

- ностроение, автоматизация, ЭВМ. №4 (22) – Брест: БГТУ, 2003.– С. 47-54.
- Головко В.А., Маньяков Н.В. Матричный нейросетевой метод обучения многослойной сети с использование адаптивного шага // Вестник Брестского государственного технического университета. Физика, математика, химия. №5 (23) – Брест: БГТУ, 2003.– С. 56-58.
 - Гладкий И.И., Маньяков Н.В., Махнист Л.П. Об одном алгоритме обучения многослойных нейронных сетей на

основе условной оптимизации ошибки // Вестник Брестского государственного технического университета. Физика, математика, химия. №5 (23) – Брест: БГТУ, 2003.– С. 25-30.

- V. Golovko, N. Maniakov, L. Makhnist. Multilayer Neural Networks Training Methodic // Proceedings of the Workshop Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2003). – Lviv, Ukraine, September 8-10, 2003. – PP. 185-190.

УДК 681.328

Дудкин А.А., Вершок Д.А.

ПОСТРОЕНИЕ ВЕКТОРНОГО ОПИСАНИЯ ТОПОЛОГИИ ИНТЕГРАЛЬНОЙ СХЕМЫ

Введение

Интерес к разработке алгоритмических и программных средств обработки и анализа изображений в микроэлектронике проявился в середине восьмидесятых годов двадцатого века в связи с увеличением степени интеграции элементов на кристалле интегральной схемы (ИС), и как следствие – возникновением трудностей при обычном визуальном контроле производства ИС. С этого момента интенсивные исследования по этой теме ведутся во многих научных центрах и, практически, всеми крупными производителями микроэлектронных устройств, такими как Hitachi, Texas Instruments, Intel, Nippon и т.д. Следует заметить, что в настоящее время на основе разработанных алгоритмов создаются коммерческие системы компьютерного зрения, позволяющие решать простейшие задачи контроля производства фотошаблонов ИС и печатных плат.

В то же время анализ состояния проблемы показывает, что чаще всего разрабатываемые алгоритмы и методы направлены на решение одной прикладной задачи, поэтому значительные успехи достигнуты в разработке специализированных алгоритмов. В настоящее время практически не развивается подход, позволяющий из набора растровых изображений слоев микроэлектронного устройства сформировать его полноценное представление в векторном виде, что в практическом плане дает возможность решать совершенно различные задачи из области микроэлектроники.

Создание такой технологии требует разработки связанного комплекса алгоритмов обработки изображений микроэлектронных устройств, начиная с этапа предварительной обработки и выделения характерных признаков и заканчивая этапом идентификации.

Изображения топологических слоев интегральной схемы, подготовленные для обработки в системах технического зрения, как правило, представляются совокупностью кадров с некоторым перекрытием соседних кадров. В связи с этим, для построения векторного описания топологии ИС необходимо сформировать единое изображение, т.е. выполнить сшивку частично перекрывающихся кадров [1]. Основной задачей при этом является определение оптимального совмещения соседних кадров в области перекрытия.

В настоящей статье мы рассмотрим задачу формирования векторного описания топологии ИС для отдельного кадра, а также построения из этих описаний общего векторного описания (ОВО) послойного представления всей топологии ИС.

1. Формирование векторного описания одного кадра

Дудкин А.А., ведущий научный сотрудник ОИПИ НАН Беларуси.
Вершок Д.А., научный сотрудник ОИПИ НАН Беларуси.

Полагаем, что отдельные кадры после предварительной обработки уже подверглись процедуре выделения контуров. В статье предлагается метод аппроксимации выделенных контуров, основанный на преобразовании Хафа (ПХ) [2]. В результате формируется векторное представление изображения в виде набора областей. Основными этапами данного алгоритма являются:

- вычисление ПХ и выделение максимумов в пространстве ПХ;
- поиск точек пересечения прямых, расположенных в поле изображения;
- поиск элементарных областей и покрытие ими изображения;
- объединение элементарных областей.

