ по их индексам. Итак, в **Хранилище** оказываются все модели проекций объектов $M(Pr_i(O_j))$.

Распознавание в РИАС системе бывает двух видов: грубое - на маленьком масштабе, без использования силуэтов (используется для быстрого выдвижения объектных гипотез – на изображении находится данный объект в данной проекции) и тонкое – на крупном масштабе, только по областям внутри силуэтов объектов (таким образом происходит подтверждение объектных гипотез по реперным особым точкам). Эти два вида алгоритмически достаточно похожи, поэтому для начала рассмотрим их общую часть.

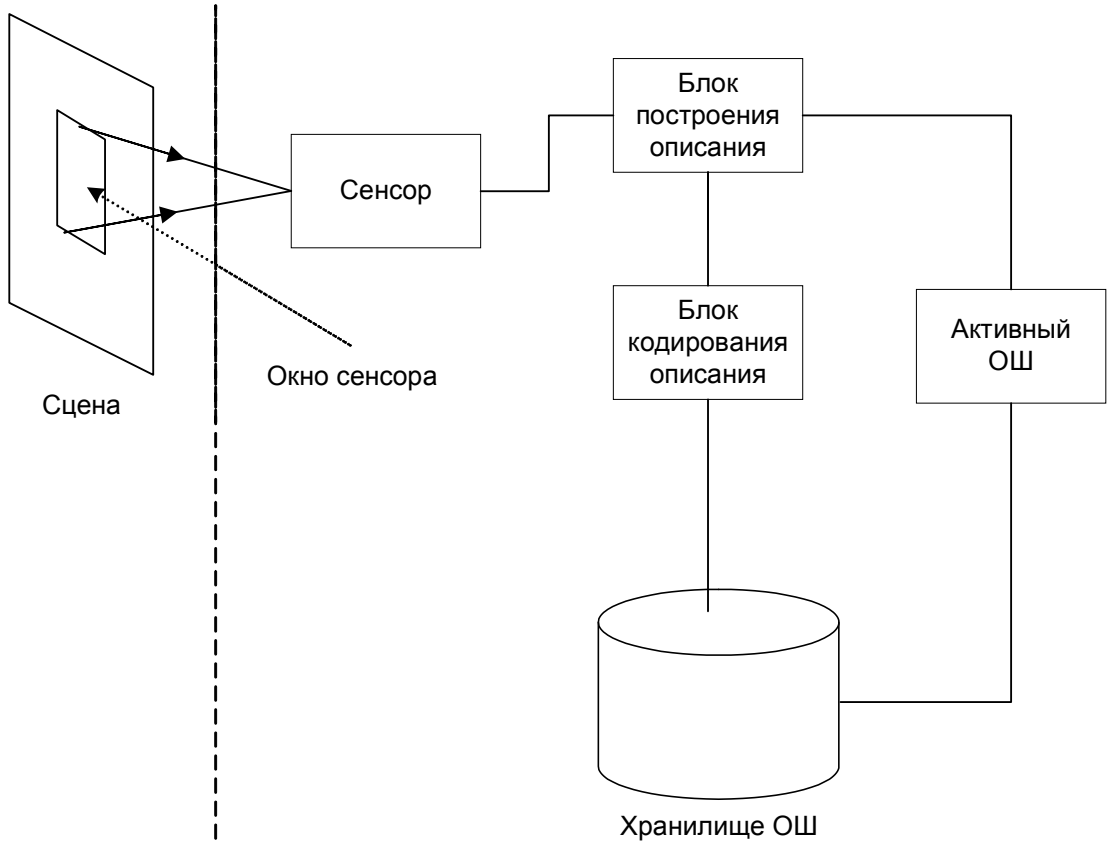

Рисунок 2-2. Общая схема распознавания

В отличие от этапа наполнения базы система в любой момент времени работает не со всем изображением в целом, а лишь с его частью попадающей в окно активного сенсора. На вход подается одно полутоновое изображение (тестовая сцена *Sc*), на котором требуется найти объекты, проиндексированные в базе. Сенсор тем или иным способом двигается по сцене (зависит от вида распознавания), изменяя при этом масштаб своего окна. Считанный сенсором участок *Im* изображения передается в **Блок построения описания**, где проводится его обработка и выделение особых точек $\{Sp_i(Im)\}$. Полученные особые точки маркируются по принадлежности к кластерам $\{KL_i\}$ полученным на этапе построения моделей.

Маркированные ОТ передаются в **Блок кодирования описания,** описываются своими конфигурациями, так же как и при наполнении базы.

Таким образом, строится описание участка *Im* сцены *Sc.* Само распознавание происходит при помощи запросов к **Хранилищу** для поиска особых точек, похожих по своей конфигурации на точки участка *Im*. При этом вычисляется рейтинг проекций объектов, чьи описания особыми точками находятся в **Хранилище**. Чем больше ОТ проекции объекта похоже на ОТ *Im,* тем выше рейтинг. На первом этапе распознавания (грубом) выбирается *Активная объектная модель*, как модель с наивысшим рейтингом,

превосходящим некоторый порог. На втором этапе происходит ее подтверждение – при тонком распознавании проверяется выше ли рейтинг областей активной модели, соответствующих реперным точкам, некоторого порога. Подтвержденная модель считается найденной на сцене. Теперь рассмотрим эти этапы более подробно.

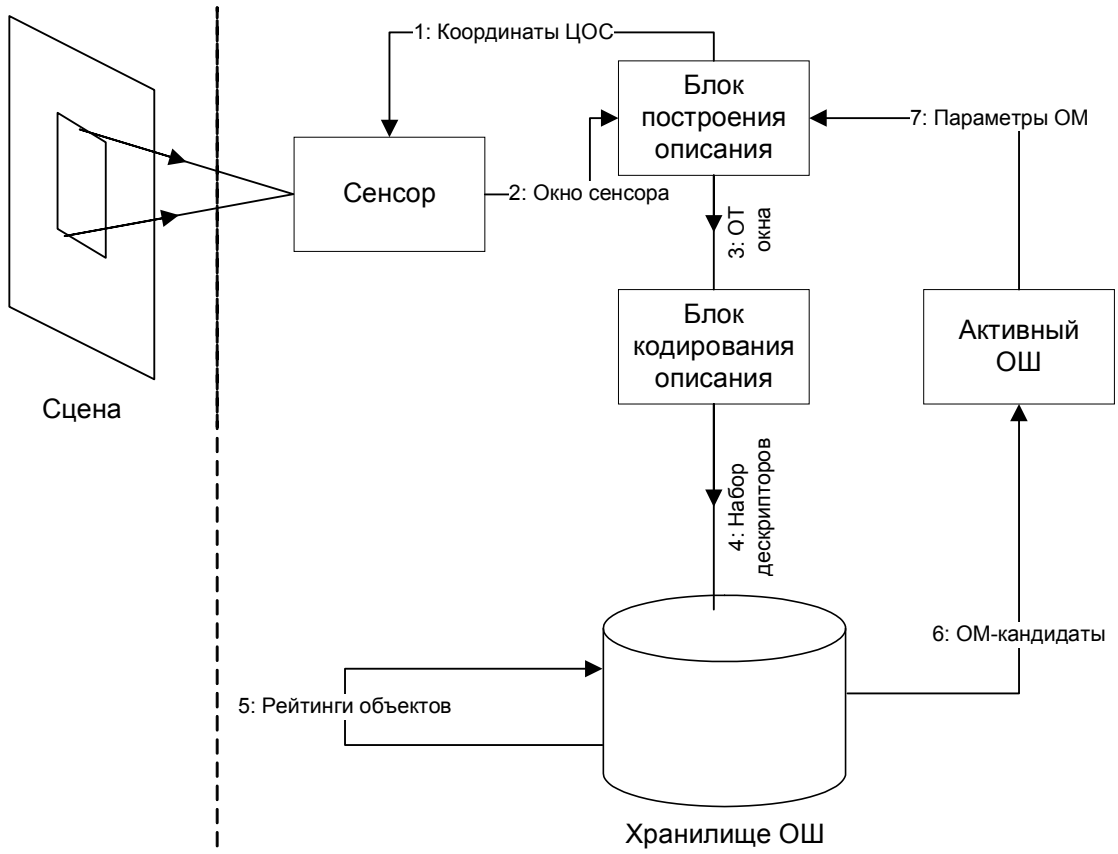

Рисунок 2-2.Выбор активной модели.

Подробная схема выбора активной модели выглядит следующим образом:

Блок построения описания задает первоначальные координаты центра (x,y) окна сенсора $WAS(x,y,s)$ и уровень масштаба изображения s. Сначала s выбирается максимальным, так чтобы все изображение попало в окно.

Сенсор делает «снимок» окна с заданными координатами и передает в Блок построения описания, где на этом участке изображения выделяются и маркируются особые точки.

Набор маркированных ОТ окна сенсора передается в Блок кодирования описания, где строится набор его дескрипторов (векторов конфигураций ОТ), т.е. представление набора в виде символьных строк.

Набор дескрипторов поступает в хранилище ОМ и каждая модель в хранилище сравнивает полученные дескрипторы с собственными, хранимыми внутри модели.

При этом вычисляется и накапливается рейтинг модели $P(Im)$ – показатель, характеризующий количество совпавших дескрипторов, а тем самым, возможность присутствия на сцене объекта *Im*.

Модель, набравший максимальный рейтинг становится активной моделью - кандидатом. Остальные модели становятся моделями кандидатами (неактивными), только если их рейтинг превысит порог активации.

Активная модель передает свои параметры в Блок построения описания, для дальнейшего подтверждения.

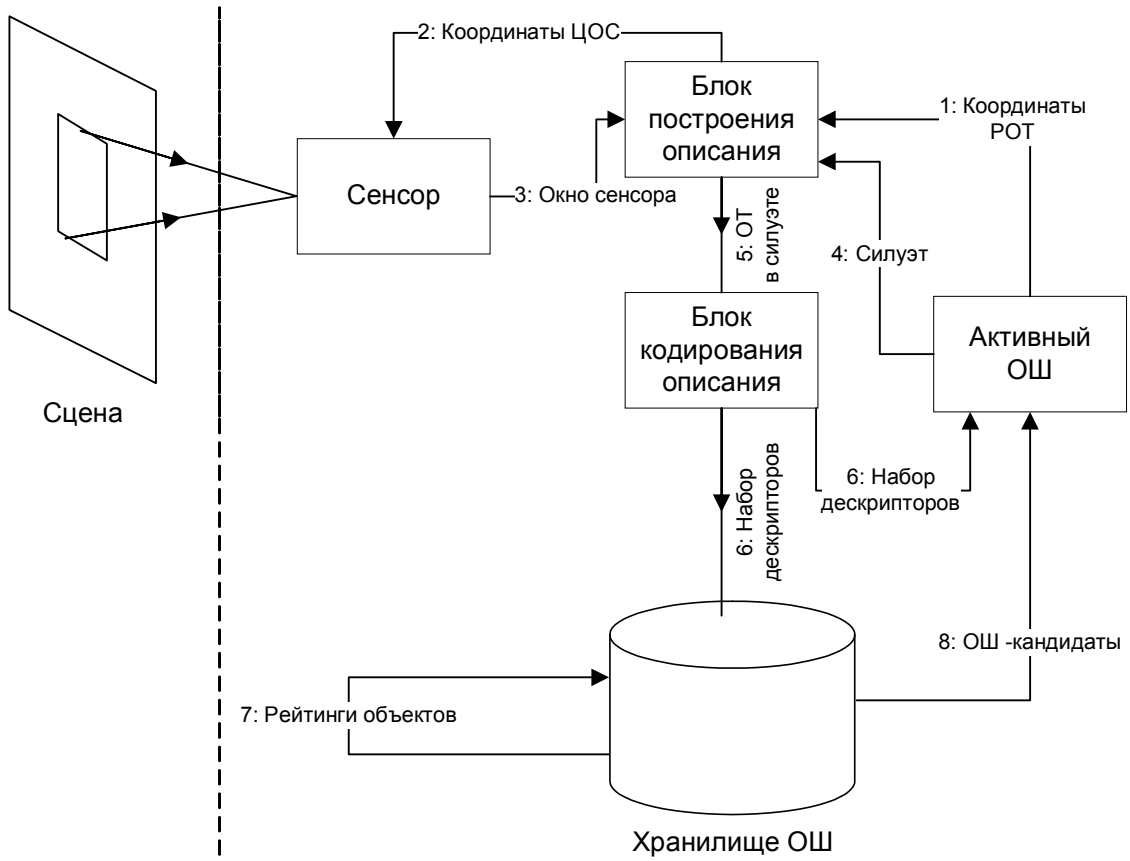

Рисунок 2-3.Подтверждение активной модели.

