

# Линейно – контурное представление бинарных изображений для анализа видео данных

А.Соболев, Б.Чудинович, А.Бовырин, В.Ерухимов, В.Писаревский

E-mail: sobolev@msu.nstl.nnov.ru, {boris.chudinovich, alexander.bovyrin, victor.eruhimov, vadim.pisarevsky}@intel.com

## Аннотация

В работе представлен алгоритм, позволяющий на основе диаграммы Вороного (или скелетона) связной фигуры построить её линейно-контурную модель, сегментирующую фигуру на «широкие» части, задаваемые контурами, и «узкие» части, задаваемые участками диаграммы Вороного. Алгоритм свободен от многих эвристик, применяемых в предыдущих алгоритмах. В конце статьи обсуждаются возможные приложения этого представления.

**Ключевые слова:** Диаграмма Вороного, скелетон, линейно-контурная модель.

## 1. Введение.

Скелетон как форма представления бинарных изображений широко распространен: он является достаточно популярным средством анализа изображения. Немаловажными достоинствами такого представления являются его компактность и простая структура. Сейчас известно множество алгоритмов скелетизации [1, 5, 6].

В [1] была предложена идея сегментации связной фигуры на растре на «длинные» (вытянутые) участки, которые должны быть представлены скелетоном и на «короткие», которые могут быть представлены контуром оставшейся части связной фигуры. Аналогичная идея была положена в основу линейно-контурной модели представления фигуры, предложенной в работе [2]. В ней вытянутые участки фигуры постоянной «толщины» с «длинной», превышающей на некоторый параметр «толщину» участка, представлены скелетоном участка, части фигуры, не удовлетворяющие критерию линейности, представлены собственно контуром этого участка и точками примыкания к скелетону. Построение линейно-контурной модели в этом алгоритме начинается с поиска «конца» линии и, если его нет, фигура разрезается до тех пор, пока не появится нужный кандидат. Весь алгоритм построен на применении значительного числа эвристик и параметров выбора участков фигуры. Аналогичные алгоритмы можно найти в [3, 4].

В данной работе представлен новый алгоритм построения линейно-контурной модели для произвольной фигуры. Среди его основных достоинств следует выделить:

1. Отсутствие эвристических параметров, что позволяет получать адекватные результаты для любых фигур.
2. Графовое представление модели, что позволяет применять различные графовые алгоритмы для идентификации и анализа деталей фигуры.
3. Использование диаграммы Вороного в качестве базиса модели.

В пункте 3 имеется в виду, в первую очередь, что в предлагаемом алгоритме наиболее трудоемкая часть построения

линейно-контурной модели выделена в отдельную задачу — построение диаграммы Вороного для данной фигуры. Это значительно облегчает реализацию нашего алгоритма, так как диаграммы Вороного всесторонне изучены и для их построения уже предложено большое число хороших алгоритмов (см. выше).

*Замечание:* Как известно, скелетон является подмножеством диаграммы Вороного. В нашем алгоритме мы опираемся на диаграмму Вороного только потому, что она, в каком-то смысле, является первичным объектом (по сравнению со скелетоном) и именно её строят наиболее популярные алгоритмы [5,6]. На самом деле в нашем алгоритме используется лишь информация, содержащаяся в скелетоне.

## 2. Основные определения и обозначения.

В статье используется общепринятая при изучении диаграмм Вороного терминология (см. [5]).

Основным объектом является *связная многоугольная фигура* (область в  $R^2$ )  $P$ , каждая связная компонента её границы  $\mathcal{P}$  — это простой многоугольник, заданный последовательностью своих вершин. *Сайтом* будем называть либо ребро фигуры  $P$ , либо её вогнутую вершину. Через  $S$  обозначим множество сайтов, задаваемое фигурой  $P$ . *Ячейкой Вороного* (ЯВ) сайта  $s \in S$  называется множество точек фигуры  $P$ , для которых этот сайт является ближайшим. *Диаграммой Вороного* (ДВ) совокупности сайтов  $S$  называется множество всех граничных точек ячеек Вороного сайтов из  $S$ . Граница ЯВ произвольного сайта представляет собой последовательность отрезков прямых и парабол, которые мы будем называть *ребрами ЯВ*, одновременно они также являются ребрами ДВ. Ребра ЯВ данного сайта мы будем называть инцидентными этому сайту, соответственно сайт мы будем называть инцидентным ребру ДВ, если это ребро инцидентно сайту. Ребро ДВ будем называть *внутренним*, если оно не имеет общих

точек с инцидентными ему сайтами, в противном случае ребро будем называть *граничным*. Два сайта, инцидентные одному ребру, будем называть *смежными*. Граничные точки ребер ДВ будем называть *узлами ДВ*.