1.1. Вычисление ПХ и выделение максимумов

Преобразование Хафа выполняется с целью выделения на контурном изображении прямых линий. Используется $\rho\theta$ - преобразование Хафа [2]. Существует возможность вычислять ПХ исходя из предварительной информации об обрабатываемых изображениях. Например, на изображениях слоев интегральных микросхем металлические проводники ориентированы строго под углами 0° , 45° , 90° , 135° относительно оси X , а топологические расстояния между ними не могут быть меньше определенного порога. Поэтому для повышения быстродействия алгоритма и точности результатов сегментации имеет смысл вычислять ПХ только для указанных углов. Это особенно полезно при аппроксимации контуров прямолинейными сегментами, с ограничениями по их ориентации и длине.

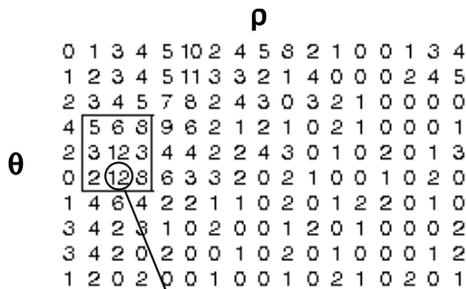
Особенностью этого этапа является также то, что исходное изображение может разбиваться на n частей, где $n = 1 \dots N$, и ПХ вычисляется для каждой части изображения отдельно. В результате получается n пространств ПХ, максимумы в которых соответствуют прямым линиям из отдельных частей изображения. Предварительные эксперименты показали, что увеличение количества пикселей изображения в n раз, увеличивает время работы алгоритма на последующих этапах приблизительно в e^n раз. Поэтому такой подход применен с целью снижения вычислительной нагрузки на последующих этапах. Каким образом достигается уменьшение вычислительной сложности, будет пояснено ниже.

Кроме того, для выделения максимумов требуется выбрать величину порога – T , что является нетривиальной зада-

чей, требующей определенного опыта, а также информации о характере и размере изображения. Разбиение можно проводить таким образом, чтобы размеры получившихся частей оставались приблизительно одинаковыми для любых размеров исходных изображений, это позволяет использовать одно фиксированное значение порога T .

Для выделения максимумов в алгоритме использовались следующие подходы:

1. Поиск абсолютного максимального значения, превышающего пороговое в некоторой локальной окрестности (рис. 1).



Выбранное максимальное значение в окне, превышающее пороговое $T=7$

Рис. 1.

Суть его состоит в поиске максимального значения, которое превышает predetermined порог T . Размер окна может выбираться любым, но не больше размера массива накопителя. В случае, если в окне имеется несколько одинаковых максимальных значений, выбирается значение, расположенное на пересечении столбца и строки, сумма значений в которых максимальна.

2. Поиск абсолютного максимального значения пространственной производной второго порядка, превышающего пороговое, в некоторой локальной окрестности. Подход аналогичен предыдущему, за исключением предварительного вычисления пространственной производной второго порядка в каждой точке массива накопителя посредством оператора Лапласа с маской размером 3×3 элемента, в которой учитывается значение четырех соседей.

Размер окна, в котором выделяется один максимум, соответствующий одной прямой, выбирается исходя из требований по точности аппроксимации контуров прямолинейными сегментами.

Результат выделения максимумов и соответствующих им прямых показан на рис. 2. Для наглядности найденные прямые наложены на исходное изображение.

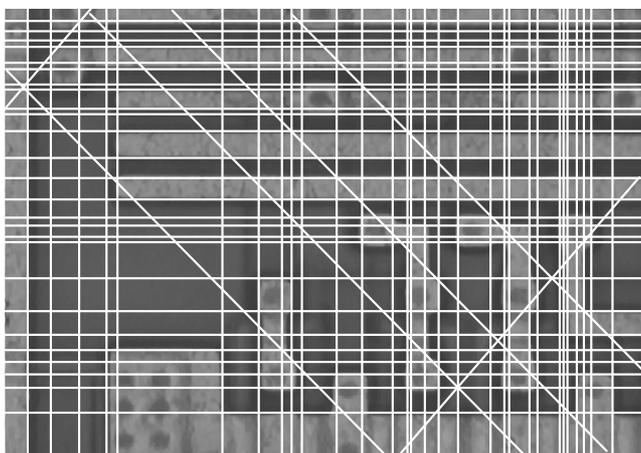


Рис. 2.

