

Санкт-Петербургский государственный университет

Математико-Механический факультет

Алгоритмы распознавания жестов мышью

Курсовая работа студента 243 группы
Верещагина Константина Алексеевича

Научный руководитель.....Литвинов Ю.В.

старший преподаватель

кафедры системного
программирования

Санкт-Петербург, 2014

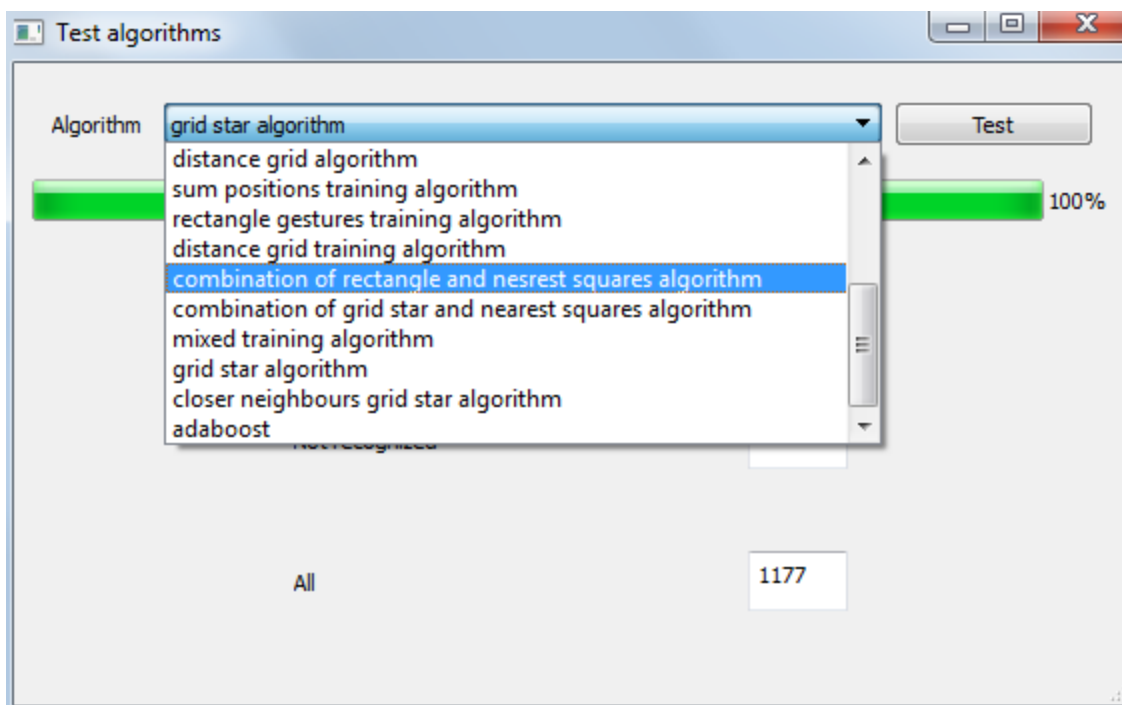
Оглавление

Введение.....	3
Основные принципы работы алгоритмов распознавания.....	4
Пример алгоритма.....	6
Grid Star Algorithm.....	7
Результаты тестирования.....	9
Метод k ближайших соседей.....	10
Заключение.....	12

Введение

В CASE-системе QReal изначально предполагалось, что добавление объектов на диаграмму должно происходить перетаскиванием с помощью операции drag-and-drop элементов из палитры. Но для того, чтобы пользователям было удобнее работать с QReal, была предложена идея распознавания жестов мышью, позволяющая обеспечить более быстрое добавление элементов.

Для этого в отдельном проекте была реализована архитектура, предоставляющая интерфейс распознавания жестов и тестовый инструментарий, позволяющий тестировать различные алгоритмы и делать некоторые выводы об эффективности работы того или иного алгоритма:



Было протестировано множество алгоритмов, но большинство из них имели определенные недостатки, такие как скорость работы, низкая точность распознавания конкретных жестов и т.д. Но при этом не были реализованы некоторые потенциально эффективные алгоритмы.

Итак, задача состоит в том, чтобы реализовать новые алгоритмы, протестировать их и сравнить точность распознавания с другими алгоритмами. Всё это может улучшить инструменты поддержки жестов мыши в системе QReal, что и является конечной целью проекта.