Фигура  $P$  считается ориентированной против часовой стрелки, такая же ориентация задается для каждой ЯВ. На рис. 1 серым цветом выделена ЯВ сайта  $S_2$ , ребра этой ЯВ обозначены через  $E_1, E_2, E_3$ . Черные «жирные» точки — узлы ДВ, стрелки показывают ориентацию фигуры и ЯВ.

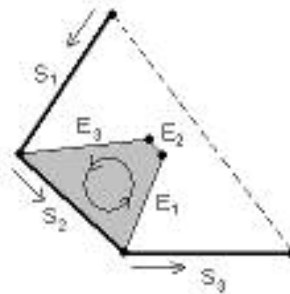


Рис.1

### 3. Построение линейно-контурной модели.

Предлагаемый алгоритм построения линейно-контурной модели (ЛКМ) для связной (многоугольной) фигуры имеет ряд преимуществ по сравнению с ранее известными ([2] – [4]): простота реализации, высокое быстродействие, представление результата в виде планарного неориентированного графа.

Фактически алгоритм состоит из двух частей: построение размеченной диаграммы Вороного (ДВ) для связной фигуры и восстановление ЛКМ из ДВ. Сложность алгоритма равна  $O(n \log n)$ , причем асимптотика определяется сложностью алгоритма построения ДВ, сложность восстановления ЛКМ из ДВ равна  $O(n)$ , здесь  $n$  — количество вершин фигуры.

Для построения ДВ можно использовать любой из общеизвестных алгоритмов (см., например, [5,6]). Специальные требования предъявляются только к реализации, а именно: математическая структура, представляющая ДВ, должна обеспечивать следующие взаимосвязи между объектами:

1. Переход от сайта к последующему и предыдущему сайтам (порядок сайтов задается ориентацией фигуры).
2. Переход от сайта к первому и последнему ребрам ЯВ этого сайта (то есть к граничным ребрам ЯВ). Порядок ребер задается ориентацией ЯВ. Например, на рис. 1  $E_1$  — первое ребро,  $E_3$  — последнее ребро.
3. Переход от каждого ребра к двум инцидентным ему сайтам.
4. Переход от каждого ребра к последующему и предыдущему ребрам в каждой из двух ЯВ, общей границей которых он является.

#### Представление ЛКМ в виде графа.

Как уже отмечалось выше, ЛКМ в предлагаемом алгоритме представляется в виде графа, так как такое представление позволяет эффективно обрабатывать построенную ЛКМ.

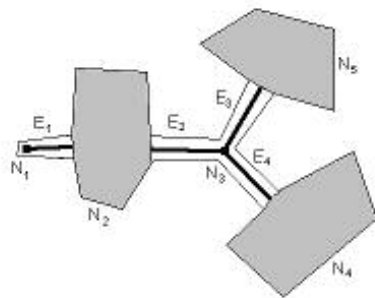


Рис. 2

Если фигура многосвязна, то в графе ЛКМ будут циклы, в частности возможны петли. Для простоты мы рассмотрим сначала случай, когда в фигуре отсутствуют «узкие кольца»; случай «узкого кольца» во

многом является исключительным, он предполагает незначительную модификацию в определении графа и будет рассмотрен отдельно.