1.2. Поиск точек пересечения прямых

На этом этапе находятся точки пересечения всех прямых в пространстве исходного изображения путем решения следующей системы уравнений для всех пар прямых:

$$x \cos \theta_1 + y \sin \theta_1 = \rho_1$$

$$x \cos \theta_2 + y \sin \theta_2 = \rho_2,$$

где ρ_1, θ_1 – параметры первой прямой;

ρ_2, θ_2 – параметры второй прямой.

Найденные точки являются кандидатами на точки, в которых направление контура, описывающего объект, меняет направление. Координаты точек пересечения округляются до целочисленных значений и, в результате, соответствуют реальным пикселям на изображении. На следующем этапе точки пересечения связываются отрезками в так называемые элементарные контуры. В результате все исходное изображение покрывается набором этих контуров, далее каждому элементарному контуру по одному из критериев присваивается цвет (объект/фон).

1.3. Поиск элементарных областей и покрытие ими изображения.

Задан плоский неориентированный граф $G = (N, R)$, соответствующий изображению, где: $N = \{n_j, j = \overline{1, n}\}$ – множество вершин графа, соответствующих точкам пересечения прямых, а $R = \{r_i, i = \overline{1, r}\}$ – множество ребер, соответствующих отрезкам, соединяющим на изображении соседние на прямой точки.

Для каждого ребра r_i задан признак e_i , является ли ребро внешним или внутренним в графе G : $e_i = 1$, если ребро внешнее; $e_i = 0$, если ребро внутреннее.

Для каждой вершины n_j графа G заданы: (x_j, y_j) – координата вершины; a_j – количество соседних к ней вершин.

Элементарной области соответствует на графе простой цикл минимальной длины. Задача построения множества элементарных областей заключается в поиске на графе всех простых циклов минимальной длины и покрытия ими графа. Она решается путем отыскания и последовательного удаления всех простых циклов минимальной длины с последующей модификацией графа.

Обозначим: $G' = (N', R')$ – текущий граф, $N(S)$ – множество всех вершин, соседних с вершинами из множества S текущего графа, $v(S)$ – мощность множества S ,

$$N(S) = \bigcup_{i=1}^{v(S)} A(s_i)$$

– множество всех вершин, соседних с S_i вершин.

Алгоритм поиска элементарных областей.

Шаг 0. Полагаем $G' = G$.

Шаг 1. Полагаем $k = 1$. Выбираем из графа G' внешнее ($e_i=1$) ребро $r_i = (n_{i1}, n_{i2}) \in R'$, вершину n_{i2} этого ребра помещаем во множество соседей первого уровня дерева решений $N_1 = \{n_{i2}\}$.

Шаг 2. Для $k = 2, 3, \dots$ строим множество

$$N_k = N(N_{k-1}) \cup_{p=1}^{k-1} N_p$$

(для каждой вершины, находящейся на уровне $k-1$, находятся соседние с ней вершины). Если при построении N_k очередная найденная вершина совпадает с n_{i1} , то простой цикл длины k сформирован.

Шаг 3. Двигаясь по дереву решений снизу-вверх от вершины n_{i1} , находящейся на уровне k , к вершине n_{i2} , находящейся на первом уровне дерева, формируем последовательность вершин $C = \{n_k, n_{k-1}, \dots, n_1\}$, где $n_k = n_{i1}$, $n_1 = n_{i2}$, которая определяет простой цикл минимальной длины (для $n_i \in N_j$ известно соответствующее множество $A(n_i^*) \ni n_i$). Одновременно проводим корректировку графа G' : для всех вершин из C соответствующие значения a_j уменьшаем на единицу; внешние ребра из R' удаляются; определяемые парами вершин внутренние ребра помечаются в R' как внешние; изолированные вершины удаляются из N' . Если граф G' не пуст, то перейти на **Шаг 1**, иначе закончить алгоритм.

В приведенном алгоритме основным этапом является построение цикла длины k для некоторой вершины n . При этом, возможно определять простые циклы и для других вершин. Для этого при выполнении операции объединения множеств на втором шаге алгоритма достаточно отмечать вершины, присутствующие на предыдущих уровнях дерева решений.

