

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/280831839>

ВЪВЕДЕНИЕ В КОМПЮТЪРНОТО ЗРЕНИЕ

Book · March 2015

DOI: 10.13140/RG.2.1.2168.3043

CITATION

1

READS

2,382

1 author:



Miroslav Galabov

St.Cyril and St.Methodius University of Veliko Tarnovo

27 PUBLICATIONS 43 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Project

System training and simulations with augmented reality. [View project](#)

Мирослав Гълъбов

ВЪВЕДЕНИЕ В КОМПЮТЪРНОТО ЗРЕНИЕ

**Велико Търново
2015 г.**

Анотация

В книгата **”Въведение в компютърното зрение”** се разглежда теорията на Компютърното зрение, състояща се от група изследвания в областта на компютърните науки, насочени към автоматична обработка на изображения от реалния свят, с цел извличане и интерпретиране на визуалната информация в тях.

Компютърното зрение е научна област от Компютърната наука, с основни задачи насочени към изграждане на системи с изкуствен интелект, включващи задачи като:

- (1) разпознаване - основният проблем е свързан с определяне кога конкретни данни са част от специфичен обект, дали тези данни определят конкретни негови свойства и действия;
- (2) определяне на движението, при което се проследява придвижването на обектите и последователността от изображения се обработва;
- (3) възстановяване на сцената – представлява изграждане на 3D модел на сцената;
- (4) възстановяване на изображението с цел премахване на шума и изясняване на изображенията чрез прилагане на различни филтри.

Книгата ще бъде полезна за студентите от специалности “Компютърни науки” и “Информатика” на факултет “Математика и информатика” на ВТУ”Св.св.Кирил и Методий”, магистри, докторанти и специалисти работещи в областта на компютърното зрение.

Рецензент: проф.д-р Маргарита Тодорова
Мирослав Гълъбов – автор, 2015

Съдържание

Въведение

ГЛАВА ПЪРВА: ОСНОВНИ ПОНЯТИЯ В КОМПЮТЪРНОТО ЗРЕНИЕ

1.1. ОСНОВНИ ПОНЯТИЯ. КЛАСИФИКАЦИЯ НА СИСТЕМИТЕ ЗА КОМПЮТЪРНО ЗРЕНИЕ. ФИЗИЧЕСКИ ОСНОВИ НА КОМПЮТЪРНОТО ЗРЕНИЕ.

1.2. АПАРАТЕН МОДЕЛ НА КОМПЮТЪРНО ЗРЕНИЕ. ИНФОРМАЦИОНЕН МОДЕЛ НА СИСТЕМА ЗА КОМПЮТЪРНО ЗРЕНИЕ.

ГЛАВА ВТОРА: МЕТОДИ ЗА ОБРАБОТКА НА ИЗОБРАЖЕНИЯ

2.1. ФИЛТРАЦИЯ НА ИЗОБРАЖЕНИЯ.

2.2. КОНВОЛЮЦИЯ.

2.3. ДИСКРЕТНО ПРЕОБРАЗУВАНЕ НА ФУРИЕ.

ГЛАВА ТРЕТА: ЕТАПИ НА ОБРАБОТКА ПРИ КОМПЮТЪРНОТО ЗРЕНИЕ

3.1. ОПРЕДЕЛЯНЕ НА РЪБОВЕ И ГРАНИЦИ ЗА ОБЕКТИТЕ В ИЗОБРАЖЕНИЕТО.

3.2. ОПРЕДЕЛЯНЕ НА ПРИЗНАЦИ. СЕГМЕНТАЦИЯ.

3.3. ОСНОВНИ ПРИНЦИПИ И МЕТОДИ НА РАЗПОЗНАВАНЕТО НА ОБРАЗИ.

3.4. КЛАСИФИКАЦИЯ НА ОБРАЗИ ЧРЕЗ ЕТАЛОНИ И ВЕРОЯТНОСТЕН ПОДХОД.

3.5. КЛЪСТЕРИЗАЦИЯ.

3.6. АНАЛИЗ И РАЗЛАГАНЕ НА 3D ВИЗУАЛНИ СЦЕНИ.

3.7. МЕТОД НА УОЛЦ ЗА АНАЛИЗ НА 3D СЦЕНИ.

ГЛАВА ЧЕТВЪРТА: ЦИФРОВО ВИДЕО

4.1. КОМПРЕСИЯ НА ЦИФРОВО ВИДЕО ИЗОБРАЖЕНИЕ.

4.2. ВИДОВЕ СТАНДАРТИ ЗА КОДИРАНЕ НА ВИДЕО ИНФОРМАЦИЯ.

4.3. СЪВРЕМЕННИ ВИДЕО ФАЙЛОВИ ФОРМАТИ.

Заклучение

Въведение

Целта на системите за компютърно зрение е да изградят модел на наблюдаваната сцена. За възприемането на информацията се използват технически устройства (камери, скенери и др.), а за обработката – компютърни системи с подходящ софтуер.

В ГЛАВА ПЪРВА са разгледани основните понятия в компютърното зрение. Направена е класификация на системите за компютърно зрение (СКЗ), дадени са апаратния и информационния модел на СКЗ.

В ГЛАВА ВТОРА е даден основният математически апарат на методите за обработка на изображения. Разгледани са най-често използваните филтри, конволюцията върху дискретни полседователности и Дискретното преобразуване на Фурие.

В ГЛАВА ТРЕТА са описани етапите на обработка при компютърното зрение. Определянето на ръбовете и границите на обектите, определянето на признаци и сегментация. Разгледани са основните методи за разпознаване на образи. Отделено е внимание на анализа и разлагането на 3D визуални сцени.

В ГЛАВА ЧЕТВЪРТА е разгледано компресирането на цифрово видео, стандартите за компресия и основните видео файлови формати.

ГЛАВА ПЪРВА: ОСНОВНИ ПОНЯТИЯ В КОМПЮТЪРНОТО ЗРЕНИЕ

1.1. ОСНОВНИ ПОНЯТИЯ. КЛАСИФИКАЦИЯ НА СИСТЕМИТЕ ЗА КОМПЮТЪРНО ЗРЕНИЕ. ФИЗИЧЕСКИ ОСНОВИ НА КОМПЮТЪРНОТО ЗРЕНИЕ.

1.1.1. Основни понятия. Класификация на системите за компютърно зрение.

Човешкият труд се разделя на физически и умствен. Машините, предназначени за автоматизация на физическия труд се наричат енергетични, а на умствения – информационни.

Интегрирането на горните две машини се наричат роботехнически. **Роботехнически системи:** идват от Робот. Роботът е конюнктурно понятие свързано с изкуствения интелект (ИИ).

Изкуствен интелект (Нилс Нилсон): Известно е пространството на решения на дадена задача, необходимо е да се намери поне едно от тези решения. Най-добре е да се намери най-ефективното решение. Развиват се методите за търсене на решения. Предефиниране на определението на Нилсон: Създаване на програми, които реализират функции на човешкият разум.

Развиват се експертни системи на основата на това определение.

Една система е с ИИ, ако отговаря на следните 5 критерия :

- 1) Изгражда ли системата модел на околната среда?
- 2) Използва ли системата този модел на околната среда за изграждане на план на действие в тази среда?
- 3) При изпълнението на плана за действие анализира ли системата алтернативни възможности?
- 4) Може ли системата да предефинира плана за действие, ако той води към непредсказуеми състояния за околната среда?
- 5) Използва ли системата минал опит за разширение и подобряване на модела на околната среда?

Основното тук е изграждане на модела на околната среда. Целта на компютърното зрение е да се изгради модела за наблюдаваната околна среда. Трябва да извлече информация за тази околна среда, за да се постигне целта.

Човек възприема най-много (90%) от входящата информация чрез зрението си. Чрез обонянието и вкуса - около (4%). Останалите проценти са за слуха.

Околна среда: съвкупност от обекти, попадащи в полето на наблюдение (възприемане) от съответната видеовъзприемаща апаратура.

Визуална сцена: двумерна картина на съвкупността от двумерни или тримерни обекти попадащи в полето на наблюдение на видеозаснемащата апаратура.

Класификация на системите за компютърно зрение може да бъде направена според [1, 12]:

- 1) Област на приложение: визуален контрол, медицина, авионаблюдение, космически наблюдения.
- 2) Сложност: прости, със средни възможности, с големи възможности.
- 3) Размерност: двумерни и тримерни.
- 4) Тип на изображението: двоично, полутоново, цветно.
- 5) Сложност на задачата: центрирани детайли, детайли в разширен безпорядък, детайли в бункер.
- 6) Използвани средства: телевизионни камери (видикони и твърдотелни камери), мултиспектрални скенери, компютърни системи.
- 7) Начин на обработка на изображението: последователна, паралелна.

Двумерни системи за компютърно зрение: когато става дума за обработка на двумерни обекти.

Класификация според типа на изображението :

- 1) двоично изображение: използват се само 2 нива – черно (1)/бяло (0).
- 2) полутоново изображение: използват се повече от 2 нива. Обикновено 256 нива на сивото. Това изображение дава по-добра представа за формата на обектите във визуалната сцена.
- 3) цветно: използват се 3 цветови канала (син, зелен и червен).

Степента на сложност нараства както с преминаване от двоично, през полутоново до цветно изображение, така и при преминаване от двумерни към тримерни сцени. Разделянето е на

малка сложност, средна сложност и голяма сложност (цветно тримерно изображение).

Класификация според сложността на задачата:

- 1) Центрирани детайли: имаме конвейер с детайли. Детайлите не се препокриват.
- 2) Детайли в равнинен безпорядък: На конвейера детайлите могат да се закриват един друг.
- 3) Детайли в бункер: Системата трябва да вземе конкретен детайл от бункера.

Класификация според използваните средства:

- 1) Телевизионните камери могат да бъдат черно-бели, полутонови и цветни.
- 2) Мултиплексорни скенери: измерват дължината на електромагнитното колебание.

Зрение: процеса на определяне какво се съдържа в наблюдаваната околна среда и къде се разполагат обектите в тази наблюдавана среда.

Трябва да се получи изображение на тази среда. То трябва да се представи по определен начин в системата. На базата на това изображение се определят обектите. Системата трябва да бъде обучена да разпознава обекти.

Трябва да се определят отношенията между обектите. Определя се центъра на тежестта на всеки разпознат обект и се определят техните отношения. Може да има неразпознати обекти. Целият този процес изисква съхраняването на голям обем информация.

Система за компютърно зрение (СКЗ): представлява съвкупност от специализирани устройства за формиране на изображението на околната среда.

От друга страна за СКЗ са необходими методи, алгоритми и програми за обработка на изображения; определяне на признаци на обектите; построяване на формално описание на наблюдаваната сцена; разпознаване на обектите и изграждане на околната среда.

2. Физически основи на компютърното зрение

Всяко тяло, което има температура над абсолютната нула генерира електромагнитно излъчване. Излъчването се движи със

скоростта на светлината. С увеличаване на вътрешната температура на тялото се увеличава и количеството на излъчваната енергия, както и излъчването преминава от по-дълги към по-къси вълни.

Ако се вземе по абциса дължината на вълната то имаме следните области (започва се от най-късите вълни):

- Гама лъчи;
- Х лъчи (рентгеново излъчване);
- Ултравиолетови лъчи;
- Видима светлина;
- Инфракчервена (отразена и термална) светлина;
- Микровълни;
- Радиовълни.

Видимата област на електромагния спектър е в рамките 400-700 нанометра. За хората основен източник на електромагнитно излъчване е Слънцето. То излъчва в много честотни диапазони : от 1 пикометър до 30 микрометра. Максимума на излъчване е около 500 нанометра. Температурата на Слънцето е 6000K.

Земята също излъчва електромагнитна енергия при средна температура 300K. Нейният максимум обаче не е в във видимата част на спектъра, а е на 9,7nm.

Видимата област се разделя на 3 подобласти :

- синьо (blue) – 400-500nm;
- зелено (green) – 500-600nm;
- червено (red) - 600-700nm.

Човешкото зрение може да се разглежда като дистанционно наблюдение (измерване от разстояние). Ако при него имаме естествен източник на електромагнитни колебания, то тогава системата е *пасивна*. Ако източника на светлина е изкуствен – системата е *активна*.

Светлината се дели на :

- 1) ахроматична: не съдържа цвят, изменя се от черно до бяло през нива на сивото. Основна нейна характеристика е интензитета на отразената светлина.
- 2) хроматична: характеризира се със цвят. Анализа на цветни изображения става на базата на трите основни цвята Red, Green

и Blue (RGB). Всеки цвят може да се получи като комбинация от компоненти по всеки един от тези основни цветове.

Трите основни цвята образуват тримерно пространство на цветовете. Ако стойностите са дискретни и ограничени, пространството е куб. Началото на координатната система е (0,0,0). Коллинеарните вектори в това пространство представят един и същ цвят, но се различават по яркостта.

За да получим всеки един от тези 3 цвята от едно хроматично изображение се използват филтри. Интегралните уравнения за реализиране на филтрите са:

$$\begin{aligned}R &= \int E(\lambda) R^1(\lambda) d\lambda, \\G &= \int E(\lambda) G^1(\lambda) d\lambda, \\B &= \int E(\lambda) B^1(\lambda) d\lambda,\end{aligned}$$

където $R^1(\lambda)$, $G^1(\lambda)$, $B^1(\lambda)$ са функции на разпределение за съответните основни цветове, а $E(\lambda)$ – спектралния състав на цвета в дадена точка.

Във всяка от трите части на видимата част на спектъра, кривата на спектралния състав има изпъкнала форма. Максимумите в отделните части са определени да са: 435,8nm (blue), 546,1nm (green), 700nm (red).

Има и други системи за анализ на цветове и изображения.

Начина на възприемане на човешкото око се доближава повече до системата HIS (Hue, Saturation, Intensity);

- цветови отенък (Hue);
- наситеност (Saturation);
- яркост (Intensity) – пропорционална на интензитета на отразената светлина.

Преминаването от едната система в другата става лесно чрез използването на следните формули:

$$I = R + G + B; H = (G - B) / (I - 3B); S = (I - 3B) / I$$

Описание на различните части от електромагнитния спектър:

- 1) Гама лъчи ($< 0.03 \text{ nm}$): Поглъща се напълно от горните слоеве на атмосферата.

- 2) Рентгенови лъчи ($0.03 - 30 \text{ nm}$): Напълно се поглъща от горните слоеве на атмосферата.
- 3) Ултравиолетови лъчи ($0.003 - 0.4 \mu\text{m}$): Вълните с дължина $< 0.3 \mu\text{m}$ напълно се поглъщат от озона в горния слой на атмосферата.
- 4) Фотографска УВ област ($0.3 - 0.4 \mu\text{m}$): Минава през атмосферата. Улавя се със фоточувствителен филм. Атмосферното разсейване е много силно.
- 5) Видима област ($0.4 - 0.7 \mu\text{m}$): Изобразяване с филми и фотодетектори.
- 6) Инфракчервена област ($0.7 - 100 \mu\text{m}$): При взаимодействие с материята (отражение) променя дължината на вълната си. Съществуват атмосферни прозорци на пропускане.
- 7) Отражена ИЧ ($0.7 - 3 \mu\text{m}$): Съдържа информация за топлинните свойства на материята. Лентата ($0.7 - 0.9 \mu\text{m}$) се улавя с филм и се нарича фотографска ИЧ област.
- 8) Термална ИЧ ($3 - 5 \mu\text{m}$), ($8 - 14 \mu\text{m}$): Принципни атмосферни прозорци.
- 9) Микровълни ($0.1 - 30 \text{ cm}$): По-дългите вълни проникват през облаците. Изображение може да бъде получено по активен начин.
- 10) Радиообласт ($> 30 \text{ cm}$): Най-дългите вълни в електромагнитният спектър.

Значението на яркостите на изображенията се определя от четири фактора:

- Геометрични свойства на обектите;
- Коэффициент на отражение на видимите повърхности;
- Осветеност на сцената;
- Позиция на наблюдателя.

Задачата за предварителна обработка на зрителна информация се заключава в сортиране на изменението на яркостта в зависимост от причината, така че да получим представяне, в което четирите фактора се отчитат независимо един от друг.

Тази цел се постига на два етапа:

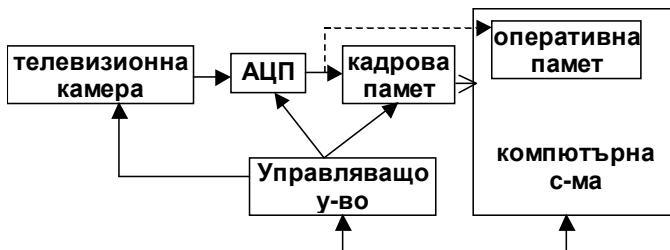
1. Съответстващите представления се базират на основата на изменение на яркостта и структурата. В резултат на тази обработка се получава така наречения първоначален **екскиз**.

2. Геометричните преобразувания върху видимите повърхности. Така се получава 2,5 мерен ексиз. И двата ексиза се построяват в равнината (координатната система) на наблюдателя.

1.2. АПАРАТЕН МОДЕЛ НА КОМПЮТЪРНО ЗРЕНИЕ. ИНФОРМАЦИОНЕН МОДЕЛ НА СИСТЕМА ЗА КОМПЮТЪРНО ЗРЕНИЕ.

1.2.1. Апаратен модел за възприемане на черно-бяло и полутоново изображение.

Изображението се заснема с телевизионна камера. Получените данни се подават на входа на Аналого-цифров преобразувател (АЦП) откъдето излиза кадровата памет. Тази кадрова памет е част от оперативната памет на системата за компютърно зрение. АЦП и кадровата памет се управляват от управляващото устройство (фиг.1.1). Всеки от тези блокове се реализира автономно.



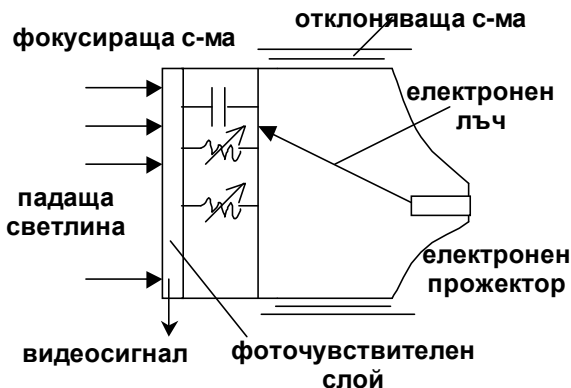
Фиг.1.1. Апаратен модел на система за КЗ

Телевизионните камери биват видикони и твърдотелни.

Принцип на работа на видикон :

Видикона има отпред светлочувствителен слой, върху който се изобразява сцената. Под този слой се намира слой от свързани със земя кондензатори и активни съпротивления. В задната част на видикона се намира устройство, което генерира електромагнитен лъч. Този лъч се насочва от друго устройство. Лъча обхожда всеки пиксел от предната част на видикона. Когато той попадне върху даден пиксел, последният отдава натрупаният

си заряд и така се отчита интензитета на светлината попаднал в пиксела. Съвкупността на интензитетите във всяка точка ни дава цялата картина ($I = f(x, y)$). Изменението на тази функция е пропорционално на падащата светлина (фиг.1.2).

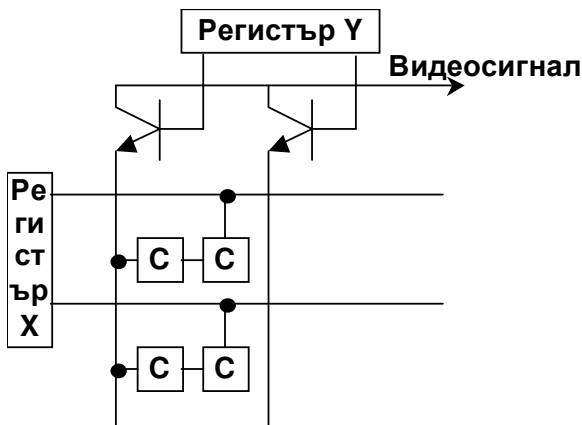


Фиг.1.2. Блокова схема на видеикон

В зависимост от интензитета на светлината падаща върху фоточувствителния слой се изменя яркостта. Елементите на фоточувствителния слой са: кондензатор и фотосъпротивление. След като се сформира потенциалния релеф, върху него се получава четене и отчитане на видеосигнала, чиито интензитет зависи от интензитета на падащата светлина. Видеосигналят се подава към АЦП. Потенциалният релеф съответства на съдържанието на визуалната сцена. На изхода се получава $I = f(x, y)$. Изменението на тази функция ще бъде пропорционално на изменението на падащата светлина.

Принцип на действие на твърдотелна камера (CCD - charge coupled device) - изгражда се като матрица от фоточувствителни елементи, които представляват МОП-кондензатор (изграден от слой Si, слой SiO₂, и метална пластина). Всеки фоточувствителен елемент се оформя като двойка зарядно свързани МОП-кондензатори (фиг.1.3). Цялата камера представлява матрица от зарядно свързани кондензатори. Кондензаторите са свързани с регистъра X \Rightarrow на тях се формира изображението като

потенциален ред. Зарядите на кондензаторите свързани с X се прехвърлят в кондензаторите, свързани с шините Y.



Фиг.1.3. Блокова схема на CCD камера.

С последователност от управляващи сигнали започва четенето на кондензаторите, свързани с Y, при което четене се получават видеосигналите, които се подават на полетата за обработка. Предимствата на CCD камерите пред видиконите:

- 1) фоточувствителните елементи са с добра геометрия;
- 2) отношението сигнал/шум е по-добро при CCD;
- 3) захранващото напрежение е много ниско.

CCD камерите могат да се използват в специални апаратури, при които пространственото квантуване е решено конструктивно. На изхода на CCD камерите имаме функция $f_1(x,y)$. Тази функция се квантува в пространството с амплитуда A . Пространствената дискретизация зависи от разрешаващата способност (256x256, 512x512) на камерите. Амплитудата на функцията f_1 се дискретизира от АЦП при обработката на полутонове $\Rightarrow 16 \div 256$ нива на символ.

Обемът на паметта за един кадър(кадрова памет) H се дава с формулата:

$$H = N^2 \log_2 m \text{ (bit)},$$

където $N \times N$ – размерност на кадъра; m - нива на сивото.

$$t = 1/(f_k * N^2) \text{ (sec)},$$

- 3) Отделяне на признаци. Сегментация.
- 4) Построяване на описание.
- 5) Интерпретация. Разпознаване.
- 6) Изграждане на модел на околната среда.

Тези стъпки могат да се разделят по два начина на нива. Едната система е с 2 нива (ниско/високо), другата система е с 3 нива (ниско/средно/високо). При първата, ниското ниво са стъпки 1,2 и 3, останалите са във високото. При втората система разделянето е на по две стъпки в ниво в същата посока както при първата система.

На входа на информационният модел постъпва функция $f_1(x,y)$, която се квантува до функция $g(x,y)$. Върху тази функция (дискретизирана) се прилага филтрация с цел подобряване на първоначалното изображение.

Сегментация: определяне на формата на различните обекти в сцената. $b(x, y)$ е двоично изображение, т.е. стойностите на функцията са само 1 или 0. Границите на ръбовете са означени с 1, а всичко останало е 0 (процес на отделяне на ръбове).

Интерпретация, разпознаване: Системата трябва да е предварително обучена, за да може да прави разпознаване.

Накрая, ако не са разпознати някои обекти, те ще участват в модела, но ще трябва да се дообучи системата за да ги разпознава следващ път. Моделът се създава на база отношението между обектите в сцената.

На основание на обработката се получават Базис от данни (БД) и Базис от знания (БЗ).

Информацията, която се получава в компютъра се дели на :

- декларативна: това са данните, които се обработват;
- процедурна: множество програми, по които става обработката.

Особености на една система за компютърно зрение по отношение на данните:

- 1) Обемът данни, които се обработва в една система за компютърно зрение е много голям. Следователно капацитета на БД е много голям.
- 2) Има йерархичност на данните.

Ако информационните единици (ИЕ) в БД придобият следните свойства те стават БЗ:

- 1) Вътрешна интерпретируемост: към ИЕ можем да се обърнем с ключ. С вторични ключове можем да се обърнем към определени полета на ИЕ.
- 2) Структурност: едни ИЕ да могат да се включват в други ИЕ. (част – цяло, обект – клас)
- 3) Свързаност на ИЕ: определя конкретни отношения за ИЕ (“пред”, “над”, “причина – следствие”, “аргумент – функция”). Тези отношения се отнасят към декларативната информация. “аргумент – функция” се отнася към процедурната информация.
- 4) Семантична свързаност: отнася се за конкретни предметни области. Близки ИЕ се анализират с цел формиране на някакво понятие.
- 5) Активност: анализирайки, конкретните данни и отношенията между тях, могат да се предизвикват конкретни действия.