Граф ЛКМ состоит из вершин и ребер ЛКМ (чтобы отличать их от ребер ДВ), имеющих внутреннюю структуру. Мы будем выделять два типа вершин: простые и сложные. Простая вершина состоит из одной точки, которая является узлом ДВ, инцидентным одному или трем и более внутренним ребрам ДВ. Сложная вершина состоит из трех и более

точек, часть из них является вершинами границы фигуры  $\mathcal{P}$ , а оставшиеся точки — это принадлежащие ДВ точки примыкания «узкой» и «широкой» частей фигуры и их ортогональные проекции на  $\mathcal{P}$ . Фактически каждая сложная вершина представляет собой многоугольную фигуру, определяемую упорядоченной последовательностью точек — вершин этой фигуры, которая содержит некоторую «широкую» часть исходной фигуры. Ребро ЛКМ представляет собой упорядоченную последовательность из двух и более точек, причем все точки кроме первой и последней являются узлами ДВ, инцидентными в точности двум внутренним ребрам ДВ, а отрезки (или параболы) между парами соседних точек являются внутренними ребрами ДВ. Первая (и последняя) точка ребра ЛКМ либо совпадает с простой вершиной (если соответствующая вершина простая), либо совпадает с одной из точек последовательности, задающих сложную вершину (если соответствующая вершина сложная). Ребро ЛКМ описывает «узкую» часть фигуры, для каждого ребра ЛКМ алгоритмом определяется средняя ширина соответствующего ему участка фигуры.

Мы будем считать, что граф ЛКМ для «узкого кольца» состоит из единственного замкнутого ребра ЛКМ (у которого нет ни первой, ни последней точек в смысле данного выше определения).

Пример приведен на рис. 2. Здесь  $N_1$  и  $N_3$  — простые вершины,  $N_2$ ,  $N_4$  и  $N_5$  — сложные вершины,  $E_1$ ,  $E_2$ ,  $E_3$ ,  $E_4$  — ребра ЛКМ (жирные линии). Контур фигуры обведен тонкой линией, выделенные серым области представляют собой «широкие» части фигуры, ребра ЛКМ соответствуют «узким» частям.

#### Выделение «узких» и «широких» частей.

Разделение заданной фигуры  $P$  на «узкие» и «широкие» части осуществляется исходя из заданного действительного параметра  $D$ . Нестрого говоря, мы считаем часть фигуры

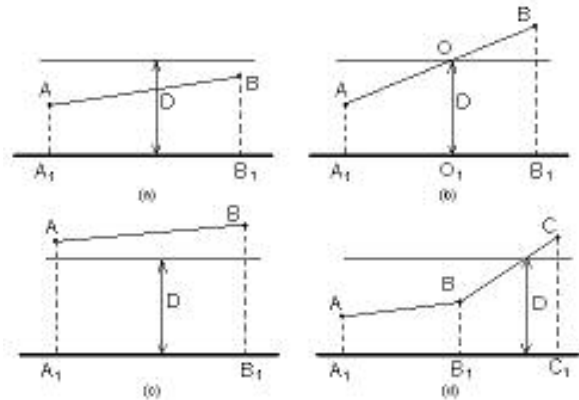


Рис. 3

«узкой», если её ширина нигде не превосходит  $2D$ . Это естественное утверждение нуждается в формализации, которая описывается далее.

Пусть  $AB$  — внутреннее ребро, инцидентное двум сайтам-отрезкам (см. рис. 3).  $A_1$  и  $B_1$  — ортогональные проекции  $A$  и  $B$  на один инцидентный сайт (он показан жирной линией),  $A_2$  и  $B_2$  — ортогональные проекции  $A$  и  $B$  на другой инцидентный сайт (на рисунке не показаны).

В случае 3(а), когда расстояния обоих концов ребра ДВ до сайта меньше  $D$ , часть фигуры  $AA_1B_1BB_2A_2$  считается «уз-

кой», то есть точки  $A$  и  $B$  принадлежат некоторому ребру ЛКМ. В этом случае мы говорим, что ребро ДВ  $AB$  и его проекции  $A_1B_1$  и  $A_2B_2$  принадлежат «узкой» части фигуры.

В случаях 3(b) и 3(c), когда хотя бы одно из этих расстояний больше или равно  $D$ , часть фигуры  $AA_1B_1BB_2A_2$  считается «широкой», при этом отрезки  $A_1B_1$  и  $A_2B_2$  принадлежат границе сложной вершины. В этом случае мы говорим, что ребро ДВ  $AB$  и его проекции  $A_1B_1$  и  $A_2B_2$  принадлежат «широкой» части фигуры.

