

Использование трехмерных анимированных изображений жестов рук для создания анимационной CAPTCHA нового типа

А.Е.Шумилов
МГТУ им. Н.Э.Баумана
Москва, Россия
e-mail: ashumilov@it-claim.ru

Аннотация

Владельцы и администраторы сайтов ежедневно сталкиваются с проблемой спама, целью которого являются реклама, получение доступа к личной информации, создание ссылок для повышения рейтингов в поисковых системах и тому подобное. Очень часто для защиты сайтов от спама и автоматических регистраций используется CAPTCHA. Хотя CAPTCHA и не гарантирует стопроцентной защиты от спама, тем не менее, применение достаточной сложной для автоматического распознавания CAPTCHA способно остановить спамера, так как большинство программ автоматического распознавания (спам-ботов) после одной или нескольких неудачных попыток прохождения теста переходят к поиску другого сайта с более слабой спам-защитой. В этой статье рассматривается один из самых сложных для автоматического распознавания вариант реализации CAPTCHA с использованием трехмерных анимированных изображений жестов рук.

Введение

Владельцы и администраторы сайтов ежедневно сталкиваются с проблемой спама, целью которого являются реклама, получение доступа к личной информации, создание ссылок для повышения рейтингов в поисковых системах и тому подобное. Очень часто для защиты сайтов от спама и автоматических регистраций используются различные варианты CAPTCHA – специальных виджетов, которые предлагают пользователю выполнить простое задание - например, отображают искаженный текст и просят его ввести. Как правило, подобные задания не вызывают трудностей у человека, но оказываются сложны для специализированного спам-бота, который после нескольких неудачных попыток прохождения теста переходит к поиску другого сайта с более слабой защитой.

Со временем технологии распознавания изображений и разрешения других видов тестов развиваются, поэтому разработчики CAPTCHA вынуждены усложнять тесты или искать иные решения. Сейчас на сайтах не редко можно встретить искаженные тексты, которые даже человеку трудно понять, что вызывает раздражение у пользователей и создает сложности для веб-мастеров. В конце 2013 г. разработчики проекта Vicarious¹ заявили, что их алгоритмы могут надежно распознавать все современные виды тестов CAPTCHA, включая популярную reCAPTCHA от компании Google, которая чаще других используется в Интернете для защиты сайтов от ботов [1]. Это заявление стало широко

¹ <http://news.vicarious.com/post/65316134613/vicarious-ai-passes-first-turing-test-captcha>

известным и породила новый рост интереса к разработке более совершенных видов САРТНА.

Разновидности САРТНА

Из всего многообразия вариантов реализации САРТНА можно выделить следующие такие группы: текстовые, звуковые, математические, логические, образные, интерактивные, анимационные и т.д.

Текстовые САРТНА. Это наиболее распространенный вариант реализации, в котором пользователю предлагается ввести в текстовое поле символы, показанные на картинке. Картинки обычно содержат помехи разного рода: символы в той или иной степени искажены, развернуты, расположены не на одном уровне, на изображение наложены цветные пятна, полосы и т.п. Чем сильнее искажена картинка, тем сложнее боту распознать ее, но и человек часто испытывает затруднения в понимании искаженного текста.



Рис. 1. Примеры простых и сложных вариантов текстовых САРТНА

Звуковые САРТНА Звуковая или аудио САРТНА обычно используется как дополнение к текстовой и позволяет пользователю пройти тест даже, если он не может по какой-то причине прочитать картинку. Аудио САРТНА обычно содержит фоновые звуки и посторонние шумы, добавленные для защиты от ботов, но затрудняющие распознавание и человеку. К тому же аудио САРТНА требует наличия оборудования воспроизведения звука на компьютере или мобильном устройстве.

Математические САРТНА. В этом варианте пользователю предлагается выполнить простейшие математические вычисления. Для человека такое задание будет необременительным, только если предлагается выполнить действия над простыми числами. Однако в этом случае существует опасность угадывания ботом правильного ответа простым перебором.

Логические САРТНА. Логические варианты реализации предлагают пользователю ответить на простой логический вопрос. Проблема в том, что пользователь может не знать языка, на котором задается вопрос. Кроме того, количество вопросов ограничено, что позволяет спамеру составить базу данных с вариантами правильных ответов.

Образные CAPTCHA. Пользователю предлагается распознать объект на изображении. При ограниченном количестве предлагаемых изображений, спамер, как и в случае с логическими вопросами, может составить свою базу данных изображений.