Представянето на знанията в системата за компютърно зрение трябва да отговаря на следните условия:

- 1) Трябва да можем да представяме в системата различни видове знания.
- 2) Системата трябва да води до ефективно съхранение на данните.
- 3) Лесна манипулация с елементите на знанието.
- 4) Знанието трябва да бъде така представено, че да бъде лесно разбираемо за човека.

Начини за представяне на знанието :

- 1) Използват се портретни модели (iconic) – изходното полутоново или цветно изображение. Тези портретни модели са много чувствителни по отношение на точката на наблюдение и осветеността. Тези портретни модели изискват голям обем памет.
- 2) Графови модели: възли, които съответстват на отделни елементи на сцената (стени, обекти) и клонове (ребра) свързващи възлите. Ребрата указват връзките между елементите в сцената. Графовите модели са по-малка

чувствителност към осветеността и ориентацията на обектите в сцената.

- 3) Семантични мрежи: развитие на граф модела. Това е ориентиран граф, възлите на който показват елементи на сцената, а ребрата именувани (смислови) отношения.

Фрейми (Мински): Фреймите са друг начин за представяне на декларативно процедурно значение. Фрейма представлява структура от данни за определяне на стереотипни ситуации. Като под ситуация се разбира визуален образ, събитие. Фреймовото представяне на знанието представлява ориентиран граф, като възлите на този граф представляват елементите на наблюдаваната сцена. Ребрата на графа дават отношенията между елементите. Всички възли на ориентираният граф на горни нива са определени, а на най-ниското ниво възлите са празни. Празните възли се запълват в зависимост от конкретната ситуация. Всеки възел може да бъде обръщане към подфрейм, при което обръщане могат да се активират процедури за изчисление на информация за конкретния възел или за даване на конкретна информация за него.

От всички представяния на знанието, най-универсално е фреймовото.

Управление на обработката на информацията :

- 1) “отдолу – нагоре” (управление на данните) - на входа на всеки етап постъпват данни, на изхода излизат резултати, които активират изпълняването на следващият етап и т.н.
- 2) “отгоре – надолу “ (управление на моделите) - прави се хипотеза, че при дадена обработка като краен резултат ще се получи резултат съответстващ на заложеният. Недостатък е възможността да се направят голямо количество тестове преди да се достигне желаният резултат. Използва се често за анализ на тримерни сцени.
- 3) Обратна връзка - анализират се резултатите на изхода на всеки етап. Ако те са удовлетворителни се минава към следващият етап. Ако не са удовлетворителни се връщаме на входа на този етап и модифицираме обработката в някаква посока с цел получаване на различен резултат.

ГЛАВА ВТОРА: МЕТОДИ ЗА ОБРАБОТКА НА ИЗОБРАЖЕНИЯ

2.1. ФИЛТРАЦИЯ НА ИЗОБРАЖЕНИЯ

2.1.1.Общи сведения

Обикновено изображенията, формирани от различни информационни системи, се изкривяват под действието на смущения. Това затруднява както визуалният им анализ от оператор, така и автоматичната им обработка с компютър. Отслабването на смущенията става чрез филтрация [2, 13]. При нея яркостният сигнал на всяка точка от оригиналното изображение, изкривено от смущения, се замяна с друго значение на яркостта, което в най-малка степен е повлияно от изкривяванията. Изображението често представлява двумерна функция с пространствени координати, която се изменя по тези координати по-бавно от смущенията, които също се явяват двумерни функции. Това позволява при оценка на полезния сигнал във всяка точка от изображението да се вземат под внимание и множеството от съседни точки, възползвайки се от определената *еднаквост* на сигнала в тях.

Идеологията на филтрацията се основава на рационалното използване на данни както от работната точка, така и от нейната околност, т.е. филтрацията не е поелементна обработка на изображения. Задачата е в това, да се намери такава изчислителна процедура, която да позволява да бъдат достигнати най-добри резултати. При решаването на тази задача се опираме на използването на вероятностни модели на изображението и смущенията и статистически критерии за оптималност. Многообразието на методите и алгоритмите за филтрация е свързано с голямото разнообразие на изображения, които се описват с различни математически модели. Освен това се използват различни критерии за оптималност.

2.1.2. Оптимална линейна филтрация. Уравнения на Винер-Хопф
Нека $x_{i,j}$ е значението на яркостта на изображението в точка с координати i,j , а модела с който се описва е:

$$y_{i,j} = f(x_{i,j}, n_{i,j}), \quad i = \overline{0, (I-1)}, j = \overline{0, (J-1)},$$

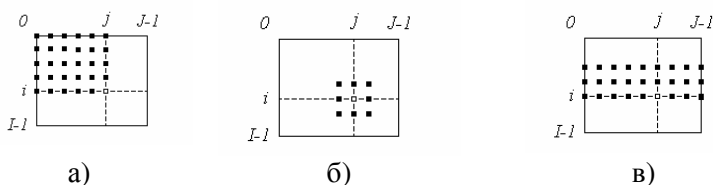
където - $n_{i,j}$ смущенията в точката с координати (i,j) ;

f - функцията описваща взаимодействието на сигнала и смущенията;

I и J - броят на редовете и колоните в изображението.

За начало на декартовата координатна система се взема горният **ляв** ъгъл на изображението. Околността на точката която се обработва може да бъде няколко типа (фиг.2.1). Според типа имаме: *казуална* (фиг.2.1а), *неказуална* (фиг.2.1б) и *полуказуална* (фиг.2.1в) филтрация. Типа се определя според взаимното положение на околността и обработваната точка. Смисълът заложен в тази класификация е следният: филтърът не може да оказва влияние на елементите от входното изображение, които не са постъпили към момента на обработка. Това разбира се оказва влияние в динамичните системи, където процесите за зависими от времето. При цифровата обработка на изображения, това не е необходимо защото изображението предварително е съхранено в паметта.

При линейна филтрация обработеното изображение, се определя чрез линейна зависимост от входните данни.



Фиг.2.1. Пример за различните видове околности

$a(i_1, j_1)$ - тегловни коефициенти, чиято съвкупност представлява **двумерната импулсна характеристика на филтъра**. Ако областта S е крайна, то импулсната характеристика е ограничена и филтърът се нарича с крайна импулсна характеристика (КИХ), в противен случай - с безкрайна импулсна характеристика (БИХ).

Най-честият критерий, които се използва за оценка на качеството на обработка е *минимума на средноквадратичната грешка*.

Системата от n_s на брой уравнения се наричат уравнения на Винер-Хопф:

$$\begin{cases} B_{xy}(k,l) = \sum_{i_1, j_1 \in S} a_{i_1, j_1} \cdot B_y(k-i_1, l-j_1), \\ k, l \in S. \end{cases}$$

където $B_{xy}(k,l)$ - корелационна функция;

n_S - брой на точките в околността S .

Ако тази система я решим относно неизвестните $a(i_1, j_1)$, то ще намерим импулсната характеристика на линейният филтър, водеща до минимална средноквадратична грешка от филтрацията.

2.1.3. Маскова филтрация

В качеството на маска се използва множество от тегловни коефициенти, дефинирани за дадената околност S , която обикновено е разположена симетрично относно обработваната в момента точка. За целта се прилага квадрат с размери 3×3 . Използват се различни разновидности на маски, като една от най-често използваните е равномерната. На фиг.2.2а е дадено зашумено изображение при отношение сигнал/шум -5dB . Резултатът от масковата филтрация е даден на фиг.2.2б, като нивото на относителната грешка е 0.395 .

Количествено ефективността на дадена обработка се оценява с подобряването на отношението сигнал/шум.



а)



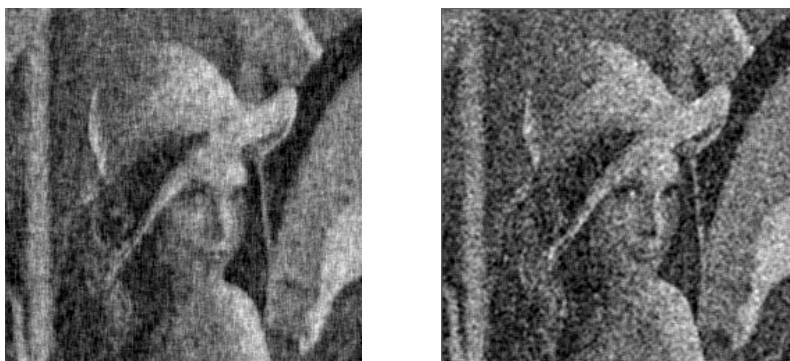
б)

Фиг.2.2. Пример за маскова филтрация при $q^2 = -5\text{ dB}$

2.1.4. Рекурентна казуална филтрация

Проблема за борба с шумовете не може да бъде решен напълно с използване на маскови филтри, тъй като там се отчита само влиянието на някаква околност на точката. Ако се увеличи размера на импулсната характеристика на филтъра, това ще доведе до увеличаване на изчислителната сложност.

Идеята тук се заключава в това да се използва двумерен БИХ филтър с такава импулсна характеристика, която да улеснява неговата практическа реализация и едновременно с това да се получи максимална ефективност на филтрацията. Типичен пример за такава филтрация е филтъра на **Калман**.



а) б)
Фиг.2.3. Пример за двумерна рекурентна филтрация

Рекурентният характер на тази филтрация е нейно положително качество. На всеки етап от обработката са необходими 3 умножения и 3 събирания, също така алгоритъмът е универсален и не зависи от отношението сигнал/шум. Тази филтрация изисква три пъти по-малко аритметични операции. Резултатите за същото изображение, но обработено с филтър на Калман за дадени на фиг.2.3а, а на фиг.2.3б за сравнение е даден резултата от масковата филтрация. Нивото на относителната грешка тук е 0,29.

Като недостатък на тази филтрация може да се посочи значителната загуба на острота в изображението, което се дължи на така наречените *динамични изкривявания* (нееднакво предаване на изхода на филтъра на отделните спектрални съставлящи на изображението).

2.1.5. Филтър на Винер за неказуална филтрация

Най-добри резултати от филтрацията се постигат при използване на неказуалния принцип, защото при него се използват абсолютно всички данни на изображението при обработката на всяка една точка. Един от най-известните варианти на линейна неказуална филтрация на изображения е филтърът на Винер. Технически този филтър се реализира чрез дискретно преобразуване на Фурие в честотната област. На фиг.2.4 е даден пример за работата на филтъра на Винер. Както и в предишните случаи експеримента е изпълнен при отношение сигнал шум - 5 децибела. Средно квадратичната грешка в този случай е 0,167 и е най-добра от всички разгледани до момента методи за линейна филтрация. За ниското ниво на остатъчните смущения говори и визуалната оценка на резултата. Това се постига със значително голяма разфокусировка на изображението дължаща се отново на динамичните изкривявания. Анализът на ефективността на този метод ще бъде непълна, ако не оценим и неговата изчислителна ефективност. Броят на умноженията за всяка точка от изображението $/256 \times 256/$ е 32, което е по-вече отколкото при останалите методи.



а)



б)

Фиг.2.4. Пример за Винеровска филтрация при $q^2 = -5$ dB

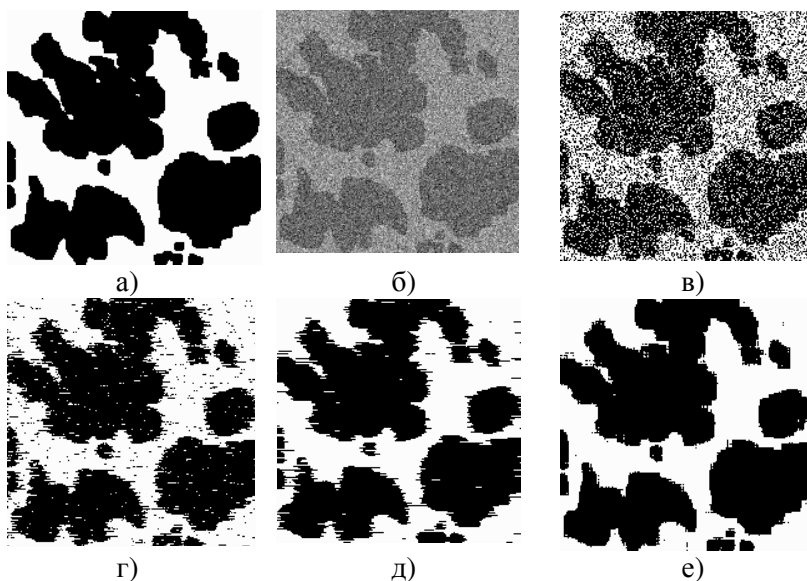
2.1.6. Беселовска филтрация

До тук разгледахме само методи за линейна филтрация. В практиката съществуват и *нелинейни процеси*. В този случай се

използва Беселовска филтрация. Още в началото е добре да кажем какви са нейните недостатъци:

1. Има високи изисквания към обема и характера на данните, които в практиката не винаги могат да бъдат изпълнени.
2. Реализирането на алгоритъма в класическия му вид води до значителни изчислителни трудности. Затова се налага да се използват алгоритми, които да преодоляват това ограничение.

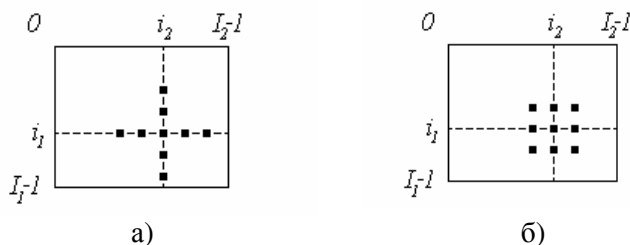
Един от тези методи е така наречената Марковска филтрация. Марковските процеси позволяват да се разработват оптимални процедури за филтрация. Резултати от проверката на този тип филтрация е дадено на фиг.2.5. На фиг.2.5а е показано тестовото изображение, на фиг.2.5б същото изображение с бял Гаусов шум, фиг.2.5в илюстрира използването на поелементна прагова обработка, а на фиг.2.5г, фиг.2.5д и фиг.2.5е са показани различни резултати от Марковската филтрация съответно от едномерна казуална, едномерна неказуална и двумерна неказуална филтрация. Вижда се, че качеството при използване на филтрация е значително по-добро. Вероятността за грешка при двумерната обработка достига 0,022, което е почти десет пъти по-малко отколкото при поелементната филтрация.



Фиг.2.5. Двуетапна Марковска филтрация на изображение

2.1.7. Медианна филтрация

Всички линейни алгоритми за филтрация водят до изглаждане на границите на промяна на яркостта. Това се дължи на факта, че



Фиг.2.6. Примерни прозорци за медианна филтрация

линейните процедури се явяват оптимални при Гаусово разпределение на сигнала и шума. Реалните изображения обаче не се подчиняват на това разпределение на вероятностите. Именно това е причината за лошото предаване на границите при линейната филтрация. Добро решение на този проблем е медианната филтрация разработена от Д.Таки през 1971 год. Трябва да се отбележи, че това е евристичен метод на обработка, чиито алгоритъм не подлежи на строго математическо описание. При използване на медианен филтър се извършва последователна обработка на всяка точка от изображението в резултат на което се образува последователност от оценки. В идейно отношение обработката в отделните точки е независима, но на практика при всяка една стъпка се използват по-рано извършените изчисления. При медианната филтрация се използва двумерен прозорец със централна симетрия като центъра е разположен върху текущата точка. На фиг.2.6 са показани два примера на най-често използваните прозорци: във вид на кръст и във вид на квадрат. Двумерният характер на прозорците позволява извършването на двумерна филтрация като за извършване на оценката се използват данните както от текущите ред и колона, така и от съседните. Обикновено се използват прозорци с нечетен брой точки.

Резултатите от експеримента илюстриращ работата на медианният филтър са дадени на фиг.2.7. Използваме прозорец с квадратна апертура и размер 3×3 . В лявата колона са дадени изображения със смущения, а в дясната резултати от медианната филтрация. На фиг.2.7а и фиг.2.7в е показано изображение със импулсни смущения, като процента на изкривявания е съответно 5% и 10%. Резултатите от обработката показват пълно премахване на смущенията в първия случай и значителното им отслабване във втория. На фиг.2.7д е показано изображение с Гаусов шум при отношение сигнал/шум -5 децибела, а на фиг.2.7е резултата от неговата медианна филтрация.

В Таблица 2.1 са дадени данните на средно квадратичната грешка и на коефициента на намаляване на шума при отношение сигнал/шум - 5 децибела, при различните методи на филтрация.

Най-голяма ефективност притежава двумерния филтър на Винер, а най-малка медианния. За сметка на това обаче както е видно от фиг.2.7е медианната филтрация в най-малка степен изглажда границите в изображението.



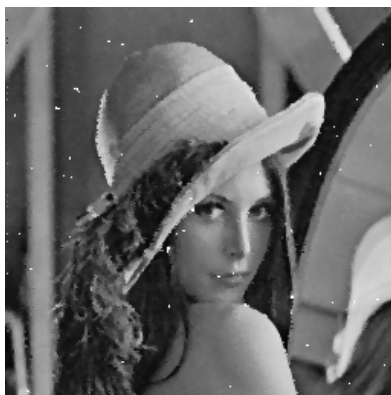
а)



б)



в)



г)



д)



е)

Фиг.2.7. Пример за медианна филтрация

Таблица 2.1. Сравнение на ефективността на потискане на шума при филтрация на изображения, $q^2 = -5$ dB.

	Масков филтър с оптимална КИХ	Масков филтър с равномерна КИХ	Двумерен рекурентен филтър	Двумерен филтър на Винер	Медианен филтър
δ_{ε}^2	0.309	0.395	0.29	0.186	0.539
γ	10.2	8.0	10.9	17.0	5.86

2.2. КОНВОЛЮЦИЯ

Нека разгледаме две крайни последователности $x(n)$ и $h(n)$ с дължина на реализациите N_1 и N_2 , т.е. $0 \leq n \leq N_1 - 1$ и $0 \leq n \leq N_2 - 1$ [3, 8, 14]. Линейна или аperiодична конволюция на двете последователности е последователността $y(n)$ определена с отношението

$$y(n) = \sum_{m=0}^n h(m)x(n-m),$$

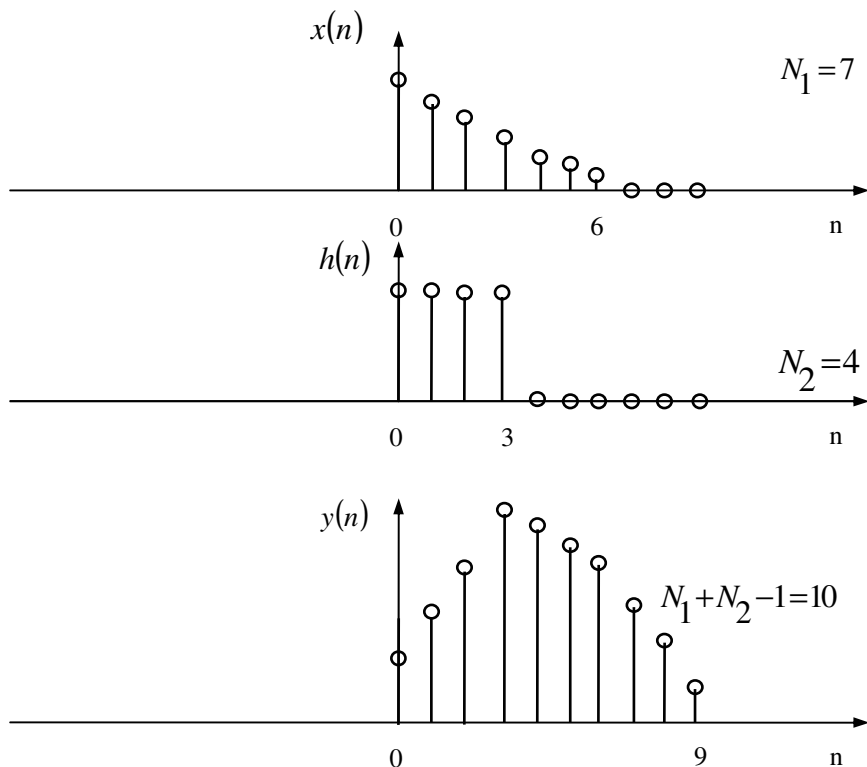
където $h(m)$ и $x(n-m)$ са равни на нула извън зададените интервали. На фиг.2.8 са дадени примери за последователностите $x(n)$, $h(n)$ и $y(n)$. Вижда се, че последователността $y(n)$ е крайна и има дължина $N_1 + N_2 - 1$ отчета.

Конволюцията на периодични последователности също е периодична и има същия период. Тъй като периодът на конволюцията $y(n)$ (фиг.2.8) е равен на $N_1 + N_2 - 1$ отчета, то за получаването на такъв период при кръговата конволюция е необходимо $x(n)$ и $h(n)$ също да съдържат $N_1 + N_2 - 1$, което се постига чрез допълнение на всяка от последователностите с необходимия брой нулеви отчета.

След това може да се намерят $(N_1 + N_2 - 1)$ - точковите Дискретни преобразувания на Фурие (ДПФ) на допълнените последователности, да се умножат и да се изпълни обратно ДПФ на произведението. В резултат се получава желаната конволюция $y(n)$.

На фиг.2.9 са илюстрирани еквивалентните периодични последователности използвани при изчисляването на кръговата конволюция. Допълнението на входните последователности с крайна дължина $x(n)$ и $h(n)$ с нулеви отчета води до изравняване на периодите на редиците и се отстранява кръговото припокриване. Всеки период на последователността $y_n(n)$ (фиг.2.9) съвпада с $y(n)$ (фиг.2.8). Разгледаният метод на изчисляване на конволюцията на две крайни последователности с прилагане на алгоритъма за ДПФ се нарича бърза конволюция. Терминът “бърза” се използва, защото ДПФ може да се изчисли бързо и ефективно, използвайки един от алгоритмите за БПФ. Може да се покаже, че даже и при не

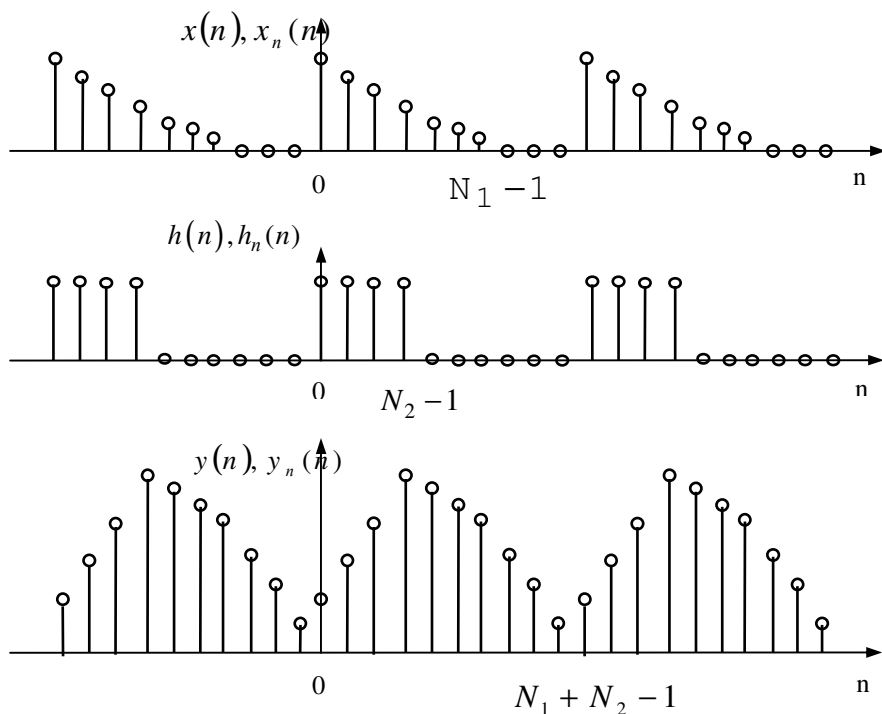
големи стойности на величината $N_1 + N_2 - 1$ (например 30) бързата конволюция е по-ефективна от пряката. Затова разгледаната методика е важна при обработка на сигналите.



Фиг.2.8. Примери за последователностите $x(n)$, $h(n)$ и $y(n)$.

При практическо приложение на метода, трябва да се отбележи, че размера на ДПФ може да не се ограничава до $N_1 + N_2 - 1$. ДПФ може да се изпълни и върху L на брой отчета, ако се удовлетворява условието $L \geq N_1 + N_2 - 1$. Възможността за избиране на произволно L е съществена, защото практическите алгоритми за изчисляване на ДПФ при различни L имат различна ефективност. Например за някои алгоритми е необходимо L да бъде степен на 2.