Далее каждой i -ой элементарной области присваивается цвет Col_i . Для этого могут применяться различные критерии определения цвета. В алгоритме опробовано два критерия. Оба критерия основаны на анализе интенсивности пикселей на изображении в месте расположения элементарной области, для которой определяется цвет.

Критерий №1 определяется отношением (2):

$$Col_i = \left\lceil \sum_{j=1}^m c_j / m \right\rceil, \quad (2)$$

где c_j – цвет j пиксела изображения, лежащего внутри области; m – количество пикселей, покрываемых i -м элементарным контуром;

$\lceil \rceil$ – округление к ближайшему целому.

В итоге, элементарная область получает среднеарифметический цвет входящих в нее пикселей.

Критерий №2 определяется отношением (3):

$$Col_i = \max \{n_k\}, \quad k = 0 \dots N, \quad (3)$$

где n_k – количество точек цвета k , покрываемых элементарной областью i .

Этот критерий присваивает элементарной области тот цвет, который соответствует максимуму гистограммы интенсивностей, построенной на данной элементарной области.

На этом этапе могут быть использованы и другие подходы определения цвета. Например, основанные на анализе текстур [3] или спектральных характеристик изображения в этой области. Пример раскраски найденных элементарных областей показан на рис. 3.

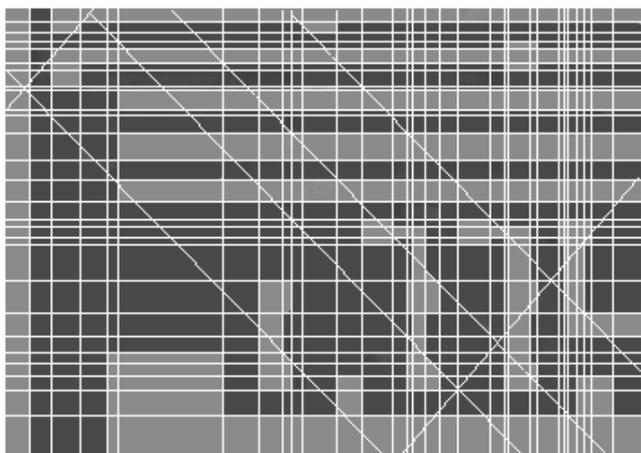


Рис. 3.

1.4. Объединение элементарных областей

Задачей объединения элементарных областей является получение нескольких, как можно больших по площади областей из набора элементарных областей одного цвета. Например, области i и $i+1$, имеющие общие ребра, и для которых выполняется соотношение $|Col_i - Col_{i+1}| \leq T$, где Col – цвет области, T – пороговое значение, объединяются в одну область. При этом общие ребра удаляются, а новой области присваивается цвет той области, которая имела большую площадь (покрывала большее количество пикселей на изображении).

При большом количестве элементарных областей их объединение становится самым трудоемким этапом с вычислительной точки зрения. Действительно, если имеется N – элементарных областей подобного цвета, состоящих в среднем из M ребер, тогда в худшем случае только для поиска двух соседних областей необходимо произвести $M \cdot (N - 1) \cdot M$ сравнений ребер.

Выход из этой ситуации может состоять, во-первых, в построении упорядоченных структур для хранения областей, ускоряющих поиск нужных ребер, и применении специальных быстрых алгоритмов поиска, а, во-вторых, в уменьшении количества ребер. Такое уменьшение достигается за счет разбиения изображения на части и выполнения всех этапов алгоритма сегментации для каждой части в отдельности. Разбиение изображения на части приводит к тому, что отдельные короткие фрагменты контуров изолируются в своей части изображения. Тогда и прямые, проходящие через эти контура, ограничиваются только своей частью изображения и не порождают дополнительных точек пересечения в других частях.

Результатом работы алгоритма является набор областей. Каждая область задается цветом и внешним контуром, который в свою очередь определяется последовательностью точек. Результат выделения с помощью алгоритма металлических проводников на изображениях слоев ИМС показан на рис. 4. На рисунке металлические проводники выделены белым цветом.



Рис. 4.

