

Лекция 2

Полярно-логарифмическое преобразование вытекло из теории обработки сигналов.

Для того, чтобы сигнал был инвариантным к амплитуде следует применить логарифмический усилитель.

Согласованный фильтр:

$$U_{\text{вых}} = \frac{U_{\text{вх}} \cdot U_{\text{э}}}{K}$$

Если подобрать ковариационную матрицу шума — то коррелятор будет оптимальным.

Если картинка испорчена — можно применить обратную фильтрацию. Если картинка «смазана» определенным

$$i_{\text{иск}} = i_{\text{вх}} * f_{\text{иск}}$$

Если применить свертку с обратным фильтром, то получим исходную картинку.

$$I_{\text{иск}} = I_{\text{вх}} \cdot F_{\text{иск}}$$

$$I_{\text{res}} = \frac{I_{\text{вх}} \cdot F_{\text{иск}}}{(F'_{\text{иск}})}$$

Восстановление изображений (Винеровская фильтрация или фильтрация Тихонова — уч. Гонсалеса и Вудса). Голографический восстанавливающий фильтр на известную восстанавливающую функцию.

Лекция 3

Методы автоматического анализа изображений (заимствованы из статистической радиотехники). При повороте и масштабировании добавляется доп. Информация, считающаяся шумом. Дополнительный шум уничтожает полезную информацию. Построение разделяющей поверхности — трудная и важная задача. Поиск инвариантных признаков. Как решаются нелинейные и линейные задачи при поиске в пространстве признаков (построении разделяющей поверхности).

Каноническое уравнение прямой задает линию в n-мерном пространстве:

$$\vec{W}^T \cdot \vec{X} = 0$$

Если классов 3, то требуется 3 разделяющих правила. Пример решения задачи в аналитическом виде. Пусть функция распределения будет гауссовым (что далеко не всегда выполняется). Применим *байесовские классификаторы* (оптимальны при нормальном распределении).

- Признаки — $x_i \dots$
- Классы — $w_i \dots$

Образ относится к классу w_i если выполняется следующее условие:

$$P(\vec{X} \vee w_i) > P(\vec{X} \vee w_j) ; \quad \forall i \neq j$$

$$\frac{P(\vec{X} \vee w_i)}{P(\vec{X} \vee w_j)} > 1$$

$$P(\vec{X} \vee w_i) = \frac{1}{(2\pi)^{n/2} |C^{1/2}|} \exp \left[-\frac{1}{2} (\vec{X} - \vec{m}_i)^T C^{-1} (\vec{X} - \vec{m}_i) \right]$$

Ковариационная матрица:

$$C_{k,l} = E[(x_k - m_k) \cdot (x_l - m_l)] = \iint_{-\infty}^{\infty} (x_k - m_k)(x_l - m_l) p(x_k, x_l) dx_l dx_k$$

По диагонали дисперсии — остальные члены — ковариации.

$$\frac{P(\vec{X} \vee w_i)}{P(\vec{X} \vee w_j)} = \exp \left[\frac{-1}{2} ((\vec{X} - \vec{m}_i)^T C^{-1} (\vec{X} - \vec{m}_i) - (\vec{X} - \vec{m}_j)^T C^{-1} (\vec{X} - \vec{m}_j)) \right]$$

Упростим (Можно посмотреть в учебнике Гонсалеса):

$$r_{ij} = \ln \frac{P(\vec{X} \vee w_i)}{P(\vec{X} \vee w_j)} = \vec{X}^T C^{-1} (\vec{m}_i - \vec{m}_j) - \frac{1}{2} (\vec{m}_i + \vec{m}_j)^T C^{-1} (\vec{m}_i - \vec{m}_j)$$

Это линейная разделяющая поверхность, построенная исходя из нормальной модели распределения образов различных классов, которые мы хотим различить. Более спекулятивный вариант — методы максимального правдоподобия. Применяются еще более спорные предположения.

1. Пусть у нас будут всего 2 класса: w_1 и w_2
2. Предлагают матрицу штрафов:

$$\begin{matrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{matrix}$$

Задача заключается в минимизации риска. Если мы предполагаем, что объект принадлежит в i-ому классу, тогда

$$r_j(\vec{X}) = \sum_{i=1}^M L_{ij} P(w_i \vee \vec{X})$$

Для минимизации применим формулу Байеса:

$$P(w_i \vee \vec{X}) = \frac{P(w_i) P(\vec{X} \vee w_i)}{P(\vec{X})}$$

С учетом нее перепишем формулу для риска:

$$r_j = \frac{1}{P(\vec{X})} \sum_{i=1}^M L_{ij} P(w_i) P(\vec{X} \vee w_i) \quad ,$$

где r_j – функция правдоподобия для класса w_i

$$r_1(\vec{X}) = L_{11} P(w_1) P(\vec{X} \vee w_1) + L_{21} P(w_2) P(\vec{X} \vee w_2)$$

$$r_2(\vec{X}) = L_{12} P(w_1) P(\vec{X} \vee w_1) + L_{22} P(w_2) P(\vec{X} \vee w_2)$$

Мы хотим, чтобы риск при отнесении объекта к первому классу был меньше, чем риск отнесения объекта ко второму классу. Тогда решение выражается как:

$$(L_{21} - L_{22}) P(\vec{X} \vee w_2) P(w_2) < (L_{11} - L_{12}) P(\vec{X} \vee w_1) P(w_1)$$

$$\frac{P(\vec{X} \vee w_1)}{P(\vec{X} \vee w_2)} = \frac{P(w_2)(L_{21} - L_{22})}{P(w_1)(L_{11} - L_{12})}$$

Лекция 4

Перцептрон (Розенблатт 1957 г.):

1. Сенсорный слой
2. Ассоциативные элементы
3. Решающие элементы

Использовалась модель нейрона Маккалока-Питса в виде сумматора (но с отсутствием пороговой функции). Модель либо в импульсном либо в непрерывном виде. В импульсном виде нейрон возбуждается при большом количестве входных импульсов и разрешается спайк последовательностью. Решающий нейрон на выходе реализует функцию суммирования возбуждения ассоциативных нейронов с некоторыми весами:

$$R = \sum_i w_i x_i$$

Если выход не соответствует

$$\tilde{x} < w_j, \text{ то } \tilde{w}(k+1) = \tilde{w}(k) - c; \quad c > 0$$

Если выходное возбуждение превышает некоторый порог

$$\tilde{x} > w_j, \text{ то } \tilde{w}(k+1) = \tilde{w}(k) + c; \quad c < 0$$

Перцептрон реализует функцию *градиентного спуска*.

$$\text{grad } f = \left[\frac{df}{dx_1}, \frac{df}{dx_2}, \dots, \frac{df}{dx_n} \right]^T$$

Покажем, что перцептрон реализует обучение методом градиентного спуска. Пусть есть функция $J(\vec{w}, \vec{x})$ – будем считать, что она выбрана правильно и при ее минимизации мы получим систему, обученную наилучшим образом.

