

УДК 681.327.22 : 003.6

М. Г. Георгиев, И. Г. Таранцев*(Новосибирск)***АЛГОРИТМ СЖАТИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ИЗОБРАЖЕНИЙ
В СИСТЕМАХ ВИЗУАЛЬНОГО КОНТРОЛЯ
РЕДКО МЕНЯЮЩЕЙСЯ ОБСТАНОВКИ**

Предложен алгоритм сжатия последовательности изображений в системах визуального контроля на базе персонального компьютера. При значительных изменениях наблюдаемой обстановки алгоритм обеспечивает сигнализацию и гарантирует сохранение изображения без видимых искажений. Коэффициент сжатия алгоритма достаточен для записи нескольких часов видеоданных на персональном компьютере стандартной конфигурации.

Введение. В системах, предназначенных для слежения за неподвижными объектами, возникает необходимость накопления визуальной информации в течение длительного периода времени. Кроме того, может ставиться задача сигнализации о критических изменениях в наблюдаемой обстановке. Как правило, системы наблюдения на базе персонального компьютера, оснащенного устройством ввода видеоизображений, обладают рядом особенностей:

- камера, ведущая наблюдение, неподвижна;
- объекты наблюдения неподвижны;
- изменения в наблюдаемой ситуации происходят редко, но информация о них крайне важна, так как именно ее получение является целью наблюдения, следовательно, потеря качества и искажение изображения недопустимы;
- в сжимаемом изображении всегда присутствует шум;
- освещенность объектов подвержена периодическим изменениям в широком диапазоне частот (мерцание искусственного освещения, свет фар проезжающих машин, смена времени суток);
- время обработки очередного кадра не должно превышать некоторой небольшой величины (порядка одной секунды);
- скорость записи видеоданных на информационный носитель (диск) практически не ограничена;
- объем носителя информации (диска) ограничен.

Целью данной работы является разработка алгоритма сжатия последовательности изображений для системы наблюдения за редко меняющейся обстановкой на базе персонального компьютера стандартной конфигурации с устройством ввода видеоизображений. Алгоритм должен обеспечивать

коэффициент сжатия, достаточный для записи нескольких часов видеоданных (25–50 раз).

Обзор методов сжатия последовательности изображений. Рассмотрим существующие алгоритмы сжатия последовательности изображений с точки зрения поставленной задачи.

Алгоритмы покадрового сжатия без потерь информации (LZW, RLE, Huffman) и с потерей информации (DCT, фрактальный метод) не подходят для решения поставленной задачи из-за слишком низкого коэффициента сжатия [1].

Алгоритмы, учитывающие при сжатии межкадровую зависимость (Cinepack, Intel Indeo, H621), также не решают поставленной задачи, поскольку предназначены для формирования равномерного потока данных для передачи видеоданных по линиям связи с ограниченной пропускной способностью. При сильных межкадровых изменениях они ограничивают объем сжатого кадра за счет уменьшения качества восстановленного изображения. При передаче неподвижной картинке эти алгоритмы продолжают поддерживать поток данных неоправданно высоким, передавая мелкие детали и фон. Данная задача предъявляет к алгоритму совершенно иные требования. Алгоритм должен передавать изменения изображения с максимальной точностью. Динамичность сцены не должна сказываться на качестве изображения. При этом суммарный объем сжатого изображения должен быть минимальным (передавать информацию о фоне нецелесообразно).

Алгоритм MPEG [2] обеспечивает высокий коэффициент сжатия при незначительных потерях качества восстановленного изображения за счет использования векторов смещения участков изображения относительно их положения в предыдущем кадре и двунаправленной интерполяции текущего кадра по предыдущему и последующим кадрам. К сожалению, нахождение векторов смещения требует большого объема вычислений и без специализированных устройств не позволяет производить сжатие с требуемой для данной задачи скоростью. Двунаправленная интерполяция также не может быть применена из-за значительных задержек во времени для оповещения о критических изменениях в наблюдаемой обстановке.