Далее из полученных векторных описаний каждого кадра необходимо получить общее векторное описание каждого слоя ИМС, а затем совместить эти описания для формирования послыонного векторного описания ИМС.

2. Построение векторного описания слоя ИС

Исходными данными для построения общего векторного описания одного слоя являются 1) векторные описания кадров слоя, 2) сформированные на этапе шивки абсолютные

координаты кадров в пределах слоя, 3) конструкторско-технологические ограничения.

Требуется скорректировать векторные описания кадров и получить ОВО слоя.

Алгоритм построения ОВО.

Шаг 1. Выбирается первый кадр из набора кадров, и соответствующее ему векторное описание добавляется в общее векторное описание слоя путем преобразования относительных координат вершин контуров в абсолютные координаты с учетом положения кадра в пределах слоя относительно точки привязки.

Шаг 2. Для добавленных векторных описаний контуров устанавливается атрибут - "активный", указывающий на недавнее добавление в общее описание.

Шаг 3. Из общего набора кадров выбирается очередной кадр, такой, чтобы он был соседним к кадру, последнему добавленному в ОВО. Для всех контуров этого кадра производится преобразование относительных координат в абсолютные с учетом положения кадра относительно точки привязки.

Шаг 4. Для выбранного кадра все содержащиеся векторные представления контуров корректируются (обрезаются по границе кадров) с учетом перехлеста соседних кадров. Пример показан на рис. 5. Для этих контуров устанавливается атрибут "активный2".

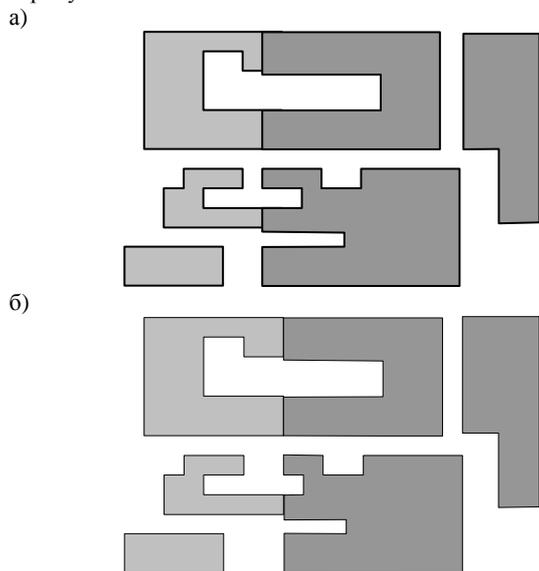


Рис. 5. Результат коррекции описания с учетом перехлеста кадров:

- а) векторные представления до коррекции;
- б) векторные представления после коррекции.

Шаг 5. Контура, помеченные как "активный2" добавляются в ОВО путем их объединения с контурами, имеющими атрибут "активный". Результат показан на рис. 6а.

Оставшиеся контура кадра добавляются в ОВО, путем их простого переноса. Результат показан на рис. 6б.

Шаг 6. Если общий набор кадров не пуст, то переходим на Шаг 2, иначе алгоритм заканчивает работу.

Установка атрибутов "активный" и "активный2" направлена на повышение скорости работы алгоритма. Такой подход позволяет не проводить полный перебор контуров в ОВО при их объединении с контурами добавляемого кадра, а вести поиск только среди контуров, расположенных в области перекрытия кадров.

На шаге 5 алгоритма выполняется также частичная проверка и коррекция объединяемых контуров на соответствие следующим нормам КТО:

- ширина проводника;
- расстояние между проводниками.

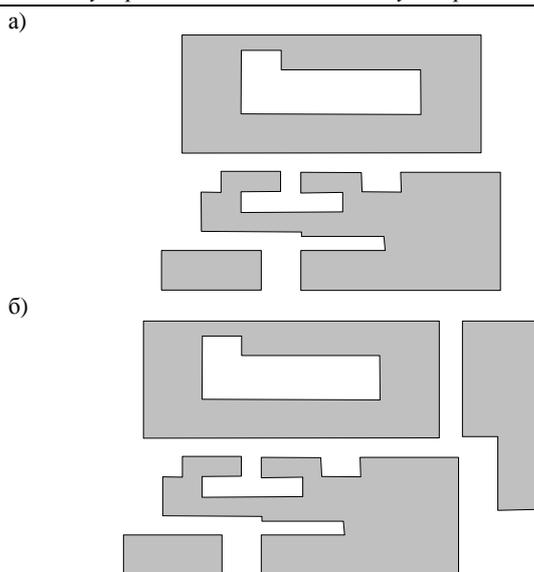


Рис. 6. Общее векторное описание:

- а) после объединения контуров;
- б) после переноса контуров.