$$\vec{w}(k+1) = \vec{w}(k) - c \left\{ \frac{dJ(\vec{w}, \vec{x})}{d\vec{w}} \right\}$$

В этом случае градиент будет иметь вид:

$$J(\vec{w}, \vec{x}) = \frac{1}{2} (|\vec{w}^T \vec{x}| - \vec{w}^T \vec{x})$$
$$\frac{dJ(\vec{w}, \vec{x})}{d\vec{w}} = \frac{1}{2} [x \text{sign}(\vec{w}^T \vec{x}) - \vec{x}]$$

Подставив, получаем формулу для обучение:

$$\vec{w}(k+1) = \vec{w}(k) - \frac{c}{2} [\vec{x}(k) \text{sign}(\vec{w}^T(k) \vec{x}(k)) - \vec{x}(k)]$$

Можно представить обучение перцептрона, как минимизацию среднеквадратичной ошибки:

$$J(\vec{w}, \vec{x}, b) = \frac{1}{2} \left[\sum_{i=1}^N (\vec{w}^T \vec{x} - b)^2 \right]$$

Более подробно рассмотрим при аппроксимации контуров отрезками прямых линий.

Взгляд притягивается к точкам, углам и может скользить по границе. Метод чампферного расстояния позволяет сопоставлять тонкие границы двух изображений. В зрительной и оптической системе изображений претерпевают аффинные и проективные преобразования.

Для описания границ можно так же использовать прямые линии (описать границу набором прямых линий). Прямые линии похожи друг на друга, поэтому нужно добавить какие-то новые структурные элементы при распознавании. Такими элементами являются углы, пересечения (X и Y образные) и дуги эллипсов. Проблема в описании.

В качестве алфавита возьмем прямые линии и углы. Используем искусственную нейронную сеть Хопфилда-Танка. *Метод исчерпывающего прохода по дереву решений*

Метод обратного распространения ошибки

Метод обучения многослойных перцептронов — метод обратного распространения ошибки (). Роммельхарт и Маклеланд использовали нелинейную модель нейрона. Это был линейный сумматор с настраиваемыми весами с нелинейным функциональным преобразователем с функцией $f(S)$:

$$f(S) = \frac{1}{1 + \exp(-S)}$$

– сигмоидная (?) функция (получила распространение как обучаемый функциональный преобразователь: Фукушима когнитрон, неокгнитрон, Роберт Хэчнелсон — учебник по нейронным сетям).

Обучение многослойного перцептрона методом обратного распространения ошибки. Нарисовали многослойный перцептрон, а нейроны последнего слоя обозначили как O_1^m . Верхний индекс — номер слоя, нижний индекс — номер нейрона... Нейрон описывается схемой сумматора (суммирует сигналы от входных нейронов с некоторыми весами и возбуждается в определенном случае в соответствии с сигмоидной функцией).

$$O_L^P = f[S_L^{(P)}] = f\left[\sum_{i=0}^{Np-1} w_{Li}^{(P)} O_i^{(P-1)}\right]$$

Задачи обучения в минимизации «энергии» ошибки (среднеквадратической ошибки):

$$E_n^m = \frac{1}{2} \sum_{k=0}^{Nm} [t_k^m(n) - O_k^m(n)]^2,$$

где n — номер итерации

Минимизация E_n^m производится методом градиентного спуска в пространстве весовых коэффициентов всех связей нейронов в сети. Приращение связи должно быть пропорционально отрицательному значению градиента ошибки.

$$\Delta w_{ki}^m = -\frac{\delta E^m}{\delta w_{ki}^m} = \frac{-\delta E^m}{\delta S_k^m} \frac{\delta S_k^m}{\delta w_{ki}^m} = \delta_k^m \cdot O_i^{m-1}$$

- этот прием замены переменной будет неявно применяться во всем выводе.

$$\frac{\delta S_k^m}{\delta w_{ki}^m} \text{ — это } \delta_k^m.$$

Эта ошибка будет распространяться «вглубь» сети.

$$\delta_k^m = \frac{-\delta E^m}{\delta S_k^m} = \frac{-\delta E^m}{\delta O_k^m} \frac{\delta O_k^m}{\delta S_k^m} = (t_k^m - O_k^m) f'(S_k^m)$$

$$\delta_j^{m-1} = \frac{-\delta E^m}{\delta S_j^{m-1}} = \frac{-\delta E^m}{\delta O_j^{m-1}} \frac{\delta O_j^{m-1}}{\delta S_j^{m-1}} = \left[\sum_{k=1}^{Nm} \frac{\delta E^m}{\delta S_j^{m-1}} \frac{\delta S_j^{m-1}}{\delta O_j^{m-1}} \right] \frac{\delta O_j^{m-1}}{\delta S_j^{m-1}} = \left[\sum_{k=1}^{Nm} \delta_k^m w_{kj}^m \right] f'(S_j^{m-1})$$

Это по определению производная сигмоидной функции. Таким образом из ошибки в одном слое мы можем определить ошибку в другом слое, и считать ошибки нейронов в соответствующих слоях.

Главный вопрос — как считать производную сигмоидной функции. Стараются пользоваться линейным участком этой функции.

Вернемся к линейным разделяющим поверхностям:

$$d(\vec{x}) = \vec{w}^T \vec{x}$$

Простые случаи:

1. Два класса: $d_i(\vec{x}) = >0 : \vec{x} \in w_i ; <0 : \vec{x} \notin w_i$
2. Большой количество классов (работам на отделение некоторого класса от всех других

- по отдельности. $d_{ij}(\vec{x}) > 0, \vec{x} \in w_i, i \neq j$
3. $d(\vec{x}) > d_j(\vec{x}) \forall \vec{x} \in w_i, i \neq j$. Этот случай сводится ко второму, если $d_{ij}(\vec{x}) = d_i(\vec{x}) - d_j(\vec{x}) - (w_i - w_j)^T \vec{x} > 0$

Рассмотрим как строится разделяющая поверхность по набору известных результатов классификации. Пусть задано по 2 представителя от двух разных классов (верхний индекс — класс, нижний — номер элемента).

$$w_1 x_{11}^1 w_2 x_{12}^1 + w_3 > 0$$

$$w_1 x_{21}^1 w_2 x_{22}^1 + w_3 > 0$$

$$w_1 x_{11}^2 w_2 x_{12}^2 + w_3 < 0$$

$$w_1 x_{21}^2 w_2 x_{22}^2 + w_3 < 0$$

Т.к. неизвестных весов, то нам необходимо всего 3 уравнения для получения решений. Про недоопределенные системы можно прочесть в Туи-Гонсалеса.

Как решают проблемы с нелинейными поверхностями? Строят нелинейные разделяющие поверхности по линейным правилам, переходя в другую систему координат. От исходных систем признаков \vec{x} переходим в пространство признаков $\vec{f}(\vec{x})$. Размерность исходного пространства — n , размерность нового пространства — k . В общем случае k может быть много больше n . Можно использовать полиномы вычисленные для признаков исходного пространства \vec{x} .