Второй этап процесса распознавания - подтверждение активной модели-кандидата:

Активная ОМ *P(O)* передает координаты своих реперных особых точек $\{RSP_i(P)\}$ в Блок построения описания для установки в них *WAS*.

Блок построения описания задает координаты *WAS* и уровень пирамиды изображения.

Сенсор передает полученное окно в Блок построения описания

Активная модель передает силуэт объекта в Блок построения описания

Блок построения описания формирует набор особых точек изображения с учетом силуэта объекта, т.е. отсекая ОТ, соответствующие фону. Полученный набор поступает в Блок кодирования описания

Блок кодирования описания строит дескрипторы и передает их в Хранилище ОШ и в Активную модель *P(O)* для вычисления рейтингов.

В Хранилище ОМ каждая модель отдельно вычисляет свой рейтинг. Если после прохождения окном сенсора всех $\{RSP_i(P)\}$ рейтинг активной модели не превышает порога подтверждения, то уровень пирамиды изображения понижается и происходит переход в шаг 1) для этого уровня. Если порог активации превышается в какой-то момент, то объект *O* считается найденным и подтверждение происходит для следующей по рейтингу модели-кандидата. Иначе считается что в данной позиции на изображении объекта *O* нет и подтверждение происходит для следующей по рейтингу модели-кандидата.

Возможно пополнение списка моделей кандидатов, если еще какие-то модели набрали рейтинг, больше порога активации. При отсутствии моделей кандидатов производится поиск активного кандидата на более низком уровне пирамиды, чем в прошлый раз.

Итак, мы рассмотрели основные виды работы РИАС системы, определили ее базовую модульную структуру и общий алгоритм ее работы. Были показаны основные отличительные особенности РИАС системы – ограничение области видимости системы при помощи окна активного сенсора, двухуровневая система распознавания, быстро находящая точки интереса на огрубленных изображениях и проверяющая их на ограниченных силуэтами объектов участках. Первая особенность позволяет сделать распознавание более интеллектуальным – каждая модель подтверждается по-своему без каких либо усилий со стороны человека оператора, что улучшает его точность. Вторая – позволяет сохранить вычислительные ресурсы за счет необходимости проверки меньшего числа особых точек на первом «грубом» этапе и подробного исследования лишь отдельных участков изображения ограниченных силуэтами. Использование силуэтов так же повышает качество разделения объект\фон, т.к. при подробной верификации на втором этапе, вносящие ошибки, особые точки фона не используются.

Теоретические оценки.

Построим теоретические оценки сложности предлагаемого алгоритма.

Лемма 1. На прямоугольном изображении размеров $M \times N$, не может быть выделено более $\frac{2M*N}{R^2}$ особых точек, если по определению ОТ в ее R окрестности не может быть других ОТ, в независимости от алгоритма выделения.

Доказательство. Покроем прямоугольник квадратами размеров $\frac{R}{\sqrt{2}} \times \frac{R}{\sqrt{2}}$. Их будет не более $\frac{\sqrt{2}M}{R} * \frac{\sqrt{2}N}{R} = \frac{2M*N}{R^2}$ штук. Каждый такой квадрат имеет диагональ длины R, поэтому по определению ОТ, в нем может располагаться не более одной особой точки (при любом положении точки внутри квадрата, он целиком попадет в ее R окрестность). Таким образом, число ОТ будет не превышать число таких квадратов, то есть $\frac{2M*N}{R^2}$. ■

Следствие 1. При уменьшении масштаба изображения в 2 раза, максимальное число ОТ, которые могут на нем располагаться сократится в 4 раза.

Доказательство. Это утверждение является элементарным следствием Леммы 1. В случае уменьшения масштаба изображения в 2 раза, его линейные размеры тоже уменьшатся в 2 раза ($M \times N \rightarrow \frac{M}{2} \times \frac{N}{2}$) , таким образом, по Лемме 1 вместо $\frac{2M*N}{R^2}$ особых точек, на нем максимально сможет поместиться $\frac{M*N}{2*R^2} = \frac{1}{4} * \frac{2M*N}{R^2}$. ■

Теорема 1. При кодировании особыми точками полутонового изображения размером $M \times N$ используемый объем информации не будет превышать $M * N$ байт при $R \geq 5$.

Доказательство. Запись одной маркированной особой точки потребует 12 байт (по 4 байта на маркер, *x* и *y* координаты). Поскольку особых точек на изображении не более $\frac{2M*N}{R^2}$, то на запись его в виде набора ОТ потребуется не более $12 * \frac{2M*N}{R^2} = \frac{24*M*N}{R^2}$ байта. Отсюда получаем $\frac{24*M*N}{R^2} < M * N \Longrightarrow R^2 > 24 \Longrightarrow R \geq 5$, т.к. оно целое. ■

Лемма 2. При одноуровневом распознавании без уменьшения масштаба, рассматриваемых изображений (классическая схема) потребуется $\frac{4M^2*N^2}{R^4} * K$ операций сравнения. Где K – число изображений в базе, $M \times N$ их размер, R – радиус пустой окрестности ОТ.

Доказательство. Пусть в базе имеется K изображений объектов размером $M \times N$. особых точек на них суммарно будет не более $\frac{2M*N}{R^2} * K$. На исследуемом изображении будет не более чем $\frac{2M*N}{R^2}$ особых точек. Таким образом для поиска объектов будет проведено $\frac{4M^2*N^2}{R^4} * K$ операций сравнения.■

Теорема 2. При предложенном подходе к распознаванию потребуется не более чем $\frac{1}{2^{2(l+1)}} * \frac{4M^2*N^2}{R^4} * K + \frac{\pi P^2*M^2*N^2*S}{R^6*2^{z-3}}$ действий для распознавания объекта на изображении. Где K – число изображений в базе, $M \times N$ их размер, R – радиус пустой окрестности ОТ, 2^l масштаб огрубления на первом шаге алгоритма распознавания, P – радиус исследуемых окрестностей реперных ОТ, $\frac{1}{2^z}$ доля реперных ОТ среди всех особых точек изображения, S - число изображений кандидатов выбираемых в конце первого шага.

Доказательство. Пусть в базе имеется K изображений объектов размером $M \times N$. На первой стадии распознавания работа идет на масштабе уменьшенном в 2^lраз, поэтому особых точек на них суммарно будет не более $\frac{1}{2^{2l}} * \frac{2M*N}{R^2} * K$. Таким образом, для сравнения всех их особых точек с точками исследуемого изображения потребуется произвести $\frac{1}{2^{2(l+1)}} * \frac{4M^2*N^2}{R^4} * K$ сравнений особых точек. На втором шаге сравниваются P окрестности реперных особых точек, которых не более $\frac{1}{2^z}$ от числа ОТ исследуемого изображения. В каждой P окрестности особых точек не более чем $\frac{2\pi P^2}{R^2}$ (вместо прямоугольника $M \times N$ покрываем квадратиками $\frac{R}{\sqrt{2}} \times \frac{R}{\sqrt{2}}$ круг радиуса P). Значит, всего придется исследовать $\frac{2\pi P^2}{R^2} * \frac{M*N}{R^2*2^{z-1}}$ особых точек на исследуемом изображении. На втором этапе выбирается S изображений кандидатов, которые рассматриваются уже в оригинальном масштабе, то есть на них содержится максимум $\frac{2M*N}{R^2} * S$. Значит, придется сделать $\frac{\pi P^2*M^2*N^2*S}{R^6*2^{z-3}}$. Итого получаем $\frac{1}{2^{2(l+1)}} * \frac{4M^2*N^2}{R^4} * K + \frac{\pi P^2*M^2*N^2*S}{R^6*2^{z-3}}$ сравнений. ■

Таким образом, подбором параметров K,l,P,S можно добиться баланса между качеством распознавания и его скоростью (числом сравнений), сделав ее выше чем у классической схемы распознавания, например выбрав $l > \frac{log_2(\frac{\pi P^2*M^2*N^2*S*R^4}{R^6*2^{z-3}*4M^2*N^2*K}}{2} - 1$.

Далее будет показана конкретная реализация распознающей системы реализующей РИАС подход и приведены результаты численных экспериментов по тестированию этой системы.

Реализация РИАС подхода.

I. Блок построения описания

Основная идея алгоритма состоит в разбиении контуров изображения на дуги окружностей и отрезки прямых. В качестве особых точек берутся центры масс

полученных дуг и отрезков. Для проведения маркировки на основе кластеризации каждой особой точке приписывается вектор, состоящий из параметров соответствующих кривых и прилежащих к ним областей изображения.

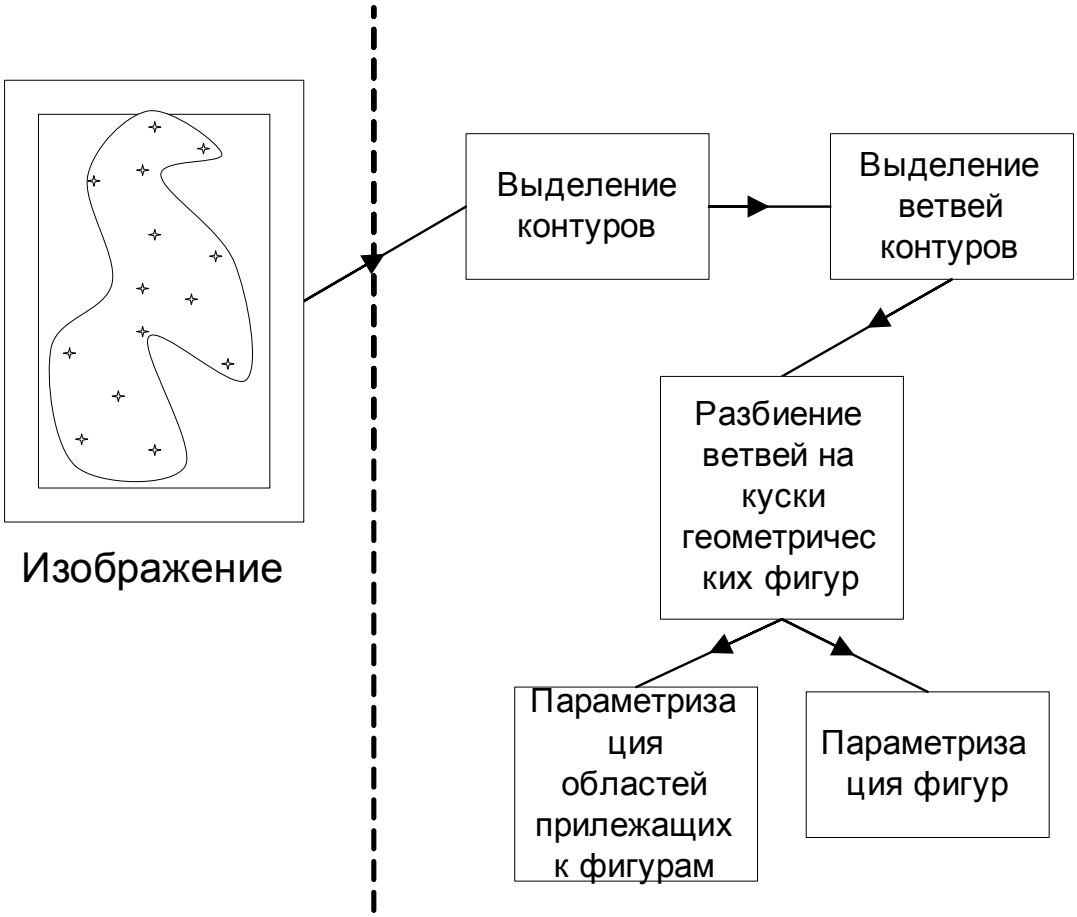

Рисунок 2-5. Схема выделения ОТ.