При реализации алгоритма возникает проблема объединения и дальнейшего представления областей, имеющих в себе вложенные структуры. В алгоритме предусмотрены два варианта представления таких областей. Первый вариант описания предусматривает представление таких областей в виде одного контура. В этом случае внешний и внутренний контуры связываются "перемычкой". Во втором случае для представления этой области используются два контура, при этом имеет значение их пространственное положение. В результате, во внутреннем векторном представлении для внутренних контуров появляется дополнительный признак.

В любом случае во время работы алгоритма все области представляются только одним контуром, содержащим перемычки, окончательное векторное представление выбирается пользователем.

В местах объединения контуров возможны возникновения "ступенек", которые также обнаруживаются и могут быть устранены.

3. Совмещение слоев

Здесь исходными данными являются: векторное описание каждого слоя, координаты контактных площадок для каждого из совмещаемых слоев и КТО.

Требуется выполнить коррекцию векторного описания слоев с учетом КТО и сформированных абсолютных координат.

Совмещение осуществляется посредством сопоставления координат контактных площадок, связывающих проводники из совмещаемых слоев.

Предлагаются два базовых алгоритма для решения задачи совмещения слоев. Первый основан на минимизации суммарной функции ошибки положения всех связывающих контактных площадок для совмещаемых слоев путем сдвига их векторных представлений относительно друг друга по осям X и Y на величину Δx или Δy . При этом полные векторные описания каждого слоя уже получены на этапе совмещения кадров слоя. Один из вариантов этого алгоритма предусматривает минимизацию функции ошибки только для выбранных кадров. В этом случае кадры для совмещения выбираются автоматически либо пользователем. Желательно выбирать удаленные друг от друга кадры, расположенные в различных частях слоя. Значения Δx и Δy выбираются исходя из неравенств:

$$\Delta x \leq \frac{Sp}{2}, \tag{4}$$

$$\Delta y \leq \frac{Sp}{2},$$

где Sp - длина стороны контактной площадки.

Второй алгоритм предусматривает наличие общего векторного описания только одного из совмещаемых слоев. Для второго слоя этапы совмещения кадров слоя в векторном формате с учетом КТО и растровых кадров не проводятся, и он представлен отдельными векторными описаниями каждого кадра. В этом случае для каждого кадра второго слоя вычисляются его абсолютные координаты положения путем минимизации суммарной функции ошибки положения связывающих контактных площадок этого кадра и контактных площадок соответствующего кадра первого слоя. Такой подход позволяет для каждого кадра второго слоя вычислить координаты положения в слое относительно заданной точки привязки и избежать трудоемкой с вычислительной точки зрения процедуры совмещения растровых кадров. Затем для второго слоя проводится этап совмещения кадров в векторном формате с учетом КТО.

После получения абсолютных координат слоев необходимо произвести уточнение положения контактных площадок с учетом КТО. Изменение положения контактной площадки может потребовать внесение значительных изменений в векторное описание слоев. Поэтому здесь могут применяться специальные алгоритмы обработки векторных представлений слоев, которые контролируют следующие конструкторско-технологические параметры:

- размер контактной площадки;
- размер контактного окна;
- расстояние между контактной площадкой и проводником;
- ширина проводника;
- расстояние между проводниками;
- расстояние от границы контактного окна до внешней границы контактной площадки.

Окончательное положение контактной площадки в каждом слое выбирается как усредненное положение связывающих площадок, если допустимое смещение D меньше или равно заданному порогу. Величина D вычисляется из следующего выражения:

$$D = |Do - Dn|, \quad (5)$$

где Do - старое положение контактной площадки;

Dn - уточненное положение контактной площадки.