Если $m=1$, то $\vec{f} = \vec{x}$, $f_i = x_i$

Если $m=2$, то $f(\vec{x}) = w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2 + w_1x_1 + w_2x_2$, $\vec{f} = (x_1^2, x_1x_2, x_2^2, x_1, x_2)$.

Полиномы легко реализовать на вычислительной машине и любую нелинейную функцию можно аппроксимировать полиномами. Необходимо решить проблему избыточности нового базиса (чтобы одни элементы нельзя было выразить через другие).

Понятие скалярного произведения:

$$(f, g) = \int_a^b f(x)g(x)dx$$

$$(f, f) = \int_a^b f^2(x)dx \quad \text{— норма функции}$$

Две функции ортогональны, если

$$(f, g) = 0$$

$$\int_a^b u(x)f(x)g(x)dx = 0$$

Ортогональной системой называется система функций, которые ортогональны.

$$\int_a^b \varphi_i(x)\varphi_j(x)dx = A_{ij}\delta_{ij}, \quad \text{где } \delta_{ij} \text{ — дельта символ Кронекера.}$$

Система функций ортонормированная, если элементы A_{ij} равны 1. Множество функций линейно независимо если не существует c_1, \dots, c_m не равных 0 для которых выполняется следующее равенство:

$$c_1 f_1(x) + c_2 f_2(x) + \dots + c_m f_m(x) = 0$$

Все образующие функции являются линейно независимыми.

Система функций называется полной, если любую кусочно-непрерывную функцию можно сколь угодно точно аппроксимировать линейной комбинацией функций, входящих в эту систему.

Функции, которыми мы хотим пользоваться для перехода между пространствами признаков должны использовать базисы, удовлетворяющие этим условиям. Существуют

специальные полиномы, которые удовлетворяют рассмотренным условиям. Ниже рассмотрены системы таких полиномов

Полиномы Лежандра

$(k+1)P_{k+1}(x) - (2k+1)xP_k(x) + kP_{k-1}(x) = 0$; По этой формуле порождаются полиномы Лежандра при $k > 1$; $P_0 = 1$, $P_1 = x$.

Показано, что $A_k = \frac{2}{2k+1}$

Функция для ортонормирования:

$$\varphi_k(x) = \sqrt{\frac{u(x)}{A_k}} P_k(x) = \sqrt{\frac{2k+1}{2}} P_k(x) ; \quad k=0 \quad .$$

Недостаток: интервал, на котором полиномы Лежандра ортогональны $[-1, 1]$.

Полиномы Лагера

$$L_{k+1}(x) - (2k+1-x)L_k(x) + k^2 L_{k-1}(x) = 0 \quad . \quad k \geq 1$$

$$L_0(x) = 1$$

$$L_1(x) = -x - 1$$

Интервал ортогональности: $(0, \infty)$

$$u(x) = e^x$$

После нормировки получаем более сложную конструкцию:

$$\varphi_k(x) = \frac{\exp(-x/2) L_k(x)}{k!}$$

Полиномы Эрмита

Ортогональны на всей числовой оси. Синтезируются при помощи функции:

$$H_{k+1}(x) - 2xH_k(x) + 2kH_{k-1}(x) = 0 \quad , \quad k \geq 1 \quad ;$$

$$H_0(x) = 1$$

$$H_1(x) = 2x$$

Ортогональны относительно $u(x) = \exp(-x^2)$

После ортонормирования:

$$\varphi_k(x) = \frac{\exp(-x^2/2) H_k(x)}{2^k k! \sqrt{\pi}}$$

Лекция 5

В конце прошлой лекции не было возмущенных возгласов насчет линейно-независимых ортонормированных функций, зависящих от единственного аргумента. Пространство признаков же известно. 1955 Гильберт и Коранд доказали, что с помощью ортонормированной системы функций одного аргумента можно построить систему ортонормированных функций для большего количества аргументов.

Пусть есть система функций:

$$\Phi_1(x), \Phi_2(x) \dots$$

ортogonalных на $[a, b]$. Системы функций от двух аргументов можно построить следующим образом:

$$\varphi_1(x_1, x_2) = \Phi_1(x_1) \Phi_2(x_2)$$

$$\varphi_2(x_1, x_2) = \Phi_1(x_1) \Phi_2(x_2)$$

$$\varphi_3(x_1, x_2) = \Phi_2(x_1) \Phi_1(x_2)$$

$$\varphi_4(x_1, x_2) = \Phi_2(x_1) \Phi_2(x_2)$$

$$\varphi_5(x_1, x_2) = \Phi_1(x_1) \Phi_3(x_2)$$

...

В этом случае система функций будет ортогональна в квадрате

$$[a, a; a, b; b, a; b, b] \text{ .}$$

Рассмотрим пример для четырех аргументов:

$$\varphi_1(\vec{x}) = \Phi_1(x_1) \Phi_1(x_2) \Phi_1(x_3) \Phi_1(x_4)$$

$$\varphi_2(\vec{x}) = \Phi_1(x_1) \Phi_1(x_2) \Phi_1(x_3) \Phi_2(x_4)$$

$$\varphi_3(\vec{x}) = \Phi_1(x_1) \Phi_1(x_2) \Phi_2(x_3) \Phi_1(x_4)$$

$$\varphi_4(\vec{x}) = \Phi_1(x_1) \Phi_1(x_2) \Phi_2(x_3) \Phi_2(x_4)$$

...

Бесконечно-размерный базис. Система будет ортогональна в гиперкубе со стороной $[a, b]$

Ортонормированные функции предполагалось строить следующим образом:

$$\Phi_i(x) = \sqrt{\frac{u(x)}{A_i}} \Phi_i(x)$$

$\vec{w}^T \vec{x} = 0$ - разделяющая поверхность. Подбираем весовые коэффициенты перед элементами базиса. Нормирующий коэффициент войдет при подборе в весовой коэффициент. Вернемся к построению разделяющих поверхностей. Машины опорных векторов (метод опорных векторов). Данный метод дает дополнительный вариант итеративного оптимизационного построения разделяющих поверхностей высокого качества. Этот метод широко применяется. Разрабатывался в конце прошлого века. Сейчас получил распространения при построении систем распознавания образов.

Когда мы имеем всего представителей двух классов представленных облаками точек измеренных в пространстве признаков. Как добиться устойчивости к шумовым искажениям. Устойчивость цифровой техники благодаря необходимости большого воздействия для переключения триггеров. Погода подобна триггеру. Подобным образом достигается устойчивость к шумовым воздействиям. В районе где действие шума велико - возможно "перескакивание" из одного класса в другой. Для сведения такого влияния к минимуму необходимо оптимальным образом расположить дискриминационные поверхности. Метод опорных векторов позволяет построить дискриминационную поверхность таким образом, чтобы она была равно удалена от самых близких элементов разделяемых классов. Самые близкие вектора являются опорными для такой дискриминационной поверхности.

