**РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ СОЦИАЛЬНЫЙ УНИВЕРСИТЕТ**

**Контрольная работа**

**по предмету «Информатика»**

**на тему: «Этапы развития и сравнительный анализ программ оптического распознавания »**

**Студентка Ильина О.О.**

**Тольятти, 2010**

**Введение**

оптическое распознавание коррекция ошибка

При работе с первоисточниками (или, как принято говорить при описании офисных технологий, с исходными документами) наиболее оптимальным может оказаться ввод какой-либо первичной текстовой информации без помощи клавиатуры. Такой способ может быть реализован получением данных из глобальных или локальных компьютерных сетей, посредством распознавания речи или оптического распознавания текста. В данной статье пойдет речь о последнем из названных способов.

Имея текст, напечатанный в типографии, на пишущей машинке, на принтере, копировальном аппарате, в факс-машине, то есть буквально текст любого происхождения, теперь - даже написанным от руки, Вы можете без помощи клавиатуры получить его в электронном виде для последующего редактирования, перевода или хранения на компьютере. Для этого Вам потребуются сканер и программа, называемая системой оптического распознавания - OCR. Каждый такой программный продукт имеет простейший автоматический режим "сканируй и распознавай", реализованный с помощью одной кнопки. Однако для того, чтобы достигнуть лучших из возможных для данной системы результатов, желательно (а нередко и обязательно) предварительно заниматься "ручной" настройкой системы распознавания на конкретный вид текста, а точнее на способ и качество начертаний букв и других знаков. Чтобы проделывать такие настройки, в меню программ обязательно отражены соответствующие регуляторы, но пользоваться ими удобнее при некотором знакомстве с принципами оптического распознавания.

**Как работает система оптического распознавания**

Фактически, когда Вы читаете (например книгу), то не представляете, какую работу делает Ваш мозг, чтобы преобразовать образцы чернильных или типографских меток в осмысленный текст. Компьютеры, однако, еще борются за решение этой одной из основных задач при создании машинного интеллекта.

Современное состояние OCR отвечает уровню, при котором довольно много систем могут быть весьма надежны, когда они имеют дело с чистым четким текстом, однако компьютерным программам все же еще далеко до того, чтобы предложить пользователю такую же безупречную точность, как предлагает другой помощник мозга человека - калькулятор. Большинство систем OCR работают с растровым изображением, которое получено через факс-модем или сканер. Для тех, кто никогда не видел OCR, обозначим скороговоркой этапы распознавания отсканированной страницы с точки зрения манипуляций над изображением текста. Делая "первый шаг", OCR должен разбить страницу на блоки текста, основанного на особенностях правого и левого выравнивания и наличия нескольких колонок. Потом эти блоки разбиваются в индивидуальные метки чернил (типографской краски и т.п.), которые, как правило, соответствуют отдельным буквам. Алгоритм распознавания делает предположения относительно соответствия чернильных меток символам; а затем делается выбор каждой буквы и цифры. В результате страница восстанавливается в символах текста (причем, в соответствующем оригиналу формате).

OCR-системы могут достигать наилучшей точности распознавания свыше 99 процентов для качественных изображений, составленных из обычных шрифтов. Хотя это число кажется почти совершенным, уровень ошибок все же удручает, потому что, если имеется приблизительно 1500 символов на странице, то даже при коэффициенте успешного распознавания 99,9% получается одна или две ошибки на страницу, требуя человеческого контроля результатов для гарантирования правильности соответствия оригиналу. Встречающиеся в жизни тексты порой весьма далеки от совершенных, и процент точности распознавания для "плохих" текстов часто недопустим для большинства приложений. Грязные изображения - здесь наиболее очевидная проблема, потому что даже малые пятна могут затенять определяющие части символа или преобразовывать один в другой.

Если документ был ксерокопирован, нередко возникают разрывы и слияния символов (такие тексты нередко возникают и при сканировании). Любой из этих эффектов может заставлять ошибаться, потому что некоторые из OCR систем полагают, что каждая соединенная черная метка должна быть одиночным символом.

Страница, расположенная с нарушением границ или перекосом, создает немного искаженные символьные изображения, которые могут путать программное обеспечение распознавания. Даже, когда изображения - чистые, странные или декоративные начертания могут вызывать проблемы, потому что они растягивают символы в различные формы для художественного эффекта. Кроме того буквы могут иметь вариации среди начертаний того же самого наименования шрифта, когда, к примеру, символы, воспроизведенные принтером HP DeskJet, отличны от символов, которые напечатаны на Apple LaserWriter.