Более подробная схема выделения и маркировки ОТ:

1) На изображении выделяются контуры методом Собеля.

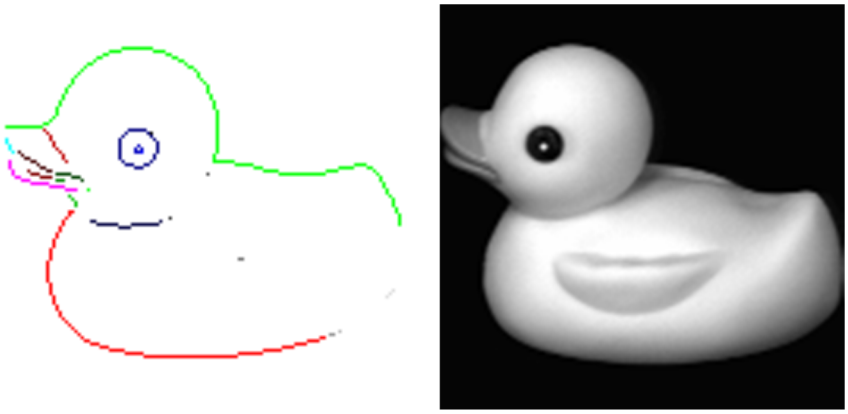

Рисунок 2-6. Ветви контура и исходное изображение.

2) Выделение ветвей. Продвигаясь вдоль контура, алгоритм рекурсивно наращивает ветви. При этом, если в какой-то момент контур обрывается, производится попытка нахождения продолжения контура по касательной (для вычисления касательной,

производится аппроксимация производной ветви разделённой разностью, построенной по последней точке и точке, отстоящей от последней на несколько точек).

Рисунок 2-7. Сращивание ветвей контуров.

Если на определенном отдалении от ветви по касательной других точек контура не находится, то ветвь с этой стороны закрывается. Иначе расширение продолжается.

3) Приближение ветвей геометрическими фигурами. Ветвь последовательно разбивается на все возрастающее число равных сегментов. Каждый сегмент приближается окружностью или прямой. Если отношения длины отрезка O1 соединяющего концы сегмента к длине отрезка O2 соединяющего середину сегмента с серединой O1 больше некоторого порога, то сегмент приближается дугой окружности, иначе – отрезком прямой. Приближение сегмента кривой производится следующим образом. По нескольким точкам сегмента проводится кривая (прямая - по двум точкам, окружность – по трём). Изображение кривой размывается путём применения некоторого фильтра (например гауссовского), и если сумма точек произведения изображения размытой кривой на изображение ветви контура больше некоторого порога, считается, что кривая найдена на сегменте. Если кривая найдена, то ее сегмент удаляется из ветви и алгоритм продолжается.

Рисунок 2-8. Сопоставление ветвям координат.

4)Каждой фигуре сопоставляется один набор из 5 координат: x, y -- координаты середины серединного перпендикуляра к отрезку, соединяющему концы дуги (назовем этот отрезок H), angle - угол между осью OX и отрезком H, len - длина дуги и curv - “кривизна” дуги, равная $\frac{2\pi h}{len}$, где h половина длины отрезка H.

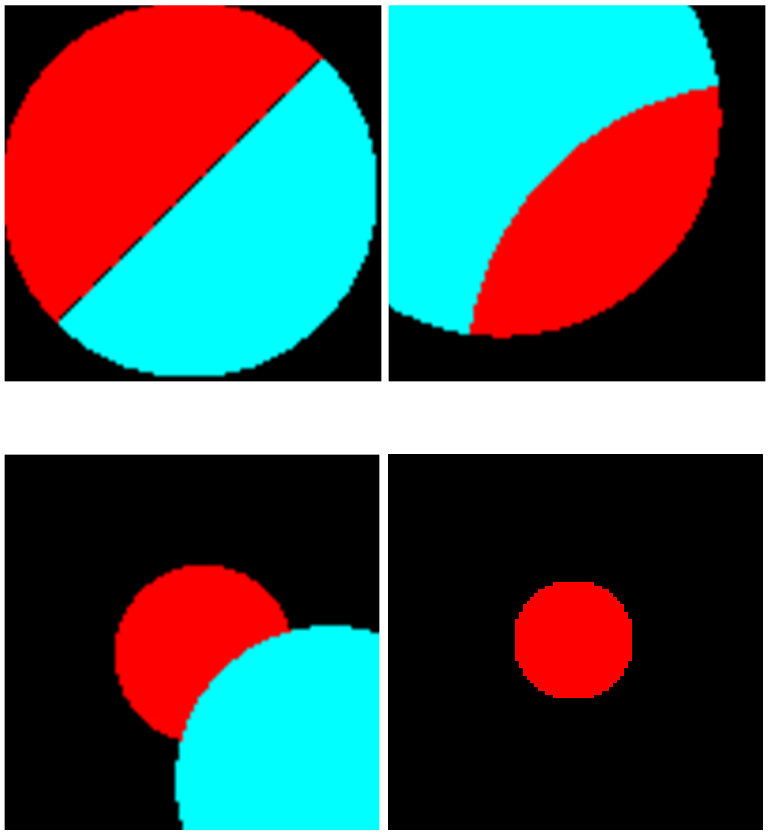

Рисунок 2-9. Сопоставление областей.

5) Каждой фигуре сопоставляется 1-2 области по следующему правилу:

если фигура – целая окружность, ей сопоставляется её внутренность

если фигура – отрезок прямой, ей сопоставляется две половины окружности, построенной на нём, как на диаметре

если кривизна фигуры достаточно близка к 0.5, что соответствует половине окружности, ей сопоставляется одна область – соответствующая половина круга

в остальных случаях, проводится окружность, перпендикулярная дуге, проходящая через её концы, фигуре сопоставляется две области, ограниченные исходной фигурой и окружностью.

Если некоторая область содержит меньше одного пикселя – она не участвует в описании. Если для некоторой фигуры нет областей – фигура исключается из описания.

Описание области (то есть, функция от области, значение которой включается в элемент описания) состоит из цветовых характеристик изображения в области.

Используется следующий способ описания:

- в качестве описания берутся первые 3 центральных средних момента яркости изображения.

Таким образом каждой особой точке ставится в соответствие вектор длины 8.

На этапе построения моделей эти вектора кластеризуются, центры кластеров запоминаются и номера кластеров сопоставляются особым точкам в качестве маркеров. На этапе распознавания по этим векторам особым точкам ставится в соответствие кластер, к центру которого они наиболее близки.

II.Блок кодирования описания

Для построения конфигураций особых точек используется простой алгоритм. Вокруг особой точки, конфигурация ОТ вокруг которой описывается, строится круг некоторого радиуса (область описания). Он разбивается на сегменты по углу (сектора) и по расстоянию относительно центра (кольца). Получается «мишень»:

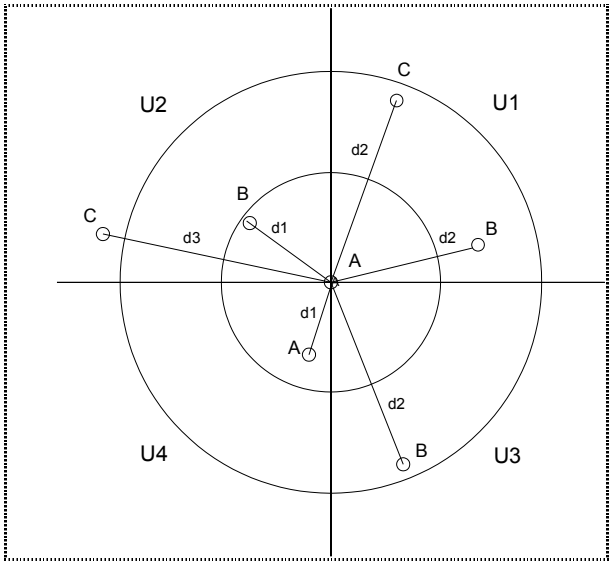

Рисунок 2-10. Здесь 3 уровня дискретизации по расстоянию *(d1, d2, d3)* и 4 уровня дискретизации по углу *(U1, U2, U3, U4),* всего 12 сегментов.

Рисунок 10. «Мишень». Здесь 3 уровня дискретизации по расстоянию (d1, d2, d3) и 4 уровня дискретизации по углу (U1, U2, U3, U4), всего 12 сегментов.

Для каждой пары [сегмент, маркер ОТ] считается количество особых точек с таким маркером попавших в этот сегмент. Таким образом получается матрица, которая «распрямляется» в вектор описания конфигурации.

III.Модуль сравнения ОТ

На вход получаем два множества особых точек (считается что первое относится к образцу). На выходе - рейтинг одного относительно другого.

Если во втором точек меньше чем 1/3 от количества в первом, то возвращается рейтинг 0.

Если в первом меньше точек чем во втором, то рейтинг считается по формуле

$$\sum_{t1\,из\,первого} \max_{t2\,из\,второго}\left(\frac{1}{1+dist(t1,t2)}\right)$$

, где dist(t1,t2) евклидово расстояние между векторами описывающими эти точки, полученными из алгоритма (1).

Иначе – по формуле $\sum_{t2\,из\,второго} \max_{t1\,из\,первого}\left(\frac{1}{1+dist(t1,t2)}\right)$.

IV.Быстрый поиск похожих ОТ в Хранилище шаблонов.

Используемое описание особых точек векторами небольшой длины позволяет пользоваться базой данных для быстрого поиска похожих ОТ. Идея состоит в построении на множестве конфигураций в базе данных пространственного индекса (spatial index). Для заданных чисел r1, r2 (r1 > r2) строятся две функции, сопоставляющие вектору из пространства R^n натуральное (или целое) число:$f_i: R^n \to R; i = 1,2$, другими словами – два разбиения пространства элементов описания на не более чем счётное число областей. Построенные функции должны (для заданной метрики над элементами описания) удовлетворять следующему свойству:

- для любых x1, x2, если $f_1(x1) = f_1(x2)$ или $f_2(x1) = f_2(x2)$, то расстояние между ними меньше r1, иначе – расстояние между ними больше r2.

Алгоритм поиска вектора, достаточно близкого к данному при наличии пространственного индекса, может быть реализован в 2 шага: поиск всех векторов, у которых совпадает хотя бы один из двух индексов с индексом данного элемента, затем отсечение лишних векторов с помощью дополнительного сравнения.

Первый шаг выполняется за время O(1) в среднем от количества элементов (в случае, если индексы реализованы как хэш-таблицы, и в случае, если этот шаг достаточно селективен, то есть, если плотность векторов в пространстве не возрастает при

увеличении количества элементов) или O(ln(n)) (в случае, если индексы реализованы, как B-деревья). Второй шаг выполняется быстро, если первый был достаточно селективным.

Результаты работы

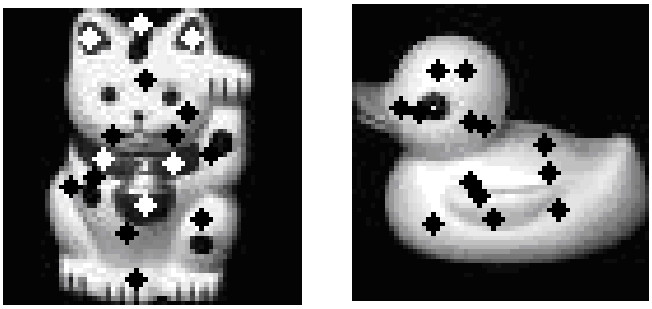

Рисунок 2-11. Особые точки на объекте выделены черными и белыми точками для лучшей видимости.