Вариант использования дополнительной информации в переопределенной системе уравнений. Когда мы хотим построить эту оптимальную линейную поверхность нами

используется следующий формализм.

$$y_i(w x_i + b) \geq 0$$

$$y_1 = 1; y_2 = -1$$

В опорных векторах разрешается равенство $y = 0$. Мы хотим максимизировать расстояние

$$y_i(w x_i + b) \geq 1$$

$$\vec{w}(\vec{x}_1 - \vec{x}_2) = 2 \quad (\text{максимальное расстояние})$$

С учетом этого:

$$\left(\frac{\vec{w}}{\|\vec{w}\|} \cdot x_{1m} + \frac{b}{\|\vec{w}\|} \right) - \left(\frac{\vec{w}}{\|\vec{w}\|} \cdot x_{2l} + \frac{b}{\|\vec{w}\|} \right) = \frac{\vec{w}}{\|\vec{w}\|} (x_1 - x_2) = \frac{2}{\|\vec{w}\|}$$

Мы должны будем минимизировать эту величину. В методе опорных векторов это величина:

$$\frac{1}{2} \vec{w}^T \vec{w}$$

Метод множителя Лагранжа

Вводится множитель Лагранжа

$$\frac{1}{2} \vec{w}^T \vec{w} - \sum_i^N \alpha_i [y_i (\vec{w}^T \vec{x}_i + b) - 1]$$

Решение получают при помощи максимизации выражения:

$$\sum_j^N \alpha_j - \frac{1}{2} \sum_i^N \alpha_i (y_i y_j x_i x_j) \alpha_j$$

После решения весовые коэффициенты разделяющей поверхности находятся следующим образом:

$$\vec{w} = \sum_i^N \alpha_i y_i \vec{x}_i$$

Таким образом при помощи метода множителей Лагранжа решается оптимизационная задача (квадратичное программирование). Система признаков: моментные инварианты. Широко используется в системах распознавания образов. Авторы Hu, Maitra Гонсалес и Вудс

Инварианты к преобразования группы подобия (сдвиг, вращение, масштабирование).

Моменты:

$$M_{pq} = \iint_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

$$M_{ij} = \sum_x^N \sum_y^N x^i y^j I(x, y) dx dy$$

M_{00} - сумма яркостей изображения

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad - \text{центры тяжести (центр масс или центроид)}$$

$$\bar{y} = \frac{M_{01}}{M_{00}}$$

Центральные моменты:

$$\mu_{pq} = \iint_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy$$

$$\mu_{00} = M_{00}$$

$$\mu_{10} = \mu_{01} = 0$$

Моменты центральных порядков, используемые авторами записываются как:

$$\begin{aligned}
\mu_{11} &= M_{11} - \bar{x} M_{01} = M_{11} - \bar{y} M_{10} \\
\mu_{20} &= M_{20} - \bar{x} M_{10} \\
\mu_{02} &= M_{02} - \bar{x} M_{01} \\
\mu_{21} &= M_{21} - 2\bar{x} M_{11} - \bar{y} M_{20} + 2\bar{x}^2 M_{01} \\
&\dots \\
\mu_{30} \\
\mu_{03}
\end{aligned}$$

Эти моменты инвариантны к сдвигу. Далее вычисляем признаки, инвариантные к масштабированию:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{1+\frac{i+j}{2}}}$$

$$I_1 = \eta_{20} + \eta_{02} ;$$

$$I_2 = (\eta_{20} + \eta_{02})^2 + (2\eta_{11})^2 ;$$

$$I_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 ;$$

...

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} - \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Еще один способ при классификации - компенсировать преобразования, а потом использовать простые признаки.

Это получение инвариантных признаков

$$P = f\{B(x, y), T(\alpha, \beta \dots)\}$$

$$P'(a, b \dots) = \text{const}[T(\alpha, \beta \dots)]$$

Можно поступить по-другому:

$$P'' = \text{const}(i)$$

$$P'' = f_1(\alpha)$$

$$P'' = f_2(\beta)$$

После компенсации будет легче выполнять распознавание.

Для изображений, имеющих среднюю ширину пространственного спектра происходит деградация корреляционного отклика при вращении и масштабировании приблизительно на 5-6%. Пространственно-спектральный состав средней ширины - спектр, характерный для изображений окружающего мира. У мелкодетальных изображений спектр широкий. Низкие частоты несут информацию о масштабе, координате и ориентации. Высокие частоты несут информацию о деталях. Эталон должен иметь узкий спектр пространственных частот. Попробуем синтезировать самый низкочастотный эталон. Раскладывая sinc-функцию в ряд Тейлора и бесконечно ее растягивая, получим что самый узкий спектр имеет бесконечно растянутая парабола второй степени. Избежать эффекта Гиббса, появляющегося из-за того, что наша парабола ограничена в пространстве, можно устранить, если предположить что фон бесконечен.

$$\begin{aligned}
F(x, y) &= [M(x-a)]^2 + (y-b)^2 + C \\
f(\tau) &= \iint_{\Omega} B(x, y) F(x+\tau, y) dx dy
\end{aligned}$$

- корреляция с нашей параболой.

Приравняем величину сдвига с сдвигу вершины параболы $\tau = -a$. Нас интересует, где будет экстремум функции корреляции. Взяв производную по параметру сдвига и приравняв ее к нулю получим:

$$\frac{df(\tau)}{d\tau} = \iint_{\Omega} B(x, y) \frac{d}{d\tau} F(x+\tau, y) dx dy = \iint_{\Omega} B(x, y) 2M^2(x-a) dx dy = 0$$

Рассмотрим функцию:

$$x_c = \frac{\iint_{\Omega} x B(x, y) dx dy}{\iint_{\Omega} B(x, y) dx dy} \quad - \text{ центр тяжести.}$$

Функция корреляции имеет максимум (единственный если функция) при совпадении вершины параболы с центром тяжести изображения (!!!). Аналогично можно найти параметр сдвига по y . Сферическая линза дает группу аффинных преобразований (A). Вращая объектив с разных ракурсов мы получаем аффинное преобразование:

$$\vec{x}' = A \vec{x}$$

Аффинные преобразования: сжатие, масштабирование, вращение, поворот, сдвиг. Как измерить составляющие аффинного преобразования. Необходимо чтобы компенсация по какому-то из преобразований не должны нарушать предыдущих. Центр тяжести в начале координат в результате аффинных преобразований не меняется! Нулевые координаты центра тяжести являются эталонными. Масштабирование и вращение не меняет азимуты. Масштабирование и вращение - нормальные делители группы (друг на друга не влияют), следовательно последовательность их применения не важна (можно применить последними). Зеркальное отражение. Вытягивание меняет азимутальные и радиальные координаты точек.