На рис. 3(d) изображен случай примыкания «узкой» —  $AA_1B_1BB_2A_2$  и «широкой» —  $CC_1B_1BB_2C_2$  частей фигуры. Примыкание происходит в точке  $B$ , которая является начальной (или конечной) точкой некоторого ребра ЛКМ, при этом последовательность точек, определяющих сложную вершину, содержит точки  $B_2, B, B_1$ .

Случай, когда одним из инцидентных сайтов для данного ребра ДВ является сайт-точка, обрабатывается аналогично (см. рис. 4). При этом проекцией любой точки на сайт является сам сайт-точка. Заметим, что этот случай содержит ещё

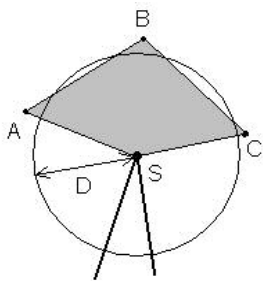


Рис. 4

один возможный вариант расположения сайта и ребра ДВ: расстояние от сайта до концевых точек ребра ДВ превосходит  $D$ , а для некоторых других точек ребра это расстояние меньше  $D$  (ребра  $AB$  и  $BC$  на рис. 4). Фактически в этом случае мы жертвуем некоторой точностью разделения фигуры на «узкие» и «широкие» части, получая при этом более

простой алгоритм. На самом деле точностью мы пожертвовали и в случае 3(b), где более точным было бы отнесение  $AA_1O_1OO_2A_2$  к «узкой» части и  $OO_1B_1BB_2O_2$  к «широкой» части. Практика показала, что такой упрощенный алгоритм в ряде случаев дает даже более предпочтительные визуальные результаты.

Для граничных ребер мы используем следующее правило: если хотя бы одно из соседних ребер принадлежит «широкой» части фигуры, то и само ребро и его проекции на инцидентные сайты принадлежат «широкой» части, в противном случае они принадлежат «узкой» части. Граничные ребра, принадлежащие «узкой» части игнорируются (т.е. не принадлежат никакому ребру ЛКМ), если они принадлежат «широкой» части, то их проекции на инцидентные сайты входят в некоторую сложную вершину ЛКМ.

*Алгоритм восстановления ЛКМ из ДВ.*

Входящими данными для алгоритма служат ДВ, удовлетворяющая описанным выше требованиям 1 – 4, и действительное число  $D$ . Сначала мы рассмотрим этот алгоритм для простейшего случая, когда заданная фигура является односвязной областью.

Схема алгоритма представлена на рис. 5, каждый блок



Рис. 5

реализуется одной функцией.

Алгоритм начинается с поиска начальной вершины, которой может быть любая сложная вершина или простая вершина, задаваемая узлом ДВ, инцидентным в точности одному внутреннему ребру. Поиск осуществляется путем последовательного перебора всех сайтов и исследования предпоследних ребер ЯВ этих сайтов. Если ребро внутреннее и принадлежит «широкой» части фигуры, то найдена сложная вершина, если ребро внутреннее, принадлежит «узкой» части и конечная точка ребра инцидентна в точности одному ребру ДВ, то найдена простая вершина. В силу односвязности фигуры начальная вершина найдется, если существует хотя бы одно внутреннее ребро ДВ. Внутренние ребра отсутствуют, если заданная фигура является треугольником или правильным  $n$  – угольником, этот простой случай следует рассмотреть отдельно.

Входящими параметрами функции построения сложной вершины являются сайт и инцидентное ему ребро, принадлежащее «широкой» части. Мы последовательно перебираем все сайты и для каждого сайта перебираем ребра его ЯВ (начиная с предпоследнего) до тех пор, пока не вернемся к начальному сайту и ребру. Если в процессе перебора мы найдем ребро ДВ, принадлежащее «узкой» части, то вызываем функцию построения ребра ЛКМ, затем переходим к смежному сайту и продолжаем перебор.