Основные принципы работы алгоритмов распознавания

Алгоритм распознавания жестов работает следующим образом: есть список идеальных жестов, с каждым из них сравнивается жест, который изобразил пользователь. Идеальный жест - жест, соответствующий некоторому объекту, который пытается изобразить пользователь. После некоторой команды программа пытается распознать фигуру, для этого в списке идеальных жестов ищется фигура, похожая на то, что изобразил пользователь. Если такой жест находится, то выполняется ассоциированная с ним команда.

Также существуют методы распознавания, использующие обучение, для этого нужна обучающая выборка - набор пользовательских жестов, в котором про каждый жест известно, к какому классу он принадлежит

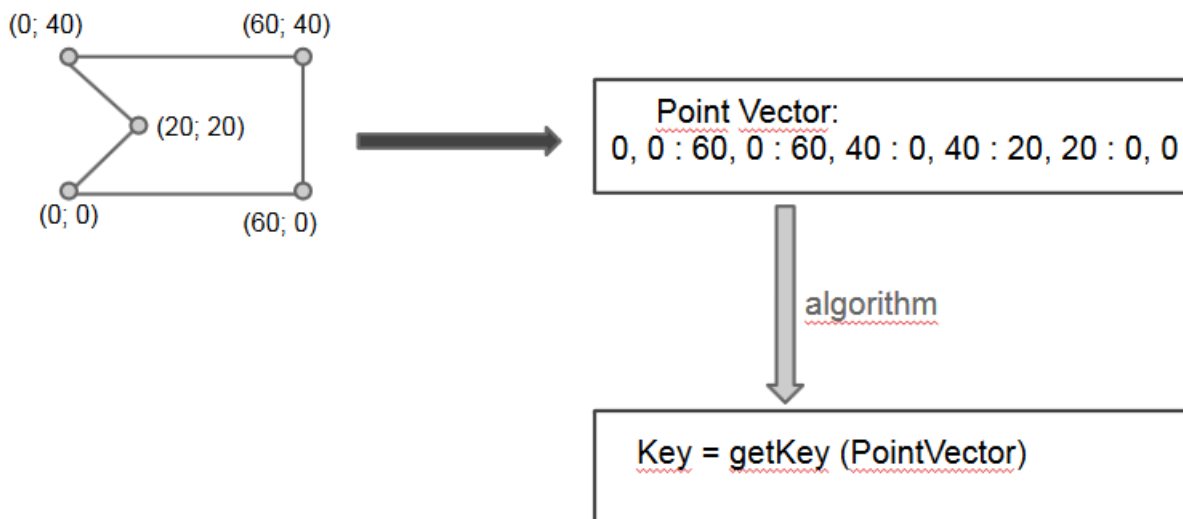
Основные этапы распознавания жестов:

1) *Построение списка точек:*

При движении мыши происходит запись координат курсора мыши и строится строка из последовательных точек жеста

2) *Построение классификатора:*

Для сравнения жестов вводится множество признаков, по которым оно производится. Это множество признаков называется классификатором. На множестве классификаторов определяется расстояние



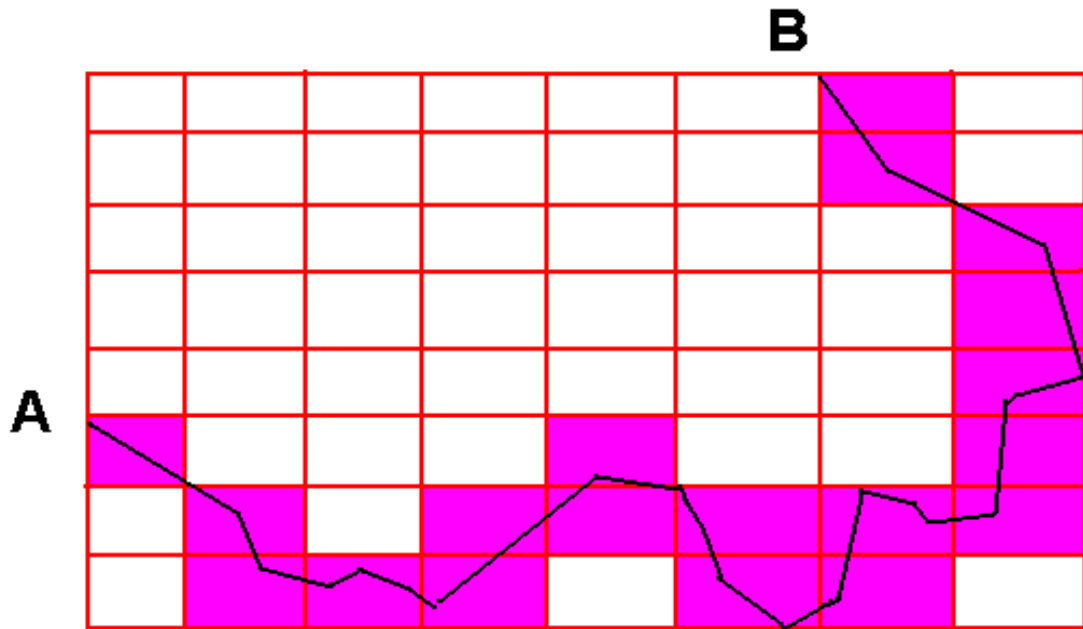
3) Выбор объекта:

Для каждого идеального жеста вычисляется расстояние до пользовательского жеста, после чего берётся идеальный жест, которому соответствует минимальное расстояние. Результат распознавания - объект, с которым ассоциирован этот жест.

Пример алгоритма распознавания жестов

Одним из простейших алгоритмов распознавания является алгоритм, использующий построение сетки вокруг жеста: жест вписывается в прямоугольник, который разбивается на ячейки фиксированным количеством горизонтальных и вертикальных линий. В классификатор входят только те ячейки прямоугольника, через которые проходит жест. В качестве метрики берём расстояние Левенштейна с весами - минимальное количество перестановок, удалений и вставок символов, необходимое для преобразования одной строки в другую. Каждой ячейке сопоставляется некоторая

последовательность символов алфавита $\{0; 1\}$, так, чтобы две любые соседние ячейки отличались одним символом.



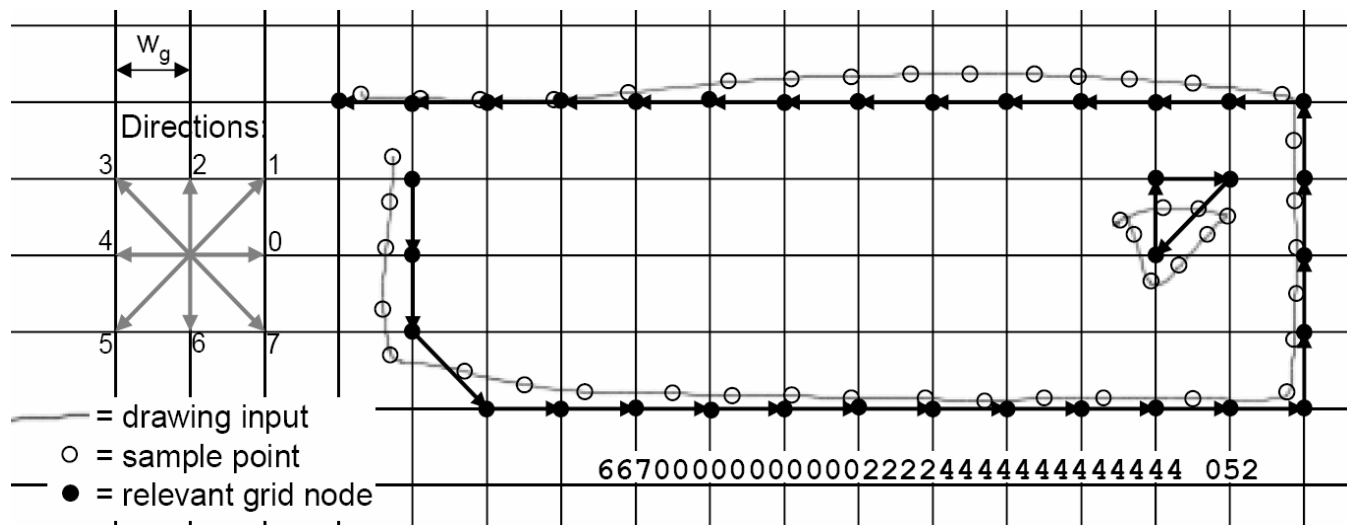
Две строки сравниваются с помощью расстояния Левенштейна, веса можно задать с помощью евклидова расстояния между точками.

Grid Star Algorithm

Одним из алгоритмов, реализованных в данном проекте, был алгоритм, основанный на комбинации построения сетки и идеи, основанной на направлениях жеста.