Общая структура алгоритма. Скорость обновления информации в системах наблюдения невелика (1–5 Гц). Если в наблюдаемую сцену проникает посторонний объект с высокой скоростью движения, его положение на соседних кадрах может значительно изменяться (объект пересекает видимую область за 2–3 кадра). В этом случае невыгодно передавать отличия от предыдущего кадра, так как площадь изменений равна удвоенной площади постороннего объекта. Выгоднее передавать отличия от некоторого фонового кадра. Кроме того, при медленных движениях объекта или при слабых изменениях самой сцены выгодно передавать отличия от предыдущего кадра. Поэтому в предлагаемом алгоритме для сравнения с текущим кадром используются два кадровых буфера: фоновый кадр и предыдущий кадр. Причем если в течение длительного времени (S_{\max} кадров) повторяется предыдущий кадр, например, после того как в комнате включили свет, то этот кадр становится фоновым, т. е. данные копируются из буфера предыдущего кадра в буфер фонового кадра. Для подсчета числа повторений используется переменная S_{count} .

Сравнение изображений проводится специальным *алгоритмом сравнения кадров*, который находит площадь отличий одного кадра от другого. Область допустимых значений площади отличий разбита на три интервала,

границы между которыми задаются двумя переменными: V_{\min} – граница между первой и второй областью, V_{\max} – между второй и третьей. Если отличия кадров пренебрежимо малы и не превышают величины V_{\min} , то в выходной поток передается пустой кадр с соответствующим флагом «Повторить фоновый кадр» или «Повторить предыдущий кадр». Если кадры отличаются, но отличия не превышают величины V_{\max} , то подвергается сжатию и передается в выходной поток разность между кадрами. Если же кадры настолько сильно отличаются друг от друга (больше V_{\max}), что их разность имеет большую амплитуду и плохо сжимается, то сжатию подвергается полный кадр и в выходной поток передается так называемый «ключевой кадр», несущий полную информацию об изображении. Использование ключевого кадра вместо разностного немного ухудшает коэффициент сжатия алгоритма, но зато значительно улучшает качество восстановленного изображения. Кроме того, при последующем просмотре записанного