Входящими параметрами функции построения простой вершины являются сайт, инцидентное ему ребро, принадлежащее «узкой» части фигуры и узел, задающий простую вершину. Так как простая вершина состоит из одной точки, то её построение состоит в нахождении всех ребер ЛКМ, выходящих из этой вершины. Для этого необходимо найти все внутренние ребра ДВ, инцидентные заданному узлу, и для каждого из них (кроме заданного) вызвать функцию построения ребра ЛКМ.

Входящими параметрами функции построения ребра ЛКМ являются сайт, инцидентное ему ребро, принадлежащее «узкой» части фигуры и точка примыкания. Мы последовательно перебираем все сайты и для каждого сайта перебираем ребра его ЯВ (начиная с предпоследнего). Перебор осуществляется до тех пор, пока очередное ребро ДВ не будет принадлежать «широкой» части, в этом случае вызывается функция построения сложной вершины, или пока начальная точка ребра ДВ не будет инцидентна одному или трем и более ребрам ДВ, в этом случае вызывается функция построения простой вершины.

Таким образом, предлагаемый алгоритм является по существу рекурсивным, односвязность фигуры гарантирует его корректность. Максимальная глубина рекурсии в системе как правило ограничена, однако эта сложность легко преодолевается. Заметим сначала, что для фигуры, имеющей  $n$  вершин, в наихудшем случае глубина рекурсии не превысит  $n$ , в среднем глубина рекурсии имеет порядок  $O(\log n)$ . Так как в подавляющем большинстве случаев для увеличения быстродействия и получения более приемлемых результатов изображения предварительно подвергают сглаживанию, то допустимой в системе максимальной глубины рекурсии обычно достаточно. Для получения универсальной программы можно, например, предусмотреть в каждой функции дополнительный параметр — глубину рекурсии, и при достижении им определенного значения не вызывать новую функцию, а запоминать параметры её вызова в отдельный массив. После построения

первой части графа ЛКМ вызывается каждая отложенная функция, и новая часть графа ЛКМ склеивается с ранее построенной.

Описанный алгоритм восстановления ЛКМ из ДВ отличается не только низкой асимптотической сложностью  $O(n)$ , но и в целом является оптимальным по числу операций. Нетрудно убедиться, что алгоритм является «полуторопроходным», то есть каждое ребро ДВ, принадлежащее «широкой» части фигуры, проходит дважды, а принадлежащее «узкой» части фигуры — только один раз.

*Обобщение на случай многосвязных фигур.*

После доработки алгоритм может быть применен и к произвольной многоугольной фигуре. Главная задача, которую для этого необходимо решить, состоит в быстром определении моментов замыкания циклов. Это нетрудно сделать, если учесть, что циклы образуют только те ребра ДВ, у которых инцидентные сайты принадлежат различным компонентам связности границы фигуры. Каждой паре компонент связности поставим в соответствие ребро-метку — ребро ДВ, инцидентные сайты которого принадлежат этим компонентам. Ребром-меткой для каждой пары компонент связности становится первое подходящее ребро ДВ; совпадение текущего обрабатываемого ребра ДВ с инициализированным ранее ребром-меткой свидетельствует о замыкании некоторого цикла.

Использование описанного выше механизма ребер-меток позволяет реализовать обобщенный алгоритм по прежней рекурсивной схеме (рис. 5). Функции поиска начальной вершины, построения простой вершины и ребра ЛКМ претерпевают минимальные изменения. Так, в функции поиска начальной вершины добавляется блок обработки ещё одного исключительного случая — «узкого кольца», в функцию построения ребра ЛКМ добавляется блок обработки ребер-меток. Существенные изменения необходимо внести только в функцию построения сложной вершины. В случае многосвязной фигуры здесь возникает ряд специфических трудностей, мы проиллюстрируем их на следующем примере.

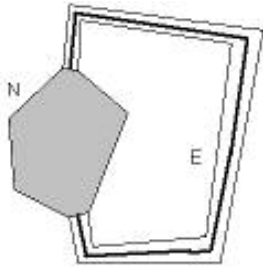


Рис. 6

На рис. 6 изображен граф ЛКМ, состоящий из сложной вершины  $N$  и ребра ЛКМ  $E$ . Если применить к этой фигуре исходный алгоритм, то он будет выполнять следующую последовательность действий: 1. нахождение начальной вершины  $N$ ; 2. построение части вершины  $N$ ; 3. построение ребра ЛКМ  $E$ ; 4. построение части вершины  $N$ ; ... Использование механизма ребер-меток позволит избежать закливания, сложность здесь состоит в том, что на этапах 2 и 4 будут строиться различные вершины.