В този случай L трябва да е степен на 2 и не по-малко от $N_1 + N_2 - 1$.



Фиг.2.9. Еквивалентните периодични последователности използвани при изчисляването на кръговата конволюция.

2.3. ДИСКРЕТНО ПРЕОБРАЗУВАНЕ НА ФУРИЕ

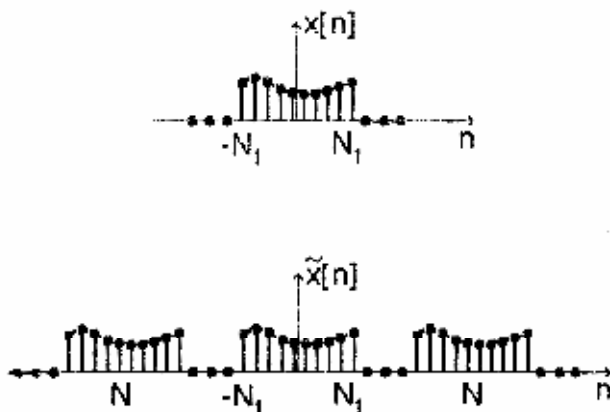
2.3.1. Дискретно преобразуване на Фурие

Коефициентите в реда на Фурие за периодичен правоъгълен сигнал може да се разглеждат като отчети на обвиващата функция и когато периодът на редицата нараства, тези отчети стават все по близо един до друг [9]. При непрекъснатите сигнали аperiodичният сигнал $x(t)$ се представя, като първо се намери

периодичен сигнал $\tilde{x}(t)$, който е равен на $x(t)$ за един период.

Когато този период клони към безкрайност, сигналът $\tilde{x}(t)$ е равен на $x(t)$, а представянето на $\tilde{x}(t)$ с ред на Фурие клони към преобразуването на Фурие на $x(t)$. Тук се прилага същата процедура за дискретни сигнали, за да получим представяне на апериодични редици с преобразуване на Фурие.

Да се разгледа обикновена апериодична редица $x[n]$, която има крайна дължина (фиг.2.10) за всяко число N_1 , тогава $x[n]=0$, ако $|n|>N_1$.



Фиг.2.10. Обикновена апериодична редица $x[n]$

От този апериодичен сигнал може да се получи периодична редица $\tilde{x}[n]$, за която сигналът $x[n]$ е един период. Ако се избере период N , редицата $\tilde{x}[n]$ е еднаква с $x[n]$ за по-голям интервал, а когато $N \rightarrow \infty$ $\tilde{x}[n] = x[n]$, за всяка стойност на "n".

Нека сега се разгледа представянето на $\tilde{x}[n]$ с ред на Фурие.

$$(2.1) \quad \tilde{x}[n] = \sum_{k=\langle N \rangle} \underline{A}_k e^{jk \frac{2\pi}{N} n},$$

$$\text{където } \underline{A}_k = \frac{1}{N} \sum_{n=\langle N \rangle} \tilde{x}[n] e^{-jk \frac{2\pi}{N} n}.$$

Тъй като $x[n] = \tilde{x}[n]$ в периода, който включва интервала $|n| \leq N_1$, удобно е да се избере интервалът на сумиране, да бъде именно този период и тогава

$$\underline{A}_k = \frac{1}{N} \sum_{n=-N_1}^{N_1} x[n] e^{-jk \frac{2\pi}{N} n}.$$

Като се използва факта, че $x[n]$ е нула извън интервала $|n| \leq N_1$ се получава

$$\underline{A}_k = \frac{1}{N} \sum_{n=-\infty}^{+\infty} x[n] e^{-jk \frac{2\pi}{N} n}.$$

Нека сега се дефинира обвивката на $N_{A_k} F_x(j\Omega)$ по следния начин:

$$F_x(j\Omega) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\Omega n},$$

$$\text{където } \Omega = k \frac{2\pi}{N}.$$

За коефициента A_k може да се запише:

$$(2.2) \quad \underline{A}_k = \frac{1}{N} F_x(jk\Omega_0),$$

където с Ω_0 е означено разстоянието $\frac{2\pi}{N}$ между отчетите.

Следователно коефициентите A_k са пропорционални на равно стоящите един от друг отчети на тази функция на обвивката. От комбинирането на формули (2.1) и (2.2) се получава:

$$\tilde{x}[n] = \sum_{k=\langle N \rangle} \frac{1}{N} F_x(jk\Omega_0) e^{jk\Omega_0 n},$$

Тъй като $\Omega_0 = \frac{2\pi}{N}$, следователно $\frac{1}{N} = \frac{\Omega_0}{2\pi}$ и

$$(2.3) \quad \tilde{x}[n] = \frac{1}{2\pi} \sum_{k=\langle N \rangle} F_x(jk\Omega_0) e^{jk\Omega_0 n} \Omega_0.$$

Когато $N \rightarrow \infty$, редицата $\tilde{x}[n]$ е равна на $x[n]$, за всяка крайна стойност на n и за $\Omega_0 \rightarrow 0$. Произведението в дясната част е периодично с период 2π . Следователно целият интервал на интегриране ще има широчина 2π и (2.3) може да се запише във вида:

$$x[n] = \frac{1}{2\pi} \int_{2\pi} F_x(j\Omega) e^{j\Omega n} d\Omega$$

и аналогично

$$F_x(j\Omega) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\Omega n}.$$

Функцията $F_x(j\Omega)$ се нарича преобразуване на Фурие за дискретен сигнал. Също така $F_x(j\Omega)$ се нарича още **спектр на $x[n]$** , защото дава информация за съставлящите $x[n]$ комплексни експоненти с различни честоти.

Бяхме приели, че $x[n]$ е редица с произволна, но крайна дължина, но формулите са валидни и за сигнали с безкрайна дължина, но трябва сумата във втората формула да бъде сходимата, а тя ще бъде ако редицата $x[n]$ е абсолютно сумируема, т. е.

$$\sum_{n=-\infty}^{+\infty} |x[n]| < \infty.$$

Дискретното преобразуване на Фурие (ДПФ) има много общи черти с непрекъснатото. Основните разлики са:

а) периодичност във времето на дискретното преобразуване $F_x(j\Omega)$;

б) краен интервал на интегриране.

Една от причините за все по-широкото използване на дискретни методи за анализ и синтез на сигнали и системи е разработването на достатъчно ефективни методи за извършване на фуриеров анализ на дискретни редици. Това вече предполага използването на компютърна и цифрова техника. За тази цел се използва ДПФ за сигнали с крайна дължина.

Нека $x[n]$ е сигнал с крайна дължина и нека съществува цяло число, такова че $x[n]=0$, извън интервала $0 \leq n \leq N_1 - 1$. Тъй като може да се състави периодичен сигнал $\tilde{x}[n]$, който е $x[n]$ за един

период. По - точно нека $N \geq N_1$ е дадено цяло число и $\tilde{x}[n]$ е периодичен сигнал с период N , такъв че

$$\tilde{x}[n] = x[n], \quad 0 \leq n \leq N-1.$$

Коефициентите в реда на Фурие се дават с формулата:

$$\underline{A}_k = \frac{1}{N} \sum_{k=\langle N \rangle} \tilde{x}[n] e^{-jk \frac{2\pi}{N} n}.$$

Ако се избере интервала на сумиране да бъде интервалът, за който $\tilde{x}[n] = x[n]$, се получава

$$\underline{A}_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk \frac{2\pi}{N} n}.$$

ДПФ на $x[n]$ се означава обикновено с $F_x(jk)$ и се дефинира с

$$(2.4) \quad F_x(jk) = A_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk \frac{2\pi}{N} n}, \quad k=0,1,\dots,N-1.$$

Множеството от коефициенти дефинирано с (2.4), обхваща ДПФ на $x[n]$.

Важността на ДПФ произтича от следните факти:

а) първоначалният сигнал с крайна дължина може да бъде възстановен от неговото ДПФ по

$$x[n] = \sum_{k=0}^{N-1} F_x(jk) e^{jk \frac{2\pi}{N} n}, \quad n=0,1,\dots,N-1.$$

б) съществува извънредно бърз алгоритъм за неговото пресмятане - Бързо преобразуване на Фурие (БПФ).

2.3.2. Двумерно дискретно преобразуване на фурие (ДПФ).
Методи за изчисляване.

Съществуват множество сигнали, които по принцип се явяват двумерни и трябва да се обработват с двумерни методи. Много от основополагащите идеи на едномерната обработка, могат да се използват и за двумерна. В същото време не всички едномерни понятия могат да се използват при двумерната обработка.

Нека $x(n_1, n_2)$ представлява двумерна последователност (n_1, n_2 - цели променливи). Това обозначение съответствува на дискретизирания непрекъснат сигнал $x(s, t)$, т.е.

$$x(n_1, n_2) = x(n_1 T_1, n_2 T_2) = x(s, t).$$

Значението на $x(n_1, n_2)$, се задава с височината на отсечките излизащи от точките с координати n_1, n_2 (фиг. 2.11).

Например цифровият единичен двумерен отскок се задава с формулата

$$u(n_1, n_2) = \begin{cases} 1, & n_1, n_2 \geq 0 \\ 0, & n_1, n_2 < 0. \end{cases}$$

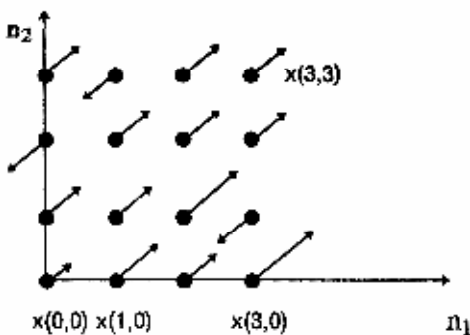
Двумерният периодичен сигнал удовлетворява условието

$$x_p(n_1, n_2) = x_p(n_1 + m_1 N_1, n_2 + m_2 N_2),$$

където N_1 - период по първата координата;

N_2 - период по втората координата;

m_1, m_2 - произволни цели числа.



Фиг.2.11. Значение на $x(n_1, n_2)$.

Както при едномерния случай, двумерният периодичен сигнал може да се представи като линейна комбинация от крайно число комплексни експоненти, като N_1 и N_2 са кратни на техните периоди.

$$(2.5) \quad x_p(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X_p(k_1, k_2) e^{j \frac{2\pi}{N_1} n_1 k_1} e^{j \frac{2\pi}{N_2} n_2 k_2}.$$

Коефициентите на Фурие $X_p(k_1, k_2)$ представляват амплитудата на $x_p(n_1, n_2)$ при $\omega_1 = \frac{2\pi}{N_1} k_1$ и $\omega_2 = \frac{2\pi}{N_2} k_2$. Значенията на коефициентите $X_p(k_1, k_2)$ може лесно да намерим, ако изчислим значението на двумерното z-преобразуване за един период на последователността $x_p(n_1, n_2)$ на честоти ω_1 и ω_2 .

$$(2.6) \quad X_p(k_1, k_2) = X(z_1, z_2) \Big|_{z_1=e^{j \frac{2\pi}{N_1} k_1}, z_2=e^{j \frac{2\pi}{N_2} k_2}} \\ = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_p(n_1, n_2) e^{-j \frac{2\pi}{N_1} n_1 k_1} e^{-j \frac{2\pi}{N_2} n_2 k_2},$$

където $0 \leq k_1, k_2 \leq N_{1/2} - 1$.

Формула (2.5) предствлява аналитичният израз на обратното двумерно дискретно преобразуване на Фурие (ДДПФ), а формула (2.6) - ДДПФ.

Съществуват три метода за изчисляване на ДДПФ.

а. При първия метод изчисляването на двумерното преобразуване се извършва чрез последователност от едномерни. За тази цел (2.6) се записва по подходящ начин, т.е.

$$X_p(k_1, k_2) = \sum_{n_1=0}^{N_1-1} e^{-j \frac{2\pi}{N_1} n_1 k_1} g_p(n_1, k_2).$$

При изменение на n_1 от 0 до $N_1 - 1$, сумата в квадратните скоби образува N_1 на брой едномерни ДПФ. Ще означим резултата от всяко с $g_p(n_1, k_2)$.

Получената формула представлява N_2 едномерни ДПФ.

Ако имаме една последователност с крайна дължина, която да означим с $x(n_1, n_2)$, нейните значения съвпадат за един период с значенията на периодичния еквивалент $x_p(n_1, n_2)$. Главното различие между $x(n_1, n_2)$ и $x_p(n_1, n_2)$ се състои в това, че преобразуването на Фурие от $x(n_1, n_2)$ се явява непрекъсната функция на честотите ω_1 и ω_2 , докато спектъра на $x_p(n_1, n_2)$ се състои от линии разположени на кратни честоти. По този начин двумерното ДПФ можем да изчислим, разбивайки го на ДПФ по редове и колони. Отначало се изчислява ДПФ за всеки ред x , резултатът се записва в междинен масив, след което се изчислява ДПФ за всеки стълб на този междинен масив. Може изчисляването да се извърши и по обратният ред. Необходимият брой комплексни умножения и изваждания е $c_{БПФ(2 \times 2)}$ и $c_{БПФ(2 \times 2)} = N_1 N_2 (N_1 + N_2)$.

б. При втория метод изчислението се извършва директно по формула (2.6).

Броят на комплексните умножения и изваждания е $c_{БПФ(2 \times 2)} = N_1^2 N_2^2$.

в. При третия метод се използва БПФ.

Двумерното ДПФ се разбива на по две по малки по размери ДДПФ, до тогава докато останат само елементарни двумерни ДПФ. Формула (2.6) може да разбием на четири суми:

- първата за четни n_1 и n_2 - $S_{00}(k_1, k_2)$;
- втората за четни n_1 и нечетни n_2 - $S_{01}(k_1, k_2)$;
- третата за нечетни n_1 и четни n_2 - $S_{10}(k_1, k_2)$;
- четвъртата за нечетни n_1 и n_2 - $S_{11}(k_1, k_2)$.

Основната клетъчна операция, която се извършва тук е с размерност 2×2 .

Броят на комплексните умножения е $C_{БПФ(2x2)} = \frac{3N^2}{4} \log_2 N$, а броят на комплексните изваждания - $C_{БПФ(2x2)} = 2N^2 \log_2 N$. Тук също е необходимо пререждане по време или честота. Сравнение между трите метода за изчисляване на ДДПФ е дадено в Таблица 2.2, като сме приели че имаме масив с размерност 1024×1024 ($N_1 = N_2$).

Таблица 2. 2. Сравнение между трите метода за изчисляване на ДДПФ

	първи метод	втори метод	трети метод
брой комплексни умножения	9 2,15.10	12 1,1.10	6 7,8.10
брой комплексни изваждания	9 2,15.10	12 1,1.10	7 2.10

ГЛАВА ТРЕТА: ЕТАПИ НА ОБРАБОТКА ПРИ КОМПЮТЪРНОТО ЗРЕНИЕ

3.1. ОПРЕДЕЛЯНЕ НА РЪБОВЕ И ГРАНИЦИ ЗА ОБЕКТИТЕ В ИЗОБРАЖЕНИЕТО.

3.1.1.Общи сведения

Изменението на яркостта на изображението определя неговото съдържание. Основните фактори, от които зависи яркостта на изображението, са:

- геометрични свойства на обектите;
- коефициентите на отражение на видимите повърхнини;
- осветеност на сцената;
- позиция на наблюдение.

Може да се добави, че от ъгъла на осветеност зависи пораждането на сенките. Обикновено различните повърхнини имат различна ориентация и коефициенти на отражение. На границите между повърхнините се получава изменение на интензитета на отразената светлина [10,17,19]. Стойностите на първата или втората производна на функцията на интензитета могат да се използват за определяне на ръбовете и границите на обектите в изображението (фиг.3.1).

В компютърното зрение, определянето на границите традиционно става чрез конволюция на сигнала с линеен филтър, обикновено отговарящ на първата или втората производна на функцията на интензитета.

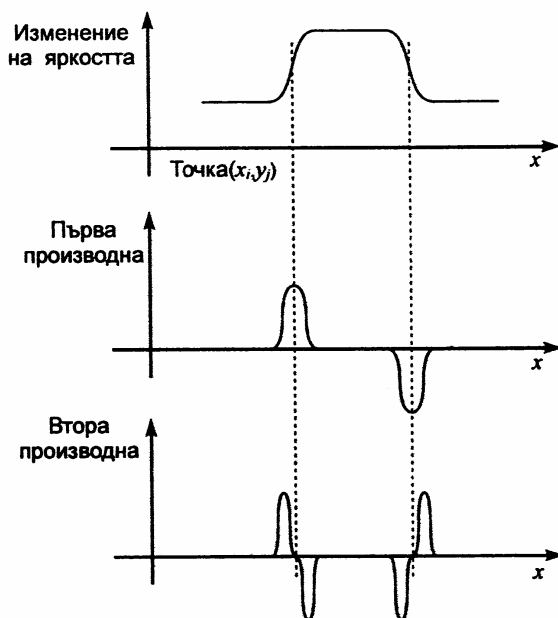
3.1.2. Градиентни оператори.

Изображението се съхранява в паметта на компютъра посредством неговото дискретно представяне. Оценката на производните за функцията се извършва посредством числови разлики, изпълнявани по определени схеми, наричани диференциални (градиентни) оператори. Градиента на функцията на изображението I се дава с вектора:

$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}.$$

Един от тези оператори е предложен от Робертс и носи неговото име. Той има следните конволюционни маски:

$$\begin{bmatrix} 0 \cdots 1 \\ -1 \cdots 0 \end{bmatrix} \text{ и } \begin{bmatrix} 1 \cdots 0 \\ 0 \cdots -1 \end{bmatrix}.$$



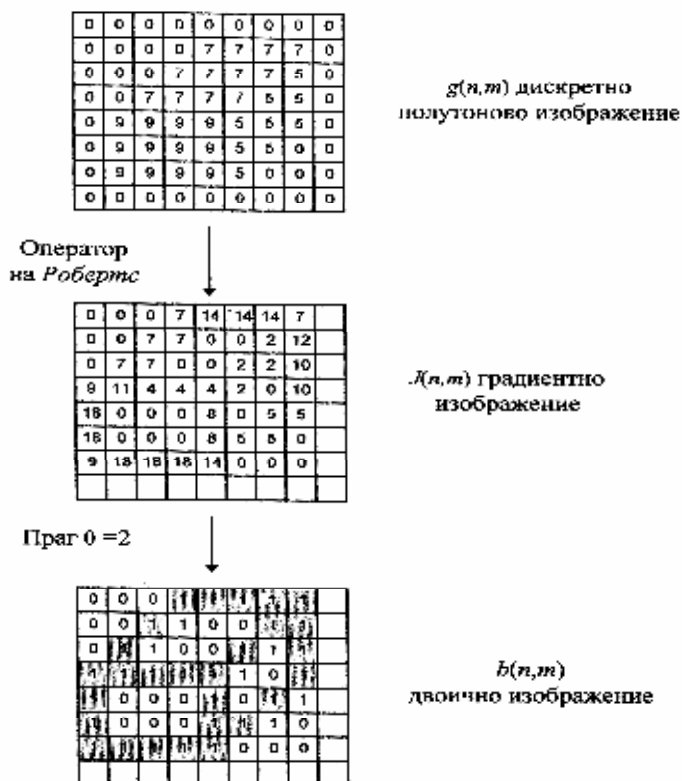
Фиг.3.1. Изменение на яркостта и производните и.

На фиг.3.2 е показано действието на диференциалния оператор на **Робертс**. Показани са получените градиентно и двоично изображение за $\theta = 2$ (където θ е праг, чиято стойност се определя от хистограмата на градиентното представяне). Вижда се как се очертават ръбовете на паралелепипед. За получаване на добро контурно изображение е необходима допълнителна обработка на двоичното изображение, например, получаване на ръбове с единична дебелина. При използването на апертура с размер 2×2 , оценките на градиента се отнасят за апроксимираната междинна точка, а не за пиксела. Използването на апертура с размер 3×3 решава проблема със въвеждането на интерполирана точка, за която се оценява стойността на градиента.

$$\begin{bmatrix} -1 & \cdot & 0 & \cdot & 1 \\ -2 & \cdot & 0 & \cdot & 2 \\ -1 & \cdot & 0 & \cdot & 1 \end{bmatrix}$$

Такъв е операторът на **Собел**. Той позволява по-точна оценка на градиента от този на Робертс. Ако се анализират диференциалните операторни формули на Собел може да се види, че те представляват скалярно произведение между вектора от стойности на дискретната функция в зададена апертура и стойностите на вектор-маска за същата апертура. Изборът на реални коефициенти затруднява изчислението. В практиката се избира най-близкият цял коефициент.

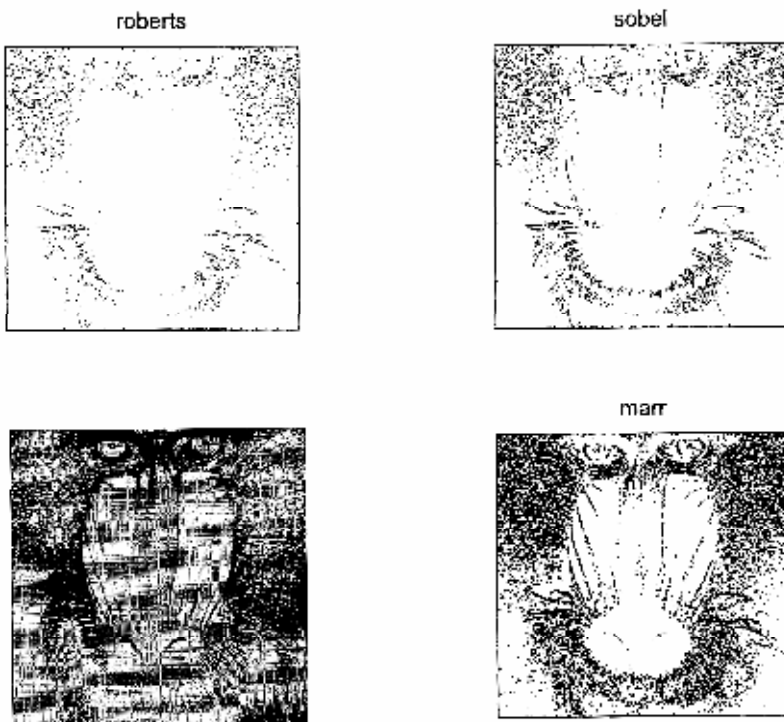
Обикновено в градиентното изображение се получават широки “хребети”, където има изменение на интензитета в полутоновото изображение. Необходимо е да се идентифицират ръбовете с тези широки хребети. За целта те се изтъняват, т.е. се **подтискат не-максимумите**. Този процес се състои в следното: сегментите се локализируют в точките на максимум на стойностите на градиента, а не-максималните стойности, перпендикулярни на посоката на сегмента, се нулират. Независимо от филтрацията, която се извършва предварително, са възможни фалшиви ръбове, породени от шум и текстура.



Фиг.3.2. Действие на диференциалния оператор на Робертс.

За да се редуцират фалшивите ръбове спрямо изтъненото градиентно изображение, се прилагат два прага в съотношение 1:2. Получават се две изображения b_1 и b_2 . Тъй като b_2 е формирано с по-висок праг, то съдържа по-малко фалшиви ръбове, но не може да има разкъсвания на контурите. Ето защо именно те се анализират във b_2 . Когато

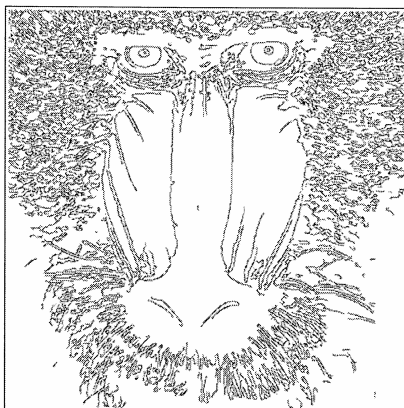
се достигне крайна точка в контура, алгоритъмът “гледа” $b1$ и съседството на точката в рамките на апертура 3×3 за евентуално свързване на контура. Алгоритъмът добавя ръбове от $b1$, докато разкъсването бъде свързано в ръб във $b2$. При използване на филтри с малка лента на пропускане се получават ръбове, показващи детайли в изображението, но са възможни и фалшиви ръбове благодарение на шум и текстура. При филтри с голяма лента на пропускане са губят ръбове, показващи детайли в изображението, но се редуцират фалшиви ръбове. Трябва да се търси компромис в зависимост от конкретната задача (фиг.3.3) [12].



Фиг.3.3.Резултати от обработката с различни оператори.