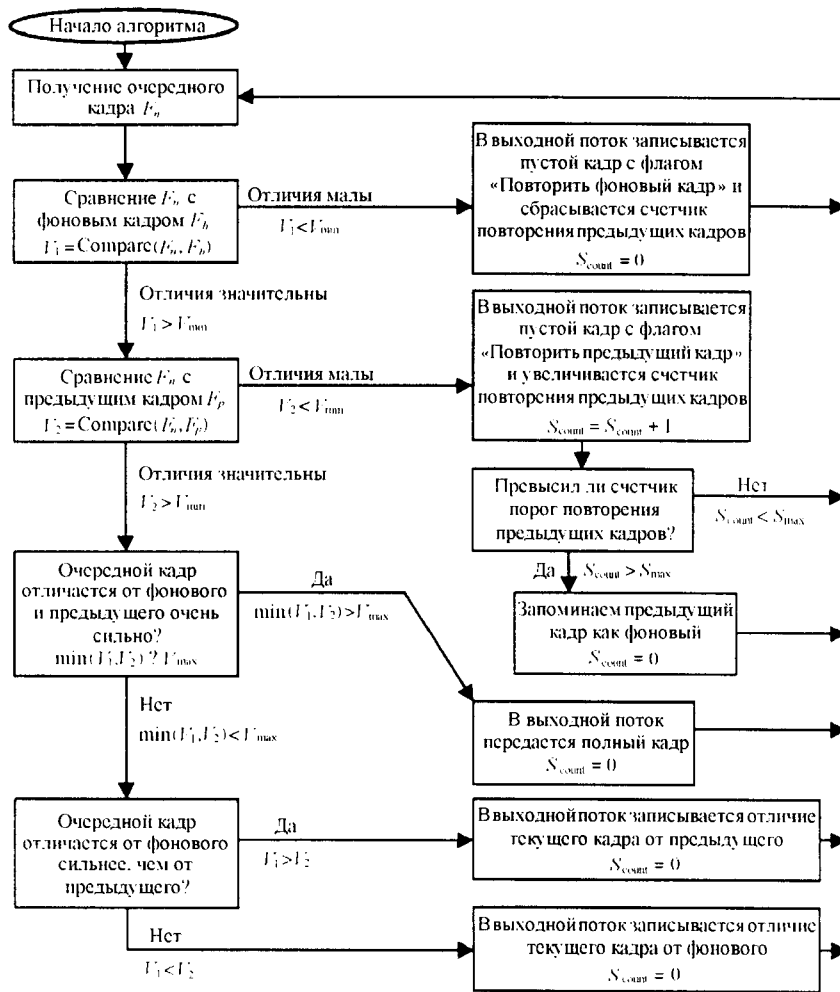


Рис. 1. Структура алгоритма сжатия очередного кадра

0	-1	0
-1	4	-1
0	-1	0

Рис. 2. Дифференцирующий фильтр

материала наличие ключевых кадров позволяет значительно ускорить позиционирование в потоке видеоданных. Мера отличия очередного кадра от предыдущего позволяет включать оповещение о критических изменениях в наблюдаемой обстановке.

Общая структура алгоритма кодирования очередного кадра представлена на рис. 1.

Алгоритм сравнения кадров. В представленном алгоритме в общем случае сжатию подвергается последовательность полноцветных изображений. Однако специально для алгоритма сравнения по исходной цветной картинке создается черно-белое изображение, которое не используется при последующем сжатии. Переход к градациям серого позволяет упростить процесс сравнения, так как вместо трех цветовых полей сравнивается только одно, являющееся наиболее информативным при восприятии изображения исследователем [3]. Перед сравнением черно-белое изображение подвергается дифференцированию пятиэлементным фильтром (рис. 2) для исключения постоянной составляющей и выделения контуров (рис. 3, *a*). Поскольку изменение освещенности наблюдаемой сцены изменяет среднее значение яркости изображения и практически не изменяет положения контуров, то дифференцирование изображения позволяет исключить влияние освещенности на алгоритм сравнения изображений. После этого производится масштабирование изображения с уменьшением разрешения (рис. 3, *b*). Степень снижения разрешения определяет минимальный размер регистрируемых изменений. Получившийся образ и используется непосредственно для сравнения изображений. Мерой отличия двух изображений является количество пикселей, яркость которых в сравниваемых образах превышает установленный порог.

Сжатие кадра. Если принято решение о сжатии очередного кадра, то изображение преобразуется в формат YUV 4 : 1 : 1, что позволяет уменьшить объем исходной информации вдвое [4]. Для зависимых (не ключевых) кадров производится вычитание из них фонового или предыдущего кадра. Затем кадр сжимается с использованием алгоритма RLE [1]. В алгоритме осуществляется независимое сжатие каждой из цветовых компонент изображения с маскированием младших битов (интерполяция нулевого порядка) для уменьшения влияния шумов и увеличения коэффициента сжатия [5]. При небольшом числе маскируемых битов алгоритм не приводит к заметным искажениям восстановленного изображения, а лишь немного сглаживает

его. Цветоразностные составляющие сигнала (U и V) можно сглаживать сильнее, чем яркостную составляющую (Y), поскольку информация о яркости играет более важную роль в восприятии изображения исследовате-



Рис. 3. Тестовое изображение, подвергнутое дифференцированию (*a*) и масштабированию (*b*)



Рис. 4. Восстановленное изображение при маскировании 2–3 бита



Рис. 5. Восстановленное изображение при маскировании 5 бит

лем, чем информация о цвете [3]. В результате экспериментов было установлено, что после сжатия с маскированием двух младших битов яркостной компоненты и трех битов цветоразностных компонент восстановленное изображение визуально не отличается от исходного (рис. 4). При сжатии с маскированием трех битов яркостной компоненты и четырех битов цветоразностных компонент восстановленное изображение содержит заметные искажения. И только при маскировании пяти битов искажения мешают восприятию визуальной информации (рис. 5). В табл. 1 приведена зависимость коэффициента сжатия тестового изображения от числа маскируемых битов.