Необходимо выяснить в каком направлении было вытянуто изображений. Будем считать корреляцию нашего параболоида с некоторым эллиптическим параболоидом. Получим периодическую зависимость, по максимумам которой можно определить направление вытягивания. При компенсации вытягивания та же корреляция станет константой.

$$\theta = \frac{1}{2} \arctg\left(\frac{C}{B}\right) + n\pi$$

$$\mu = \pm \sqrt{(D + \sqrt{C^2 + B^2}) / (D - \sqrt{C^2 + B^2})}$$

$$B = \iint_{\Omega} B(\rho, \varphi) \rho^2 \cos(2\varphi) d\rho d\varphi$$

$$C = \iint_{\Omega} B(\rho, \varphi) \rho^2 \sin(2\varphi) d\rho d\varphi$$

$$D = \iint_{\Omega} B(\rho, \varphi) \rho^2 d\rho d\varphi$$

$$A = A_1 A_2 A_3 A_4 A_5 A_6$$

- A_1 и A_2 - сдвиги по осям
- A_3 и A_4 - вытягивание
- A_5 - масштабирование
- A_6 - вращение
- A_7 - зеркальное отражение

Проинтегрировал по углу в радиальных координатах можно получить проекцию изображения на радиус (функция, описывающаяся как расстояние точки от начала координат). При масштабировании функция и центр тяжести меняется в m раз. Таким образом можно померить коэффициент масштабирования независимо от класса изображений.

$$\rho_c = \frac{\iint_{\Omega} \rho B(\rho, \varphi) d\rho d\varphi}{\iint_{\Omega} B(\rho, \varphi) d\rho d\varphi} \quad - \text{ центр тяжести в полярных координатах.}$$

Можем ли мы применить такие-же проекции для измерения угловой координаты? Необходимо использовать функции \sin и \cos .

Лекция 6

Универсальным эталоном была квадратичная парабола. Центр тяжести перенесен в начало координат. Целесообразно рассматривать признаки объекта на подпространстве, в котором. Таким пространством можно считать проекцию изображения на азимутальную координату в полярной системе координат. (проекция изображения на окружность). Проекция изображения на окружность содержит всю необходимую информацию, которую можно извлечь, однако это проекция имеет особенность по сравнению с декартовой системой координат - проекция на окружность циклическая. Как тогда применять наш универсальный метод? В качестве универсальной эталонной функции следует использовать синус.

Из опубликованных исследований известно, что параметры вращения и зеркального отражения могут быть однозначно определены по параметрам нескольких синусоид, имеющих некратную частоту.

Фаза первой гармоники несет информацию о вращении. Первая гармоника может быть неявно выражена. Для квадрата только четвертая гармоника. Если взять 3 соседних хорошо выраженные гармоники (кроме первой, чтобы они были не кратными), мы можем измерить при помощи этих гармоник параметры вращения и зеркального отражения изображений.

Если проецировать изображения на окружность и взять Фурье-Спектр:

$$\varphi_k + 2\pi i_k = \varepsilon \Psi_k + \varphi_k$$

$$\varphi_n + 2\pi i_n = \varepsilon \Psi_n + \varphi_n$$

$$\varphi_m + 2\pi i_m = \varepsilon \Psi_m + \varphi_m$$

ε показывает есть ли зеркальное отражения (если есть, то фазы меняются на противоположные).

Если попарно вычесть уравнения друг из друга получим систему уравнений:

$$(\varphi_k - \varphi_n) + 2\pi(i_k - i_n) = \varepsilon(\Psi_k - \Psi_n) + \varphi(k - n)$$

$$(\varphi_k - \varphi_m) + 2\pi(i_k - i_m) = \varepsilon(\Psi_k - \Psi_m) + \varphi(k - m)$$

Во-первых:

$$k - n = -1$$

$$k - m = 1$$

т.к. гармоники расположены по обеим сторонам от k -ой.

Во-вторых, поскольку эти гармоники соседние и разность фаз, набегаящая между соседними гармониками меняется от 0 до 2π , $\varphi \in (0, 2\pi)$. Для соседних гармоник вторая часть уравнений не наблюдается. Можно упростить до:

$$(\varphi_k - \varphi_n)(k - m) - (\varphi_k - \varphi_m)(k - n) = \varepsilon[(\Psi_k - \Psi_n)(k - m) - (\Psi_k - \Psi_m)(k - n)]$$

Откуда получаем выражение для параметра зеркального отражения:

$$\varepsilon = \frac{2\varphi_k - \varphi_n - \varphi_m}{2\Psi_k - \Psi_n - \Psi_m}$$

И подставив, найдем поворот.

$$\varphi = \varphi_k - \varphi_m - \varepsilon(\Psi_k - \Psi_m)$$

При помощи описанных выше способов можно посчитать все параметры аффинного преобразования, которое записывается в виде:

$$\vec{x}' = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \vec{x}$$

Главным недостатком является необходимость выделять объект из фона. Это существенное ограничение может быть преодолено в фабричных условиях (контраст объекта с фоном). Можно использовать описание объектов при помощи структурных элементов (контуров), которые могут быть описаны при помощи прямых линий и кривых второго

порядка. Докажем, что алфавит структурных элементов, содержащий прямые линии, окружности и эллипсы.

При дифференцировании парабол (нашего эталона) получаем прямую линию - структурный элемент, который имеет бесконечно узкую спектральную полосу. Он так же инвариантен как и парабола инвариантен к аффинным преобразованиям. Пример такого фильтра: оператор Прюитт. Этот эталон хорош, так как имеет узкую спектральную полосу. При использовании свертки считается, что изображение вне скользящего окна равно 0.

Структурный анализ происходит на макроизображениях. На микроизображениях анализируются текстуры.

Как текстуры распознаются в зрительной системе животных и человека? Текстуры распознаются в виде совокупности текселов (зернышек, похожих на маски Лавса). В зрительной системе имеются специальные детекторы зернышек текстуры. Эти детекторы работают на пределе зрительного разрешения. Для анализа одновременно передается небольшое количество (70 бит информации). Перефокусировка глаза происходит достаточно быстро. При предъявлении зрительного стимула на доли секунды мы не успеваем заметить детали изображения и видим только наиболее масштабные отличия. Если картинку наблюдать дольше, то исследование объекта глазом будет более детально. На высоком уровне распознавание текстур определяется шириной зоны внимания (на низком уровне - количеством палочек и колбочек).

В зрительной системе человека есть специальные детекторы текстур. Поскольку текстура анализируется на пределе разрешения как зернышко имеющая самую простую форму (круги). Самые простые - эллиптические детекторы. В зрительной системе имеются детекторы - лапласиан-гауссианы. Для очень удлиненных объектов объединяется информация о соседних текселах, имеющих близкую ориентацию. В работах Vistnes'a представлены варианты аппроксимации текселов различной ориентации.

На пределе пространственного разрешения от детектора типа "Мексиканская шляпа" остается парабола. Такие детекторы были применены для подсчета статистики различных видов текселов на изображении. Можно применять для выделения границ оператора Прюитт. Это анализ на микро-уровне. Для разделения совокупности контуров на локальные образы на макро-области. Можно применить механизм зон внимания.