В противном случае окончательное положение контактной площадки выбирается исходя из требования внесения минимальных изменений в векторные описания слоев.

В результате работы алгоритмов формируется: векторное описание слоев и уточненные координаты контактных площадок.

УДК 681.3

Горбашко Л.А.

СТЕГАНОГРАФИЧЕСКОЕ СКРЫТИЕ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ПРЕОБРАЗОВАНИЯ ФУРЬЕ

Введение

Стеганография исследует скрытую передачу данных в маскирующем сигнале. Для скрытия информации могут быть выбраны любые файлы: текстовые, HTML, графические, звуковые, dll- библиотеки, но чаще используются графические и звуковые файлы, т.к. при встраивании сообщения в эти типы файлов не увеличивается размер файла со скрытой информацией.

На данном этапе выявляются также отсутствующие контактные площадки. Рассмотрим множество $\{P_i / i = 1, 2, \dots, k\}$ слоев, где слой i расположен над слоем $i + 1$. Множество КП слоя i , соединенных с КП слоев $i - 1$ и $i + 1$ обозначим как P_i^{pred} и P_i^{succ} соответственно, $P_i = P_i^{pred} \cup P_i^{succ}$, $i = 1, \dots, k$. Для слоя 1 имеем $P_1^{pred} = \emptyset$ и для слоя k - $P_k^{succ} = \emptyset$. Нарушение равенства $P_i^{pred} = P_{i-1}^{succ}$ or $P_{i+1}^{pred} = P_i^{succ}$ свидетельствует о об отсутствии или наличии лишней КП.

Заключение

Предложен эффективный по точности алгоритм аппроксимации выделенных контуров отрезками различной длины, основанный на преобразовании Хафа. Алгоритм формирует векторное описание топологии одного кадра видеоизображения. Предложен алгоритм объединения векторных описаний каждого кадра в общее векторное описание слоя и алгоритм совмещения этих описаний для формирования многослойного представления топологии ИС.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Ваткин М.Е., Дудкин А.А.* Алгоритм совмещения частично перекрывающихся кадров изображения // Анализ цифровых изображений. Минск: ОИПИ Национальной академии наук Беларуси, 2003. Вып. 2. С. 25-32.
2. *Duda R. O., Hart P. E.* Use of the Hough transformation to detect lines and curves in pictures // Communication of the Association for Computing Machinery. 1972. Vol. 15. №. 1. P. 11-15.
3. *Denzler J.* Texture region Extraction. "http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/DENZLER1/node11.html". In CVonline: On-Line Compendium of Computer Vision [Online]. R. Fisher (ed). Available: "http://www.dai.ed.ac.uk/CVonline/". [9.07.2001].
4. *L. G. Brown.* A survey of image registration techniques // ACM (Assoc. Comput. Mach.) Comput. Surv. 1992. № 24. P. 325-376.
5. *Д.А. Вершок, А.А. Дудкин, А.Г. Калюта, А.М. Селиханович.* Система цифровой обработки изображений слоев интегральных микросхем // Идентификация образов. Минск: Ин-т техн. кибернетики НАН Беларуси, 2001. № 2. С. 72-87.
6. *Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong.* A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry // Artificial Intelligence Journal. 1995.V.78. P. 87-119
7. *William J. Chimmitt, Jr., and Laurence G. Hassebrook.* Scene reconstruction from partially overlapping images with use of composite // J. Opt. Soc. Am. 1999. № 16(9), P. 2124-2135.
8. *Прэнт У.* Цифровая обработка изображений: Пер. с англ. М.: Мир, 1982. Кн. 2. 480 с.

В стеганографии принята следующая терминология.

Контейнер (изображение- носитель, container, cover image, carrier) – файл для встраивания данных.

Скрытое сообщение – сообщение, встраиваемое в контейнер. Стего-образ, стего (stego-image) – изображение со скрытой информацией.

Все известные методы цифровой стеганографии можно

Горбашко Лариса Ашотовна, ст. преподаватель кафедры интеллектуальных информационных технологий Брестского государственного технического университета. Беларусь, БГТУ, 224017, г. Брест, ул. Московская, 267.