Рисунок 2-12. Особые точки на объекте выделены черными и белыми точками для лучшей видимости.

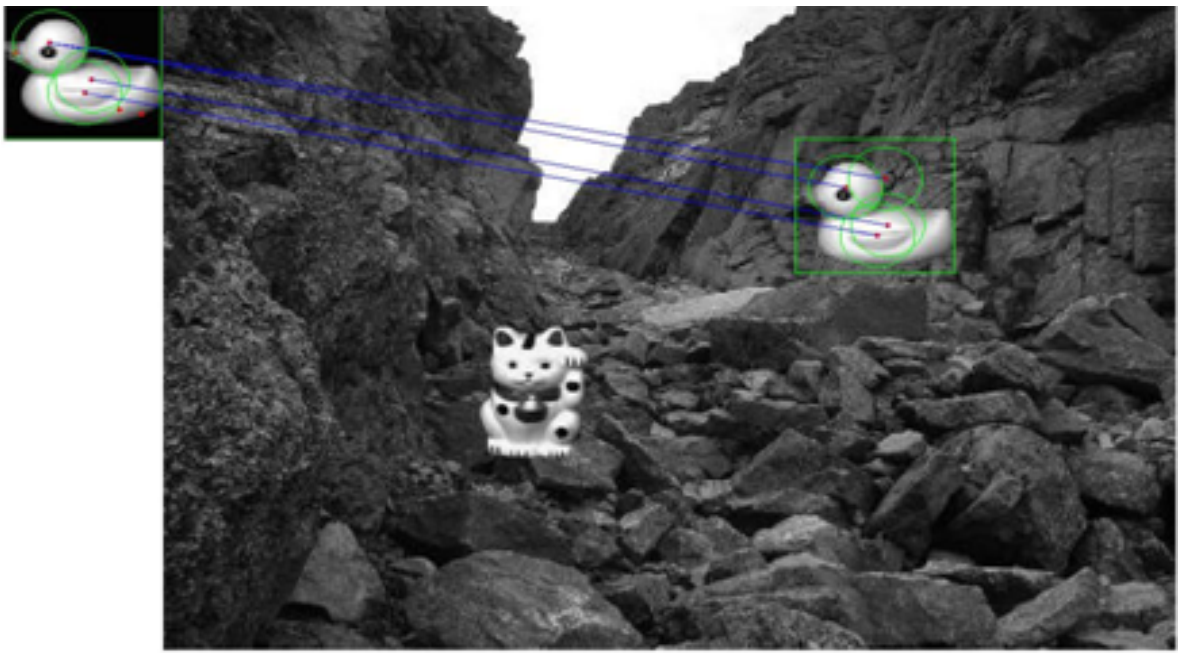

Рисунок 2-13. Результат работы программы, реализующей предложенный метод в системе Matlab. Линиями соединены соответствующие особые точки.

Рисунок 2-14. Результат работы программы, реализующей предложенный метод в системе Matlab. Линиями соединены соответствующие особые точки.

Тестирование показало неплохие результаты распознавания объектов на сценах со сложным фоном. За счет использования низкой детализации изображений на этапе поиска активной модели и использования векторов конфигураций ОТ как индексов в базе данных активная модель ищется достаточно быстро (численные результаты ниже).

Подтверждение тоже происходит быстро т.к. не смотря на увеличенную детализацию

особых точек внутри силуэта не много по сравнению с количеством ОТ в базе. Использование активного сенсора позволяет эффективно решать проблему отделения объекта от фона. При нахождении на сцене хотя бы одной РОТ какого-либо объекта сенсор начинает искать следующие реперные точки этого объекта в тех местах, где они располагаются на объекте. Поэтому при правильном распределении РОТ на объекте (оператор должен располагать их дальше от границ объекта, чтобы на них не влиял фон) для первичного нахождения объекта на сцене достаточно найти хотя бы одну реперную точку на сцене, а при подтверждении фон отсекается при помощи силуэта, проверяемого объекта.

В тестовой БД хранятся описания 100 изображений, 2128 фигур, 4075 областей, 24613 пар фигур.

Сравнение одного изображения с 99 другими – 1 мин. 35 сек.

Сравнение двух изображений алгоритмом быстрого поиска – меньше 1 сек.

Сравнение одного изображения с 99 другими алгоритмом быстрого поиска – меньше 1 сек.

Тестирование производилось на компьютере с ЦПУ AMD Athlon 1700+, объём ОЗУ: 512Мб.

Выводы.

Было дано описание РИАС подхода к распознаванию. Описана возможная реализация компонент процесса. Показаны результаты полученной в результате системы реализованной в системе Matlab. Таким образом были подтверждены сделанные ранее утверждения о преимуществах РИАС подхода к распознаванию. Далее будет продемонстрировано описание процесса построения объектного классификатора использующего РИАС подход и системы автоматизирующей этот процесс.

Глава 3.

Процесс построения классификатора.

Помимо задачи максимизации точности классификатора и скорости его работы, так же остро стоит и задача быстрого построения классификатора для конкретного набора объектов и фонов (предметной области). Одним из основных результатов данной работы является разработка процесса построения РИАС классификатора, позволяющего использовать компоненты классификаторов построенных ранее для других предметных областей, то есть решается задача эффективного приложения этого метода к практическим задачам построения распознающих систем для конкретных предметных областей.

Ключевой особенностью предлагаемого процесса является расширяемая библиотека алгоритмов – набор версий компонент программного модуля классификатора, разработанных для различных предметных областей. Индивидуальные компоненты разрабатываются для конкретной предметной области, только если возможности уже существующих компонент для нее недостаточны – таким образом, достигается высокая эффективность работы процесса построения классификатора за счет массового повторного использования программного кода.

Опишем предлагаемый процесс построения и верификации объектного классификатора использующего РИАС схему. Предполагается, что в подразделение, реализующее данный процесс, обращается Заказчик, желающий получить программный модуль - классификатор, (возможно встраиваемый – dll библиотека, возможно исполняемый – exe приложение с графическим интерфейсом) распознающий объекты его предметной области. Особенностью предлагаемого процесса является построение классификатора из готовых компонент (возможно с добавлением некоторых созданных специально для этого проекта), а не создается полностью с нуля, то есть используется мощный framework для разработки РИАС классификаторов.

Краткое описание этапов.

В процессе ПиВОК можно выделить три основных этапа:

Подготовка и первичный анализ: На основании первоначальных данных *Заказчика* определяется набор изображений объектов, фонов и тестовых изображений моделируемой предметной области (тестовая выборка) и критерии оценки качества распознавания (например, допустимый процент ошибок и время обработки одного изображения). Производится первоначальная разметка силуэтов объектов, высчитывается

его ориентация, удаленность, они размечаются особыми точками и индексируются в БД. Так же индексируются фоны и тестовые изображения. Создается большое число семейств конфигураций (семейством называется множество с одинаковым набором версии модулей подсистем, но разными наборами их параметров), среди которых после массового тестирования на распознавание выбираются отдельные конфигурации или семейства, показавшие наилучшие результаты в этом тесте, называемые конфигурациями-кандидатами.

Тестирование и корректировка: Проводится «тонкое» тестирование полученных конфигураций кандидатов на тестовых изображениях. Составляются отчеты и ведутся логи. Производится корректировка параметров алгоритмов конфигурации, если не удается достичь приемлемых по критериям качества результатов, изменяются версии алгоритмов или принимается решение о необходимости разработи новых алгоритмов на исследовательском этапе. Все эти решения отражаются в итоговом отчете, который передается заказчику на рассмотрение.

Исследование: Если существующих в системе алгоритмов не хватает для качественного распознавания объектов данной предметной области, разрабатываются новые алгоритмы, эффективно работающие с ними. Новые алгоритмы должны иметь такие же интерфейсы, как и старые.

Подробное описание процесса мы будем вести по типу Rational Unified Process – введем основные роли, задачи, артефакты, и опишем их взаимосвязи.

Роли.

Библиотекарь – Отвечает за работу с базой данных объектов. Вносит в нее проекции объектов, выделяет на них силуэты объектов, определяет их ориентацию относительно объекта, размечает их особыми точками по запросу Архитектора.

Архитектор – Отвечает за создание работоспособной конфигурации предметной области, на основании которой строит окончательный модуль распознавания, который передается Заказчику как результат работы. При необходимости принимает решение об исследовательском этапе или о полном закрытии проекта, если создание классификатора потребует слишком обширных дополнительных исследований, которые заказчик не согласен оплачивать.

Тестировщик – Отвечает за тестирование отобранных Архитектором конфигураций кандидатов, предоставляет ему отчеты о проведенных тестах и первичный анализ полученных в их процессе результатов.

Исследователь – Отвечает за создание новых версий модулей системы, заточенных под конкретную предметную область, в случае недостаточности уже имеющихся версий.

Артефакты.

Описание предметной области: Текстовый документ, в свободной форме описывающий какие объекты на каких фонах должна распознавать система. Может дополняться набором изображений, на которых заказчик обязательно хотел бы видеть хорошие результаты работы.

Критерии качества распознавания: Числовые параметры, определяющие, при каких условиях Заказчик будет считать модель предметной области удачной.

Тестовая выборка: Набор изображений типичных объектов и фонов предметной области, а так же сцен с этими объектами и фонами, на которых будет проводиться настройка и тестирование системы. Может быть частично или полностью получена от заказчика, или быть получена по Описанию из иных источников (подбор изображений на определенную тему в интернете, например).

Контуры объектов: Набор битовых масок контуров объектов с изображений из *Тестовой выборки* хранятся в базе с привязкой к породившим их проекциям объектов.

Объектная разметка тестовых сцен: набор битовых масок (0 – фон, 1 - объект) соответствующих тестовым сценам. Получается автоматически для сцен, созданных при помощи системы, и полуавтоматически для сцен, полученных от заказчика, при выделении контуров объектов.

База алгоритмов (модулей): Набор алгоритмов (версий), из которых можно строить схему распознавания. Вообще является заранее данным, но в случаях, когда имеющейся набор является недостаточным для нужд Заказчика, он может быть изменен во время первой фазы

Математиком — добавлены новые модули, созданные под конкретную предметную область и критерии качества. Для каждой компоненты в общей модели распознавания этот набор свой.

Конфигурация предметной области: Набор версий модулей системы и их параметров.

Конфигурации кандидаты: набор конфигураций предметной области, отобранных для тонкого тестирования.

Набор конфигураций кандидатов: Конфигурации показавшие наилучшие результаты в распознавании типа объект\фон.

База данных элементов описания: База, содержащая элементы описания изображений тестовой выборки. Содержит в себе силуэты тестовых объектов, их пространственные конфигурации, наборы особых точек и описания их конфигураций для объектов и фонов, для разных масштабов и проекций.

Отчет о первичном тестировании: Для каждой КПО содержит результаты разделения объект\фон (в виде битовой маски, где 0 соответствует фон, а 1 объекты) для каждой тестовой сцены из *Тестовой выборки* и процент ошибок для каждой сцены(получается при помощи сравнения результатов разделения с *Объектной разметкой тестовых сцен*) и для КПО в целом.

Логи: Записи, автоматически создаваемые системой во время распознавания. Содержат информацию о внутреннем состоянии системы в процессе работы. По ним можно восстановить почему система приняла то или иное решение в отношении тестового объекта.

Отчет о тонком тестировании: Для каждой *конфигурации кандидата* содержит данные о работе отдельных модулей системы (их логи), результаты распознавания для каждой тестовой сцены(какие объекты, в каких проекциях и где найдены) процент ошибок конфигурации на каждой сцене и в среднем по *тестовой выборке*.