На базата на гореизложеното през 1963г. е създаден алгоритъмът на **Canny** (Jonh) (фиг.3.4), който се състои от следните етапи:

- входното изображение се изглажда с конкретен 2D Гаусов филтър;
- изгладеното изображение се диференцира с конкретен оператор;
- изтъняват се получените градиентни преходи, като се използва метода за подтискане на не-максимумите. Ръбовете се локализируют в точките на максимум на стойностите на градиента, а не-максималните стойности, перпендикулярни на посоката на ръба се нулират;
- спрямо полученото изтънено изображение, се прилагат два прага за извличане и свързване на ръбове в контури.



Фиг.3.4. Резултат от обработката с оператора на Canny.

Методите за получаване на градиентното изображение, използващи **втората производна**, са чувствителни към шумовете в изображението и ги усилват, тъй като дори много малки локални пикове на първата производна, водят до пресичане на нулата за втората производна.

Втората производна на функцията I се дава с нейният *Лапласиан*.

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}.$$

Обикновено преди прилагане на Лапласиан оператора изображението се филтрира с Гаусов филтър за намаляване на шума. Те използват факта на “*пресичане на нулата*” при втората производна за ефективно търсене на изменението на яркостта на изображението. Този оператор се нарича LoG (Laplacian of the Gaussian):

$$\nabla^2 (G \otimes I) = \nabla^2 G \otimes I$$

Сравнявайки разгледаните до тук градиентни оператори могат да се направят следните изводи:

1. Алгоритмите, използващи апертура 3x3 са по-шумоустойчиви от тези с апертура 2x2.
2. Операторът на **Робертс** води до разкъсвания в изображението.
3. Операторът на **Собел** дава по-добри резултати, в сравнение с подобните на него. Недостатък на този оператор е, че при импулсни шумове в еднородни области, открива фалшиви ръбове.
4. Въведените процедури в алгоритъма на **Canny** за предварителна Гаусова филтрация, локализация на интензитетните преходи и дострояване на разкъсванията в контурите водят до добри резултати в отделянето на ръбове и граници за обектите в изображението.
5. Методите, реализирани на базата на втора производна (**LoG**), водят до фалшиви ръбове и граници за обектите поради случайните пресичания на нулата.

3.1.3. Методи чрез анализ на спектъра на входното изображение в областта на пространствената честота.

Границите и ръбовете в изображението водят до получаването на големи стройности на пространствените честоти f_x и f_y след преобразуването на Фурие на функцията на интензитета $f(x,y)$.

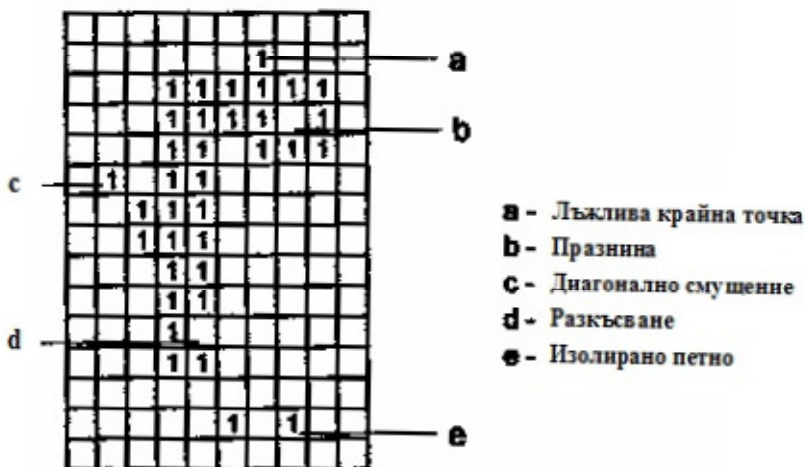
Следователно, за отделянето на ръбове и граници, е необходимо да се отделят високочестотните съставлящи от спектъра на входното изображение. Именно те носят информация за границите и ръбовете в изображението. Извършва се обратна трансформация в пространствената област. Получаването на двоичното изображение става чрез използването на конкретен праг. Освен трансформацията на Фурие се използват още трансформации на Уолш, Адамар и др. Споменатите трансформации имат три основни свойства, на които се основава използването им за отделяне на ръбове в изображението:

1. Запазване на енергията – енергията на трансформационните коефициенти е равна на енергията на входното изображение.
2. Компресиране на енергията – обикновено по-голяма част от енергията е компресирана в относително малък брой трансформационни коефициенти (10%).
3. Декорелация – корелацията между елементите на монохромните входни изображения обикновено е твърде висока (0.95). Чрез прилагането на трансформациите се достига висока декорелация.

Отделните ръбове и граници в двоичното изображение обикновено имат дебелина повече от 1 пиксел (2-3 пиксела), ако не са възприето някои специални методи за изтъняване (например, алгоритъма на Canny). Необходимо е тази дебелина да се редуцира в единична. Структурата и формата на обектите в изображението се запазва, но се намалява обема на данните. Обектите с единична дебелина са по-удобни за отделяне на топологични признаци, за сегментация и анализ. При изтъняването не трябва да се получават изменение на дължината на линиите и разкъсванията. За изтъняването се използват следните стратегии:

1. Изтриват се граничните елементи, докато се получи линия с единична дебелина;
2. Подчертаване на осевата линия между граничните точки. Отрицателно влияние при този метод могат да окажат

смушения (фиг.3.5), които предварително трябва да се филтрират.



Фиг.3.5. Смушения в двоичното изображение.

3.2. ОПРЕДЕЛЯНЕ НА ПРИЗНАЦИ. СЕГМЕНТАЦИЯ.

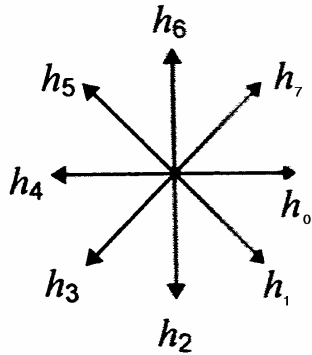
3.2.1. Определяне на топологично-особени признаци.

В резултат на предварителната обработка се отделят ръбовете и границите на обектите с единична дебелина. Получава се скелетът на обектите в изображението. Информацията за обектите се съдържа в тяхната структура. Структурата се характеризира от топологично-особените признаци, определящи свързаността на линиите. Признаците и линиите /сегментите/ се отделят и формират в съответни множества, на основата на които се извършва формално описание, анализ и разпознаване на обектите. Двоичното изображение **b** с размерност $N \times M$ се състои от две групи елементи. Тези, които принадлежат на скелета на изображението и образуват подмножеството **b₁** и подмножество **b₀** от елементи на фона.

$$B(i,j) = \begin{cases} 1 \cdot \cdot \cdot 3a \cdot \cdot b_1 \\ 0 \cdot \cdot \cdot 3a b_0 \end{cases}$$

5	6	7
4	$b(i,j)$	0
3	2	1

a)



b)

Фиг.3.6. Посоки и техните номера

За търсенето на конкретни конфигурации от точки от изображението се анализира локалното пространство на конкретен център $b(i, j) \in b_1$, определено от апертура 3×3 . Определят се връзките на $b(i, j)$, съгласно схемата на фигура 3.6.

$$H = \{h_i\} \dots i \in (0, 1, 2, \dots, 7)$$

$$\text{като } h_i = \begin{cases} 1, & \text{ако има единичен съсед} \\ 0, & \text{ако съседът е нула} \end{cases}.$$

За всеки елемент $b(i, j)$ се изчислява показател на свързаността му:

$$r(b(i, j)) = \sum_{i=0}^7 h_i.$$

Този показател определя свързаността на $b(i, j)$ със съседните му елементи, принадлежащи на скелета. Търсят се два вида

конфигурации от скелета: *характерни топологични признаци и сегменти.*

Множеството на признаците P определя топологично-особените точки, които ограничават сегментите в изображението:

$$P = \{P_1, P_2, \dots, P_t\},$$

където t показва броят на отделените признаци.

Възможна е следната стратегия за откриване на признаци:

1. Въвежда се праг θ (обикновено θ е равно на 2).
2. Проследява се скелетът на изображението.
3. За характерни признаци се определят елементите, в които става смяна на посоката на проследяване (признак “ъгъл” – елемент b на фиг.3.7).
4. За характерна се приема точка, от която излизат три или повече сегмента с последователно свързани елементи, чиито брой е по-голям или равен на θ (точка c).
5. Вътрешният елемент от линията (точка e), не се анализира за характерен признак.

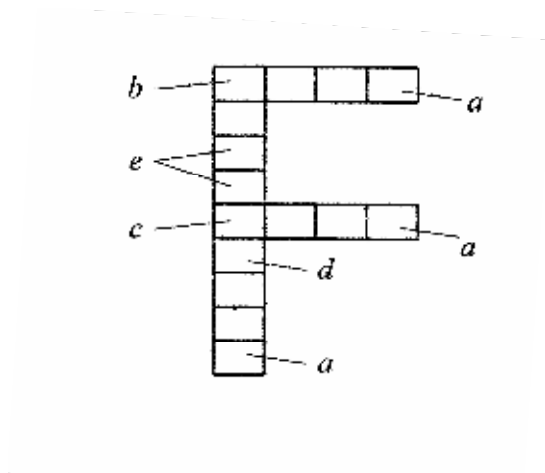
Топологичните признаци могат да се разглеждат като входни образи, а синтезиращите признаци да се използват за тяхната класификация.

3.2.2. Отделяне на синтезиращи признаци.

Формулираме следната задача: дадено е множество от входни образи $P = \{P_1, P_2, \dots, P_t\}$. Всеки образ се задава на черно-бяло рецепторно поле с размер $n \times n$. Образът може да се представи като двоичен вектор /последователност на редовете на полето отгоре-надолу/ $r_1, r_2, \dots, r_{n \times n}$, където $r_i = 0$, ако съответния елемент е бял или $r_i = 1$, ако съответния елемент е черен. Например за фиг.3.8 се получава векторът: $\{1111110000100001000010000\}$.

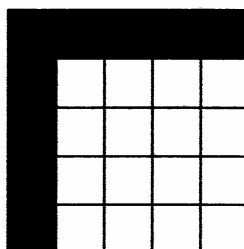
Трябва да намерим минимален набор от синтезиращи признаци $F = \{F_1, F_2, \dots, F_q\}$, който да удовлетворява условието:

$$P_i = \bigcup_{\alpha=W} F_{\alpha} \cdots i \in Z = \{1, 2, \dots, t\} \cdot W = \{1, 2, \dots, q\}.$$



Фиг.3.7. Конфигурации от точки.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25



Фиг.3.8. Двоично изображение на рецепторно поле 5x5.

Основното предположение е, че ако образите имат общи признаци, те се съдържат и в техните логически произведения (пресичания):

$$F_{\alpha} = P_i \cap P_j, \dots i \neq j, \dots i, j \in Z, \dots \alpha \in W.$$

Block, *Nilson* и *Duda* предлагат последователен и паралелен алгоритъм за отделяне на множество синтезиращи признаци, основани на анализа на логическите пресичания на образите. И в двата алгоритъма се задава праг, изхождайки от евристични съображения. Броя на получените признаци зависи от зададения праг. Това затруднява използването на алгоритмите.

Abdali предлага интересен алгоритъм за отделяне на синтезиращи признаци. Той се основава на метод на включване на атрибутните вектори. Основната идея е, че ако група атрибути по еднакъв начин участват в входните образи, те се отделят като признаци. Нека разгледаме този алгоритъм за конкретен пример от образи по *Abdali*, дадени на фиг. 3.9а. Матрицата на образите е дадена на фиг.3.9b. Всеки ред от нея показва как даден атрибут r_i ($i \in 1,2,\dots,9$) участва в съответните образи – с единица /черно/ или нула /бяло/. Редът за даден атрибут се нарича още атрибутен вектор. Последният стълб от тази матрица $\|r\|$, показва броя на единиците за съответния атрибутен вектор. Организира се матрица на включване на атрибутните вектори. Атрибутните вектори се подреждат в матрицата по нарастване на техните модули, като нулевите не се отчитат. Ако даден атрибутен вектор се включва или е равен на друг, то на съответното място в матрицата се записва единица, иначе се записва нула. Векторът, спрямо който се изпълнява отношението включване или равенство, се отбелязва с буквата **d**. На първия етап тези вектори не се разглеждат в операцията включване или равенство с останалите. Векторът с най-малък модул r_8 се проверява за включване с останалите. Очевидно е, че $r_8=r_8$ и на съответното пресичане на реда със стълба се записва 1. r_8 се включва в r_6 . На съответното пресичане се записва 1 и редът за r_6 се отбелязва с **d**. r_8 не се включва в други атрибутни вектори и на съответните места се записват нули. За атрибутните вектори, които не са отбелязани с буквата **d**, се извършват горните проверки за включване или равно. Единиците от всеки ред определят конкретен признак. Например, първият ред определя признак $F_1=\{000001010\}$, вторият $F_2=\{100110000\}$ и т.н.. Получените признаци са показани на фиг. 3.9д. По получените признаци не могат да се възстановят всички входни образи. Образите P_7 и P_8 не могат да бъдат възстановени от признаците F_1, F_2, F_3, F_4 . Потенциални признаци за възстановяване на всички входни образи трябва да се търсят за редовете, белязани с буквата **d**. За целта към матрицата на включване на атрибутите се добавят още два списъка В и С (фиг.3.9е). Списъкът В включва всички образи, в които рецепторът съответстващ на този ред присъства. По същество, това е списъкът, еквивалентен на атрибутния вектор.

Списъкът С първоначално е празен и служи за записване на образите, в които присъствието на дадения рецептор е взет под внимание, чрез някои от вече получените признаци. Ако за даден ред не се получи съвпадение за В и С, самият ред се отбелязва със *s* и се формира допълнителен признак по разгледания по-горе метод, т.е. съответният атрибутен вектор се проверява на отношението включване или равенство. Буквата *e* пред *d* за даден ред, показва, че зависимият атрибутен вектор е равен на друг и за него допълнително изследване не се прави. Определя се допълнителен признак F_5 , който се добавя към множеството признаци. Полученото множество признаци възстановява изходните образи.

3.2.3. Сегментация.

Смисъла на процеса на сегментация е в отделянето на фигурата от фона, чрез разбиване на изображението на области, избора на които се определя, така че те да съответстват на реалния обект или на негови части.

Сегментите свързват топологичните точки от множеството Р и се организират в множество L:

$$L = \{L_1, L_2, \dots, L_q\},$$

където *q* показва броя на отделените сегменти. Критерии за наличието на сегмент в дадена посока е съществуването на връзка:

$$L_i = \{(b(i, j)b(i + k, j + k)) / b(i, j), b(i + k, j + k) \in b_1, \dots k \in \{-1, 0, 1\}\}$$

Фримен е предложил верижен код за описание на линейни изображения с помоща на числата от фиг.3.6. Например, за кривата на фиг.3.10 се записва следния верижен код: 000000222444444. За вектора на два съседни елемента, имащи връзка, се определя разстояние *d* по формулата:

$$d = \begin{cases} 1 \cdot za \cdot h_i, \dots i = 0, 2, 4, 6 \\ \sqrt{2} \cdot za \cdot h_i, \dots i = 1, 3, 5, 7 \end{cases}.$$

Всеки сегмент свързва два топологични признака. За него се определя набор от атрибути:



P₁ P₂ P₃ P₄



P₅ P₆ P₇ P₈

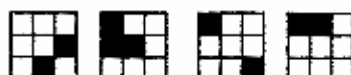
a)

	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	r
r ₁	1	1	1	0	1	1	1	1	7
r ₂	1	1	1	0	0	1	1	1	6
r ₃	0	0	0	0	0	0	0	0	0
r ₄	0	0	1	0	1	1	1	1	5
r ₅	0	0	1	0	1	1	1	1	5
r ₆	1	0	0	1	0	1	1	1	5
r ₇	0	0	0	0	0	0	0	0	0
r ₈	1	0	0	1	0	1	0	0	3
r ₉	1	1	1	0	1	0	0	1	5

b)

	r ₈	r ₄	r ₅	r ₆	r ₉	r ₂	r ₁
r ₈	1	0	0	1	0	0	0
r ₄		1	1	0	0	0	1
d r ₅							
d r ₆							
r ₉					1	0	1
r ₂						1	1
d r ₁							

c)



F₁ F₂ F₃ F₄

d)

	r ₈	r ₄	r ₅	r ₆	r ₉	r ₂	r ₁	B	C
r ₈	1	0	0	1	0	0	0	1, 4, 6	
r ₄		1	1	0	0	0	1	3, 5, 6, 7, 8	
e d r ₅								3, 5, 6, 7, 8	
s d r ₆				1	0	0	0	1, 4, 6, 7, 8	1, 4, 6
r ₉					1	0	1	1, 2, 3, 5, 8	
r ₂						1	1	1, 2, 3, 6, 7, 8	
d r ₁								1, 2, 3, 5, 6, 7, 8	3, 5, 6, 7, 8; 1, 2

e)



F₅

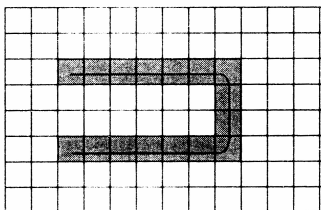
f)

Фиг.3.9. Отделяне на признаци по *Abdali*.

$$L_k = \{k, P_n, P_e, LS, BRE, CODE\},$$

където k е номер на сегмента, $k \in 1, 2, \dots, q$; P_n – номер на началния за сегмента топологичен признак, $n \in 1, 2, \dots, t$; P_e – номер на крайния за сегмента топологичен признак, $e \in 1, 2, \dots, t$; LS – дължина на сегмента; BRE – брой на кодовите елементи в сегмента; $CODE$ – кодова верига с дължина BRE .

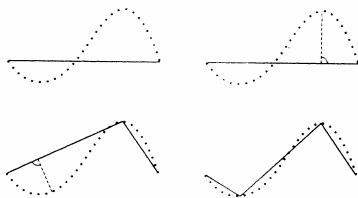
Необходимо е да се извърши класификация на сегментите L до отсечки и дъги.



Фиг.3.10. Верижен код по Фримен.

Дуда и Харт предлагат прост алгоритъм за определяне на линейни сегменти в двоичното изображение:

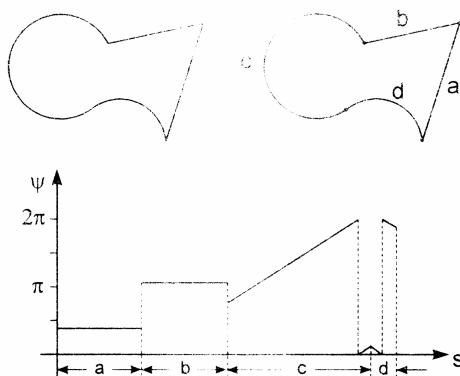
1. Избират се крайните точки в дадена последователност. Между тези точки се прокарва права апроксимираща линия.
2. За всяка точка от последователността се изчислява разстоянието и до апроксимиращата линия. Ако тези разстояния са по-малки от даден праг то отиваме на точка 4.
3. Намира се най-голямото от разстоянията и съответната точка се взема като възел, спрямо който последователността от точки се апроксимира до два нови линейни сегмента.
4. Ако има последователност от точки и съответна апроксимираща линия за изследване, то преминаваме към точка 2.
5. Край.



Фиг.3.11. Определяне на линейни сегменти в двоично изображение

На фиг.3.11 е показано действието на алгоритъма. Недостатък на този алгоритъм, е че случайни точки в двоичното изображение могат да доведат до лоши резултати.

Turner предлага алгоритъм за апроксимация на последователността от точки чрез прави линии и дъги. Използва се $\psi - s$ крива (фиг.3.12) за намирането на прави линии и дъги в двоичното изображение. ψ е тангентата спрямо точките от последователността. Функцията ψ се изчертава срещу s -дължината на съответната последователност от точки. Вижда се, че функцията е периодична с прекъснати скокове между 0 и 2π . Хоризонталните участъци на $\psi - s$ кривата съответствуват на прави линии в изображението, а наклонените на дъги.



Фиг.3.12. Апроксимация на последователност от точки чрез отсечки и дъги.

3.3. ОСНОВНИ ПРИНЦИПИ И МЕТОДИ НА РАЗПОЗНАВАНЕТО НА ОБРАЗИ

Основните задачи, които се решават при компютърното разпознаване на образи са [6, 11]:

- Представяне на измерените данни за обектите, които подлежат на разпознаване;
- Предварителна обработка на данните;
- Отделяне на характерни признаци за обектите;
- Построяване на описания;
- Обучение;
- Намиране на решаващи правила (функции), на основата на които се извършва класификацията.

За някои от задачите би направено конкретно разглеждане. Данните за обектите се формират обикновено като вектори или матрици. Да предположим, че буквено-цифрови символи се представят в поле (апертура) с размери 5x5 - фиг.3.13.



Фиг.3.13. Образи.

Когато полето $n \times m$ не е голямо ($n, m = 2, 3, 4, 5, 6, 7, 8$) по-ефективно оказва представянето на горните образи като вектори (в случая последователно по редове):

$$P_1 = (1111110000111111000010000),$$

$$P_2 = (1111110000111111000011111),$$

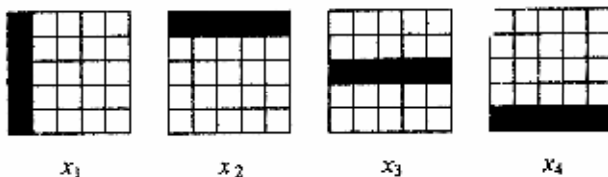
$$P_3 = (1000010000100001000011111).$$

За входното множество от образи може да се определи набор от признаци, на основата на които се описват образите. За образите на фиг.3.13 като признаци могат да се разглеждат вертикалните и хоризонталните линии (фиг.3.14). Векторите на образите ще бъдат:

$$P_1 = (x_1 x_2 x_3 \overline{x_4}) = (1110),$$

$$P_2 = (x_1 x_2 x_3 x_4) = (1111),$$

$$P_3 = (x_1 \overline{x_2} \overline{x_3} x_4) = (1001).$$



Фиг.3.14. Общи признаци.

Вижда се, че значително намалява размерността на векторите за образите (от 25 до 4). Като се има предвид, че на основата на тези признаци да се синтезират $2^n - 1$ образа, то ползата в търсенето и отделянето на признаци и описанието на образите посредством тях е очевидна. Признаци, характеризиращи даден клас от образи се наричат вътрешно-класови (x_1 , x_2 и x_3 за P_1). Признаци, определящи разликата между образите от различни класове се наричат междукласови (x_4 за P_1 и P_2). Последните са много полезни за класификацията, а първите за разкриване структурата на образите от даден клас.

Когато признаците се изразяват с реални числа, то векторът на даден образ може да се разглежда като точка в n -мерното евклидово пространство. В този случай образите от даден клас ще съответствуват на близки точки и ще образуват област, съответстваща на този клас.

Принципите на класификация на образи могат да се сведат до следните три: **посредством еталони; на основата на общите признаци за даден клас; използвайки клъстери.**

Класификацията на образи посредством *еталони* означава, че за всеки клас ω_i ($i \in 1, 2, \dots, t$, където t е брой на класовете) се формира еталон. Възможно е за класа да се определи набор от еталони, отчитащи допустими отклонения в рамките на класа:

$$\omega_i = (e_i \cdot OR \cdot e_i^1, e_i^1, \dots, e_i^r)$$

където e_i , - единствен еталон, формиран за класа, а e_i^j ($j = 1, 2, \dots, r$) - множество еталони, определени за класа.

Всички еталони за различни класове се запомнят в паметта на системата. Обектът, който подлежи на разпознаване се сравнява последователно с еталоните и се отнася към този клас, с чийто еталон е съвпаднал. Очевидно е, че този принцип е чувствителен към шумове. Той може да се използва успешно за класификация на стилизовани обекти.

Принципът на *общите признаци* предполага, че образите от даден клас ω_i , имат общи свойства:

$$\omega_i = (x_1, x_2, \dots, x_n)$$

В паметта на системата се запамятват признаците, характеризиращи отделни класове. За обекта, който подлежи на разпознаване се отделят признаци. Тези признаци се сравняват с признаците за съответните класове. Обектът се отнася към класа, за който се е получило най-голямо подобие. Този принцип изисква добри методи за отделяне на признаци.

Принципът на *клъстеризация* се използва когато всеки образ се представя като вектор от признаци (в общия случай n на брой), които са реални числа. Образите от даден клас образуват компактна област (клъстер) в n -мерното евклидово пространство на признаците:

$\omega_i \rightarrow K_i = (a_1, a_2, \dots, a_r)$ където K_i – клъстер; a_j ($j = 1, 2, \dots, r$) - точки на образите, образувачи клъстера в n -мерното пространство на признаците – $a_i = (x_1^i, x_2^i, \dots, x_n^i)$.