Разработка алгоритмов, которые позволяют распознавать символы, несмотря на эти проблемы - трудная задача. Разработчики должны сбалансировать потребность в гибкости ПО с требованием его точности. Если программное обеспечение не достаточно гибко, то оно будет неточно сегментировать символ, когда будет сталкиваться с различными вариациями начертания. С другой стороны, слишком много гибкости может также вызывать ошибки. К примеру, отличие между "b" и "h" в нижнем регистре не очень большое, и гибкий алгоритм может спутать их.

**Деление текста на символы**

Одна из наиболее недорешенных задач в OCR - сегментация отдельных символов. Это происходит потому, что границы между буквами часто нечеткие, штрихи соседних букв соприкасаются, и это мешает делить слова на символы. Например, две или более буквы распознаются как одна, или одна - как две или три. Существуют шрифты, у которых, несмотря на общее хорошее качество печати, встречаются склейки (например, в шрифте Tense). Порою, увидев отсканированное изображение, в котором буквы сливаются на большей площади листа с текстом, можно заранее сказать, что данный текст не распознается корректно, и сэкономить свое время, даже не запустив программу на распознавание.

**Образ страницы и распознавание по шаблонам**

Программное обеспечение OCR обычно работает с большим растровым изображением страницы из сканера. Изображения со стандартной степенью разрешения получаются сканированием с точностью 300x300 пикселей на дюйм. Изображение бумажного листа формата A4 (11 формата) при этом разрешении занимает около 1 Мбайт памяти. Изображения c более тонким разрешением возможны с применением более дорогих сканеров, но они часто непрактичны для OCR-приложений из-за большой требуемой памяти для изображений или длительности самого процесса сканирования. Кроме того, увеличение разрешения сканера не приносит пользы, если качество оригинала недостаточно хорошее. Анализируя изображения, исходящие из факсимильных машин, подбирают разрешение, проверяя отличия от имеющихся шаблонов символов (самая маленькая их величина при самом близком соответствии).

По этой причине, большинство систем имеет шаблоны, созданные для различных начертаний. После нескольких слов, программное обеспечение определяет основное используемое начертание и ищет соответствующие пары только с этим начертанием. В некоторых случаях программное обеспечение использует численные значения частей символа (пропорций), чтобы определить новый шрифт. Это может улучшать эффективность распознавания до других стилей печати, типа курсива или жирного начертания слова, найденного на странице.

Таким образом, при распознавании по шаблонам представление описания похоже на представление входных объектов, и описание сравнивается с ними непосредственно.

Программа распознавания TypeReader фирмы ExperVision использует машинно-зависимые алгоритмы, чтобы найти наиболее важные пиксели для различения символов. Берется 30 различных вариантов символа (которые исходят из 30 различных документов) и в каждом из этих примеров анализируется по 100 пикселей, чтобы определить то, какие из них наиболее вероятно должны быть характеристикой для специфического символа (к примеру, пиксели на нижней части "A" всегда чисты и т. д.). Краевые элементы изображения вдоль границ символа часто исключаются этим анализом, потому что они могут быть темны в чистых изображениях, но чисты в страницах с пониженной четкостью. Машинный алгоритм ранжирует 100 пикселей от более до менее непротиворечивых для каждого из символов.

Однако этого шага не достаточно из-за подобия между буквами. Например, в нижнем регистре "h", "k", "f", "l", и "t" имеют общие длинные вертикальные штрихи слева, и пиксели вдоль этого штриха не будут передавать различия этих символов. По этой причине, программное обеспечение фирмы ExperVision находит 24 пикселя, которые являются наиболее непротиворечивыми у всех символов в алфавите, и удаляет их из списка для индивидуальных символов. Пиксели, которые оставлены, должны с наибольшей вероятностью быть уникальными.

Таким образом, требуется создать шаблон целостного описания символа, так чтобы любое изображение буквы в него попадало, а любые допустимые изображения других букв - нет. В чистом виде шаблонное описание может применяться только для распознавания печатных символов. Заметим, что рукописные шрифты тоже распознаются с применением шаблонов (только "более хитрых"), но одновременно со структурным подходом.