Сжатие и восстановление каждого кадра производятся с использованием изображений предыдущего и фонового кадров. Так как восстановление происходит с потерей информации, важно обеспечить полное совпадение сохраняемого в специальном буфере предыдущего кадра при сжатии и восстановлении. Для этого необходимо при сжатии обновлять содержимое буфера предыдущего кадра не прямым копированием в него нового кадра, а восстановлением только что сжатых данных. Такая процедура гарантирует, что при восстановлении любого кадра последовательности состояние буфера будет точно таким, каким оно было при его сжатии, и позволит избежать накопления искажений в разностных кадрах, возникающих из-за

Таблица 1

Зависимость коэффициента сжатия от числа маскируемых битов

Количество маскируемых битов яркостных компонент	Количество маскируемых битов цветоразностных компонент					
	0	1	2	3	4	5
0	2,08	2,08	2,13	2,13	2,22	2,27
1	2,17	2,17	2,22	2,27	2,32	2,38
2	2,44	2,44	2,44	2,50	2,70	2,70
3	2,85	2,85	2,85	2,94	3,12	3,22
4	3,33	3,44	3,44	3,57	3,70	4,00
5	4,00	4,17	4,17	4,35	4,55	5,00

потери информации при сжатии RLE. Заметим, что для буфера фонового кадра это условие выполняется автоматически, так как в него копируется информация только из буфера предыдущего кадра.

Восстановление изображения. Алгоритм восстановления изображения представлен на рис. 6. При отображении кадр преобразуется из формата YUV в формат RGB.

Сравнение со стандартными алгоритмами сжатия последовательностей изображений. Была проведена серия измерений с целью сравнения работы разработанного алгоритма с другими алгоритмами сжатия: "Cineraack Codec by Radius" версии 1.10.0.6 и "Intel Indeo Video R3.2". Сравнение с алгоритмом MPEG не проводилось из-за очень медленной скорости сжатия последовательности изображений. При проведении измерений для параметров настройки каждого из алгоритмов подбирались значения, обеспечивающие визуально одинаковое качество восстанавливаемого изображения. Использованные при сравнении тестовые последовательности изображений были сняты работающей в стандарте NTSC видеокамерой с частотой 1 кадр в секунду и отображали как неподвижные, так и динамически изменяющиеся сцены. Результаты измерений представлены в табл. 2. Из таблицы видно, что при сжатии последовательностей редко изменяющихся изображений, когда число повторяющихся кадров велико, коэффициент сжатия разработанного алгоритма значительно выше, чем у других алгоритмов.