По каким параметрам наша зрительная система отделяет поверхности друг от друга? Человеческая кожа имеет специфический оттенок. В rg пространстве (нормирование r и g компоненты к b) область соответствующая цвету кожи человека - это сосиска на планковской кривой. Она хорошо отделяется двумя параболой. Цвет определяется количеством пигмента (меланина) и количеством кровеносных сосудов. Метод выделения лиц на основе цвета работает при недостаточной освещенности. При хорошей освещенности кожу лучше распознавать HSV пространстве на основе анализа H компоненты (оттенка).

В OpenCV есть выделение лиц по методу Виолы-Джонса, вейвлетами Хаара, структурным методом (взаимному расположению деталей). Надежен, но не всегда достаточно быстро.

Лекция 7

Невозможно скомпенсировать проективные преобразования при помощи структурных элементов. Когда расстояние до объекта соизмеримо с расстоянием до самого объекта - проективное преобразование. В исследовательской работе по автоматическому наведению крылатой ракеты на цель считается, что инерциальная навигационная система ракеты настолько хороша, что выводит ракету на прямую видимость цели. В памяти есть либо эталонное изображение, либо эталонное описание цели. Ракета видит цель - корректирует ошибку инерциальной навигационной системы и достаточно точно попадает в уязвимую точку цели.

Как без использования гироскопа? Если на земле - на колесах датчики, которые считают кол-во поворотов + известна длина окружности колеса. По кол-ву поворотов каждого колеса можно определить тип и расстояния движения. аналогичным образом работают датчики вычисления пути на спидометре.

Для летающих целей используют гироскопы - они сохраняют направление оси вращения. Земля - тоже гироскоп. Согласно исследованиям - стабильно развитие жизни возможно благодаря Луне. Гироскоп из-за Луны более стабилен. Условия вращения такие, что они способствовали развитию жизни. Более ранний вариант стабилизации - маятниковые стабилизаторы (ФАУ-1 и ФАУ-2). В Исакиевском соборе - маятник Фуко.

В свое время американцы разместили на территории ФРГ баллистические ракеты ближнего радиуса действия, имевшие наведение без распознавания образов. Дальнометы позволяют строить трехмерную модель местности (радиолокация). Крылатые ракеты летят низко (чтобы было не видно) и медленно.

К чему все это? Расстояние от крылатой ракеты до цели не велико и мы получаем проективное преобразование, а оно мешает.

Почему не использовать эллиптический параболоид в качестве универсального эталона для компенсации проективных преобразований? Потому что он не инвариантен к этим преобразованиям. Какие плюсы у нас остаются? При проективном преобразовании смещается, например, центр тяжести.

Можно отследить параметры проективного преобразования например отслеживая смещения центра тяжести при применении маленьких проективных преобразований в 2 противоположные стороны. Находя максимальную разность смещения центра тяжести и делая маленькие контр-преобразования аналогично градиентному спуску можно найти параметры исходного преобразования.

Показано, что когда проективное преобразование почти компенсировано начинаются бифуркации, о фактически мы уже получили необходимую компенсацию преобразования. Вот и пригодились методы оптимизации для измерения параметров проективного преобразования.

По параметрам внутренней асимметрии под действием проективного преобразования можно измерить в каком направлении оно изменяется быстрее всего. Если положить это эталонное направление на ось x мы можем компенсировать вращение. Если у нас есть 2 таких направления. Взяв оба направления мы можем померить еще и зеркальное отражение.

Структурные методы сопоставления изображений позволяют решить задачу сопоставления тогда, когда корреляция абсолютно не работает (для снимков, сделанных в разные сезоны при помощи разных датчиков и т.д.).

На прошлой лекции рассматривали анализ текселов для сопоставления. С увеличением масштаба изменяются свойства изображений (контур в другом месте и т.д.). Так работают SIFT и аналогичные методы. Используем зоны внимания. Считается что в центре зоны внимания выделяется больше всего информации о свойствах объекта.

Структурные методы распознавания образов появились в 70х годах прошлого века.

Методы, рассматривающие изображения, как нечто целое (корреляционные и признаковые) не работают! К этому моменту были разработаны компьютеры, в которых использовалась математическая лингвистика. Хотели придумать формальные методы структурного описания (автоматные грамматики), которые легко реализовывались на конечных автоматах (автоматы Милли и Мура). Такие автоматы использовались для построения простых вычислительных устройств в больших вычислительных машинах. Как устроен процессор? Автоматы Милли и Мура позволяли аппаратно, а не программно реализовывать простые арифметические операции. Автоматная грамматика легко реализуется микропрограммными автоматами. К сожалению оказалось, что живой язык содержит исключений больше чем правил. Описать живые языки формальными грамматиками не получилось, так же как и описать многообразие окружающего мира. Дальнейшие 2 десятилетия шли по линии попыток применения формальных правил для решения проблем автоматического распознавания изображений. Что имеется в виду под формальными правилами? Начали появляться книжки по ИИ, содержавшие описание автоматической игры в шахматы и другие игры, автоматического доказательства теорем и т.д. Использовались суперкомпьютеры, в которых использовались знания, однако знаний слишком много, чтобы заложить туда достаточно знаний. Такие программы реализовывались в крупных научных программах и военных проектах. Однако суперкомпьютер на боевой самолет не поставишь. Для помощи летчикам используют экспертные системы для помощи летчикам.

Распознавание образов

Классификация не работает с объектами, меняющими форму. Метод внутренних расстояний - метод автоматического распознавания образов с помощью инвариантных признаков, подсчитанных хитрым образом. Что мы имеем в виду, говоря, что форма немного изменяется? Этот метод ориентирован на работу в таких случаях. Это гистограммный метод. Вместо гистограммы расстояний длины хорд к каждой точке считается гистограмма кратчайших расстояний от каждой точки контура для каждой точки контура. Такую гистограмму целесообразно делать не одномерной (ибо она слишком гладкая), например брать длину ломаной или не ломаной хорды и соответствующие этой хорде направления касания с линией контура. Получим двумерную гистограмму. Дополнительная информация оказывается очень продуктивной. Различающая способность значительно улучшается.

ЕМ (Expectation maximization) - метод ожидания максимизации (ОМ). У нас нет достоверных данных о том, какие пиксели относятся к фону, а какие к объекту и нет данных о статистике. Но задали начальные данные - какие данные точно фон, а какие - к объекту. Ковариационная матрица:

$$E = [(\vec{x} - \vec{m}) \cdot (\vec{x} - \vec{m})^T] = \begin{pmatrix} E[(x_1 - m_1)^2] & \dots & \dots & \dots \\ \dots & E[(x_2 - m_2)^2] & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & E[(x_n - m_n)^2] \end{pmatrix}$$

Преобразование Каруна-Лоэва

Матрицы имеют собственные вектора. Чем отличается собственный вектор от несобственных? Собственный вектор подчиняется уравнению:

$$(A) \vec{x} = \lambda \vec{x}$$

Если матрица не вырождена и имеет размер $n \times n$, то она будет иметь n отличающихся собственных векторов и соответствующих им собственных чисел.