**Структурный подход**

Самая продаваемая в мире система OCR - Caere OmniPage Professional использует алгоритм, который не должен настраиваться на индивидуальное начертание, потому что он основан на нахождении общих специфических особенностей символов. Эта система содержит 100 различных "экспертных систем", которые в действительности являются только алгоритмами для идентификации 100 различных символов: верхнего и нижнего регистра от "A" до "Z", записи чисел и символов пунктуации. Каждая из этих экспертных систем ищет "особенности" начертаний типа "островов", "полуостровов", точек, прямых оттисков и дуг. Экспертные системы также рассматривают горизонтальные и вертикальные проекции оттисков буквы и обращают внимание на основные особенности в созданных кривых, суммируя в них число темных пикселей.

Очевидно, что "t" всегда состоит из жирного вертикального штриха, поперечного с горизонтальным штрихом. Дизайнер шрифта может включать в него засечки или сдвигать расположение пересечений, но человек может без труда выяснять и игнорировать эти отличия. Основанные на шаблоне подходы должны создать шаблоны для каждого возможного шрифта (программа ExperVision, к примеру, использует 2100 таких начертаний). Caere напротив пробует находить сущность каждого символа на основе структурного подхода.

Нечеткий текст может стать специфической проблемой для этих структурных алгоритмов, потому что отсутствующий пиксель может разбивать длинный штрих или кривую. Аналогично, дополнительное пятно грязи могло закрывать петлю записи числа "5" и заставлять ее выглядеть наподобие "6" согласно основанному на особенности начертания алгоритму. Алгоритм на основе шаблонов здесь не имеет таких проблем, потому что остальные пиксели в "5" выровнялись бы правильно.

**Контекстное распознавание**

Люди способны быстро различить на бумаге "h" и "b" еще и потому, что они знают контекст слова, в котором встречаются эти буквы. По этой причине программное обеспечение системы OCR включает словари для помощи алгоритмам распознавания. Словари предоставляют справки во многих случаях, но быстро отказывают, когда, скажем, программное обеспечение сталкивается с именами собственными, которые не находятся в словаре.

Корпорация Xerox имеет один из наиболее сложных пакетов программ с контекстным анализом. Пакет, называемый Lexifier (сокращение для "лексический классификатор"), содержит большинство главных правил записи буквенных структур или принятых образцов номеров телефонов. Он помогает интерпретировать имена собственные типа Lexifier, которые кажутся совершенно правильно английскими, но не находятся в словаре. Этот эффект особенно заметен в российской программе FineReader, который чаще, чем в среднем по всем символам, ошибается в словах, которые отсутствует в его словаре (к примеру, названия фирм ему даются тяжело).

Таким образом, алгоритм распознавания в самом общем виде состоит в последовательном выдвижении и проверке гипотез, причем порядок их выдвижения управляется заложенными в программу знаниями об исследуемом предмете и результатами проверки предыдущих гипотез (производится дополнительный анализ объекта в рамках выдвинутой гипотезы).

Основное требование к предварительной обработке - не потерять о входном объекте существенную информацию. Поскольку для выделения целого требуются его части, а для нахождения частей требуется целое, целостный процесс восприятия может происходить только в рамках гипотезы о воспринимаемом объекте - в целом.

Если программа должна приближаться к качеству восприятия текста человеком, то, скорее всего, она может успешно использовать алгоритм, "подсмотренный" у человека-читателя. Читая предложение, человек узнает буквы, воспринимает слова, связывает их в синтаксические конструкции и понимает смысл предложения. Все процессы происходят одновременно, влияя друг на друга, а окончательное решение принимается на основе полного учета их результатов.

Как было показано в примерах, целостное описание класса объектов восприятия должно удовлетворять двум свойствам: во-первых, все объекты данного класса должны удовлетворять этому описанию, во-вторых, ни один объект другого класса не должен удовлетворять описанию.

Процесс выдвижения и проверки гипотез явно отражен в алгоритмах программы (каждая гипотеза имеет числовую оценку или операцию сравнения). Обычно гипотезы выдвигаются последовательно, объединяются в список и сортируются на основе предварительной оценки гипотезы. Окончательный выбор гипотезы делается в рамках контекста, с привлечением, возможно, дополнительных источников знания.