Интерактивная CAPTCHA. В этом случае пользователю предлагается выполнить одно или несколько действий, как правило, с помощью компьютерной мыши, или сыграть в мини-игру. Такие CAPTCHA действительно сложны для распознавания ботами, особенно, пока они новые. Но проблема в том, что обычно на их выполнение пользователь затрачивает значительное время, большее, чем на решение текстовой CAPTCHA.

Анимационная CAPTCHA. В этом варианте пользователю предлагается распознать движущееся изображение (чаще всего – это бегущая строка символов). Однако есть и другие, более сложные варианты. Например, азиатские разработчики представили вариант теста с движением изображения на сильно зашумленном фоне². Анимационная CAPTCHA действительно сложна для распознавания ботами, но вместе с тем и может вызывать трудности у людей с дефектами зрения.

Анимационная CAPTCHA с использованием жестов рук

Основная идея этого вида CAPTCHA заключается в том, чтобы показывать пользователю последовательность легко узнаваемых жестов, которые он мог бы воспринимать как последовательность символов или слов. Для обеспечения повышенной сложности автоматического распознавания демонстрируемого жеста предлагается использовать трехмерную интерактивную визуализацию.

В качестве научной базы жестовой визуализации и распознавания, лежат многолетние исследования в рамках научной школы МГТУ им. Н.Э.Баумана под руководством Ю.Н. Филипповича (<http://it-claim.ru/Projects/Gesture/index.htm>), направленные на создание жестомимического интерфейса.

Реализация проекта создания анимационной CAPTCHA с использованием жестов рук требует решения следующего спектра задач:

1. Разработка трехмерной модели руки для эффективного отображения в различных браузерах и на различных устройствах,
2. Разработка алгоритмов моделирования движений руки,
3. Разработка интерфейса создания жестов и перевода текста в жесты,
4. Проработка механизмов защиты CAPTCHA от взлома,
5. Отбор распространённых жестов и разработка средств (подсказок) для их лучшего понимания человеком.

Разработка трехмерной модели руки для отображения в браузере

Трехмерную модель руки можно создать в одном из популярных графических редакторов, таких как Autodesk 3d Max, Autodesk Maya, Blender. Для реализации 3D моделей была выбрана графическая библиотека WebGL, которая позволяющую создавать

² <http://www.aiportal.ru/news/animated-captcha.html>

3d-графику на языке JavaScript и поддерживается большинством современных веб-браузеров. Для демонстрации трехмерной модели в браузере выбрана кроссбраузерная библиотека Three.js [5], которая поставляется в одном JavaScript файле и может быть подключена к странице в любом месте. Созданные в графическом редакторе модели были преобразованы к формату, который может использоваться с этой библиотекой, например, COLLADA или собственный формат Three.js.

Строение человеческого скелета с точки зрения компьютерного моделирования очень хорошо описано в книге Стефани Рис «Анимация персонажей в 3D Studio MAX». Кисть руки автор описывает следующим образом: «Кисть руки содержит двадцать семь костей... Восемь костей неправильной формы в основании кисти руки называются запястьем (carpal). Они аналогичны предплюсневым костям ступни. Ладонь руки состоит из пяти длинных тонких костей с закругленными концами, которые называются пястью (metacarpal) и аналогичны плюсне ступни. Наконеч, пальцы, включая большой, образованы четырнадцатью короткими костями с прямоугольными концами, это — фаланги, они аналогичны фалангам ступни».

Для исследования эффективности САРТНА на первом этапе была разработана менее детализированная модель кисти руки, которая содержит только 17 костей. Одна из них определяет положение кисти в пространстве, другая позволяет немного изменить форму ладони около мизинца, а все остальные кости определяют положение пальцев. Кость, расположенная в ладони у основания большого пальца в созданной модели считается аналогом ближней к ладони фаланги всех остальных пальцев.