Таким образом, при использовании исходного алгоритма в случае многосвязной фигуры типичной является ситуация, когда одна сложная вершина строится различными функциями (на самом деле функция одна, но вызывается на разных глубинах рекурсии), как следствие возникает задача отождествления всех этих вершин. По-видимому, наиболее эффективным её решением является недопущение этой ситуации,

для чего в функции построения сложной вершины можно использовать механизм отложенных функций (который уже предлагалось использовать для контроля глубины рекурсии). Он состоит в том, что параметры функций построения ребер ЛКМ, для которых эта вершина является начальной, запоминаются, а сами функции вызываются только после построения всей вершины. При этом все те начальные ребра ДВ этих ребер ЛКМ, которые инцидентны различным компонентам связности границы фигуры, становятся ребрами-метками, тем самым возможность повторного построения этой вершины исключается.

Описанный обобщенный алгоритм восстановления ЛКМ из ДВ отличается от исходного, главным образом, использованием ребер-меток. Нетрудно убедиться, что, если количество ребер-меток не превосходит числа  $p$ , то сложность обобщенного алгоритма будет превосходить сложность исходного не больше, чем в  $p$  раз. Минимизировать сложность можно путем более эффективной работы с ребрами-метками. Наиболее кардинальное решение этой задачи предполагает наложение дополнительных требований на структуру, представляющую ДВ: наличие индексации для компонент связности границы фигуры и указание для каждого сайта индекса компоненты связности, которой он принадлежит. При этом, предполагая, что ребра-метки рассредоточены равномерно по компонентам связности, мы сразу уменьшаем количество операций в среднем в  $m(m-1)/2$  раз (здесь  $m$  — число компонент связности). Также количество ребер-меток можно уменьшать, удаляя уже ненужные. Критерий «ненужности» определяется деталями реализации алгоритма, которые здесь не обсуждались, однако можно добиться такой реализации, при которой ненужным становится один раз использованное ребро-метка.

#### 4. Приложения ЛКМ.

В этом параграфе представлены некоторые актуальные задачи, для решения которых может быть с успехом применён описанный выше алгоритм построения ЛКМ.

##### *Дефектоскопия при производстве печатных плат.*

Оптический метод тестирования заключается в визуальной проверке качества изготовления печатных плат и предупреждает выявление таких дефектов, как нарушение топо-

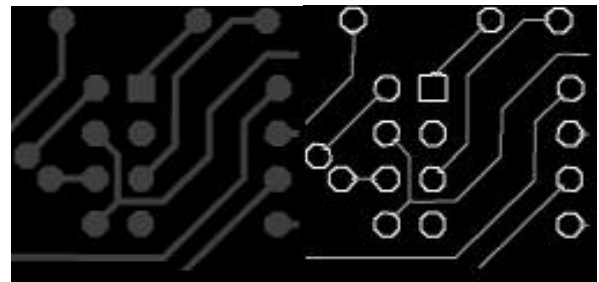


Рис. 7

логии схемы, наличие неоднородностей и посторонних включений, обрывы дорожек и т. д. Сейчас, в основном, визуальный контроль проводится высококвалифицированным оператором с помощью специального дорогостоящего оптического оборудования. Трудоемкость этой операции можно значительно снизить путём применения различных алгоритмов машинного зрения для автоматического поиска дефектов и

отклонений от эталона. Так как ширина дорожек на печатной плате обычно меньше её узловых частей и известна априори, то целесообразность построения ЛКМ становится очевидной. С помощью кусочно-линейной аппроксимации построенной модели нетрудно получить прямолинейные участки дорожек для дальнейшего их сравнения с эталонными.