ТЗ на разработку модуля: содержит требования к функциональности новой версии одного из модулей системы, результаты работы его предшественников, анализ их ошибок. На его основе математик пытается создать новую версию одного из модулей системы, более соответствующую предметной области проекта.

Окончательный модуль распознавания.

Конечный результат работы, который передается заказчику. Представляет из себя библиотеку и файл базы, так же может дополняться исполняемым файлом, содержащим графический интерфейс.

Задачи.

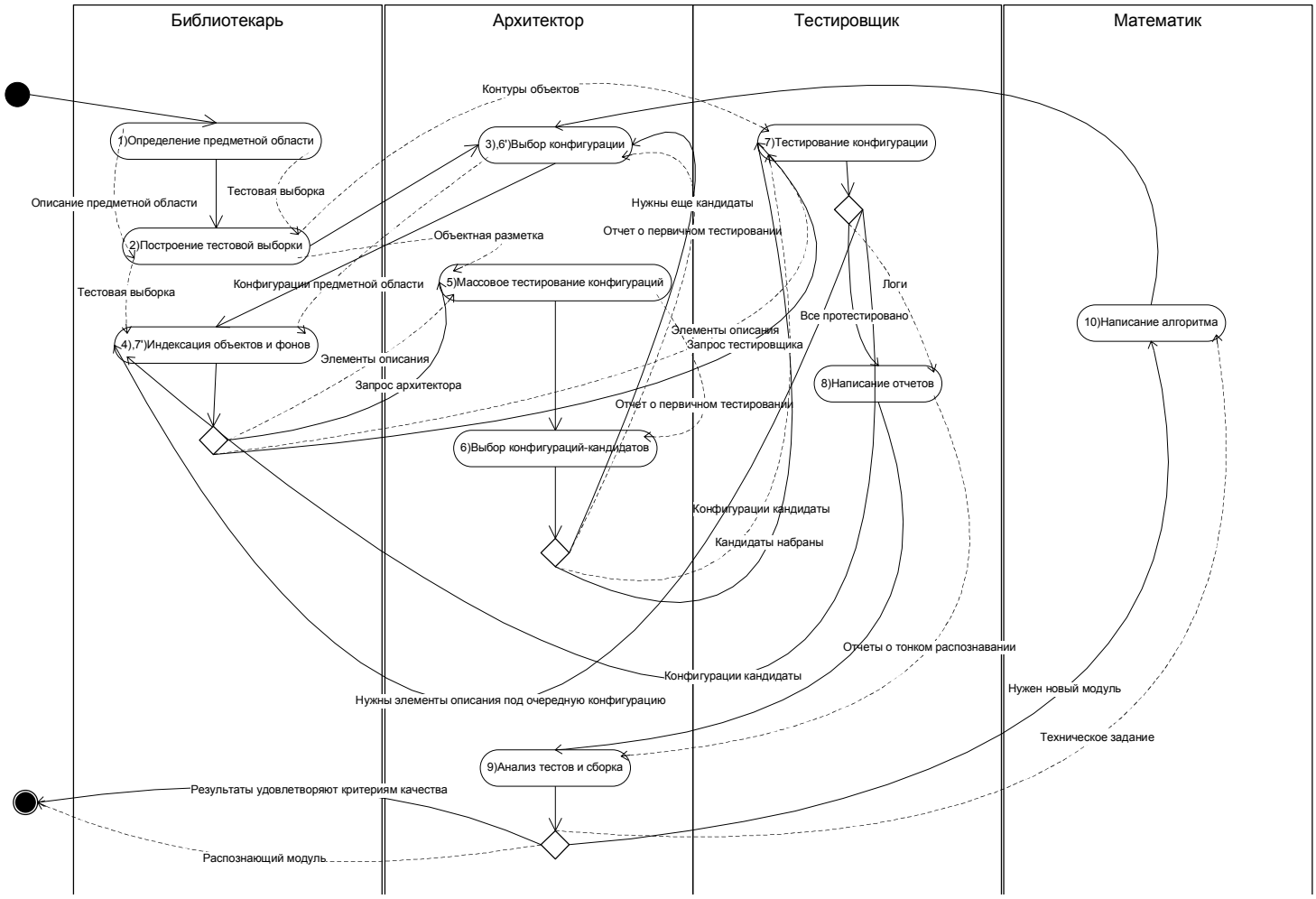

Рисунок 3-1. Диаграмма активности предлагаемого процесса в нотации UML 2.0. Сплошными стрелками обозначается передача управления, пунктирными – объектов.

Библиотекарь

1)Определение предметной области. Согласовывает с заказчиком предметную область, критерии качества модели.

Входные артефакты: нет

Выходные артефакты: Описание предметной области, Критерии качества распознавания,Тестовая выборка

1)Заказчик объясняет в общих словах предметную область – характерные объекты и фоны(*Описание предметной области*).

2)Заказчик предоставляет библиотекарю изображения характерных объектов и фонов. (опционально) (*Тестовая выборка*)

3)Заказчик предоставляет библиотекарю набор числовых параметров которым должна удовлетворять система (скорость работы, число одновременных запросов в секунду, процент распознавания и т.д.) (*Критерии качества распознавания*)

4)Библиотекарь оценивает полученные данные с точки зрения возможностей системы. При необходимости модифицирует *Описание предметной области* и *Критерии качества распознавания* и возвращается к 1 пункту для согласования изменений с Заказчиком.

2)Построение тестовой выборки. Строит набор тестовых изображений, на основании соответствующих документов.

Входные артефакты: Описание предметной области, Тестовая выборка

Выходные артефакты: Тестовая выборка, Контуры объектов, Объектная разметка тестовых сцен

1)Поиск в интернете или иных доступных источниках изображений объектов и фонов предметной области (Если ранее они не были получены от заказчика или их недостаточно)

2)Выделение контуров на изображениях объектов и сохранение их как двоичных масок в базе, сохранение объектной разметки этих изображений.(*Контуры объектов, Объектная разметка тестовых сцен*)

3)Сохранение вырезанных объектов на черном фоне в базе, как эталонных проекций объектов, и добавление фонов с вырезанными объектами в базу фонов, сохранение исходных изображений заказчика как тестовых сцен.

4)Для каждого объекта и его контура автоматически достраиваются необходимые масштабы (их количество параметр системы в целом)

5)Если есть трехмерное изображение объекта, то просто берутся его необходимые проекции методом рендеринга с определенной позиции, при этом эта позиция запоминается, иначе 5’.

5’)По известным проекциям восстанавливается, используя сторонние компоненты, трехмерная модель объекта.

6’) Если полученная модель не нравится, то выбираются другие базовые проекции и опять пункт 5’ иначе далее пункт 5

6) При помощи фотошопа или другой подобной программы объединяются изображения объектов в разных проекциях и масштабах и изображения фонов, таким

образом увеличивается число тестовых сцен, автоматически в базу добавляется объектная разметка этих сцен.(*Объектная разметка тестовых сцен*)

3)Индексация объектов и фонов. По требованию Архитектора для конкретной конфигурации системы выделяет, описывает конфигурации особых точек и индексирует их в базе.

Входные артефакты: Конфигурация предметной области, Тестовая выборка

Выходные артефакты: База данных элементов описания

1)Получение набора модулей и их конфигурационных файлов (конфигурации системы) от Архитектора.

2)Запуск модуля выделения и маркирования ОТ для всех эталонных объектов и фонов, сбор полученных данных в БД.

3)Запуск модуля описания конфигураций ОТ для всех ОТ хранящихся в базе для данной конфигурации системы. Занесение полученной информации в базу (*База данных элементов описания*)

4)Передача полученной базы описанных ОТ архитектору

Архитектор

1)Выбор конфигурации. Выбирает набор версий алгоритмов и их параметры.

Входные артефакты: База алгоритмов, Отчет о первичном тестировании(опционально)

Выходные артефакты: множество Конфигураций предметной области

1)Выбор версий алгоритмов подсистем.

2)Выбор возможных границ для каждого параметра подсистем.

3)Запуск генерации множества конфигураций с одинаковым набором версий подсистем, но разными наборами параметров, попадающими в заданные на прошлом этапе границы.

4)Сохранение полученных Конфигураций предметной области.

2)Массовое тестирование конфигураций. Первичное тестирование большого числа конфигураций на распознавание объект\фон.

Входные артефакты: Объектная разметка тестовых сцен, База данных элементов описания

Выходные артефакты: Отчет о первичном тестировании

1)Выбор набора конфигураций для тестирования.

2)Выбор набора тестовых изображений, объектов и фонов (тестовой предметной области).

3)Индексация тестовой предметной области для каждой конфигурации.

4)Запуск процедуры распознавания объект\фон.

5)Сохранение результатов тестирования в *Отчете о первичном тестировании*

3)Выбор конфигураций-кандидатов. Выбирает наиболее перспективные конфигурации для настройки и подробного «тонкого» тестирования.

Входные артефакты: Отчет о первичном тестировании, Конфигурации предметной области

Выходные артефакты: Конфигурации кандидаты

1)Анализ отчета о первичном тестировани.

2)Выбор Конфигураций кандидатов.

4)Анализ тестов и сборка. Анализирует отчеты Тестировщика о тестировании конфигураций кандидатов. Принимает решение о необходимости Исследовательского этапа и обсуждает этот вопрос с Заказчиком.

Входные артефакты: Конфигурации кандидаты, Критерии качества распознавания, Отчет о тонком тестировании

Выходные артефакты: ТЗ на разработку модуля, Окончательный модуль распознавания

1)Анализ *отчетов о тонком тестировании* предоставленных Тестировщиков.

2)Выбор лучшей из конфигураций кандидатов по *Критериям качества распознавания*, сборка ее в *Окончательный модуль распознавания* и передача заказчику вместе с *отчетом о ее тонком тестировании*, если ни одна из конфигураций не подходит под критерии заказчика, то переход к пункуту 3)

3)Сообщение заказчику о необходимости исследовательского этапа, а значит увеличении сроков и бюджета проекта.

4)Привлечение Математика к исследовательскому этапу (написание *ТЗ*) или сворачивание проекта, в зависимости от решения заказчика.

5)Если исследовательский этап успешно проведен, то переход к *Выбору конфигурации*.

Тестировщик.

1)Тестирование конфигурации. Тестирует конфигурации на тестовых сценах. Пытается подобрать параметры алгоритмов для максимального удовлетворения Критериям качества. Логи при этом ведутся автоматически.

Входные артефакты: Конфигурации кандидаты, База элементов описания, Контуры объектов

Выходные артефакты: *Логи*

1)Выбор конфигурации для тестирования

2)Выбор набора тестовых сцен и эталонных изображений объектов и фонов.

3)Запуск процедуры распознавания.

4)Изменение параметров. Если критерии качества удоволетворены, то *Написание отчетов*, иначе 1). Если не удается после большого числа подбора параметров приблизиться к удовлетворению Критериев, то *Написание отчетов*.

2)Написание отчетов. На основании Логов пишет отчет о работе конкретной конфигурации, анализирует работу подсистем. Передает его архитектору и преступает к тестированию следующей конфигурации-кандидата.

Входные артефакты: *Логи*

Выходные артефакты: Отчет о тонком тестировании

1)Анализ процента ошибок модулей.

2)Анализ производительности.

3)Анализ процента ошибок системы в целом.

4)Написание общего отчета и передача его Архитектору. Далее 1) с новой конфигурацией, пока они не кончатся.

Математик.

1)Написание алгоритма. На основании технического задания Архитектора разрабатывает новую версию того или иного алгоритма системы, более подходящего для данной предметной области чем уже имеющиеся в базе алгоритмов системы.

Входные артефакты: ТЗ на разработку модуля, конфигурации кандидаты, Отчет о тонком тестировании, Тестовая выборка

Выходные артефакты: *База алгоритмов*

Описание архитектуры предлагаемого классификатора.