Клъстерите за класовете имат конкретно пространствено разположение. Ако клъстерите не се препокриват, то за обекта, който подлежи на разпознаване се изчислява до кой клъстер е най-близо (геометричното разстояние е най-малко – d_{\min}) и се причислява към съответния клас. Когато клъстерите се препокриват се използват вероятностни методи за класификация. Препокриване се получава в резултат на шумове. Намаляването им се достига с увеличаване на броя и качеството на признаците за образите. Следователно за класификацията на образите е важно съответните им клъстери да бъдат компактни и разделими. Фиг.3.15 илюстрира трите принципа на разпознаване.

Трите принципа могат да се илюстрират за образите, показани на фиг.3.13 по следния начин:

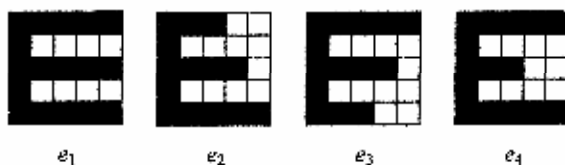
1) 25-разрядният двоичен вектор за всеки образ може да се разглежда като еталон за съответния клас. За даден клас могат да се използват няколко еталона фиг.3.16. Векторът на образа, който подлея класификация (входният образ) се сравнява с тези еталони и съответно се класифицира.



Фиг.3.15. Принципи за разпознаване на образи.

2) Чрез суперпозициите се определят кои са общите признаци на съответните класове. За входния образ се отделят признаци. Строи се неговата суперпозиция, която се сравнява със суперпозициите за класовете и съответно се извършва класификация.

3) Ако стойността на всеки елемент от полето 5×5 (за полутонови от 0 до 255, а за двоични - 0 и 1) се разглежда като признак, то в 25-мерното пространство на признаците се формират клъстери за съответните класове. За входния образ се изчислява разстоянието до съответните клъстери. Определя се минималното разстояние и съответно се класифицира образа.



Фиг.3.16. Няколко еталона за един клас.

Разгледаните принципи на разпознаване се реализират посредством три основни метода: **евристичен, математически (геометричен) и структурно-лингвистичен (синтактичен)**. Методите взаимно не се изключват и често се използват комбинирано. *Евристичният метод* се прилага за решение на конкретни задачи на основата на опита и интуицията на експерта, използвайки главно принципите на разпознаване посредством еталони и общи признаци за класовете. Методите за извличане на признаци и правилата за класификация се определят до голяма степен от характера на решаваните задачи. *Математическите методи* предлагат общи формализми за класификация на образи, използвайки принципите на еталоните, на общите признаци и клъстеризацията. *Лингвистичният метод* на разпознаване използва принципа на общите признаци. Признаците, които се отделят за обектите в изображения използват като терминален речник на формален език за описание на обекти. Правилата за описание се дефинират от конкретна граматика. Този метод се нарича още *синтактичен*, тъй като анализът и разпознаването се извършват на основата на синтаксиса на описанията, определен от граматиката. Често се използва и понятието *структурно-лингвистичен* метод, поради това, че описанията съдържат информация за структурата на обектите, което е особено важно за анализа и разпознаването на двумерни и тримерни обекти от визуалното изображение в системите за компютърно зрение (фиг.3.15).

За да се реализира даден метод на разпознаване е необходимо системата предварително да бъде **обучена**. Два метода на обучение се използват: *с учител и без учител*. При обучението *с учител* са

известени броят на класовете и алгоритъмът на класификация, които се съобщават от учителя. Алгоритъмът се представя във вид на правила, според които се извършва класификацията, като се определят и имената на класовете. Всеки клас се свързва с представителен образ (или образи). Обучението *без учител* се извършва въз основата на множество примери, като алгоритъмът на класификацията не е известен. На основата на информацията за образите от обучаващото множество от примери се търсят признаци и правила за класификацията. Критериите за определянето на класовете се посочват от човека-експерт. За класовете се определят условни номера, които могат да бъдат заменени с конкретни имена. Обучението завършва, когато за примерите от обучаващото множество се получава приемлива класификация. За проверка на обучението (с учител или без учител) се извършва *тестване с помощта* на множество от образи. Ако резултатите от тестването са приемливи се минава към разпознаване на входни образи. В процеса на тестване и разпознаване на входните образи е възможно дообучение на системата. За по-добра работа на разпознаващата система, важно е обучаващото и тестовото множество да имат добра *представителност* на възможните образи за класовете.

3.4. КЛАСИФИКАЦИЯ НА ОБРАЗИ ЧРЕЗ ЕТАЛОНИ И ВЕРОЯТНОСТЕН ПОДХОД

Чрез използване на ниско честотно филтри (НЧФ) могат да бъдат премахнати високочестотните шумове и малки структури в изображенията, които не носят информация. Използват се и нелинейни филтри. За подтискане на импулсни шумове се прилага преди всичко медианен филтър. Филтрацията се извършва като правоъгълен прозорец с размери 3×3 се движи по изображението и стойността на централния пиксел се определя по дадена зависимост от околните. Като резултат се намалява изменението на стойностите на граничните пиксели и така се получава изображение близко до оригинала. Прозореца с такива размери позволява лесно изчисляване на логическите функции, проверяващи качеството на централната точка. Филтрите

осигуряващи това се наричат пространствено-инвариантни филтри.

За премахване на импулсни смущения в практиката обикновено се използва медианна филтрация. Получените резултати, показват, че за типове изображения, където е най-важно да се запази формата, медианната филтрация не е ефективна. Характеристиката на работа на линейния филтър е по-добра по отношение на неясните очертания. Върху линейно филтрираното изображение се прилага и “прагова” (threshold) филтрация. “Праговият” алгоритъм е основан на статистическата теория и определя дали пикселът е повреден от импулсен шум или не. Това става по следните формули:

$$(3.1) \quad b_{i,j} - T_0 < \frac{T_{i_{\min}} - S_{\max}}{3} ,$$

$$(3.2) \quad b_{i,j} - T_0 < \frac{T_{i_{\max}}}{3} ,$$

където $b_{i,j}$ - елементите на прозореца с размерност 3×3 ;

T_0 - централният елемент на прозореца;

$T_{i_{\max}}, T_{i_{\min}}$ - максимален и минимален елементи в прозореца;

S_{\max} - максимален елемент в цялото изображение.

Ако условието (3.1) е изпълнено, централния елемент (пиксел) е равен на нула. Ако е вярно (3.2) централния пиксел е равен на S_{\max} .

Ако и двете условия не се изпълняват, то той запазва стойността си. Ако зависимостта между обработваното и записаното предварително еталонното изображение е линейна то:

$$(3.3) \quad E = a + bO ,$$

където a и b са коефициенти. След тяхното определяне, формула (3.3) добива вида

$$E - m_E = R_{EO} \frac{\sigma_O}{\sigma_E} (O - m_O) ,$$

където σ_E и σ_O са съответно средноквадратично отклонение на еталона и обработваното изображения;

R_{EO} - коефициент на взаимна корелация;

m_E, m_O - математическо им очакване.

Тъй като лявата част на това уравнение и σ_E са константи, се получава

$$R_{EO} \sigma_O \left(\begin{bmatrix} o_{11} & o_{12} \dots & o_{1N} \\ o_{21} & o_{22} \dots & o_{2N} \\ \dots & \dots & \dots \\ o_{N1} & o_{N2} \dots & o_{NN} \end{bmatrix} - m_O \right) = const.$$

От този израз следва, че обработваното изображение при тези условия се характеризира чрез m_O , σ_O и R_{EO} . Тъй като m_O , σ_O не са устойчиви признаци, то за информативен признак при разпознаването се избира **коефициента на взаимна корелация**. Той се пресмята по формулата

$$(3.4) \quad R_{EO} = \frac{B_{EO} - m_E \cdot m_O}{\sqrt{\sigma_E^2 \cdot \sigma_O^2}},$$

където B_{EO} е взаимно корелационната функция на еталона и обработваното изображение;

σ_E^2, σ_O^2 - дисперсия на еталона и обработваното изображение.

Позовавайки се на централната гранична теорема данните за еталона и обработваното изображение са разпределени по нормален закон, и определения по формула (3.4) коефициент на взаимна корелация характеризира само линейната вероятностна зависимост между двете величини.

Взаимно корелационната функция се изчислява по формулата:

$$B_{EO} = \sum_{i=1}^N \sum_{j=1}^N (e_i - m_E)(o_j - m_O) p_{ij},$$

където p_{ij} е вероятността за съвместно настъпване на събитията $E=e_i$ и $O=o_j$. Математическото очакване и дисперсията се намират чрез стандартните изрази:

$$m_E = \sum_{i=1}^N e_i p_i, m_O = \sum_{j=1}^N o_j p_j,$$

$$\sigma_E^2 = \sum_{i=1}^N (e_i - m_E)^2 p_i, \sigma_O^2 = \sum_{j=1}^N (o_j - m_O)^2 p_j,$$

където $p_{i(j)}$ е вероятността случайната величина $E(O)$ да приеме стойност $e_i(o_j)$.

В базата данни се записва само едно изображение на обработваното, независимо под какъв ракурс. По този начин многократно се намалява обема на необходимата база от данни. Щом при разпознаването се получи $R_{EO} \geq R_{EOtr}$, където R_{EOtr} е “праг” за разпознаване, то приетото изображение принадлежи на този клас изображения.

Избирането на R_{EOtr} става след определянето на практическата граница на разсейване на коефициента на взаимна корелация, съответно $\Gamma\Gamma_{R_{EO}}$ - горна граница на разсейване и $\Delta\Gamma_{R_{EO}}$ - долна граница на разсейване, между които с вероятност γ , са заключени Р.100% от получените стойности. Изчисляването им се извършва по следните формули:

$$\Gamma\Gamma_{R_{EO}} = m_{R_{EO}} + \kappa \cdot \sigma_{R_{EO}}$$

$$\Delta\Gamma_{R_{EO}} = m_{R_{EO}} - \kappa \cdot \sigma_{R_{EO}},$$

където $m_{R_{EO}}$ е математическото очакване на R_{EO} ;

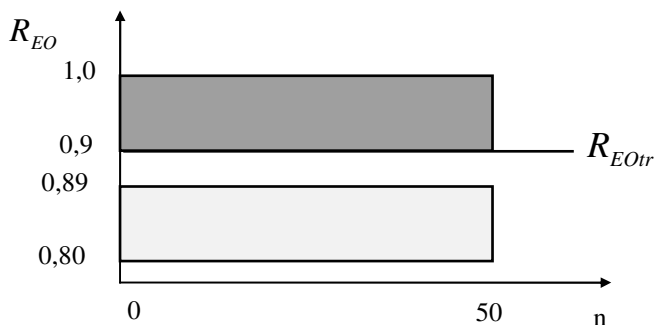
$\sigma_{R_{EO}}$ - средноквадратично отклонение;

κ - коефициент, определящ се от таблица .

Средноквадратичното отклонение се пресмята по формулата:

$$\sigma_{R_{EO}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (R_{EO_i} - m_{R_{EO_i}})^2},$$

където n е обема на извадката.



Фиг.3.17. Граници на R_{EO} .

На базата на определените граници на R_{EO} се избира $R_{EOir}=0.9$ (фиг.3.17). При постъпване на неизвестно изображение то се отнася към съответния клас, за който по гореописания метод R_{EO} между еталона и неизвестното изображение е по-голям или равен на 0,9, с вероятност 0.95.

3.5. КЛЪСТЕРИЗАЦИЯ

3.5.1.Клъстеризация

От предишната тема стана ясно, че пространствената и съдържателната организация на клъстерите оказват съществено влияние на класификацията на образите на основата на минималното евклидово разстояние. Следователно формирането на клъстерите е важен етап за класификацията на образите.

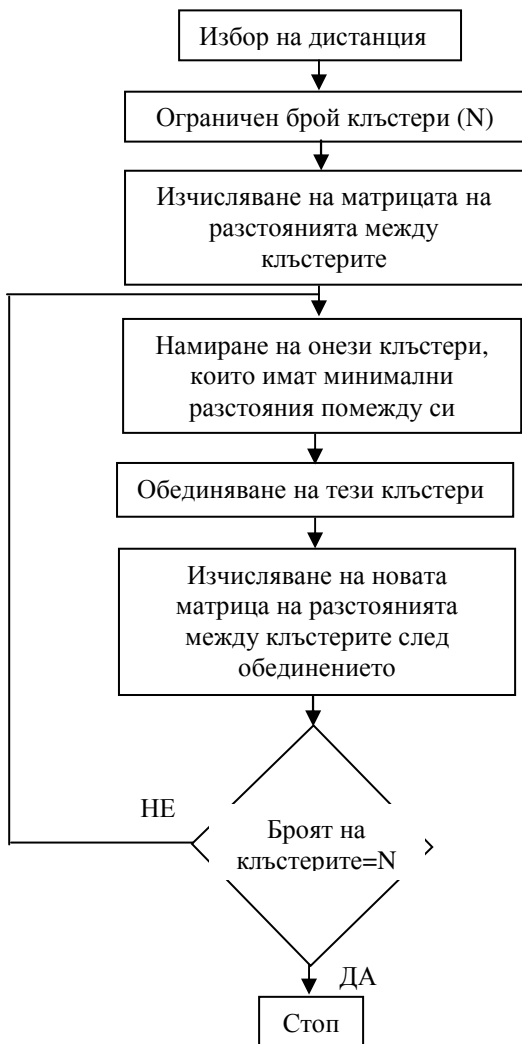
Клъстеризацията е групиране на данни с еднакви характеристики. Разделя се на йерархична и нейерархична клъстеризация.

При йерархичната клъстеризация подобността на клъстерите се оценява с използване на минимално разстояние. Минималното разстояние между клъстерите се дава с обединени клъстери след повтарящи се процедури от стартовите клъстери към финалния ограничен брой от клъстери.

При нейерархичната клъстеризация в началния етап случаен брой от клъстери се избира временно. Броят на елементите във всеки

клъстер се проверява по избраните параметри или минимално разстояние и се реорганизируют в съответни клъстери с висока отделимост. Пример за такава клъстеризация е ISODATA.

1. Всички елементи се дислоцират в близки клъстери чрез изчисляване на дистанцията между елементите и клъстерите.



Фиг.3.18. Алгоритъм за ефективна клъстеризация.

2. Центъра на тежестта на всички клъстери се пресмята отново и горната процедура се повтаря получаване на сходство.
3. Ако броя на клъстерите е под предварително зададения и разстоянието между клъстерите е под зададения праг, то клъстера се смята за завършен.

Постановка на задачата: имаме множество от образи $\{a_1, a_2, \dots, a_q\}$. Всеки образ се характеризира от n признака $\{x_1, x_2, \dots, x_n\}$. Необходимо е входното множество $\{a\}$ от образи да се раздели на m непресичащи се клъстъра(групи) така, че всеки образ от даден клъстър да е по-близо до образите в този клъстър отколкото до всеки образ от останалите клъстери. За клъстеризация се използва алгоритъма за **ефективна клъстеризация**, който се реализира като показател за качеството на клъстеризация (фиг.3.18).

Този показател се изчислява по следната формула :

$$J = \sum_{j=1}^m \sum_{a \in K_j} \|a - Z_j\|^2,$$

където m – брой на клъстърите, които сме задали;

K_j – клъстър съдържащ подмножество от образи;

Z_j – вектор от средните стойности на елементите в клъстъра K_j

$$Z_j = \frac{1}{r} \sum_{a \in K_j} a,$$

където r – брой на образите в клъстъра;

Показателят J определя сумата от квадратите от разликите на образите до центровете на техните клъстери. При минимизация на тази сума се постига ефективна клъстеризация.

Алгоритъм за клъстеризация :

- 1) Определят се m центъра на клъстерите : $Z_1 \dots Z_m$. Обикновено се вземат първите m образа.
- 2) $i = 1$ - параметър на цикъла (итерации).
- 3) Определят се най-близките образи до центровете, т.е. формират се m клъстъра $K_1^i, K_2^i, \dots, K_m^i$ - това са клъстърите формирани на i -тата итерация.

- 4) За клъстерите получени в точка 3 се определят нови коригирани центрове. Изхожда се от условието, че сумата от квадратите на разстоянията между новия център на зададения клъстер и образите в него да бъде минимална:

$$J_j = \sum_{a \in K_j^i} \|a - Z_j^{i+1}\|^2, \cdot \cdot j = 1, 2, \dots, m.$$

- 5) Ако се получат едни и същи центрове за клъстерите в две последователни итерации:

$Z_j^{i+1} = Z_j^i$ за всички $j=1, 2, \dots, m$ се преминава към 6 иначе се връщаме към 3.

- 6) Край

Задачата за клъстеризация се използва в медицината и други области за взимане на решения.

Алгоритъма за клъстеризация по същество представлява обучение без учител.

3.5.2. Линеини решаващи (разделящи) функции

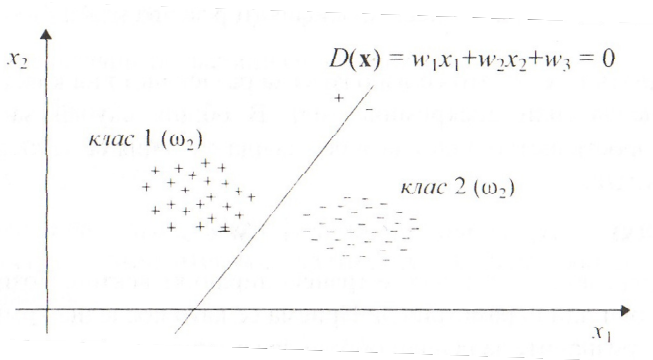
Ако определените клъстери /класове/ в n -мерното евклидово пространство са добре разделими, то съществуват хиперповърхнини, които могат да се разглеждат като граници за съответните класове. Възниква задачата за определяне и оптимизация на уравненията на хиперповърхнините. На фиг.3.19 са показани два класа, разделящата ги линия и нейното уравнение в общ вид.

Където $D(x)$ е линейна функция; $w_1, w_2 \cdot u \cdot w_3$ са коефициенти; x_1, x_2 – координати за съответния образ. За конкретни коефициенти и конкретни стойности за признаците на входния образ $x = (x_1, x_2)$ се изчислява функцията $D(x)$. Ако $D(x) > 0$, то образът се отнася към клас ω_1 . Ако $D(x) < 0$, то образът се отнася към клас ω_2 .

Ако $D(x) = 0$, се получава неопределеност и образът може да се отнесе към единия или другия клас. Предлага се находчива функция за класификация на образи в случая на два класа:

$$D'(x) = -\frac{1}{2}(\text{sign}D(x) - 3),$$

$$\text{където } \text{sign}(D(x)) = \begin{cases} +1, \dots \text{ако } D(x) \geq 0 \\ -1, \dots \text{ако } D(x) < 0 \end{cases}.$$



$$D(\bar{x}) = w_1x_1 + w_2x_2 + w_3 = 0$$

Фиг.3.19. Линейна разделяща функция.

Получената стойност за $D'(x)$ определя направо номера на класа.

Ако $D(x) > 0$ то $D'(x) = -1/2(1-3) = 1 \rightarrow \omega_1$;

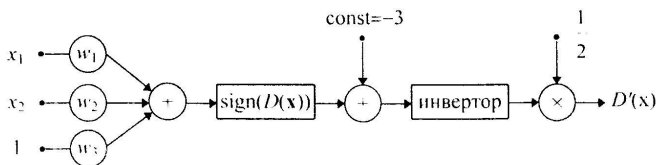
Ако $D(x) < 0$ то $D'(x) = -1/2(-1-3) = 2 \rightarrow \omega_2$.

На фиг.3.20 е показана схема на линейния класификатор в случай на два класа.

Функцията $D(x)$, която се използва за разделянето на класовете се нарича решаваща. В общия случай линейната решаваща функция се записва посредством формулата:

$$D(x) = w_1x_1 + w_2x_2 + w_nx_n + w_{n+1},$$

където $w=(w_1,w_2,\dots,w_n)$, а w^t е транспонираният вектор. Този вектор се нарича теглови или параметричен. Приема се като последна компонента на вектора на признаците за всички образи да се записва 1 т.е. $x=(x_1,x_2,\dots,x_n)$, а с w_{n+1} се допълва вектора на коефициентите: $w=(w_1,w_2,\dots,w_n,w_{n+1})$.



Фиг.3.20. Линеен класификатор за два класа.

Когато класовете са повече от два, използването на метода на линейните решаващи функции, също е справедлив. Обикновено се разглеждат три случая.

Ако са формирани клъстерите, между тях има разделяща линия, която има конкретна функция(уравнение).

В зависимост от $D(x)$, образът на обекта се отнася към единия или другия клас. Въпросът е как да се получи уравнението на разделящата повърхнина за n -мерното пространство (разделяща линия за двумерното пространство).

Общ вид на разделящата (решаваща функция) :

$$D(\bar{x}) = \sum_{i_1=1}^n \sum_{i_2=i_1}^n \dots \sum_{i_r=i_{r-1}}^n w_{i_1 i_2 \dots i_r} x_{i_1} x_{i_2} \dots x_{i_r} + D^{r-1}(x),$$

където r показва степента на нелинейност, а $D^0(\bar{x}) = w_{n+1}$.

Работи се n признака в n -мерното пространство и може да се запише :

$$D(\bar{x}) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_k f_k(x) + w_{k+1},$$

$$D(\bar{x}) = \sum_{i=1}^{k+1} w_i f_i(x) = w^t x',$$

където $w = (w_1, w_2, \dots, w_k, w_{k+1})$;

$$x' = (f_1(x), f_2(x), \dots, f_k(x), 1).$$

Това е векторно записване на решаващата функция. Винаги можем да подберем такова множество от функции, че да апроксимираме каква да е сложна решаваща функция. В зависимост от r и n получаваме голям брой коефициенти :

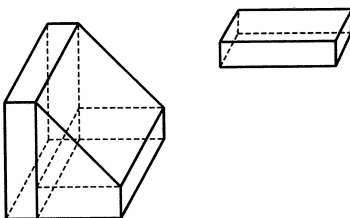
$$N_w = C_{n+r}^r = \frac{(n+r)!}{r!n!}.$$

За практическите приложения в повечето случаи не се използват всички членове на разложението на решаващата функция. Достатъчно е използването само на членовете със старшите степени на елементите на вектора x . Както се вижда решаващите функции зависят от много аргументи, чиито брой се определя от използваните признаци за входното множество образи. За конструирането на решаващи функции се използват ортогонални или ортонормирани функции (на основата на полиномите на Лъжандър, Лагер и Ермит).

3.6. АНАЛИЗ И РАЗЛАГАНЕ НА 3D ВИЗУАЛНИ СЦЕНИ.



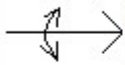


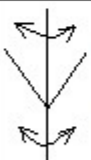

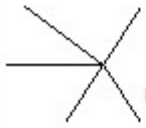
Обикновено анализът се извършва на основата на полученото *двоично контурно изображение* на сцената. Проблемът, който възниква като задача от областта на изкуствения интелект е да се интерпретира двумерната картина като тримерна сцена. Получените резултати в тази област, макар и скромни могат да се разглеждат като теоретична основа за решаването на поставения проблем [15,16,20]. Първите научни публикации в това направление са на *Робертс*. След получаването на двоично контурно изображение той предлага анализът на обектите в сцената да се извърши чрез *тримерни многостенни модели*. Всеки обект от сцената се разглежда като съставен. Целта е той да се представи като конструиран от прости тримерни модели. На фиг.3.21 сцената се представя посредством композиция от паралелепипед и призма.

Когато телата в сцената се допират или закриват едно от друго възниква въпросът за тяхното *разделяне*, т.е. *декомпозиция* на сцената до отделни тела. За тази цел Гузман предлага програма, наречена *SEE* (виждам), определяща контурите принадлежащи на дадено тяло и отделяща тялото като цяло. Гузман обръща внимание на *възлите* (*топологичните признаци*), които се формират от съединяването на ръбовете на телата. Той разглежда сцени, съдържащи *плоскостенни* тримерни тела. Наборът от обобщени типове възли и обясненията за тях са показани на фиг.3.22.



Фиг.3.21. Сцената представена посредством композиция от паралелепипед и призма.