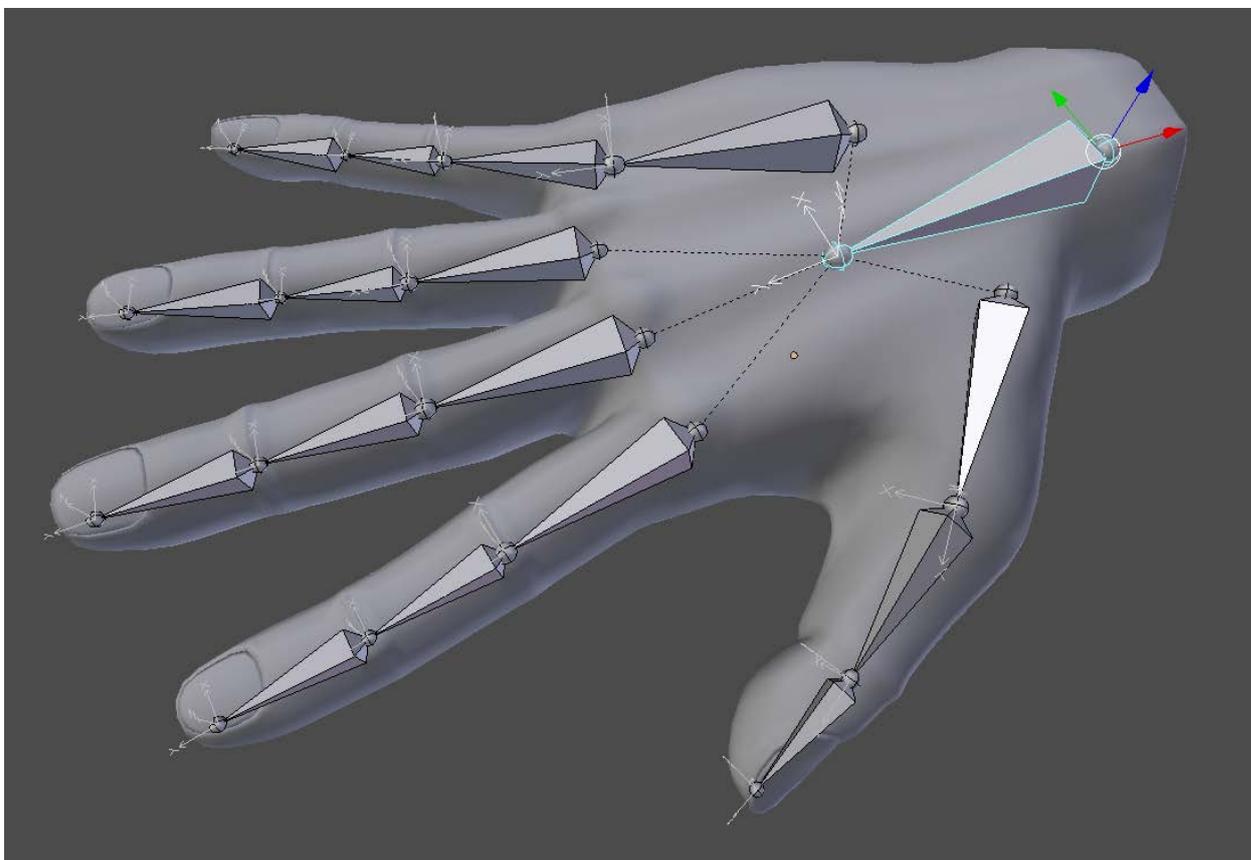


Рис.2.Скелетная модель кисти руки.

Разработка алгоритма моделирования движений руки

Трёхмерные графические редакторы позволяют создавать также и анимацию. Но эта анимация не предусматривает возможности ее динамического изменения и не может быть использована при создании САПТСНА, так как в этом случае при добавлении нового жеста в алфавит для него пришлось бы вручную создавать и хранить анимации перехода руки из положения этого жеста в положения всех остальных жестов алфавита. В связи с этим было принято решение анимировать руку программно, используя прямой доступ к положению костей скелетной модели. Это может быть сделано с помощью библиотеки Tween.js, обладающей простым, но мощным API. Однако, при использовании Tween.js возможны коллизии, когда объекты (в нашем случае, пальцы руки) пересекают друг друга и проходят сквозь друг друга при движении. Поэтому было решено разработать собственный алгоритм моделирования движений и обработки коллизий для создания реалистичной динамической анимации.