Далее будет дано описание модулей РИАС классификатора и их интерфейсов, но сначала нужно определить каким образом будут использоваться эти модули, поэтому ниже расписываются два основных сценария использования классификатора с описанием основных действий каждого модуля.

Первичное распознавание

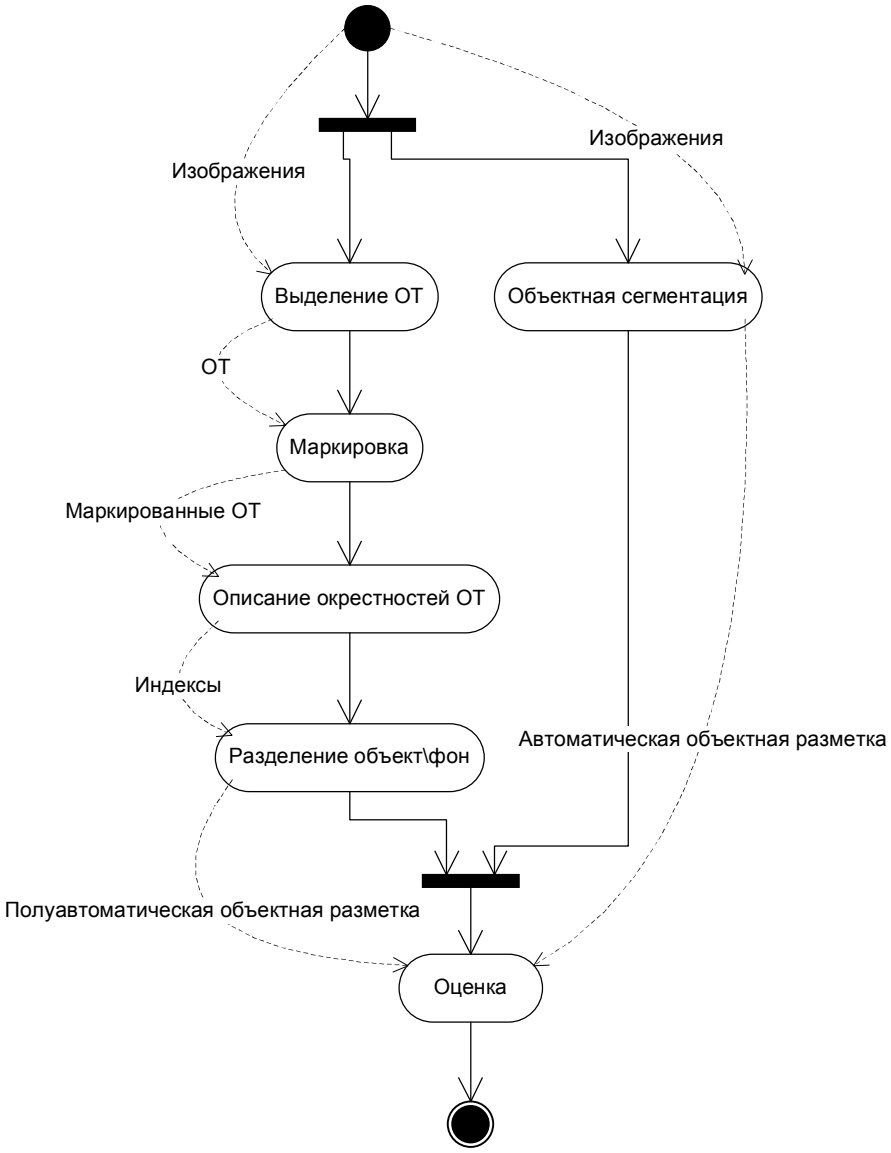

Рисунок 3-2. Диаграмма активности работы классификатора при первичном распознавании в нотации UML 2.0. Сплошными стрелками обозначается передача управления, пунктирными – объектов.

1) Изображения обрабатываются оператором на полуавтоматическом модуле объектной сегментации. Выделяются контуры объектов, по ним модулем

строится объектная сегментация сцены. Силуэты и карта сегментации помещается в хранилище.

2) На изображениях выделяются ОТ
3) После обработки *всех* изображений модулем выделения ОТ, полученные особенности кластеризуются и маркируются номерами своих кластеров. Кластеры разделяются на «объектные» и «фоновые»
4) У маркированных ОТ описываются окрестности, по описаниям строятся индексы
5) По маркерам на изображениях проводится автоматическая объектная сегментация
6) Проводится сравнение автоматической и полуавтоматической объектных сегментаций (процент ошибок, список неверно классифицированных ОТ…)
7) Результаты сравнения и изображения с отмеченными «объектными» и «фоновыми» ОТ подаются на выход.

Тонкое распознавание

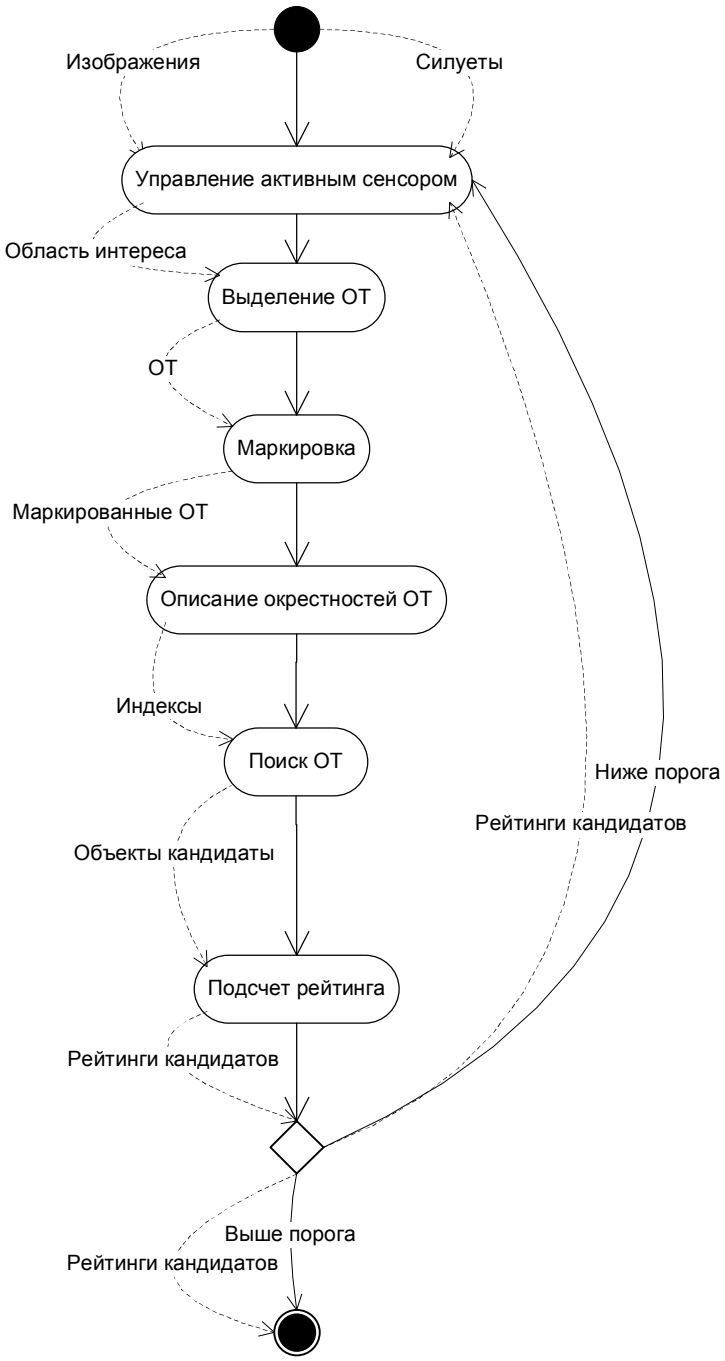

Рисунок 3-3. Диаграмма активности работы классификатора при тонком распознавании в нотации UML 2.0. Сплошными стрелками обозначается передача управления, пунктирными – объектов.

1) Сенсор выбирает область интереса на изображении и масштаб
2) В области интереса выделяются ОТ
3) ОТ маркируются маркерами кластеров находящихся в хранилище
4) Окрестности маркированных ОТ описываются, по ним строятся индексы
5) По индексам в хранилище ищутся похожие ОТ объектов
6) По их «похожести» объектам начисляется рейтинг
7) Если рейтинг какого-то из кандидатов выше порогового значения то все заканчивается, иначе опять передается в модуль позиционирования сенсора

Описание модулей системы.

Далее будут приведены описания модулей классификатора. Строгое описание их интерфейсов будет дано в приложении.

Модуль выделения ОТ. Выделяет на некоторой области изображения особые точки по тому или иному алгоритму. На алгоритм выделения накладываются следующее ограничение – особые точки, получаемые при его помощи, должны удовлетворять следующим условиям – их число на изображении должно быть относительно небольшим (не более нескольких сотен на изображение со сложным фоном и несколькими объектами), их положение на проекции объекта должно несильно изменяться при поворотах этого объекта, изменении масштаба и освещенности.

Входные данные: область изображения, на которой следует выделить ОТ, параметры специфичные для применяемого алгоритма выделения (например, порог для метода Кэнни выделения контуров, как в приведенной выше реализации этого модуля)

Выходные данные: множество координат особых точек

Модуль маркировки ОТ. Строит маркеры особых точек по тем или иным характеристикам прилегающих к ним частей исходного изображения. Сначала по тем или иным характеристикам изображения в окрестности особых точек строятся вектора фиксированной длины, приписанные к соответствующим особым точкам. Никаких ограничений на алгоритм получения этих векторов нет. Затем эти вектора кластеризуются, и номера кластеров, соответствующих векторам описания ОТ, приписываются этим ОТ как маркеры.

Входные данные: область изображения, координаты особых точек на ней, параметры алгоритма построения векторов описания окрестностей ОТ, параметры кластеризации.

Выходные данные: множество координат особых точек с соответствующими им маркерами (набор маркированных ОТ)

Модуль построения индексов ОТ. Строит вектора описывающие окрестности особых точек на основании расположения остальных ОТ в этих окрестностях. Примером может служить алгоритм «мишени» из реализации этого модуля, описанной выше. Особенностью алгоритма этого модуля является то, что он работает только с особыми точками, а не с участками изображения, результирующие вектора конфигурации ОТ должны быть практически инвариантны относительно незначительных изменений координат особых точек (то есть при изменении координат особых точек из окрестности

некоторой видимой ОТ, чей вектор конфигурации строится, на некоторый ε вектор конфигурации не должен измениться больше чем на O(ε) по метрике, которая будет использоваться в дальнейшем при сравнении векторов конфигураций на похожесть).

Получившиеся в результате работы вектора используются как индексы в базе данных хранилища моделей, отсюда название модуля.

Входные данные: множество особых точек для индексирования, параметры алгоритма, типа используемой метрики и радиуса окрестности используемой для построения векторов.

Выходные данные: множество проиндексированных особых точек, то есть особых точек представляющих из себя пару (координаты, вектор индекса).

Модуль полуавтоматической объектной разметки изображения. Помогает оператору выделять контуры объектов на изображении, после выделения автоматически заносит их битовые маски в хранилище. Выделение контуров происходит в полуавтоматическом режиме, как в фотошопе.

Входные данные: изображение.

Выходные данные: контуры объектов и ручная объектная разметка изображения.

Модуль разделения объект\фон. Производит автоматическую объектную сегментацию изображения на основе данных об особых точках объектов и фонов их хранилища. На первом этапе разделяет все особые точки изображения на объектные и фоновые. На втором этапе пытается наложить силуэты объектов на объектные ОТ, таким образом, чтобы внутрь силуэта попало как можно больше особых точек похожих на ОТ соответствующего объекта и как можно меньше непохожих объектных и фоновых точек. Наложенные на изображение силуэты и формируют автоматическую объектную сегментацию изображения.