Имената на възлите са определени, изхождайки от геометричните конфигурации на линиите във възела. Предлагат се връзки за контурните области, определени от даден възел, които могат да принадлежат на едно тяло. Например, областите определени от възела СТРЕЛКА често принадлежат на едно тяло. Възел Т обикновено показва закриване на едно тяло от друго и т.н. Три етапа са характерни за работата на програмата на Гузман. На първия етап програмата определя (разпознава) типа на възлите и възможните връзки между областите на възлите. Тази локална информация се анализира на втория етап и се предлагат хипотези за отделяне на телата в сцената. За тази цел се използват множество *евристики*. Първата от тях е: *две области принадлежат към едно и също тяло, ако между тях има поне една връзка*. Очевидно е, че тази евристика не е достатъчно силна и са възможни ситуации, например при съвпадение на възли, за които ще се получат неправилни връзки - фиг.3.23а. Втората евристика гласи: *две области принадлежат към едно и също тяло,*

	'L' възел, в който се свързват две линии
	'Вила' - разклонение от 3 линии с ъгли по-малки от 180^0 и има връзки между областите
	'стрелка' възел от три линии, единият от ъглите е по-голям от 180^0 и има връзки между областите
	'Т' възел - разклонение от 3 линии, две от които са колинеарни
	'К' - разклонение от 4 линии, две от които са колинеарни, а другите две са разположени от едната им страна
	'Ψ' - признак, който е разклонение от 4 линии, две от които са колинеарни, а другите две са разположени от двете им страни и има връзка между областите
	'ПИК' - възел, образуван от 4 или повече линии, Y координатата на възела е по-голяма от Y-координатите на съседните признаци и има връзка между областите
	'Мулти' - признак, образуван от 4 или повече линии и неприличащ на никой от горните

Фиг.3.22. Набор от обобщени типове възли и обясненията за тях.

ако между тях има две връзки - областите (AB) и (DE) за сцената на фиг.3.23в. С тази евристика не се допускат грешки характерни за първата, но тя се явява много силна за области имащи по една връзка, например C, F, R от фиг.3.23в. Отслабването и се извършва с евристиката: *отчитайки обединяването на области на основата на предишната евристика като цяло да се извърши обединяването на области, имици връзки с него* - областите (AB) със (C) и съответно получаваме (ABC). Следващата евристика продължава отслабването и гласи, *ако дадена област има само една връзка с друга област, те се обединява* - (ABCFR)(DE). Последната евристика гласи: *ако от съседните върхове се получава противоречива информация за връзките между областите, то се извършва допълнителен анализ на тези връзки, като някои от тях се отхвърлят*. Например, не се установяват връзки по отношение на линия, която в единия край образува външен възел тип СТРЕЛКА, влиза във възел тип L, или се явява закриваща във връх тип T.

В процеса на определянето на връзките се формират *графи*. Всеки възел на графа с номер съответствува на конкретна стена, а дъгите определят връзките между стените. На следващия етап възлите на всеки граф се поглъщат в едно ядро, определящо конкретно тяло - фиг.3.24. На фиг.3.25 е показана сцена *TRAIL*, използвана от *Гузман*. Програмата *SEE* за тази сцена дава следния резултат:

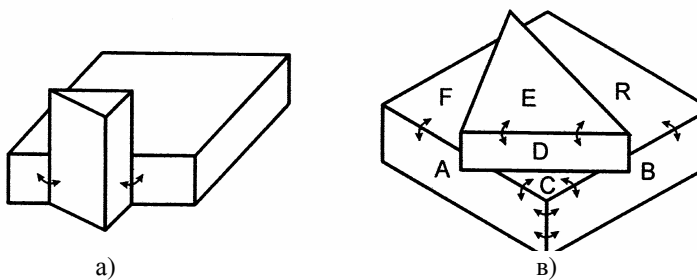
(BODY-тяло, IS-е)

(BODY1 IS:6:2:1)

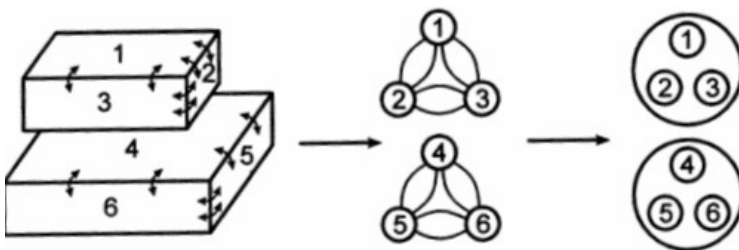
(BODY2 IS:11:12:10)

(BODY3 IS:4:9:5:7:3:8:13)

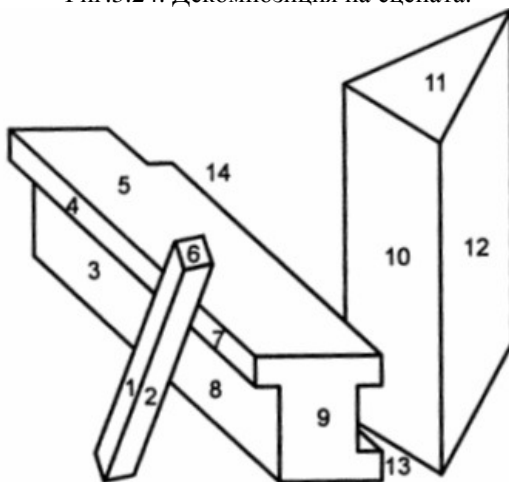
Както се вижда тази програма с успех извършва декомпозиция на тримерни сцени до обектите, съдържащи се в нея. Не е трудно, обаче, да се конструират примери на сцени, за които програмата *SEE* допуска грешки. Направените експерименти с програмата показали, че тя работи успешно за сцени с тела, чиито върхове се получават от пресичането на три плоски стени (тристенни върхове) и телата не съдържат отворстия. В този смисъл програмата на *Гузман* може да се разглежда като евристична теория, съгласно която, ако възелът се явява изображение на



Фиг.3.23. Използване на множество евристики.



Фиг.3.24. Декомпозиция на сцената.



Фиг.3.25. Сцена "TRAIL" по Гузман.

тристенен връх, той съдържа съответните връзки между областите на тялото. Тя оказва силно влияние на следващите разработки в тази област. Но на тази теория ѝ липсват дълбоки семантични корени, разкриващи причините за формиране на линии и връзки в тримерния свят и ограниченията, които налага този свят.

Дадена линия на изображението може да възникне поради следните причини, определени от физическата структура:

- 1) Линията може да бъде граница на сянка;
- 2) Тя може да се разглежда като пукнатина между два изместени обекта;
- 3) Ребро за две видими стени;
- 4) Граница между обекта и това, което се намира зад него (фона).

Независимо един от друг *Huffman* и *Clowes* предложили систематичен, семантичен подход при анализа на тримерни сцени. Те приемат ред ограничения и изхождат от физическия смисъл на линиите. Ограниченията за линейните изображения са следните:

- 1) Обектите в сцената са плоскостенни, в които всеки връх се образува от пресичането на три стени;
- 2) Обектите са в такава (основна) позиция, че нейното незначително изменение не води до изменение на типовете на върховете в линейното изображение;
- 3) Точката на наблюдение е избрана така, че нейното незначително изменение не води до изменение на типовете на върховете в линейното изображение.

Първото ограничение е много важно и силно, тъй като по същество за изображението се избягват върхове от типа К, У, ПИК, МУЛТИ, които се формират от пресичането на повече от три плоскости - фиг. 3.22. Възможни са върхове от тип L, ВИЛА, СТРЕЛКА, Т, което значително опростява разглеждането. Осветеността на сцената се приема такава, че *не се появяват сенки*. Линиите на изображението се класифицират в *три типа*, съгласно с техния физически смисъл, определящ се от структурата. Всеки тип се отбелязва с конкретна марка:

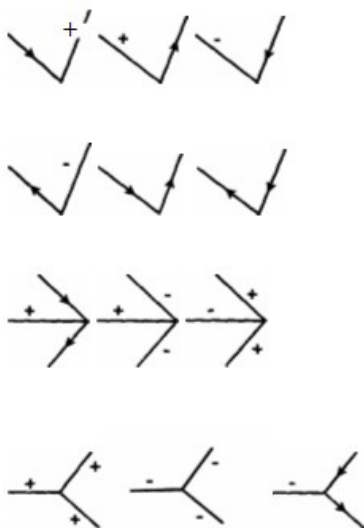
- 1) *Изпъкнала линия* (ръб), определяща се от две видими стени се отбелязва с "+";
- 2) *Неизпъкнала линия*, определяща се от две видими стени, се отбелязва с "-";

- 3) *Гранична линия*, определяща се от една видима стена, закриваща друга стена се отбелязва с една от стрелките \rightarrow или \leftarrow . Посоката на стрелката е такава, че при движение по нея закриващият обект остава отлясно.

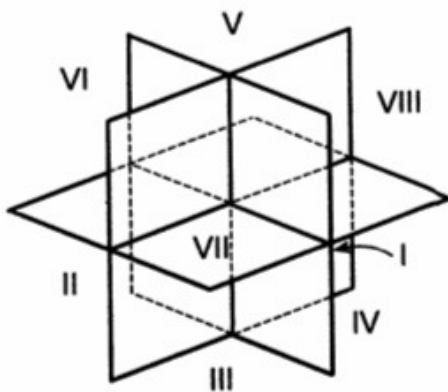


Фиг.3.26. Нереални върхове.

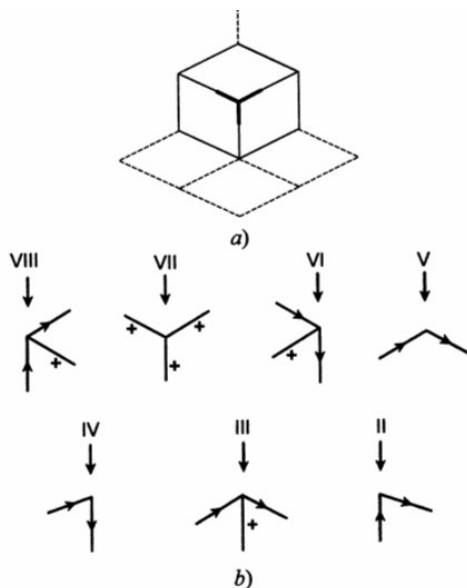
Линиите могат да се разделят на *гранични* и *вътрешни*. Граничните линии разделят области, принадлежащи на различни обекти, а вътрешните линии разделят области, отнасящи се към един и същ обект. От своя страна вътрешните линии се разделят на *изпъкнали* и *неизпъкнали*. Съгласно с приетата маркировка, всяка линия може да се маркира по един от четирите начина. Следователно, връх, имащ две линии може да се маркира по $4^2=16$ начина, а връх с три линии по $4^3=64$ начина. Приемайки горните ограничения, че линейното изображение съдържа върхове от тип L, ВИЛА, СТРЕЛКА, Т, то възможният брой различно маркирани върхове ще бъде $1 \cdot 4^2 + 3 \cdot 4^3 = 16 + 192 = 208$. Но някои от тези върхове са невъзможни за реалните изображения, както е показано на фиг.3.26. От това множество различно маркирани върхове за реалните 3D изображения са характерни 16: шест за връх L, три за връх ВИЛА, три за връх СТРЕЛКА и четири за връх Т (фиг.3.27). Това се доказва, изхождайки от ограничението, че всеки връх се получава от пресичането на три плоскости. Пресичането на три плоскости разделя пространството на осем части (октанта) - фиг.3.28. На фигурата е приета конкретна номерация на октантите. Един или повече октанти се запълват с материал, даващ реален обект. Образуваният централен тристенен връх се наблюдава от незапълнените октанти. Той се изобразява като се маркират линиите, влизащи в него в зависимост от физическия им смисъл в структурата.



Фиг.3.27. 16 характерни върхове за реалните 3D изображения.



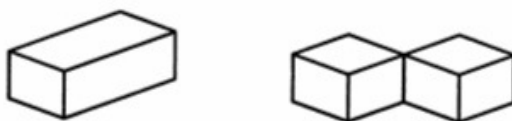
Фиг.3.28. Пресичането на три плоскости разделя пространството на осем части (октанта).



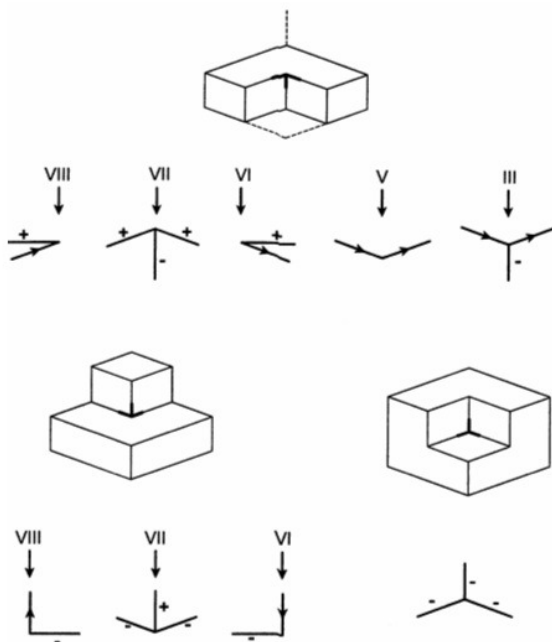
Фиг.3.29. а) Запълване на първи октант; б) Резултат от наблюдението от останалите седем незапълнени октанти.

На фиг.3.29а е показано запълването на един октант, а именно първия. Резултатът от наблюдението от останалите седем незапълнени октанти е показан на фиг. 3.29b, започвайки от осмия към седмия, шестия и т.н. Интересно е да се отбележи, че типът на върха и маркировката не се изменят при изменение на точката на наблюдение в дадения октант, макар че ъглите между линиите се изменят. Анализирайки получените типове върхове и техните маркировки се вижда, че те се свеждат съответно до един връх ВИЛА, СТРЕЛКА и *L*, с конкретни маркировки. Изключва се разглеждането на обекти, получени в резултат на запълването на два, четири или шест октанта. В този случай или не се получава централен връх за наблюдение, или той се получава в резултат на пресичането на повече от три повърхнини (фиг.3.30). По разбираеми причини трябва да се изключи и запълването на осемте октанта. На фиг.3.31 е показано запълването на три, пет и седем октанта. Резултатът от наблюдението от свободните октанти дава по две нови маркировки за върхове тип ВИЛА и СТРЕЛКА и пет нови маркировки за връх тип *L*. Частичното закриване на

приетите основни типове линии във връх тип Т, дава четири различни маркировки за този връх. С това се показва получаването на 16-те различно маркирани върхове за реалните изображения - фиг.3.27. Вижда се, че в *реалния физически свят* са силно ограничени комбинациите на маркировка на линиите, образувачи различни върхове. Процентното отношение на това ограничение е показано в Таблица 3.1.



Фиг.3.30. Запълване на два октанта.



Фиг.3.31. Запълване на три, пет и седем октанта.

Това е *фундаментален* въпрос, който принципно и ресурсно (по време и памет) опростява автоматичния анализ на изображенията.

Таблица 3.1 Процентното отношение на комбинациите на маркировка на линиите, образуващи различни върхове

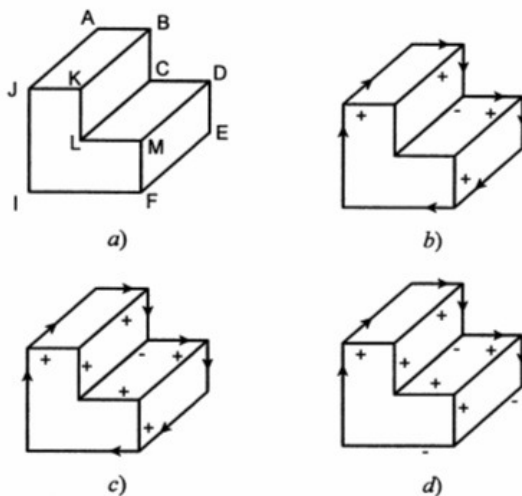
Признак (върх)	Брой възможни маркировки	Брой маркировки в резултат на реалните ограничения	Процентно съотношение
L	16	6	37.5%
ВИЛА	64	3	4.7%
СТРЕЛКА	64	3	4.7%
T	64	4	6.2%

Вместо маркирането на линиите на върховете да се извърши при използване на знания за физическата структура на реалните обекти, може и обратното - използвайки знанията за възможната маркировка на върховете и съгласуването ѝ за съседни върхове, да се получи информация за реалната структура на сцената и обектите в известна степен отговаряща на естествената интуитивна интерпретация. Може да се използва следната стратегия при маркирането на изображението:

- 1) Маркират се граничните линии по посока на часовниковата стрелка;
- 2) Линиите, влизащи във върхове с гранични линии, се маркират съгласно възможното множество маркирани върхове (фиг.3.27). Съгласува се маркировката на всеки два съседни върха, изхождайки от важното свойство, че линия, свързваща два върха не може да има различна маркировка в двата края.

Да разгледаме пример за маркиране на изображението, дадено на фиг.3.32. Маркировката на граничните линии е показана на фиг.3.32в. Върховете В, D, F, J са тип СТРЕЛКА с по две гранични линии. Съгласно с множеството допустими марки (фиг.3.27) третата линия може да има марка само "+". За върха С от типа ВИЛА с две гранични линии, третата линия може да бъде маркирана само с "-". На вътрешните върхове К и М, тип ВИЛА двете линии са маркирани с "+". Третата линия също трябва да бъде маркирана с "+". Двата плюса за върха L се потвърждават и от минуса за този връх. Ако СТРЕЛКА има един минус другите две линии трябва бъдат с "+". На фиг.3.32с е показана окончателната маркировка. Анализирайки маркировката може да

се разкрие структурата на обекта. Ако предположим, че обектът лежи на дадена повърхнина, то линиите EF и FI могат да се маркират с "-", което не противоречи на останалите маркировки (фиг.3.32d).

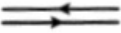










Фиг.3.32.Маркировка на линиите.

3.7. МЕТОД НА УОЛЦ ЗА АНАЛИЗ НА 3D СЦЕНИ

Уолц (*Waltz*) обобщил и разширил идеите на *Huffman* и *Clowes* за анализ на сцени, съдържащи плоскостенни тела като:

- 1) Разглежда сцени, не само с тристенни върхове, но и с върхове, образувани с повече от плоски стени К, У, ПИК, МУЛТИ;
- 2) Допълва множеството възли на Гузман (фиг.3.22) с нови върхове с имена XX, КА, КХ ;
- 3) Отчита сенките, получени от обектите и отворстията в обектите, ако има такива;
- 4) Увеличава набора от символи за маркиране на линиите, отчитайки повече информация за физическата същност на реалната структура;

Маркировка	Обяснение
	Гранични линии
	Изпъкнали линии
	Линии, определящи границите на сянка. Стрелката сочи вътрешността на сянка.
$\frac{R_1}{R_2}$ - 	Неразделима, неизпъкнала линия. R_1 и R_2 принадлежат на едно и също тяло.
$\frac{R_1}{R_2}$ - 	Разделима, неизпъкнала линия, образувана от две тела. Ако обектът ($Об(R_1)$) е разположен над обекта ($Об(R_2)$), то ($Об(R_2)$) се явява опора за ($Об(R_1)$).
$\frac{R_1}{R_2}$ - 	Разделима, не изпъкнала линия, образувана от две тела. Ако област R_1 се намира по-горе от област R_2 , то обектът ($Об(R_2)$) закрива обекта ($Об(R_1)$), или нито ($Об(R_1)$) се явява опора за ($Об(R_2)$).
$\frac{R_1}{R_2}$ - 	Разделима, неизпъкнала линия, образувана от три тела, нито ($Об(R_1)$), нито ($Об(R_2)$) не могат да бъдат опора за друго тяло.
$\frac{R_1}{R_2}$ - 	Линия от тип "пукнатина". Обектът ($Об(R_2)$) се намира пред обекта ($Об(R_1)$), ако област R_1 е разположена по-горе от областта R_2 .
$\frac{R_1}{R_2}$ - 	Линия от тип "пукнатина". Обектът ($Об(R_2)$) се явява опора за обекта ($Об(R_1)$), ако областта R_1 е разположена над областта R_2 .

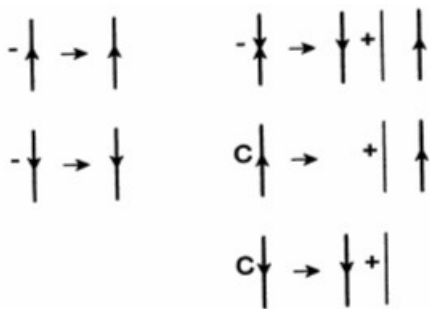
Фиг.3.33. Маркировка на линиите по Уолц.

- 5) Освен за линиите, въвежда някои маркировки за областите, например, за тяхната осветеност;
- 6) За маркировка на линиите предлага бърз, последователен алгоритъм за филтрация.

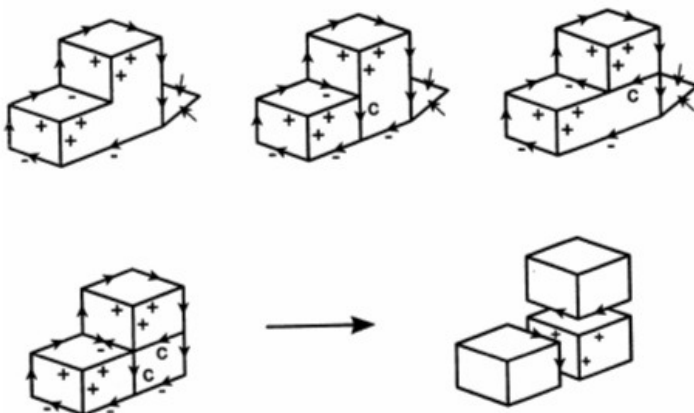
Целта на предложената система е да се построи точно описание на тримерната сцена по нейното двумерно изображение. Освен

физическите съответствия, определени от марките, системата определя и някои *отношения* за обектите: "*пред*", "*зад*", "*поддържане*" и т. н. На фиг.3.33 са показани приетите 11 маркировки на *Уолц* и тяхната физическа интерпретация. Смисълът на образуване на разделимите линии (техните съставлящи) е показан на фиг.3.34, а на фиг.3.35 са дадени маркировки на сцени с едно, две и три тела, отчитайки сянката. Тези сцени потвърждават физическата интерпретация на марките. Раздвижването на телата в последната сцена показва разделимостта на съответните типове линии и нейното разлагане до отделни тела, при използване информацията на маркировката (поточно чрез проследяване контурите с гранични линии). Всяко тяло се оказва оградено със символ стрелка.

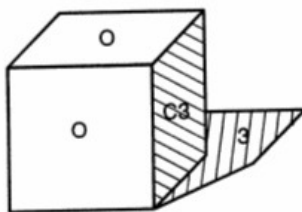
Отчитането на *сенките* в сцената води до по-точното определяне на пространственото разположение на обектите. Това разширява ограниченията и прави анализа по-прост. По-нататъшното подобряване на описанието на сцената е отчитането на осветеността на областите на обектите. Разглеждат се три случая (фиг.3.36): областта е осветена (О); областта е засенчена (З); областта се самозасенчва (СЗ). За линиите, разделящи тези области са възможни $3^2=9$ комбинации на осветеност. Всяка комбинация е възможна за 11 типа линии, определени по-горе. Следователно възможният брой комбинации става $11 \times 9 = 99$. Но срещаните комбинации в реалния свят са не повече от 50. Например, не е реална комбинацията за не изпъкналия ръб едната област да бъде осветена, а другата затъмнена.



Фиг.3.34. Разделими линии и техните съставлящи.



Фиг.3.35. Маркировка на линии с едно, две и три тела. Илюстрация за разделяне на линии.



Фиг.3.36. Самозасенчване и сянка в сцената.

Уолц анализира и определя физически възможните комбинации на марки за линиите на възлите. В Таблица 3.2 са дадени за съответните възли:

- 1) Възможният брой комбинации от марки;
- 2) Физически срещаните комбинации;
- 3) Отношението между възможните и срещаните комбинации.

Анализирайки тези данни и сравнявайки ги с данните от Таблица 3.1 могат да се направят следните изводи:

- 1.С увеличаване на средствата за по-подробно описание на сцената силно растат физическите ограничения;
- 2.Физическите ограничения позволяват да се изгради "ограничена" база знания (каталог от маркирани възли), но достатъчна за надеждния анализ на сцените;

3.Сравнявайки ограниченията за често срещаните възли тип ВИЛА и СТРЕЛКА се вижда, че възелът СТРЕЛКА внася значително повече ограничения и по-малко нееднозначности. С това се обяснява тезисът, получен в резултат на експеримента на програмата на Гузман, че *стрелките* дават *по-надеждни* връзки за областите;

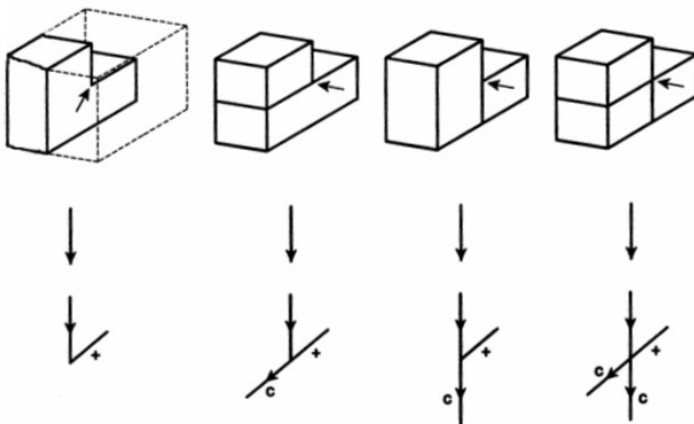
Таблица 3.2. Физически възможните комбинации на марки за линиите на възлите.