В данном случае наиболее важным является положение костей пальцев, для упрощения работы с ними был создан класс Finger, который группирует кости, относящиеся к одному пальцу, рассчитывает и хранит положение осей, вокруг которых поворачиваются кости при сгибании пальца и матрицы поворота этих костей. Матрицы поворота необходимо сохранять, так как в Three.js при каждом изменении положения кости необходимо указывать ее расположение относительно начальной позиции, а не относительно предыдущей. Класс также содержит методы rotateN для поворота ближней к ладони фаланги пальца в плоскости ладони и rotate для сгибания любого сустава пальца. Фрагменты исходного кода приведены ниже:

```
function Finger(name, bonesArray, palmBoneName, thumbAngle) {
  palmBoneName = palmBoneName || '';
  thumbAngle = thumbAngle || 0;
  var checkPalm = Boolean(palmBoneName);
  var bones = [];
  for (var i = 0; i < bonesArray.length; i++) {
    if ((checkPalm) && (palmBoneName == bonesArray[i].name)) {
      bones[3] = bonesArray[i];
    }
    else if (bonesArray[i].name.substring(0, name.length) == name) {
      var temp = bonesArray[i].name.substring(name.length + 1);
      switch (temp)
      {
        case '':
          bones[0] = bonesArray[i];
          break;
        case '001':
          bones[1] = bonesArray[i];
          break;
        case '002':
          bones[2] = bonesArray[i];
          break;
      }
    }
  }
  var fingerVector = new THREE.Vector3();
  fingerVector.getPositionFromMatrix(bones[0].matrix).normalize();
  var hAxis = new THREE.Vector3(fingerVector.z, 0, -
  fingerVector.x).normalize();
  var vAxis = new THREE.Vector3(0, 1, 0);
  if (thumbAngle != 0) {
```

```

varthumbMatrix = new THREE.Matrix4();
thumbMatrix.makeRotationAxis(fingerVector, thumbAngle * Math.PI / 180);
hAxis.applyMatrix4(thumbMatrix);
vAxis.applyMatrix4(thumbMatrix);
}
varhMatrix = new THREE.Matrix4();
varvMatrix = new THREE.Matrix4();

this.rotateH = function(angle) {
varrMatrix = new THREE.Matrix4();
hMatrix.makeRotationAxis(vAxis, angle * Math.PI / 180);
rMatrix.multiplyMatrices(hMatrix, vMatrix);
bones[0].setRotationFromMatrix(rMatrix);
};

this.rotateV = function(angle, idx) {
idx = idx || 0;
varrMatrix = new THREE.Matrix4();
if (idx == 0) {
vMatrix.makeRotationAxis(hAxis, angle * Math.PI / 180);
rMatrix.multiplyMatrices(hMatrix, vMatrix);
} else {
vartempMatrix = new THREE.Matrix4();
tempMatrix.makeRotationAxis(hAxis, angle * Math.PI / 180);
rMatrix.multiplyMatrices(hMatrix, tempMatrix);
}
bones[idx].setRotationFromMatrix(rMatrix);
};
}

```

Разработка интерфейса создания жестов и перевода текста в жесты

Для того чтобы модель руки могла воспроизводить в виде жестов заданные символы, необходимо создать алфавит жестов. Алфавит должен хранить для каждого символа углы расположения костей относительно друг друга. Для более удобного создания и пополнения алфавита жестов было принято решение разработать графический редактор жестов. С помощью редактора жестов можно также создавать разные алфавиты для различных целевых аудиторий. На рис. 2 показано окно прототипа редактора жестов, позволяющее управлять движением пальцев руки.

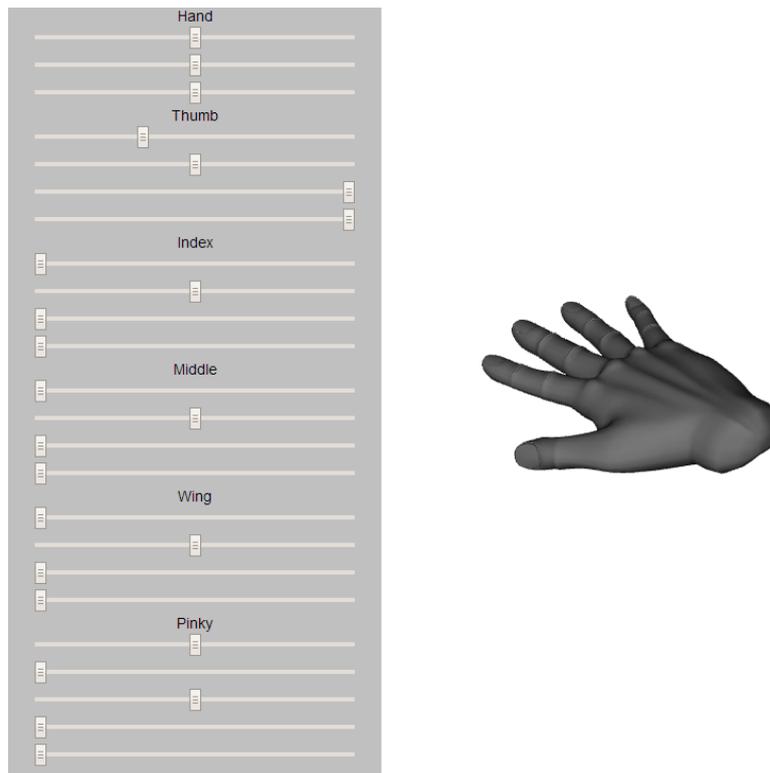


Рис.2. Интерфейс управление движением пальцев руки.