Входные данные: изображение, особые точки изображения, особые точки объектов и фонов из хранилища, силуэты объектов.

Выходные данные: автоматическая объектная разметка изображения.

Модуль оценки качества объектной сегментации. Проводит сравнение ручной и автоматической объектной разметки и вычисляет меру их похожести. В качестве таковой, например, может выступать сумма количества ошибок первого и второго рода или величина обратная к сумме площадей несовпадающих участков разметок.

Входные данные: ручная объектная разметка изображения, автоматическая объектная разметка изображения.

Выходные данные: степень похожести разметок.

Модуль поиска ОТ. Ищет в хранилище ОТ, с индексами близкими по некоторой метрике к данному. Одна из возможных быстрых реализаций этого модуля была описана выше.

Входные данные: индекс особой точки, для которой мы ищем похожих собратьев, множество индексированных ОТ из хранилища, метрика по которой считается похожесть.

Выходные данные: набор похожих ОТ и их похожесть (расстояние до индекса целевой ОТ по заданной метрике), проекции объектов, которым они принадлежат.

Модуль подсчета рейтинга. Обновляет рейтинг проекций объектов относительно тестовой сцены, на которой производится поиск объектов. На вход получает похожесть ОТ проекции на ОТ, исследуемого в данный момент, участка сцены и на основании этих данных тем или иным способом увеличивает (или уменьшает) рейтинг этих проекций, при этом никаких ограничений на то, что такое рейтинг, у этого модуля нет, единственное его понимание рейтинга должно соответствовать пониманию *модуля отсечения*.

Входные данные: целевая ОТ, множество похожих на нее ОТ и их похожесть(как число), соответствующие им проекции объектов, параметры алгоритма подсчета рейтинга (если есть).

Выходные данные: рейтинги проекций, полученных на вход.

Модуль отсечения. Определяет по рейтингу проекции присутствует ли она на исследуемой сцене или нет. Может принять решении о прекращении распознавания если у нескольких проекций очень высокий рейтинг (например сумма рейтингов по всем проекциям выше определенного порога), в таком случае на выход подаются данные о проекциях с высоким рейтингом – в каких местах на сцене находятся объекты с высоким рейтингом и каков этот рейтинг, или же может принять решение о продолжении распознавания.

Входные данные: проекции объектов с рейтингами, пороги отсечения.

Выходные данные: проекции объектов с их расположением на сцене и рейтингами.

Модуль управления активным сенсором. Управляет положением окна сенсори уровнем масштаба в зависимости от хода распознавания и активной модели – кандидата. Перемещает сенсор по предполагаемым местам расположения реперных особых точек текущей активной модели-кандидате. При необходимости принимает решение о смене активной модели и соответственно дальнейшего хода распознавания, при этом поиск начинает происходить внутри предполагаемого расположения силуэта объекта, соответствующего новой активной модели (например если все РОТ текущей активной модели уже проверены, но рейтинг еще ни одной проекции не достиг порога

распознавания). Так же может принимать решение о переходе а другой уровень масштаба, если на текущем слишком много шума от лишних ОТ (удаление) или наоборот не хватает деталей (приближение).

Входные данные: координаты РОТ текущей модели кандидата, рейтинги проекций объектов, набор проекций кандидатов, которые могут быть активными и их рейтинги

Выходные данные: область изображения для исследования (участок с центром в центре окна сенсора взятый в определенном масштабе)

Отдельно скажем несколько слов о *хранилище*. Это база данных содержащая данные о проиндексированных объектах (проекции объектов, фоны), моделях объектов (особые точки, силуэты), ходе распознавания (данные о рейтингах проекций), объектной разметке изображений, версии алгоритмов модулей системы и списки их параметров. Его структура рассматриваться не будет, т.к. она является специфичной для конкретной реализации классификатора.

Конфигурация классификатора.

Важнейшей особенностью предлагаемого классификатора его версионность – классификатор состоит из набора независимых изолированных модулей с определенным интерфейсом и произвольным алгоритмическим наполнением. Для каждого модуля предлагается хранить несколько версий алгоритмов реализующих его функциональность, таким образом лишь изменением версий алгоритмов можно получать совершенно разные классификаторы, подходящие для разных предметных областей. Каждая версия алгоритмов имеет свой набор параметров, эти параметры могут быть различны для различных версий одного и того же алгоритма, в отличие от интерфейсов которые они должны реализовывать. Такая модульность позволяет легко получать новые виды классификаторов просто добавляя свои версии алгоритмов его модулей, при этом нет необходимости переписывать весь код классификатора заново, достаточно изменить только, не устраивающие по каким то причинам, модули и взять остальные из базового набора. Построение классификатора при использовании предложенного процесса происходит как подбор готовых версий модулей и их параметров и, лишь при необходимости, создания новых версий. Такой подход значительно увеличивает скорость разработки новых классификаторов за счет повторного использования надежного кода уже готовых версий алгоритмов вместо написания и тестирования всего кода заново. Таким образом, для однозначного воспроизведения результатов некоторого

классификатора, построенного по предлагаемой схеме, достаточно знать лишь версии составлявших его алгоритмов и их параметры – конфигурацию классификатора. Такой подход облегчает исправление ошибок по запросу клиентов купивших классификатор – все версии алгоритмов когда либо использовавшиеся для построения классификаторов хранятся, поэтому для воспроизведения ошибки достаточно знать лишь конфигурацию клиентского классификатора.

Выводы.

Было дано описание процесса построения РИАС объектного классификатора, произведено его разбиение на задачи и распределение по ролям. Так же было дано описание архитектуры классификатора и описание его модулей. В результате было показано преимущество предложенного подхода к построению классификатора ввиду уменьшения времени на построение классификатора и упрощения воспроизведения результатов распознавания с его помощью.

Заключение.

В дипломной работе произведен обзор подходов к распознаванию, использующих модели. На его основе проведена классификация подхода к распознаванию использующего активный сенсор, в разработке и апробации которого, автор принимал активное участие.

Произведена унификация РИАС подхода, уточнена используемая терминология и произведено описание этапов распознавания с ее использованием.

Описана конкретная реализация подхода и проведены численные эксперименты, проверяющие качество распознавания с ее помощью.

Разработан процесс построения РИАС классификатора, произведено его разбиение на задачи и распределение по ролям. Построено UML описание информационной системы автоматизирующий этот процесс.

Описана архитектура РИАС классификатора и произведено его разбиение на модули. Показаны преимущества модульного подхода к построению классификатора в плане скорости разработки и легкости воспроизводимости результатов.

Возможно следующее развитие результатов этой работы.

- Разработка новых версий алгоритмов модулей классификатора (у некоторых модулей еще нет ни одной реализованной версии). Их тестирование и применение к конкретной достаточно узкой предметной области.
- Реализация системы автоматизирующей процесс построения классификатора.
- Улучшение РИАС подхода к распознаванию за счет использования других методов сравнения изображений, например использовать Support Vector Machine на векторах конфигураций особых точек.
- Сменить задачу распознавания объектов ,например, задачу построения панорамы по отдельным снимкам с сохранением РИАС подхода. Для этого, например, можно попробовать адаптировать SIFT особые точки к предлагаемому подходу.

Результаты работы опубликованы:

1) Mednikov D.I., Milovidov A., Sergunin S.Yu., Kumskov M.I. “Object identification on grayscale images with complicated background using the active sensor model”, РОАИ – 2008, труды конференции.

2)D. I. Mednikov, A. Milovidov, S. Yu. Sergunin, M. I. Kumskov “Identification of stable description elements using an active sensor”, Pattern Recognition and Image Analysis, Volume 19, Number 1 / Март 2009 г.

Список литературы.

1) Thomas M. Breuel, “View Based Recognition”, IAPR Workshop on Machine Vision Applications, 1992

2) Schweitzer, H., Kulkarni, S.R., “Computational Limitations Of Model-Based Recognition”,
Intl. J. Intelligent Systems(13), 1998, pp. 431-443

3) Mian, A.S.; Bennamoun, M.; Owens, R.A.” A novel algorithm for automatic 3D model-based free-form object recognition”, IEEE international conference on systems, man & cybernetics, 2004

4) Azad, P.; Asfour, T.; Dillmann, R., “Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization”, Intelligent Robots and Systems, 2006

5) Fang Fang; Xudong Ma; Xianzhong Dai “Mobile Robot Localization Based on Topological Model with Tiered Features” Mechatronics and Automation, Volume 1, Issue , 2006 pp. 520 - 525

6) Crandall, D. Felzenszwalb, P. Huttenlocher, D. “ Spatial priors for part-based recognition using statistical models” Computer Vision and Pattern Recognition (CVPR), 2005, Volume: 1, pp. 10- 17

7) Moreno, P. , Bernardino, A. , Santos-Victor, J., “Model Based Selection and Classification of Local Features for Recognition Using Gabor Filters”, ICIAR 06, pp. 181-192

8) J. Huang, B. Heisele, and Blanz V. “Component-based face recognition with 3D morphable models”, 4th Conference on Audio- and Video-Based Biometric Person Authetication, 2003

9) Xiaoguang Lu, Dirk Colbry, Anil K. Jain, "Three-Dimensional Model Based Face Recognition," *icpr*, pp. 362-366, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 1, 2004

10) Xiaoguang Lu, Rein-Lien Hsu, Anil K. Jain, Behrooz Kamgar-Parsi, Behzad Kamgar-Parsi “Face Recognition with 3D Model-Based Synthesis”, Springer, Berlin / Heidelberg 2004

11) Hans van Kuilenburg, Marco Wiering, and Marten den Uyl “A Model Based Method for Automatic Facial Expression Recognition”, Machine Learning: ECML, 2005

12) X. Liu, T. Chen.” Video-based face recognition using adaptive hidden markov models”. Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003.

13) L. Prevost, C. Michel-Sendis, A. Moises, L. Oudot, M. Milgram, "Combining model-based and discriminative classifiers : application to handwritten character recognition," p. 31, Seventh International Conference on Document Analysis and Recognition (ICDAR'03), 2003, Volume 1, p. 31

14) Jie Zou, Nagy G. Evaluation of model-based interactive flower recognition 17th International Conference on Pattern Recognition (ICPR'04) Volume: 2, pp 311- 314

15) Canton, Cristian “Human Model and Motion Based 3D Action Recognition in Multiple View Scenarios”, XIV European Signal Processing Conference, 2006

16) M.G. Suraj, D.S. Guru “Appearance based Recognition Methodology for Recognising Fingerspelling Alphabets” International Joint Conference on Artificial Intelligence (IJCAI), 2007

17) Aravind Sundaresan; Amit RoyChowdhury; Rama Chellappa, “A hidden Markov model based framework for recognition of humans from gait sequences”, ICIP, 2003

18) Justin Muncaster, Yunqian Ma “Hierarchical Model-Based Activity Recognition With Automatic Low-Level State Discovery” Journal of Multimedia, 2007, Volume 2, Issue 5, pp. 66-76

19) Lowe, D. G., "Object recognition from local scale-invariant features," Proceedings of International Conference on Computer Vision, 1999, pp. 1150-1157

20) Марр Д., «Зрение. Информационный подход к изучению представления и обработки зрительных образов», Радио и связь, 1987

21) С. Сергунин, А. Миловидов, В. Лозинский, Д. Кравченко «Примеры построения алфавитов описания окна активного сенсора в РОАС системе.» // Доклады 12-й Всероссийской конференции «Математические методы распознавания образов» (ММРО-12-2005) 20-26 ноября 2005, г. Звенигород (Московская обл.) стр. 447 – 450.

22) С.Ю. Сергунин, М.И. Кумсков «Свойства модели объекта в системе распознавания с активным сенсором». // Доклады 12-й Всероссийской конференции «Математические методы распознавания образов» (ММРО-12-2005) 20-26 ноября 2005, г. Звенигород (Московская обл.) стр. 441 – 444.