Признак (Връх)	Брой възможни маркировки	Брой маркировки в резултат на реалните ограничения	Процентно отношение
L	2.50×10^3	80	3.2 %
ВИЛА	1.25×10^5	500	0.4 %
СТРЕЛКА	1.25×10^5	500	0.4 %
T	1.25×10^5	70	$5.6 \times 10^{-2} \%$
ψ	6.25×10^6	300	$4.8 \times 10^{-3} \%$
K	6.25×10^6	100	$1.6 \times 10^{-3} \%$
X	6.25×10^6	100	$1.6 \times 10^{-3} \%$
МУЛТИ	6.25×10^6	100	$1.6 \times 10^{-3} \%$
ПИК	6.25×10^6	10	$1.6 \times 10^{-4} \%$
КА	3.12×10^8	30	$9.6 \times 10^{-6} \%$

При анализа на възлите приоритет трябва да имат тези, които внасят по-големи ограничения. Броят на различно маркираните възли, които се срещат на границата на сцената с фона е значително малък по сравнение с общия брой. Това се използва при определяне на стратегията на маркиране - най-напред да се маркират възлите, граничещи с фона. За формиране на каталога на различно маркираните възли, Уолц използва метода на *Huffman*. Той запълва октантите последователно с едно, две и повече тела и наблюдава конкретния възел от свободните октанти. На фиг.3.37 е показан пример на запълване на съответните октанти с едно, две и три тела и маркировката на централния възел, гледан от седми октант.

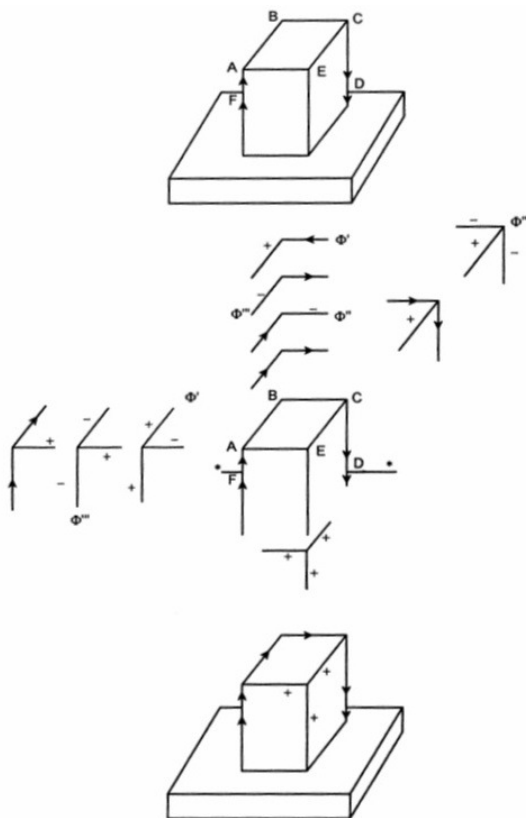
При маркировката на възлите Уолц използва *последователен филтриращ алгоритъм*. Прилага се последователно за всички

възли. За първия избран възел се отчитат всички възможни маркировки за него. Маркировката на втория избран възел се съгласува с първия. За третия, маркировката се съгласува с втория и т.н. Този процес може да се разглежда като *права филтрация*. След определянето на маркировката на даден възел се разглежда *обратното разпространение* на ограничаване (филтрация) на маркировки последователно за предшестващите възли. Алгоритъмът е итерационен и спира работа, когато престанат да отпадат маркировки за възлите. Да разгледаме пример за възлите A, B, C, D, E, F от сцената на фиг.3.38. Върховете се проследяват в реда, в който са записани. За илюстрацията се използва ограниченото множество от възможни маркировки по *Huffman* - Таблица.3.2. За стрелката A се отчитат трите възможни маркировки. Следващият връх е от тип L . За него четири маркировки се съгласуват с маркировките за предишния възел (A). За стрелката C се съгласуват две маркировки с маркировката на B . Маркировката на C налага обратна филтрация за B , като се отхвърля една маркировка (отбелязана с Φ'). Обратната филтрация за B налага обратна филтрация за A . Върхът D е от тип T . Колинеарните линии за връх, тип T , се маркират само със стрелки, в дадения случай съгласувани с маркировката на възел C . Третата линия на възел D е отбелязана със звезда, което означава, че за нея е възможна съответна маркировка, която за приетото разглеждане не е съществена. Възелът D налага филтрация за възел C , за който остава една единствена маркировка. От своя страна възелът C налага филтрация за възел B , за който остават две възможни маркировки. До тук обратната филтрация спира. Възелът E е от тип ВИЛА. Ако една от линиите на този възел е отбелязана с $+$, то и всички останали линии трябва да бъдат маркирани с $+$, т.е. възможна е една маркировка. Възелът F е от типа T . Посоката на стрелките за колинеарните линии се съгласува с маркировка за възел A . F налага филтрация за възел A , за който остава една маркировка. При повторния преглед и съгласуване на маркировките за възлите отпада една - единствена маркировка за възел B . По-нататъшно изменение на маркировката не настъпва и процесът се прекратява. Възлите еднозначно са маркирани – фиг.3.38.



Фиг.3.37. Запълване на октантите с различен брой тела и маркировка на централния възел.

Някои предлагат по-бърз релаксационен алгоритъм за маркировка на възлите. При него съгласуването на маркировката за даден възел се извършва с всички негови съседи. Процесът се прекратява, когато престанат да се изменят маркировките за възлите. Този алгоритъм е удобен за паралелна реализация. На фиг.3.39 се илюстрира работата на алгоритъма. Маркировката на възела *A* се съгласува с тези за *F*, *B*, *E*. За *A* и *E* се определя по една маркировка, а за *B* се определят две маркировки. Маркировката на *C* се съгласува с тази за *B*, *D* и *E*. За *C* се определя една маркировка. Отпада една от маркировките за *B*. За всички възли, които се анализират, се определя по една маркировка - фиг.3.39. До този момент се разглеждаше интерпретацията на визуални сцени, съдържащи само плоскостенни тела. Ако се допусне обектите да съдържат и *криволинейни повърхнини*, то естествено е да се направи опит за прилагане на разгледаните подходи за анализ и на сцени, съдържащи криволинейни обекти. В структурно отношение броят на типовете възли ще нарасне, тъй като всяка линия от даден възел може да бъде права или крива (изпъкнала или неизпъкнала). Ако, например, възелът съдържа три линии, то в този случай възможните комбинации са $3^3=27$, както това е показано за възел тип ВИЛА на фиг.3.40.

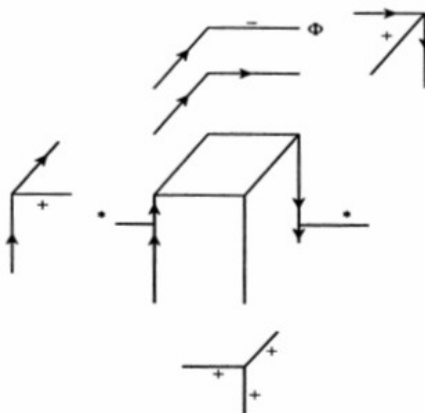


Фиг.3.38. Последователен алгоритъм за маркировка чрез филтрация.

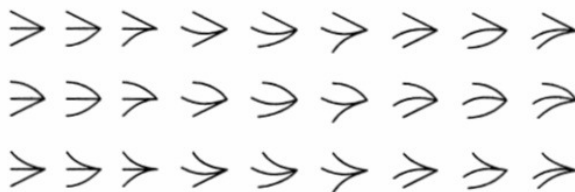
Нараства и физическият смисъл на линиите. При изображенията на криволинейните тела линията може да съответствува на контур на гладка криволинейна повърхнина. За маркиране на такава линия се използват две последователни стрелки $\rightarrow \rightarrow \rightarrow$, определящи заграждащ контур на криволинейната повърхнина. При обхождане на контура по посока на стрелките, дясната страна от повърхнината закрива лявата част на самата повърхнина - фиг.3.41а.

Turner използвал подхода на Уолц за анализ на сцени, съдържащи и криволинейни повърхнини (елипсоиди, конуси и хиперboloиди).

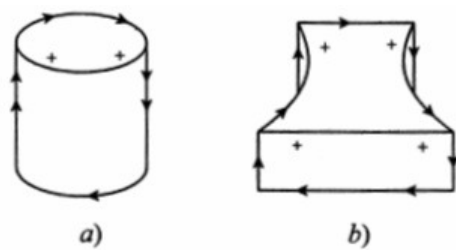
При формирането на речника на възлите, той изменя последователно всяка плоска стена от възела до криволинейна. Генерирани са повече от пет хиляди типа възли. Принципна трудност, която възниква е, че за криволинейните обекти *марката на линията може да се смени по средата* (фиг.3.41в). Смяната на марката предполага съгласуването на конкретна двойка възли, което се отчита в речника. *Turner* използва алгоритми за маркиране на изображения, подобни на описаните по-горе.



Фиг.3.39. Релаксационен филтриращ алгоритъм за маркировка.



Фиг.3.40. Признаци образувани и от криволинейни ръбове.



Фиг.3.41. Тела съдържащи криволинейни повърхнини и маркировка на ръбове.

ГЛАВА ЧЕТВЪРТА: ЦИФРОВО ВИДЕО

4.1. КОМПРЕСИЯ НА ЦИФРОВО ВИДЕО ИЗОБРАЖЕНИЕ

4.1.1. Общи сведения

Видео изображенията се използват в два формата:

- композитен – с три канала: R, G и B;
- компонентен – с три канала: яркост Y, сигнали за цветност (R-Y) и (B-Y).

В цифровата обработка на изображения се използва предимно компонентната форма.

За обработка на видео изображенията с компютър те трябва да се *дискретизират*. За целта всяка една компонента се дискретизира отделно. Яркостта - с честота 13.5MHz, а сигналите за цветност със същата честота или с по-малка. Така се дефинират схемите за дискретизация 4:4:4, 4:2:2, 4:2:0 и 4:1:1 [5].

4.1.2. Методи за компресия

Намаляването на *битовата скорост* (потокът от данни, които трябва да се предаде за единица време) посредством поддискретизация на сигналите за цветност е възможно само до известна степен, без това да доведе до забележимо влошаване качеството на изображението. По нататъшното постигане на по-малък битов поток става с прилагането на "компресия" на цифровите видеосигнали.

В действителност компресията се изразява в намаляване количеството данни, които описват изображението. При това се премахва част от данните така, че да не се влошава субективно качеството на изображението. Това се постига с прилагането на редица особености на системата за визуално възприятие на човека (Human Visual System - HVS), голям принос в изследването на която има "Комитетът за създаване на национална телевизия" (National Television System Committee - NTSC) на САЩ, проведени през 1950г. във връзка с въвеждането на цветна телевизия.

Компресия без загуби

Най-важната компресия без загуби е премахването на интервалите за хоризонтален и вертикален бланк (черно), което очевидно не може да повлияе на качеството на изображението.

Компресия със загуби

Намаляването на битовата скорост е със загуби, когато отстранената при компресията информация се загубва и оригиналното изображение се възстановява (при обратното преобразуване от цифров в аналогов сигнал) само приблизително.

При компресията на цифровия видеосигнал обикновено се комбинират различни начини,, така че да се постигне значително голяма компресия - от 3:1 до 100:1.

Методи на компресия

Днес се прилагат много методи на компресия. Някои от най-популярните са JPEG, MPEG и DV. Тези системи използват дискретната косинус трансформация (Discrete Cosine Transform - DCT). При DCT видеосигналите се преобразуват от времевата в честотната област, което може да се разглежда като възпроизвеждане на честотния спектър на видео сигнала. След като сигналът вече е преобразуван в честотната област, е възможно да се премахне част от информацията - например високите честоти, без това да доведе до субективно влошаване на качеството на изображението. Степента на компресия се определя от това каква част от информацията, в случая високи честоти, се отстранява.

По-ниска честота на под-дискретизация или по-голяма стойност на компресия

Въпросът кой случай е по-благоприятен - по-ниска честота на под-дискретизация и по-малка стойност на компресия или по-висока честота на под-дискретизация и по-голяма компресия за постигане на една и съща битова скорост, е все още предмет на дискусия сред експертите в областта мултимедиите.

Изследванията и анализите: дали да се прави по-ниска дискретизация на сигналите за цветност при по-малка компресия; или да се стартира с по-вече информация за цветността в началото и след това да се извърши по-силна компресия, показват, че качеството на изображението е едно и също (малко по-добро за 4:2:2), докато битовата скорост стане изключително ниска. Тогава схемата 4:2:0 има малко преимущество.

Крайното заключение обаче е, че най-целесъобразно е прилагането на схемата на дискретизация 4:2:2. Взета е предвид преди всичко съвременната технология, която е свързана с многократна

обработка на изображението, със създаване на ефекти, нанасяне на надписи и т.н.; използването на компютри, които третира по един и същи начин яркостите и цветовете.

При цифровия сигнал, всяко преобразуване от един стандарт в друг, влияе на сигнала, с което се влошава качеството на изображението. Ето защо важно е и да не се преминава от една схема на под-дискретизация към друга по време на процеса на видео производството. Или с други думи , ако се започне със схема на дискретизация 4:1:1 (4:2:0), тази схема да се запази докрай.

4.1.3. Компресия на цифровите видеосигнали

Възможната степен за намаляване честотата на под-дискретизация на сигналите за цветност зависи от особеностите на визуалните възприятия на човек, след което настъпва недопустимо влошаване на качеството на изображението. По - нататъшното намаляване на битовата скорост става с прилагане на компресия на цифровия видеосигнал.

- Стандарт **JPEG** - Осигурява компресия до 4:1 без субективно влошаване на качеството на изображението. Възможен е монтаж с точност до един кадър.

- Стандарт **MJPEG** - разширение на стандарт JPEG. Също осигурява точност на монтажа до един кадър.

- Стандарт **MPEG** - В настоящият момент се използват три стандарта MPEG ; MPEG-1,-2 и -4.

MPEG-1 стандартът е публикуван през 1991г. Той е предназначен за приложение в мултимедиите и компютрите. Използва разделителната способност на формата SIF (source input format) 350x240 пиксела. MPEG-1 е проектиран да създаде оптимална скорост на предаване на данните от приблизително 1.5 Mb/s за видео и звукови битови потоци.

MPEG-2 предлага пълна разделителна способност на изображението, отговаряща на специфицираната по стандарта CCIR-601 и възможности за компресия на видеосигнал, получен с презредова телевизионна развивка.

MPEG-4 използва една четвърт от разделителната способност, специфицирана с формата SIF и е предназначен за работа с ниски битови скорости, като телеконференции и Internet.

- Стандарт **DV** (Digital Video Cassette -DVC) Цифров формат за видеозапис, предназначен за битови нужди. Използва части от JPEG при кодирането на видео сигнала. Базира се на дискретизация 4:2:0 за страните с ТВ стандарт 625/50 и 4:1:1 - за страните с ТВ стандарт 525/60. Битовата скорост за видео данните е 25 Mb/s. Компресията е 5:1.

4.1.4. Стандарт за компресия JPEG

Принципи на компресията JPEG

Стандартът JPEG е първия стандарт за компресия на изображения. През 1988г. Joint Photographic Experts Group (JPEG) създава стандарт за компресия на неподвижни изображения за нуждите на печатарската индустрия, работа с компютри, електронната фотография.

Като стандарт за компресия на неподвижни изображения, JPEG представлява кодиране вътре в кадъра (Intra-frame). Стандартът предлага широки граници за вариране на компресията - броят на битовете хоризонтално и вертикално може да се изменя то 1 до 12. Обикновено кодирането се извършва с 8 битова дума. Цветовете се кодират поотделно. Поради наличието на различни източници, цветните компоненти могат да бъдат представени като червен, зелен, син - RGB (Red, Green, Blue); яркост, сигнали за цветност за червения и синия цвят - Y/Gb/Gr (Luminance/R-Y/B-Y); яркост, синьо-зелен, жълт, виолетов и черно - Y/C/M/K (Yellow/Cyan/Magenta/Black). Компресията се базира на "Директно косинус преобразуване (Direct Cosine Transform).

DCT преобразуване

DCT е математически метод за преобразуване без загуби на информацията за яркостта на отделните пиксели в кадъра, в изобразяващи я (олицетворяващи) коефициенти в честотната област. След като информацията вече е в честотната област, коефициентите се подлагат на обработка, която не е вече без загуби. Известна част от информацията за изображението се губи, която повече не може да се възстанови. Резултантните данни се подлагат на кодиране с ентропия.(Кодиране с ентропия означава изразяване на повтарящи се многократно елементи от изображението с един кратък код, който е признак на повторение).

За да се осъществят горепосочените преобразования се прави следното. Компонентните сигнали - сигналът за яркост (Y) и сигналите за цветност (R-Y, B-Y) се кодират поотделно, като независими едно от друго изображения. Всяко изображение се разделя на блокове, състоящи се от 8 x 8 пиксели (осем в хоризонтална и осем във вертикална посока). Всеки отделен блок се подлага на дискретно косинус преобразуване, в резултат на което яркостта на всеки отделен пиксел се превръща в 64 синусоидални напрежения с различни честоти (аналогично на честотния спектър на сигнала). С помощта на редица преобразования, тези синусоидални напрежения с различни честоти се подреждат по следния начин: в горния ляв ъгъл на пиксела се разполага напрежение с нулева честота. В останалите 64 пиксела надясно и надолу се разполагат синусоидалните напрежения с увеличаваща се честота. Целта е разположението на напрежения с различни честоти да има пространствено измерение. Или казано с други думи: в резултат на тези действия в блока се намират не пиксели с различна яркост, а пиксели с различна честота, подредени по определен начин.

Пространствено напрежение на различните честоти дава възможност те да се кодират с различен брой битове: ниските честоти се кодират с по-голям брой битове, а високите честоти - с по-малък (малка битова дума). На практика се извършва намаляване или загуба на информацията с висока честота, което означава намаляване броя на битовия поток за единица време (една секунда) - намаляване на битовата скорост.

И така компресията се изразява в "изрязване" на високите честоти (намаляване детайлността на изображението) до определена степен в зависимост от стойността на компресия.

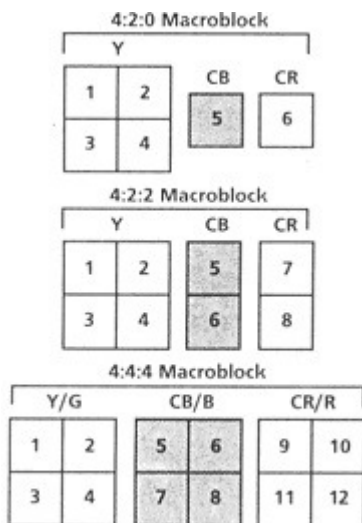
По нататък блоковете се обединяват в макро блокове по начина показан на фиг.4.1.

За структурата на дискретизация 4:4:4 макро блокът на сигнала за яркост и макро блоковете на сигналите за цветност се изграждат от по четири блока, респективно: четири блока за Y, четири блока за R-Y и четири блока за B-Y.

За структурата на дискретизация 4:2:2 макро блокът на сигнала за яркост се изгражда от четири блока; макро блоковете на

сигналите за цветност - от по два блока, респективно: два блока за R-Y и два блока за B-Y.

За структурата на дискретизация 4:1:1 (4:2:0) макро блокът на сигнала за яркост се изгражда от 4 блока; макро блоковете на сигналите за цветност - от по един блок, респективно: един блок за R-Y и един блок за B-Y.



Фиг.4.1. Обединяване на блоковете в макро блокове

Характерни особености на стандарта

- При компресия до 20:1 изображението е използваемо, въпреки че влошаването на изображението е видимо. При компресия над 20:1, изображението е неизползваемо;

- Причинната за тези относително ниски стойности на възможна компресия е, че при JPEG не се използва корелацията на видеосигнала във времето (еднаквост на съседни кадри). Ето защо е създаден стандарт за кодиране с JPEG на поредица от кадри или полета. Този стандарт, известен като Motion JPEG (MJPEG), фактически е разширение на стандарта JPEG. Всички motion JPEG системи имат точност на монтажа до един кадър (или даже до поле), тъй като всяко изображение (кадър, поле) се кодира поотделно.

4.1.5. Стандарт за компресия MPEG

Стандартите MPEG са създадени от експертната група Moving Pictures Experts Group (MPEG), основана през 1988г. Целта на експертите е да създадат стандарт за компресия на движещи се изображения [18].

Принципът на стандарта MPEG е следният.

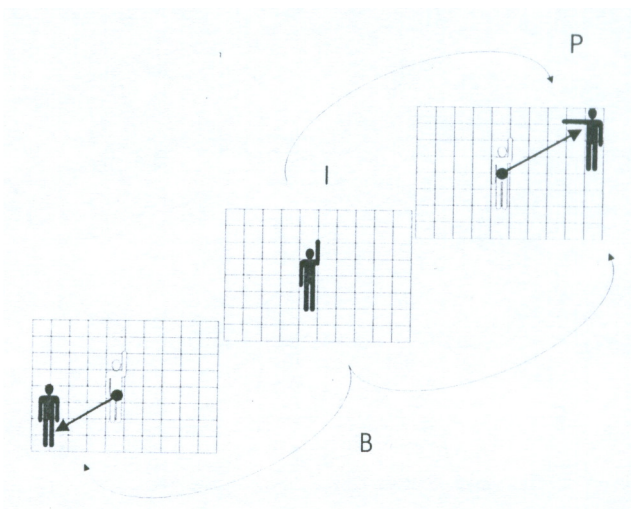
- Като първа стъпка стандартът MPEG-2 определя прилагането на JPEG компресия, т.е. извършване на компресия вътре в един кадър - наречена "Intra-frame" (или Intra-field) компресия (компресия в кадъра). В резултат на тази компресия се получава "ключов" кадър, наречен "I" кадър.

- Втора стъпка е извършването на компресия, като се сравняват съседни кадри, в които се намират движещи се обекти. Използва се еднаквостта на съседните кадри, разликите между които са само местоположенията на движещите се обекти. При това "I" кадърът, за който вече е направена JPEG компресия, се приема за основен ("ключов") кадър, спрямо който се сравняват съседните кадри- следващ и предхождащ го кадър. Определят се местоположенията на движещите се обекти в съседните кадри спрямо местоположенията им в ключовия кадър. Новите местоположения се дефинират с помощта на вектори, наречени "Вектори на движението", които показват големината и посоката на преместването на движещите се обекти.

- Така, като се сравняват два съседни кадри, в които има движещи се обекти, информацията от следващия кадър след ключовия кадър, ще се състои само от вектори на движението, указващи новите местоположения на движещите се обекти. Компресията се бележи с "P", кадърът се нарича "предсказан кадър" - "Predicted" и също се бележи с "P". Компресията е "междукadroва" - "Intraframe" - фиг.4.2.

- Възможно се да се извърши сравнение между три кадъра - настоящия, следващия и предишния. Отново след сравнението им, се определя местоположението на движещите се обекти - този път в предишния и в следващия кадър след ключовия кадър. Информацията в тях също се изразява с вектори на движението, указващи посоката и големината на преместване на движещите се обекти спрямо основния, ключов "I" кадър. Компресията се

бележи с "В" (Bi-directional predicted frame). Кадрите също се бележат с "В" - фиг.4.2.



Фиг.4.2. Принцип на стандарта MPEG

Очевидно е, че между кадровата компресия в случай на "В" компресия е по-голяма, отколкото в "Р" компресия, защото обхваща два кадъра, а не само един.

И така MPEG компресия може да се изгради от всяка подходяща комбинация от I, P и B кадри. Тази възможност се дефинира с така наречената "група от изображения" - GOP (Group of Pictures). Големината на GOP определя степента на компресия. Очевидно е, тук се появява проблем при монтажа - невъзможно е извършването на монтаж с точност до един кадър.