**Коррекция ошибок**

Подход в построении шаблонов в программе TypeReader настроен так, чтобы найти базовые пиксели в широком диапазоне документов. Разорванные символы находятся автоматически, потому что программное обеспечение избегает пикселей в указанном доверительном интервале, которые могут исчезать в плохих изображениях. Программное обеспечение только должно найти места, в которых произошло разбиение или размывание символов. Caere также имеет частный алгоритм для разорванных символов.

Все программы оптического распознавания позволяют проводить проверку ошибок пользователем, используя одновременное исходное изображение в окне экрана: нет необходимости консультироваться с бумажной версией. К примеру, программа TypeReader просматривает строки текста для поиска ошибок клавишей табуляции (программное обеспечение помещает отсечение изображения за текстом так, чтобы Вы могли сравнивать текст с изображением без того, чтобы переместить взгляд). Ручной поиск ошибок в русских программах распознавания происходит аналогично.

Кроме того, уже к 1993 году была продемонстрировано, что можно делать компьютерные программы, использующие многие "человеческие" способы анализа текстового распознавания: OCR обучаются, чтобы делать лучше работу с неразборчивыми рукописными символами и метками, подобно зачеркиванию или подчеркиванию. Но пока дальше демонстрационных трюков эти достижения не слишком пошли.

**Увеличение скорости бесклавиатурного ввода документов в технологиях электронного документооборота**

Скорость распознавания порядка одного документа в минуту, считая накладные расходы на коррекцию, не позволяет говорить об удовлетворительном уровне автоматизации ввода больших потоков бумаг. Ведь скоростные сканеры Fujitsu в настоящее время позволяют вводить документы со скоростью 20-50 документов в минуту. Адекватные скорости оптического распознавания до недавнего времени не были достигнуты.

Однако эта скорость, не имеющая первостепенного значения для отдельно взятого "писателя", использующего бесклавиатурный ввод материалов первоисточников, необходима в архивах.

Еще в 1995 году были предложены два радикальных способа решения этой проблемы.

Одним из них стало использование всех ресурсов Pentium-процессоров в системах распознавания. Для этого фирмой Intel при участии российских разработчиков были созданы специальные библиотеки для программ OCR (в рамках широко разрекламированной программы NSP - позднее замененной программой MMX).

Как было сказано, при структурном подходе описание представляется в виде графа, узлами которого являются элементы входного объекта, а дугами - пространственные отношения между этими элементами. При этом производится преобразование входного объекта в представление, удобное для дальнейшей обработки (для "посвященных" можно упомянуть, к примеру, здесь векторизацию изображения, преобразование Фурье), либо в получение всевозможных вариантов сегментации входного объекта, из которых путем выдвижения и проверки гипотез выбирается правильный (построение графа линейного деления слова, предварительное выделение объектов при распознавании формы).

Именно такие операции (свертки, преобразования Фурье и др.) были оптимизированы на низком уровне для Pentium-процессоров, а соответствующие библиотеки бесплатно предоставлены фирмой Intel разработчикам. Еще более кардинальный способ был реализован в России, когда была создана установка распознавания (на базе OCR фирмы "Окрус"), использующая параллельные вычисления наиболее трудоемких операций на транспьютерах. Но дальнейшего развития этот эксперимент не получил.

**Заключение**

Главный вывод в данном разговоре об OCR - это то, что описываемые системы многое умеют и являются полноправными системами в пакете программ "электронного офиса". Их надо иметь.

Если же делать выводы о закономерностях в построении программ OCR, то можно сказать следующее.

Шаблонное описание проще и эффективней в реализации, но, в отличие от структурного, не позволяет описывать объекты с высокой степенью изменчивости. В простейшем случае, шаблонное описание может применяться для распознавания печатных символов, а структурное - для рукописных. Отметим, что все современные российские программы распознавания - Autor, CuneiForm и FineReader - являются структурно-шаблонными. По-видимому, только сочетание этих двух методов обеспечивает приемлемую надежность. Целостность процесса восприятия программных продуктов OCR предполагает, что все исследуемый объект должен представляться и обрабатываться по возможности весь сразу, а источники знания должны работать по возможности одновременно: каждая фраза подвергается распознаванию, словарной и контекстной обработке (для создания обратной связи от контекстной обработки к распознаванию).

**Список литературы**

1. Новиков Ф., Яценко А. Microsoft Office 2000 в целом. СПб., 1999.

2. Рабин Ч. Эффективная работа с Microsoft Office 2000. СПб., 2000.

3. Айден К., Фибельман Х., Крамер М. Аппаратные средства РС. СПб., 1997