Поэлементное перемножение двух векторов - это скалярное произведение. Скалярное

произведение векторов - это проекция одного на другой. При умножении на новый базис - мы получаем проекцию на новый базис. Старый базис может быть обязательно ортогонален. Если базис не ортогонален (одна компонента является линейной комбинацией другой), то компоненты коррелируют, а следовательно такое описание избыточно. Преобразование Каруна-Лоэва позволяет сделать оптимальное сжатие.

Собственное число - аналог дисперсии. Оно характеризует, насколько вдоль данного базиса меняются анализируемые вектора. Мы не рассматриваем элементы базиса, соответствующие маленьким собственным числам. Если мы кодировали изображения вектором длины n , то ковариационная матрица имела n собственных векторов, и после перевода в новый базис мы представили каждый вектор векторов длина n . Не все компоненты нам нужны, следовательно не сильно меняющиеся компоненты нового пространства можно отрезать и на этом можно основать новый способ сжатия, позволяющий не использовать с учетом статистики изображения некоторые компоненты (которые не изменяются).

Преобразование Каруна-Лоэва - удобный инструмент манипулирования с базисом. С помощью него мы можем более "умно" считать расстояния в векторном пространстве. Наиболее распространенной метрикой является евклидово расстояние:

$$s = \sqrt{\sum_i^N (x_i - y_i)^2}$$

Манхэттэнская метрика:

$$S = \sum_i^N (x_i - y_i)$$

и т.д. Можно использовать метрику, учитывающие статистические свойства пространства. Например, расстояние Махалонобиса:

$$s(\vec{x}, \vec{c}) = (\vec{x} - \vec{c}) B^{-1} (\vec{x} - \vec{c})^T$$

где B - ковариационная матрица.

Лекция 8

Рассмотрим частный случай метода ожидания максимизации, описанного на прошлом занятии. Это метод кластеризации k-средних.

Алгоритм:

1. $ic=0$ обнулили счетчик итераций
2. $C_j(ic)=c_j$ инициализируем набор центров. Например равномерно распределим их по пространству признаков.
3. Выполняем классификацию всех векторов признаков, которые имеются. Относим из к j-ому классу. $x_i(ic) \rightarrow w_j$ если $dist(x_i, c_j) = \min_j$ по всем кластерам. $dist$ - это расстояние в пространстве признаков (например, евклидово).
4. Корректируем центры кластеров, назначенных на нулевой итерации. Назначаем центры кластеров на новой позиции, которые являются математическим ожиданием (средним) всех векторов, принадлежащих кластеру. $C_j(ic) = E(x_i \in w_j)$.
5. Сравниваем положения кластеров на текущей итерации и их положения на предыдущей итерации. Если практически не изменилось - останавливаемся. $\|C_j(ic) - C_j(ic-1)\| < T \forall j$; где T - заданный порог.

Как определить количество центров? Можно адаптивно добавлять и удалять кластеры. Для ковариационной матрицы векторов кластера посчитаем собственные числа и собственные вектора. Если собственное число окажется слишком большим, то это означает, что в каком-то направлении в базисе собственных векторов кластер будет очень сильно вытянутым. В этом направлении (вытянутости) кластер можно поделить на 2 центра и повторить процедуру кластеризации и измерения разойдутся к этим двум центрам. Процедуру можно повторять. Если какие-то кластеры достаточно маленькие, а расстояние между ними мало (расстояние между ними сравнимо с размерами векторов в кластерах), то вероятно, они представляют один кластер.

Чем плохи методы кластеризации по гистограммам? Вследствие того, что на картинке имеется множество промежуточных вариантов (слишком много информации в модах гистограммы усредняется) и статистическое их разделение становится невозможным. Поэтому пытаются сделать итеративную кластеризацию. Внутри связанных объектов можно использовать гистограммную кластеризацию. Этот метод - итеративная гистограммная кластеризация Аландера.

Иногда используют так называемое графовое разбиение Шин. Имеем совокупность векторов измерений (совокупность отсчетов). Представим эту совокупность как вершины графа (v_i, v_j, v_k) и связываем их ребрами графа (E_{ij}, E_{ik}, E_{jk}) . Граф представляется как совокупность вершин и ребер $G=(V, E)$. Каждому ребру приписывается вес, пропорциональный сходству двух измерений, которое это ребро соединяет. Далее этот граф разбивают на 2 несвязанных подмножества путем разрывания связей с минимальными весами. Получают 2 подмножества (2 кластера). Эта операция называется " $cut(A, B)$ ":

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \rightarrow \min$$

Для того, чтобы кластеры получились не слишком маленькие сумму нормируют:

$$Ncut(A, B) = \frac{cut(A, B)}{\sum_{v_i \in A} v_i} + \frac{cut(A, B)}{\sum_{u_i \in B} u_i}$$

Графы, соответствующие измерениям в пространстве признаков являются слишком громоздкими, поэтому на практике данный метод не используется.

Экспертные системы не универсальны. Если мы хотим построить универсальную систему распознавания образов, то мы должны заложить в нашу систему минимальное

количество общих правил:

1. Проецируемые изображений подвергаются проективным преобразованиям.
2. Информация о расположении структурных элементов (можно получить инвариантно к вращению, сдвигу и масштабированию).
3. Сопоставление должно быть взаимно-однозначным. Если объекты прозрачны мы одновременно видим то что находится на передней и задней поверхности и мы получим неоднозначное сопоставление структурных элементов, поэтому оговорено, что случай в природе встречается не часто, а многие виды не адаптированы.
4. Наименьшим изменениям подвергаются границы объектов.
5. Правило загораживания (исходя из того, что объекты непрозрачны).
6. Наблюдаемый мир неоднородный (если бы он был однородны - нечего было бы распознавать - поле со случайными вариациями яркости). Мир организован иерархически и мы эксплуатируем это при интерпретации изображений. Наблюдаемая сцена делится на объекты, объекты делятся на подобъекты в зависимости от фиксации зоны внимания и т.д.

Вернемся к распознаванию образов в пространстве признаков.

Для хорошего распознавания необходимо сделать кластеры компактными и разнесенными (для этого в лабораторных работах рассматривается внутриобластная и межкластерная дисперсия).