На фиг.4.3 е показана последователността на извършване на MPEG компресията, както и структурата на входящите и изходящи цифрови данни за видеосигнала.

Първата стъпка е разделянето на кадъра на отделни блокове, състоящи се (всеки един от тях) от 8 пиксела и 8 реда.

Втората стъпка е създаването на макро блокове. За пример е взета структурата на дискретизация 4:2:0.

Макро блоковете изграждат една серия - **slice**.

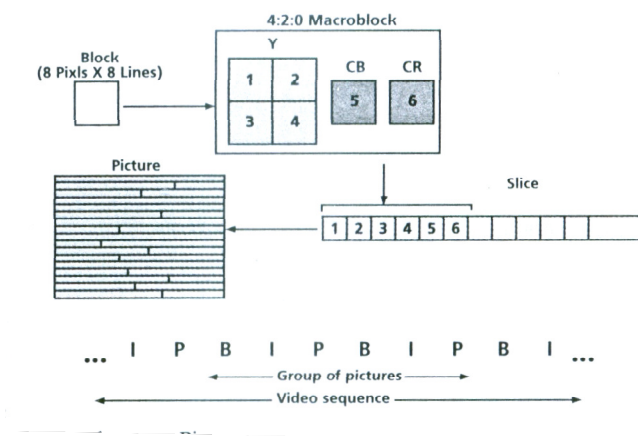
Сериите (slices) оформят един кадър - **picture**.

Кадрите (Pictures) се групират в група - **GOP (Group of Pictures)**

Групите от кадри (Group of Pictures), оформят видео потока - **video sequence**.

За описаната схема на последователност на действията трябва да се отбележи следното:

- компресията JPEG се извършва след създаването на серийния поток от макро блокове (slice);
- компресията MPEG се извършва след изграждането на кадъра (Picture).



Фиг.4.3. Последователност на извършване на MPEG компресията.

Приложения на MPEG-2

Стандартът MPEG-2 първоначално е предназначен за компресия на видео сигнала с цел запаметяване и разпространение. Концепцията е била, че компресия на видео сигнала трябва да се прави върху крайния видео материал - монтиран, обработени напълно готов за излъчване. По тази причина и стандартът MPEG-2 осигурява висока стойност на компресия и недостатъчна точност при монтирането. Но с разпространението на MPEG-2 необходимостта от монтиране на компресиран видео материал става реалност. Това налага да се започне работа върху по-нататъшното усъвършенстване на

стандарта и прилагането му за нуждите на монтажа - постигане на точност до един кадър (поле).

Най-очевидният начин е да се декомпресира MPEG-2 потокът и монтажът да се извърши с нормален видео сигнал, след което той отново да се компресира по стандарта MPEG-2. Недостатъкът е: влошаване на качеството на изображението, върху което, в смисъл на влошаване, оказват процесите на декомпресия и компресия.

Друг метод е да се монтира директно MPEG-2 потокът. Както беше описано, MPEG-2 видео сигналът съдържа "I", "B" и "P" кадри. I кадрите са самостоятелни и позволяват директен достъп до тях в потока и монтиране. Така, MPEG-2 поток, съдържащ изцяло "I" кадри може да бъде монтиран по същия начин, както и некомпресиран видео сигнал. Недостатъкът е, че "I" кадрите изискват голям брой битове и следователно не се постига достатъчна компресия на сигнала за нуждите за нуждите на ТВ разпръскване, както и реално предимство спрямо стандарт на компресия JPEG. "P" и "B" кадри изискват по-малък брой битове, но не предлагат директен достъп до тях в битовия поток (като "предсказани" кадри, те съдържат само информация за векторите на движението и изискват информация от "I" кадри, за да бъдат декодирани). Така се стига до параметъра компресия спрямо директен достъп.

Трябва да се подчертае, че схемата на дискретизация 4:2:2 позволява да се запази по-високата разделителна способност на сигналите за цветност, необходима при работа със специални режими на врязване и обработка на изображението; вкарване на специални ефекти, надписи, т.н., спрямо 4:4:1 (4:2:0).

Особености на MPEG-2 стандарта

MPEG-2 стандартът използва компресия със загуби. Следователно се очаква влошаване на качеството с увеличаване на компресията. MPEG-2 с различните си levels и profiles предлага битова скорост от 3 до 100 Mb/s. Съвременните ТВ програми използват видео сигнали кодирани в границите от 6 до 15 Mb/s при пълна разделителна способност по CCIR (720 x 480). Качеството на кодираното видео изображение е професионално. Следователно винаги може да се избере битова скорост и разделителната

способност на видео изображението, като се има предвид предназначението му и широчината на разполагаемата честотна лента.

Кодирането с MPEG-2 технология позволява интелигентно превключване между кадри и полета, когато се кодират статични изображения.

Бъдещото развитие на MPEG-2

Въпреки, че MPEG-2 се наложи като стандарт, в настоящия момент се разработват нови версии.

MPEG-4 е предназначен за приложения, които работят с малко битови скорости, каквито са видео по Интернет, мобилните мултимедии и интерактивните видео игри.

MPEG-7 е най-новият стандарт, и е насочен към идентифициране и търсене на мултимедийно съдържание, което съществува в различни форми днес. Тези нови стандарти могат да станат важни за разпръскването, компютрите и телефонната индустрия.

4.1.6. Стандарт за компресия DV

Международен консорциум на електронни фирми, в които влизат Panasonic, Philips, Sony и Thomson, започва работа в началото на 90-те години върху нова схема за компресия, предназначена за следващата генерация видеокамери. Целта на консорциума е да се обедини натрупаният опит на водещи в света компании, както и да се приложат съвременни технологии, в областта на видео изображенията. Така се създава стандартът DV (Digital Video).

Стандартът DV използва схема на дискретизация 4:1:1, при която за сигналите за цветност се взема всеки четвърти отчет (sample) от отчетите за сигнала за яркост. Това означава четири пъти намаляване на хоризонталната разделителна способност на изображението за цветовете, което съответства напълно на аналоговата цифрова телевизия. А именно: честотната лента (определя хоризонталната разделителна способност) на сигналите за цветност се ограничава изкуствено до 1.5 MHz, което е четири по-малко от честотната лента на сигнала за яркост - 6 MHz. Вертикалната разделителна способност на сигналите за цветност

остава същата, както за сигнала за яркост и в двата случая - на DV стандарта и аналоговата телевизия.

Стандартът прилага JPEG - вътрешно кадрова компресия, със стойност на компресията 5:1.

На тази база е създаден битовият формат на видеозапис DV (Digital Video). Този формат е пример за миниатюризация на електронните устройства.

Така се оформят три вида DV камери: miniDV (битови камери), DVCAM (промишлени камери) и DVCPRO (професионални).

Параметри на формата DV :

- Широчина на магнитната лента - 1/4 инч (6.35 mm);
- Покритие на магнитната лента – метално;
- Широчина на видео пътеката - 10 μm ;
- Структура на дискретизация - 4:1:1;
- Стандарт на компресия - JPEG, т.е. само вътрешно - кадрова (intra-frame) компресия;
- Стойност на компресията - 5:1;
- Битова скорост - 25 Mb/s;
- Вид на видеосигнала - компонентен.

Предимства на формата

• Видеокасетите са изключително малки. Това означава лека и компактна апаратура, както и ниска себестойност;

• Вътрешно - кадровата компресия е предназначена за монтаж с точност до един кадър. тук не съществува нито един от проблемите на монтажа, присъщи на системите MPEG;

• Изключително важно е, че по начало DV форматът е проектиран като "разширяем" формат, т.е.битовата скорост може да се увеличи от 25 Mb/s до 50 Mb/s (стандартна телевизия - SDTV) и 100 Mb/s (телевизия с висока разделителна способност - HDTV).

Недостатъци на формата

Основен недостатък на формата е малката широчина на видео пътеката - 10 μm . Тя не осигурява необходимата стабилност на записа. В резултат на това се получават голям брой пропадания на видео сигнала (drop-out).

Форматът не осигурява удовлетворяването на основни изисквания за професионално качество на изображението, като:

чувствителност; отношение сигнал/шум; диапазон на яркостите; цветови диапазон; отсъствие на изкривявания, известни като "артефакти"; многобройни звукови пътечки.

4.2. ВИДОВЕ СТАНДАРТИ ЗА КОДИРАНЕ НА ВИДЕОИНФОРМАЦИЯ.

4.2.1. H.261

H.261 е стандарт за кодиране на видеоинформация, който е публикуван от ITU (International Telecom Union) през 1990г. Този стандарт е предназначен за предаване на данни със скорост кратна на 64 Kbit/s в ISDN мрежи с фиксирано времезакъснение и понякога се нарича **px64** (където p може да приема стойности в границите от 1 до 30). Като алгоритми за компресия в този стандарт се използва комбинация от дискретно косинусово преобразуване (DCT) и компенсация на движението. H.261 поддържа две разделителни способности: QCIF (Quarter Common Interchange Formal) и CIF (Common Interchange Formal). При формат CIF, разделителната способност е 325 x 288 пиксела за канала Y и 176 x 144 за каналите Cr и Cb, а при формат QCIF разделителната способност е 176x144 пиксела за канал Y и 85x72 за каналите Cr и Cb. Използва цветово описание YCrCb 4.2.0 като всеки пиксел се описва с осем бита. Скоростта на кодиращия алгоритъм е оптимизирана за скорости между 40 Kbits/s и 2 Mbits/s.

4.2.2. H.263

H.263 е ITU стандарт за кодиране на видео, създаден през 1995/6 година. Първоначално е разработен с използване в нискоскоростна среда до 64Kbits/s, но днес това ограничение е отпаднало.

H.263 поддържа пет разделителни способности. Като допълнение към QCIF и CIF, поддържани от H.261 са добавени SQCIF, 4CIF и 16CIF. QCIF е приблизително на половината разделителна способност на QCIF, 4CIF и 16CIF са съответно с 4 и 16 пъти по-голяма разделителна способност от CIF.

Поддържането на последните две разделителни способности означава, че кодекът е съпоставим с останалите стандарти за кодиране на видео като MPEG.

Стандартът също като H.261 използва DCT и алгоритъм за компенсация на движението. В сравнение с по-стария стандарт H.261 има много подобрения в областта на компенсацията на движението, което позволява постигане на по-голямо компресиране и подобрения при коригирането на грешки. На фиг.4.4 е дадена блок-схема на алгоритъма за кодиране.

Основните стъпки за компресиране, които се използват в стандарта H.263 са следните:

1) Входните видеоданни попадат в блок за анализ параметрите на движението (Motion Estimation) и блок за компенсация на движението (Motion Compensation). Целта на обработката е да бъдат кодирани само разликите между текущия кадър и предходния. В процеса на анализ на параметрите на движението, кодекът намалява размера на компресирания файл, като отчита само областта от кадъра, която се е променила в сравнение с предходния. Блокът за компенсация на движението формира матрица от пиксели с размерност 16×16 (макроблок) . От своя страна всеки макроблок се състои от четири матрици с размерност 8×8 .

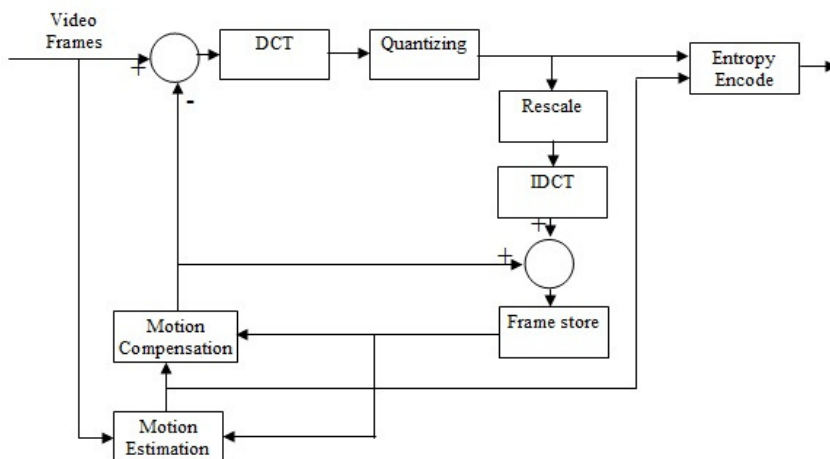
2) Всеки макроблок на текущия кадър се получава като разлика между собствените стойности на макроблока и съответните стойности на макроблока в предишния кадър.

3) Върху така получените данни за всяка матрица се извършва кодиране, като се използва дискретно косинусово преобразование (Discrete Cosine Transformation - DCT). Получава се матрица с амплитуди на честотите на сигнала с размерност 8×8 . Особеното при нея е, че амплитудите на високите честоти са близки до нула.

4) Матрицата на амплитудите се преобразува във вектор с помощта на зиг-заг сканиране, като се започва от горния ляв ъгъл. По този начин амплитудите съответстващи на ниските честоти се получават в началото на вектора, а тези на високите (със стойности близки до нула) в края на вектора.

5) Следващата стъпка е квантуване (Quantizing). За целта се извършва поетапно деление на всеки от елементите на вектора на квантуващ вектор и последващо закръгляване да цяло число. Чрез промяна на стойностите на вектора на квантуване може да се увеличава или намалява значението на отделните честоти и да се

управлява качеството на изображението. Полученият след квантуването вектор съдържа много последователни нули в областта на високите честоти.



Фиг.4.4. Блок-схема на алгоритъма за кодиране на стандарта H.263

6) На следващия етап в блока Entropy Encode се извършва ентропно кодиране на вектора. Ентропното кодиране представлява разделяне на вектора на двойки (n,m), където n указва броя на пропуснатите нули, а m - значението, което трябва да бъде записано в следващия байт. Например вектор: 47, 93, 0, 0, 0, 9, 6, 0, 0, 4,...ще бъде записан като: (0,47), (0,93), (3,9), (0,6), (2,4)...

7) След това върху резултата се прилага кодиране с помощта на алгоритъм на Хофман. Получава се последователност от битове, която се предава.

По време на целия процес на кодиране текущия кадър се съхранява в буфер (Frame Store). При разкомпресиране се изпълнява обратната последователност от действия.

4.2.3. HuffYUV

HuffYUV представлява кодек, компресиращ видео без загуба на информация. Използва се за временно запазване на

прихваната видеоинформация преди редактиране. При този формат се използва само вътрешнокадрово кодиране, което дава възможност за достъп до всеки кадър.

Алгоритъмът на кодека работи по следния начин:

- предсказва се стойността на всеки пиксел като се използват стойностите на съседните пиксели;
- изчислява се разликата между предсказаната и действителната стойност;
- резултатът се компресира с помощта на алгоритъма на Хофман - като за всеки канал се използва различна таблица.

При разкомпресиране се постъпва по обратния начин - извлича се разликата за пикселите от потока от данни и с помощта на стойностите на съседните пиксели се възстановява оригиналната стойност.

Видеофайловете кодирани с HuffYUV обикновено са двойно по-големи от тези, кодирани с използване на M-JPEG или DV.

4.2.4. Microsoft Video 1

Това е оригиналният кодек използван във файловия формат AVI. Кодирането на видеосигнал с този кодек е бързо, но качеството не е високо. Работи с блокове от пиксели с големина 4x4 поради което входната информация трябва да има размери по височина и ширина, делима на 4.

Подходящ е за първоначално компресиране с цел тестване и преди да се избере по-добър кодек. В Microsoft Video 1 се поддържат 8-битов и 16-битов цвят и може да се използва за кодиране на видео с ограничен брой цветове. Големината на компресирания видеофайл е малка, но не толкова, колкото с използване на други кодекци.

4.2.5. DivX 5.1

DivX е най-популярната видео технология, която към настоящия момент се използва от над 50 млн. потребители. Кодекът е проектиран за пълноекранно и пълнокадрово възпроизвеждане на видео. Дава възможност за бързо кодиране и декодиране в реално време на видео с висока разделителна способност без използване на специален хардуер. DivX 5.1 декодира MPEG-4 Sample Profile и MPEG-4 Advanced Sample

Profile видео. По този начин кодекът успешно може да бъде използван в съвременни преносими устройства, като например портативни възпроизвеждащи устройства. DivX5.1 използва най-съвременната технология за премахване на ненужна видеоинформация, наречена психовизуално моделиране (Psychovisual Modeling). Посредством използване на знания за визуалната система на човека, кодекът прилага оценка на психовизуална сложност (Psychovisual Complexity Rating - PCR) за всеки кадър и премахва данните, които не могат да се видят от човешките очи при пълноскоростно възпроизвеждане на видео. По този начин намалява значително обемът, заема от видеофайла и се повишава качеството на видеоизображението.

4.2.6. XviD

Представява MPEG-4 кодек с отворен код, което е причина за неговата голяма популярност. Осигурява качество съпоставима с останалите MPEG-4 кодеци при висок битрейт. Влошаване на изображението има при нисък битрейт (под 500 Kbit/s) или бързо променящи се сцени във видеото. Големината на видео файла, който се получава е по-малка от тази, която се получава при използване на DivX кодек. Все още съществуват проблеми с поддържането и възпроизвеждането от различни програми.

4.3. СЪВРЕМЕННИ ВИДЕО ФАЙЛОВИ ФОРМАТИ

Форматът на видеофайла зависи от използваното апаратно и програмно осигуряване. Днес най-голямо разпространение имат файловите формати AVI, MOV, MPEG и ASF, които могат да се използват в съчетание с различни кодеци.

4.3.1. AVI (Audio Video Interfaced)

AVI представлява файлов формат разработен от фирмата Microsoft. При него секторите за видео данни се редуват със сектори за звукови данни, по такъв начин, че при възпроизвеждане да е необходим минимален буфер за данни. Форматът се използва най-често при записване и редактиране на видео.

Предимство е, че не е необходим допълнителен софтуер, за да бъде възпроизведен такъв файл по Windows. Най-често за

компресиране в такъв формат се използва M-JPEG, но се поддържат и други кодеци.

Недостатъците му са следните:

- несъвместимост с технологиите за потоково предаване на данни;
- не се поддържа от други операционни системи, т.е. необходим е допълнителен софтуер.

Напоследък форматът AVI се заменя с ASF.

4.3.2. MOV

Форматът е разработен от фирмата Apple и поддържа файлове, които могат да се използват за CD или Интернет базирани мултимедийни приложения. Този формат поддържа няколко аудио и видео канала. Видео изображенията в този формат са с високо качество, но обема на файловете е много голям. Поддържа различни кодеци, включително MPEG-4.

4.3.3. MPEG

MPEG форматът за видео е разработен от независимия комитет Moving Picture Experts Group (разработил и методите за компресиране MPEG). Форматът MPEG използва един от трите стандартни алгоритъма за компресиране: MPEG-1, MPEG-2 или MPEG-4. Той предлага както по-добро компресиране от AVI формата, така и по-добро аудио и видео качество. Голяма част от съвременните редактори на цифрово видео дават възможност за запазване на видеоинформацията във файлов формат MPEG.

4.3.4. ASF (Active Streaming Format)

ASF представлява нов файлов формат, създаден от фирмата Microsoft. Предназначен е да осигури поточно предаване и синхронизирано възпроизвеждане на различни типове мултимедийни файлове - видео, анимация, графика, звук, MIDI и текст в среди с ниска пропускателна способност. Една от отличителните черти на формата е, че осигурява мащабируемост при възпроизвеждане. ASF файл може да съдържа основен поток, осигуряващ минимално качество и допълнителни потоци информация с по-добро качество, в случай, че потребителя може да обработи допълнителни данни, а мрежата има достатъчно пропускателна способност. Едновременно с разширението ASF, което може да се използва както за видео, така и за аудио файлове

се използват и разширенията WMV(Windows Media Video) за видео и WMA (Windows Media Audio) за звук.

Заклучение

В основата на зрителното възприятие са процесите на получаване, представяне, обработка и разпознаване на информация, отразяваща свойствата на наблюдавания от човека свят.

В тази книга, са обхванати широк кръг от теми на компютърно зрение. Като се започне с формирането на изображения, предварителната им обработка за отстраняване на шума или размазването и разделенето на региони. Множество изображения могат да бъдат асоциирани с резултатите, използвани за оценка на движение, проследяване на хора, реконструиране на 3D модели, или да се слоят изображения в по-атрактивни и интересни композиции и интерпретации. Изображенията могат да се анализират и да произвеждат семантични описания на тяхното съдържание. Въпреки това, разликата между компютъра и ролята на човешкия фактор в тази област все още е голяма и е вероятно да остане така за много години.

В изложението се прилага широка гама от математически техники. Те включват дискретна математиката, като например обработка на сигнала, вариационни подходи, триизмерна и проективна геометрия, линейна алгебра и метод на най-малките квадрати. Разгледани са теми от дискретна математика и компютърни науки, като графични алгоритми, комбинаторна оптимизация и дори техники за извличане на информация от бази данни. Тъй като много проблеми в компютърно зрение са вероятностни се използват статистически заключения, както и техники за машинно обучение, за да се получат вероятностни модели на големи обеми от данни за обучение.

Компютърното зрение се базира на широко разнообразие от под-дисциплини, което го прави предизвикателство при разработване на един учебен курс. За сметка на това невероятната широчина и техническата сложност на компютърното зрение е това, което привлича много хора на това изследователско поле.

Какво е бъдещето на компютърното зрение? Една от най-новите тенденции в компютърно зрение е да се използва огромното количество визуални данни в Интернет като източник за изучаване на визуални модели на сцени и обекти, като например разпознаване на реч, машинен превод, синтез на музика и

компютърна графика (рендиране на изображения и анимация с отчиране на движение). Повечето традиционни количествени техники в компютърно зрение, като например оценка на движение, подобряване на изображението, както и ефикасни техники за статистическа обработка продължават да се развиват. Някои техники, като функция на съвпадение и структура от движение, могат да бъдат приложени над почти произволни колекции от снимки на статични сцени. Това води до развитие на 3D моделиране върху интернет масиви от данни, които отново са свързани с визуалното разпознаване.

Какъвто и да е резултатът от тези изследователски усилия, компютърно зрение вече има огромно влияние в много области, включително цифрова фотография, визуални ефекти, медицинско образна диагностика, безопасност и наблюдение, както и уеб-базирано търсене. Обхватът на проблемите и техники, присъщи на тази област, в съчетание с математиката и полезността на получените алгоритми гарантира, че това остава една интересна област на проучване за години напред.

Литература:

1. Гочев Г. Компьютерно зрение и невронни мрежи. София, изд. на ТУ, 1998.
2. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. Новосибирск, Изд-во НГТУ, 2000.
3. Опенхайм А., А.Уилски, Я.Йънг. Сигналы и системы. София, изд. Техника, 1993.
4. Прэтт У. Цифровая обработка изображений, т.1, Москва, изд. "Мир", 1982.
5. Смикарова Ст., Цв. Георгиев. Мултимедийни системи и технологии, Авангард принт, Русе, 1999г.
6. Bishop C. M. Pattern Recognition and Machine Learning. Springer, New York, NY, 2006.
7. Bleyer M., Gelautz M., Rother C., and Rhemann C. A stereo approach that handles the matting problem via image warping. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami Beach, FL, 2009.
8. Bovik A. (ed.). Handbook of Image and Video Processing, Academic Press, San Diego, 2000.
9. Bracewell R. N. The Fourier Transform and its Applications. McGraw-Hill, New York, 2nd edition, 1986.
10. Forsyth D. and Ponce J. Computer Vision: A Modern Approach. Prentice Hall, Upper Saddle River, NJ, 2003.
11. Gonzales R. C. and Woods R. E. Digital Image Processing. Prentice-Hall, Upper Saddle River, NJ, 3rd edition, 2008.
12. Marr D. VISION. A computational Investigation into the Human Representation and Processing of visual information. NY, WH Freeman and Co, 1987.
13. Paris S., Kornprobst P., Tumblin J., and Durand F. Bilateral filtering: Theory and applications. Foundations and Trends in Computer Graphics and Computer Vision, 4(1):1–73, 2008.
14. Pratt, W. K. Digital Image Processing. Wiley-Interscience, Hoboken, NJ, 4th Edition, 2007.
15. Ramesh Jain, Rangachar Kasturi and Brian Schunk. Machine Vision .McGraw Hill 1995.

16. Richard I. Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision, 2004.
17. Richard Szeliski. Computer Vision: Algorithms and Applications, Springer, 2010.
18. Richardson I. E. G. H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. Wiley, 2004.
19. Ritter G. X. and Wilson J. N. Handbook of Computer Vision Algorithms in Image Algebra. CRC Press, Boca Raton, 2nd edition, 2000.
20. Trucco E. and Verri A. Introductory Techniques for 3-D Computer Vision. Prentice Hall, Upper Saddle River, NJ, 1998.