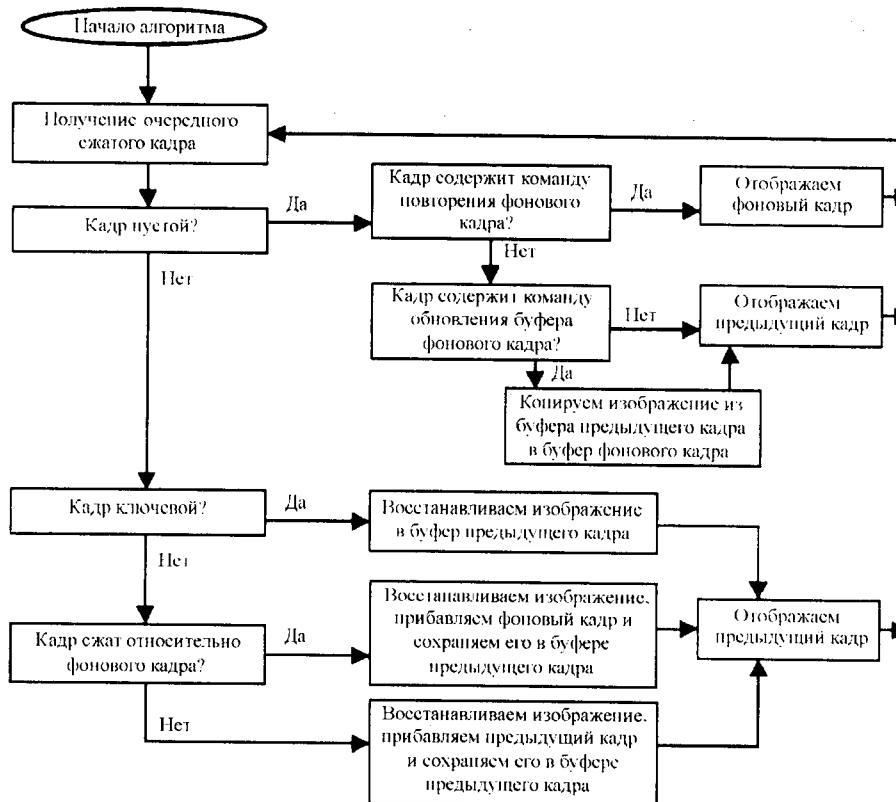


Рис. 6. Алгоритм восстановления изображения

Таблица 2

Сравнение эффективности алгоритмов сжатия видеопоследовательностей

Номер последовательности	Коэффициент сжатия			
	Предлагаемый алгоритм	Алгоритм "Intel Indeo Video R3.2"	Алгоритм "Cinepack Codec by Radius"	Процент неповторяющихся кадров
1	27.3	9.9	22.8	9.1
2	14.0	9.8	10.2	22.7
3	13.7	9.8	9.8	40.9
4	7.3	9.8	13.1	50.0
5	5.8	9.9	18.8	50.0
6	7.1	9.9	17.4	59.1

Заключение. Предлагаемый алгоритм сжатия последовательности изображений обеспечивает коэффициент сжатия, достаточный для сохранения последовательности редко изменяющихся изображений длительностью до нескольких часов на персональном компьютере стандартной конфигурации. При этом алгоритм гарантирует восстановление изображения без видимых искажений при значительном изменении наблюдаемой обстановки. Скорость работы алгоритма достаточна для обработки в реальном времени данных с частотой кадров 1–5 Гц. Алгоритм обладает рядом параметров, позволяющих легко настраивать его для использования в каждом конкретном случае. Алгоритм нечувствителен к изменению освещенности сцены, однако не всегда удается избежать ложного срабатывания алгоритма сравнения кадров при движении источников освещения. Движение источника освещения приводит к появлению на объектах бликов и к перемещению теней, что воспринимается алгоритмом как движение объектов и приводит к сохранению бесполезной информации.

СПИСОК ЛИТЕРАТУРЫ

1. Murray J. D., van Ryper W. Encyclopedia of Graphics File Formats. Sebastopol, CA: O'Reilly & Associates, Inc. 1994.
2. Ряхин А. Видеостандарт MPEG // 625. 1996. № 6.
3. Измайлов Ч. А., Соколов Е. Н., Черноризов А. М. Психофизиология цветового зрения. М.: Изд-во МГУ. 1989.
4. Jack K. Video Demystified: a Handbook for the Digital Engineer. Solana Beach: High Text Publ., Inc., 1995.
5. Прэтт У. Цифровая обработка изображений. М.: Мир, 1982. Т. 2.

Поступила в редакцию 16 сентября 1998 г.