23) D. I. Mednikov, A. Milovidov, S. Yu. Sergunin, M. I. Kumskov “Identification of stable description elements using an active sensor”, Pattern Recognition and Image Analysis, Volume 19, Number 1 / Март 2009 г.

Приложение

Эскизы системных сценариев

Теперь опишем сценарии использования системы автоматизирующей процесс описанный выше. Роли будут те же, что и в описании процесса.

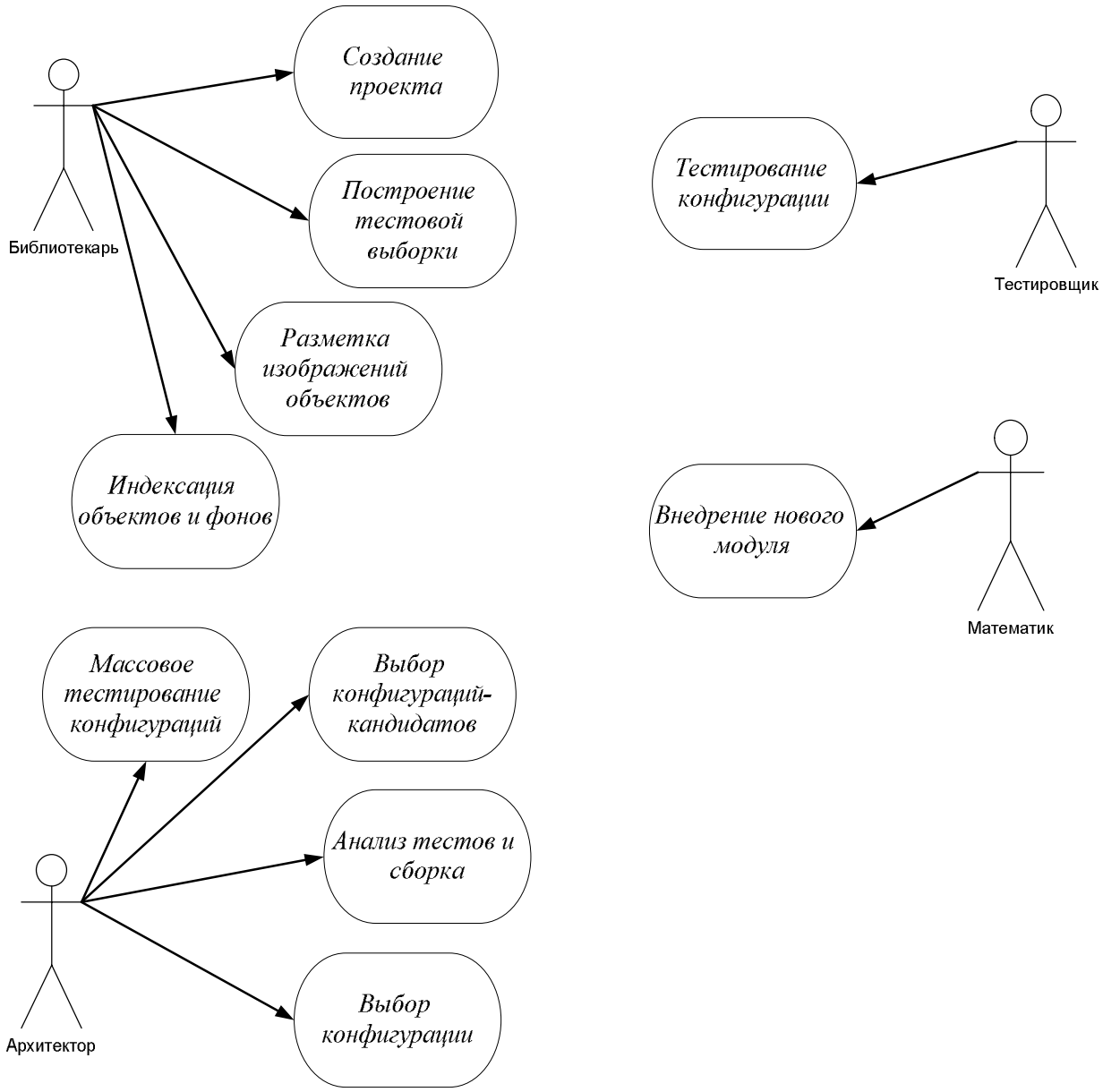

Рисунок П-1. Диаграмма сценариев использование системы автоматизирующей предлагаемый процесс.

Библиотекарь

Создание проекта.

1)Создание файла проекта под данный заказ.

2)Внесение файла содержащего численные требования заказчика к работе системы в

проект.

Построение тестовой выборки. Строит набор тестовых изображений, на основании соответствующих документов.

1)Выделение контуров (отдельный сценарий) на предоставленных заказчиком изображениях объектов и сохранение их как двоичных масок в базе

2)Сохранение вырезанных объектов на черном фоне в базе, как эталонных проекций объектов, и добавление фонов с вырезанными объектами в базу фонов, сохранение исходных изображений заказчика как тестовых сцен.

3)Для каждого объекта и его контура автоматически достраиваются необходимые масштабы (их количество параметр системы в целом)

4)Если есть трехмерное изображение объекта, то просто берутся его необходимые проекции методом рендеринга с определенной позиции, при этом эта позиция запоминается, иначе 4’

4’)По известным проекциям восстанавливается, используя сторонние компоненты, трехмерная модель объекта.

5’) Если полученная модель не нравится, то выбираются другие базовые проекции и опять пункт 4’ иначе далее пункт 4

5) При помощи фотошопа или другой подобной программы объединяются изображения объектов в разных проекциях и масштабах и изображения фонов, таким образом увеличивается число тестовых сцен.

6)Сохранение описания расположения файлов изображений в специальный файл конфигурации предметной области.

Разметка изображений объектов. Выделяет силуэты объектов.

1) Выбор метода автоматического распознавания контуров и запуск автоматического выделителя. Контуры появляются прямо на обрабатываемом изображении.

2) Удаление ненужным кусков контуров методом удаления их контрольных точек. Так же должен быть инструмент позволяющий удалить все контуры в определенной области, как ластик в Пэинте.

3) Добавление недостающих ветвей контуров, как "магнитное выделение" в фотошопе.

4) Подгонка формы полученных ветвей методом перетаскивания контрольных точек. (Как они будут выбираться на контурах и как будет апроксимироваться контурная линии при их перемещении это отдельный вопрос)

5) Сохранение полученного результата.

Индексация объектов и фонов. По требованию Архитектора для конкретной конфигурации системы выделяет, описывает конфигурации особых точек и индексирует их в базе.

1)Выбор конфигурационного файла (или их набора) конфигурации системы полученного от архитектора в меня выбора конфигурации.

2)Запуск модуля выделения ОТ для всех эталонных объектов и фонов, сбор полученных данных в БД

3)Запуск процедуры классификации и маркирования ОТ

4)Запуск модуля описания конфигураций ОТ для всех ОТ хранящихся в базе для данной конфигурации системы. Занесение полученной информации в базу

Архитектор

Выбор конфигурации. Выбирает набор версий алгоритмов и их параметры.

1)Выбор версий алгоритмов подсистем – в меню конфигурации модулей системы для каждого модуля можно выбрать из выпадающего списка необходимую версию алгоритма.

2)Выбор параметров алгоритмов – для каждого алгоритма можно вызвать меню, содержащее его возможные параметры, после выбора необходимых значений для сохранения полученной конфигурации алгоритма, необходимо нажать «сохранить», для сброса изменений – «отменить».

3)Сохранение полученной конфигурации системы в файл.

Массовое тестирование конфигураций. Первичное тестирование большого числа конфигураций на распознавание объект\фон.

1)Выбор набора конфигураций для тестирования из списка.

2)Выбор набора тестовых изображений, объектов и фонов (тестовой предметной области).

3)Запуск процедуры индексации тестовой предметной области для выбранных конфигураций.

4)Запуск процедуры классификации объект\фон для каждой конфигурации.

5)Тестирование процедуры распознавания объект\фон на тестовых изображениях..

6)Сохранение результатов тестирования в виде таблицы с процентом успешных распознаваний и списком изображений, на которых были допущены ошибки.

Выбор конфигураций-кандидатов. Выбирает наиболее перспективные конфигурации для настройки и подробного тестирования в следующей фазе.

1)Выбор конфигураций (или их семейств) показавших качественно лучшие результаты в таблице, полученной при Массовом тестировании.

2)Сохранение их как конфигураций кандидатов.

4)Анализ тестов и сборка. Анализ отчетов Тестировщика, сборка финального билда классификатора.

1)Получает доступ к отчетам Тестировщика о тестировании конкретном наборе конфигураций- кандидатов

2)Сборка объектного модуля из конкретной конфигурации кандидата.

Тестировщик.

Тестирование конфигурации. Тестирует конфигурации на тестовых сценах. Пытается подобрать параметры алгоритмов для максимального удовлетворения Критериям качества. Логи при этом ведутся автоматически.

1)Выбор файла конфигурации для тестирования

2)Выбор файла описывающего набор тестовых сцен и эталонных изображений объектов и фонов.

3)Запуск процедуры тонкого распознавания. Далее опять 1) с новыми параметрами или новой конфигурацией пока все конфигурации не будут проверены.

Математик.

Внедрение нового модуля.

1)Регистрация в системе нового модуля.

2)Ввод возможных значений его параметров.

3)Запуск создания нового меню конфигурирования полученного алгоритма.

Системные сценарии использования заказчика в случае наличия GUI

Опишем сценарии использования графического интерфейса прилагаемого к получаемому классификатору.

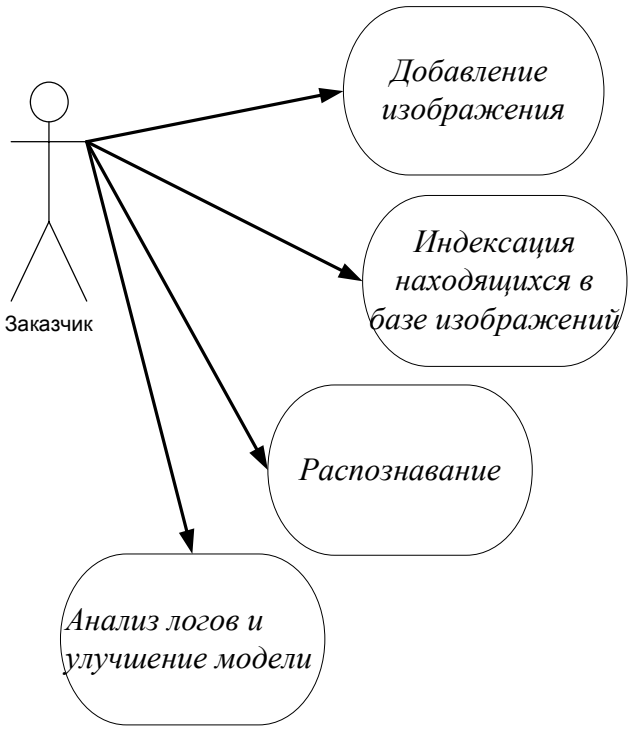

Рисунок П-2. Диаграмма сценариев использование объектного модуля классификатора при наличие в нем GUI.

Добавление изображения.

1)Выбор пути к новому(отсутствующему в базе) изображению или их набору.

2)Выбор типа изображения-\ий (объект\фон)

3)Добавление его в базу и разметка ОТ

Индексация или переиндксация находящихся в базе изображений

1)Все изображения лежащие в базе индексируются

Распознавание

1)Выбор объекта или их набора из находящихся в базе для распознавания

2)Выбор сцены или набора сцен для поиска объектов

3)Выбор параметров типа необходимости ведения логов

4)Запуск процедуры распознавания, ее вывод содержит, какие объекты найдены на каких сценах

Анализ логов и улучшение модели

1)Если заказчика не устраивает как система работает на определенных сценах, он присылает разработчикам логи и необходимые изображения для улучшения модели. Это оформляется как отдельный проект.