#### Распознавание жестов.

Как известно, сейчас ведутся обширные исследования в области повышения эффективности человеко-машинного интерфейса. Для поддержки «нестандартных» каналов ввода информации используется широкий спектр аппаратуры. Хорошо зарекомендовали себя так называемые «виртуальные перчатки», которые с помощью нескольких десятков датчиков позволяют отслеживать положение кисти руки в пространстве и регистрировать движение каждого пальца. Подобные устройства достаточно эффективны, а среди их недостатков можно отметить высокую стоимость и некоторое неудобство в использовании: пользователь не всегда комфортно чувствует себя, окутанный проводами. Поэтому сейчас активно разрабатываются алгоритмы отслеживания 3D конфигурации кисти на данных, полученных с одной и более камер или с так называемых дальномеров (активные измерения). Для успешного распознавания жестов необходимо находить характерные элементы кисти – пальцы, ладонь. Для

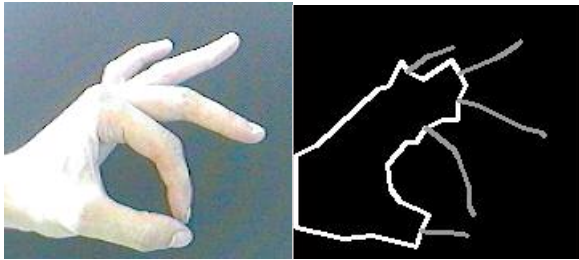


Рис. 8

этого очень удобно применять ЛКМ, при этом пальцы будут линейной частью модели. Таким образом, можно легко определить их направление, конфигурацию и характеристики движения (см. рис 8).

#### Видеонаблюдение.

В последнее время цифровые системы видеонаблюдения активно доминируют на рынке средств обеспечения безопасности по сравнению с аналоговым оборудованием видеоконтроля. Одним из ключевых преимуществ цифровых систем является возможность автоматической



Рис. 9

обработки видео информации и, следовательно, более качественный мониторинг событий. В большинстве таких систем оператор оповещается сигналом тревоги, если средняя интенсивность определённой области изображения резко измени-

лась. Такой незамысловатый алгоритм приводит к частым ложным срабатываниям системы. Так как фигура человека имеет характерные узкие и широкие участки (туловище, руки, ноги, шея, голова), то анализ ЛКМ может помочь детектировать и идентифицировать человека. Кроме того, предлагаемый алгоритм может помочь при решении такой безусловно сложной и актуальной задачи, как распознавание поведения человека.

#### 5. Заключение.

Очевидно, что применение ЛКМ может существенно повысить надёжность алгоритмов, применяемых при оцифровке карт, медицинской диагностики, определения положения различных устройств на конвейере, оптического распознавания символов и т.д. ЛКМ допускает простые обобщения на основе анализа поведения расстояния от скелетона до границы фигуры как функции длины границы, что позволяет построить различные новые типы декомпозиции фигуры. Однако до сих пор ЛКМ не получили широкого распространения, что связано, по-видимому, с их близостью к диаграммам Вороного (действительно, скелетон является частным случаем ЛКМ, соответствующем параметру  $D = 0$ ). Последние, несмотря на наличие глубокой теории и очевидные выгоды от своего применения, также используются недостаточно активно (здесь речь идет о геометрическом подходе, а не о растровом), так как задача написания *надёжной и универсальной* программы скелетизации (по любому из известных алгоритмов) очень сложна.

#### Список литературы:

- [1] Rosenfeld A., Pfaltz J., Sequential operations in digital picture processing. 1966. Journ. ACM, v.13, n.4. 1966.p. 471-494.
- [2] Васин Ю.Г., Башкиров О.А., Рудометова С.Б. Математические модели структурированного описания графических изображений. Автоматизация обработки сложной графической информации. Межвуз. сб., ГГУ, Горький, 1984, с. 92-110
- [3] Shapiro, B. Pisa J., Sklansky J., Skeleton generation from x,y boundary sequences, CGIP, v.15, 1981, p.136-153.
- [4] Martinez – Perez M. P., Jimenez J., Navalon J. L. A Thinning Algorithm Based on Contours. CVGIP v.39, n. 3. 1987, p.186-201.
- [5] D.T. Lee «Medial Axis Transformation of a Planar Shape», IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4, No. 4, 363-369, July 1982
- [6] S. Fortune. A sweepline algorithm for Voronoi diagrams. Algorithmica, 2:153-174, 1987.