Защита жестового САРТСНА от взлома

Относительно небольшое ограниченное количество символов в алфавите может позволить злоумышленнику создать собственную базу данных относительных углов костей, соответствующих символам алфавита. Поэтому имеет смысл рассмотреть возможность преобразования сформированной анимации в формат GIF для демонстрации его пользователю в качестве САРТСНА. В дальнейшем также предполагается отработать и другие методы защиты – повышение полигональности модели, изменение текстур, добавление шумов и т.д.

Понимание жестового САРТСНА человеком

В настоящее время количество общепринятых или интуитивно понятных жестов руки достаточно невелико, поэтому использование только таких жестов диктует достаточно небольшой размер алфавита. Алфавит может быть расширен в случае использования на специализированных сайтах, предназначенных для людей, владеющих каким-либо жестовым языком, например, на сайтах, предназначенных для подводников, музыкантов, спортсменов и т.д..

Тем не менее, подобная САРТСНА применима на любых сайтах, так как в задании теста жесты могут быть показаны несколькими моделями рук, а также скомбинированы с вращающимися моделями букв латинского или кириллического алфавита. Таким образом, может быть зашифрован цифробуквенный код, который применяется в большинстве современных САРТСНА. Кроме того, жестовый САРТСНА может содержать несколько моделей рук и предлагать пользователю выбрать определенный символ на основании того, какая именно в данный момент используется модель.

Социальная значимость проекта

Большинство современных САРТСНА предлагает альтернативный вариант тестирования, которым может воспользоваться человек, который по той или иной причине испытывает затруднения с прохождением основного варианта. Обычно в качестве запасного варианта человеку предлагается прослушать аудиозапись и указать цифры и буквы, которые в ней прозвучали. К сожалению, люди с нарушениями органов слуха лишены такой возможности, поэтому единственная их возможность пройти тестирование – правильно ответить на основное задание. Использование жестового САРТСНА позволило бы решить эту проблему, предоставив альтернативный вариант теста, в значительной степени отличающийся от основного, но при этом не накладывающий ограничений по слуху.

Практическая значимость разработки и возможность коммерциализации

Защита от сайтов от спама не только серьезный вызов для исследователей и разработчиков знают, это еще и достаточно распространенный бизнес. Например, Akismet (<http://akismet.com>), популярная система по отлову спама (часто используется как плагин для WordPress), фиксирует более 18 миллионов спам-комментариев в день. В сумме эта система отловила 20 миллиардов комментариев за все ее историю. Система Mollom (<http://mollom.com/>), которая создана с тем же предназначением, отлавливает более миллиона спам-комментариев в день, эта же система подсчитала, что более 90% комментариев – спам³.

Руководитель отдела веб-поиска компании "Яндекс" Александр Садовский в своем напутнем выступлении на IBC Russia 2013 отметил как значимое событие разработку "Яндекс" новой САРТСНА с использованием кириллических символов. Кроме того, он предположил, что это может существенно изменить рынок в этой области.

Для оценки значимости новых видов САРТСНА часто используют тариф распространенного заработка в Интернет – пользователи, осуществляющие ввод вручную, зарабатывают в среднем около 1 доллара за 1000 каптч.

Заключение

У предлагаемого варианта САРТСНА хорошие перспективы использования. Во-первых, пользователю предлагается качественное, легкое для восприятия трехмерное изображение знакомого жеста, распознавание которого не вызовет у него затруднений и не потребует длительного времени. Во-вторых, этот вариант реализации САРТСНА относится к анимационной группе, которая считается самой сложной для автоматического распознавания.

У проекта также могут быть перспективы развития в части создания тематических (настраиваемых администраторами) серий жестов для различных профессиональных и социальных групп, в том числе и с ограниченными возможностями. Развитие мобильных технологий позволяет потенциально использовать технические возможности устройств для повторения жестов вместо набора, а также применения соответствующих алгоритмов для задач идентификации и авторизации пользователей.

³ По материалам статьи <http://habrahabr.ru/post/120851/>

На данный момент мной выполнена концептуальная разработка проекта, проведен анализ предметной области, разработана трехмерная модель руки и портирована в WebGL, создан интерфейс управления ее движением с использованием библиотеки Three.js, ведутся работы по развитию данного проекта в рамках выпускной дипломной работы.

Ссылки

1. Журнал «Хакер» <http://www.hacker.ru>.
2. Стефани Рис. Анимация персонажей в 3D Studio MAX.
3. David Bushell. In Search Of The Perfect CAPTCHA. <http://coding.smashingmagazine.com/2011/03/04/in-search-of-the-perfect-captcha>.
4. Jeff Atwood. CAPTCHA Effectiveness. <http://www.codinghorror.com/blog/2006/10/captcha-effectiveness.html>.
5. <http://threejs.org/>