Сначала на жест накладывается решетка, но, в отличие от предыдущего алгоритма, на этот раз фиксирован размер квадратной ячейки, а не количество линий. Каждой точке жеста ставится в соответствие узел решётки, расположенный ближе остальных. После этого, аналогично строке точек, строится строка узлов решётки. Так как каждый штрих жеста непрерывен, то понятно, что для каждого

узла, не являющегося концом строки, можно указать следующий за ним один из восьми возможных узлов, таким образом определив для него одно из восьми направлений от одного узла к другому.



Конечная строка представляет из себя набор направлений из каждой ячейки в следующую. Метрика - нормализованное расстояние Левенштейна.

Особенностью алгоритма является его значительная зависимость от множества параметров:

1) Классификатор одного жеста будет сильно меняться при разном масштабе из-за фиксированного размера ячейки

2) Размеры ячейки

3) Расположение точек на жесте - если расстояние между точками будет значительно превышать размеры ячейки, то узлы будут слишком далеко друг от друга, в таком случае невозможно определить направление

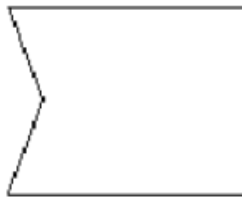
Решением проблемы будет приведение жестов к одному размеру, так, чтобы $\max(\text{height}, \text{width}) = 1000$. Экспериментально установлено,

что в таком случае максимальная точность распознавания жеста достигается при длине ячейки, равной 0.1 от максимального размера жеста, т.е. 100.

Проблема с точками решается добавлением новых точек там, где их слишком мало. Также удаляются близко расположенные точки, чтобы устранить возможные помехи.



Результаты тестирования алгоритма



pic 1 Accept Event Action



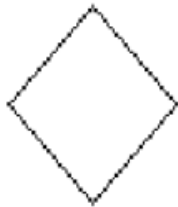
pic 2 Accept Time Event Action



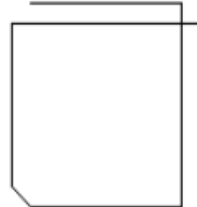
pic 3 Activity Partition



pic 4 Comment



pic 5 Decision Node



pic 6 Diagram



pic 7 Initial Node



pic 8 Input Pin



pic 9 Send Signal Node

Рис.1	Рис.2	Рис.3	Рис.4	Рис.5	Рис.6	Рис.7	Рис.8	Рис.9	Всего
108	120	118	119	147	93	138	122	110	1075 из 1177

Метод к ближайших соседей

Следующий алгоритм, реализованный в данном проекте, отличается от остальных алгоритмов тем, что является не классификатором, а методом распознавания, то есть вообще в него можно подставлять любой классификатор.

Для работы алгоритма необходима обучающая выборка. В этом алгоритме не нужно искать ближайший идеальный жест для пользовательского жеста, вместо этого ищутся его k ближайших соседей среди жестов обучающей выборки. Результатом распознавания является объект, к которому относятся большинство из k соседей.

Для тестирования алгоритма были использованы разные классификаторы, для каждого из них наблюдалось улучшение точности распознавания.

1) Распознавание пользователем

Рис.1	Рис.2	Рис.3	Рис.4	Рис.5	Рис.6	Рис.7	Рис.8	Рис.9	После	До
115	142	133	128	147	122	147	122	121	1177	1177

2) Расст. Левенштейна + сорт. по координатам

106	81	133	95	109	101	99	58	85	867	492
-----	----	-----	----	-----	-----	----	----	----	-----	-----

3) Количество ячеек в прямоугольнике

91	99	133	92	86	56	123	54	44	778	685
----	----	-----	----	----	----	-----	----	----	-----	-----

4) Количество ячеек в полосе

96	140	130	114	143	65	130	47	32	897	647
----	-----	-----	-----	-----	----	-----	----	----	-----	-----

5) Grid Star

107	129	128	125	146	106	132	122	111	1106	1075
-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------

Заключение

Были релизованы новые алгоритмы распознавания жестов, проведены тесты на точность распознавания. Полученные результаты показывают, что эти алгоритмы достаточно эффективны, точность распознавания порядка 93-94%.

Таким образом, становится возможным улучшение системы распознавания жестов в QReal с помощью этих алгоритмов.

Литература

1. Многоштриховые жесты в проекте QReal /Осечкина М.С. Курсовая работа, СПбГУ, Мат-мех, каф. системного программирования, 2011
2. Gesture recognition based on manifold learning, Heeyoul Choi, Brandon Paulson, Tracy Hammond