Внутрекластерная дисперсия

$$\bar{D}^2 = \sum_i^N (\sigma^2) \rightarrow \min$$

Собственные числа ковариационной матрицы говорят нам о направлении наибольшего удлинения кластера. Когда мы перешли в ортогональное пространство новый базис в пространстве признаков характеризуется ортогональностью, а значит компоненты признаков, отложенные в этом базисе взаимно независимы, а это значит что ковариационные матрицы будут диагональными.

$$\begin{matrix} \vec{X} \rightarrow \vec{X}^* \\ \vec{w}^T \cdot \vec{X}^* \end{matrix}$$

Получается, что мы трансформируем новое пространство пращников - сжимаем его в тех направлениях, в которых кластеры имеют наибольшее удлинение (делаем кластеры более компактными). Это формально обоснованный процесс. Во многих учебниках (например в учебнике Туи-Гонсалеса 70х годов) можно встретить следующий способ выбора диагональных коэффициентов w_{kk}

$$w_{kk} = \frac{1}{\sigma_k^2 \sum_{j=1}^N (1/\sigma_j^2)}$$

$$w_{kk} = \frac{1}{\sigma_k} \left(\prod_{j=1}^N \sigma_j \right)^{\frac{1}{N}}$$

Можно строить максимумы весовых коэффициентов, такую, чтобы достигался минимум энтропии. Можно показать, что если мы выбираем диагональную матрицу весовых коэффициентов обратных собственным числам, то энтропия так же будет минимизироваться.

Рассмотрим методы максимизации междкластерных растираний. Общий подход: различающая способность кластера i относительно класса j :

$$u_{ij} = \ln \frac{P(\vec{X} | w_i)}{P(\vec{X} | w_j)}$$

Средняя различающая способность:

$$I(i, j) = \int_x P(\vec{X} | w_i) \ln \frac{P(\vec{X} | w_i)}{P(\vec{X} | w_j)} dx$$

$$I(j, i) = \int_x P(\vec{X} | w_j) \ln \frac{P(\vec{X} | w_j)}{P(\vec{X} | w_i)} dx$$

Полная различающая способность:

$$I_{ij} = I(i, j) + I(j, i) = \int_x [P(\vec{X} | w_i) - P(\vec{X} | w_j)] \ln \frac{P(\vec{X} | w_i)}{P(\vec{X} | w_j)} dx$$

Откуда взять жэти вероятность - большой вопрос.

Лекция 9

При сопоставлении ключевых точек с дескрипторами SIFT сопоставляются изображения, на которых больше всего совпавших ключевых точек.

Bag of words - метод, разработанный для упрощения сопоставления ключевых точек (евклидово расстояние - долго!). Суть метода в предварительной кластеризации имеющихся точек.

Можно использовать информацию о пространственном расположении точек. Они должны быть расположены вблизи эпиполярной линии. Могут быть откинута точки, не удовлетворявшие требованиям геометрической близости двух картинок. Анализ посвященной эпиполярной геометрии.

Для вычисления параметров преобразования необходимо точек:

- Подобия: 2
- Аффинное: 3
- Проективное: 4

Рассмотрим развитие метода SIFT. На его основе были созданы современные мощные методы классификации изображений. Ошибка не превышает 5-10% при тысячах классов.

Современные методы классификации пошли по пути описания всего содержимого картинки поделенного на локальные фрагменты дескрипторами аналогичными SIFT-дескрипторами, учитывавшими яркостные гистограммы. Длина такого вектора признаком может содержать несколько тысяч компонентов. Этот метод получил название Dense-SIFT (плотный SIFT).

Слабость SIFT-дескрипторов - они локальны. Они разрушаются при небольшом изменении условий наблюдения. Дэвид Ловэ почти преодолел этот недостаток путем отбраковки неправильных дескрипторов с использованием информации о взаимном пространственном расположении. Современные методы пытаются сделать SIFT-дескрипторы "не такими локальными". Следующий шаг после Dense-SIFT - построение HOG-дескрипторов (гистограмм ориентаций градиента). Как это работает, если мы ищем на изображении человека? Используем вертикальноориентированное окно, которое делится на небольшое количество клеточек. Изображение в таком скользящем окошке описывается длинным вектором признаком, составленным с помощью конкатенации векторов признаков, описывающих направления градиентов. Когда дескрипторы строятся в перекрывающихся подблоках - результат получается более устойчивым к ракурсным изменением. Масштабные дефекты преодолеваются сканированием окошками разных размеров, а если изменяется ракурс - масштабировании происходит только в конкретных направлениях. Деление на перекрывающиеся подблоки позволяет получить некоторую инвариантность к таким типам изменений. В чем недостаток по сравнению с методом SIFT? От геометрии сканирующего окна зависит качество распознавания. Лежащий пешеход не распознается.

Следующим шагом был метод, основанный на анализе взаимного расположения частей. Используются штрафы за смещения найденной части относительно ее эталонного положения. Есть ограничения - части расположены симметрично относительно вертикальной оси окошка, либо на ней. При обучении классификатор смещает эталонное положение этих частей в среднее положение по результат детектирования частей на изображениях в большой обучающей выборке.

Что делать с "лежащим" человеком. Разработчики предлагают делать распознавание с разных ракурсов (а в этот ракурс мы можем включить эталоны лежащего и стоящего человека). Система автоматически делит совокупность распознаваемых изображений на подмножества отдельных "ракурсов". Т.е. используются окошки разных типов, учитывающие разные эталоны. При обучении этого классификатора в качестве параметров системы задается

количество "ракурсов" с которыми классификатор будет работать и обучающая программа поделить исходную выборку на подмножества изображений, соответствующие разным "ракурсам" (величинам удлинения окошка в разных направлениях). При обучении предъявляются помеченные изображения (image.net etc.), т.е. на картинках уже выделена область, содержащая объект. Обучение такой системы для 40 классов производилось 2 дня на одноядерном процессоре.

Нейронные сети с иерархической секционированной корреляцией и глубоким обучением. Сеть осуществляет свертку элементов изображения с набором фильтров, которым обучена сеть. Разработчики объединили каскады Виолы-Джонса и HOG-Методы. Суть в обучении при помощи каскадов Виолы-Джонса, использующие не вейвлеты Хаара, а HOG-дескрипторы. Метод называется HOG-каскады. Это, вероятно, наиболее новый умный метод.

Рассмотрим методы нахождения границ на изображениях

$$y'(x) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y(\Delta x)}{\Delta x}$$

Наша Δx - в случае изображения.

Фильтры Робертса, Прюитта и Собеля дают толстые контуры. Обработка при помощи Лапласиан-Гауссиана позволяет выделить контуры более точно:

$$\nabla^2 = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$$

Фильтр в форме Мексиканской шляпы. Обычно сумму элементов в масках фильтров делается нулевой, чтобы не было отклика фильтра на местах константной яркости. Этот фильтр можно считать как DOG-фильтр - разность двух гауссовых фильтров. Границы соответствуют точкам, равные 0 (вторая производная в точке максимума).

Фильтры первого и второго порядка описываются соответственно дифференциальными уравнениями первого и второго порядка. Как описать полученное контуры? Один из подходов - нахождение на контурах точек наибольшей кривизны и считаем, что эти точки соответствуют изломам контуров и эти точки можно соединять отрезками прямых линий. Другой вариант: итеративный алгоритм, учитывающий некоторое максимальное отклонение, которое считается для некоторой хорды контура. Если порог превышен, то делаем эту точку - точкой деления контура и проводим 2 хорды, по которым так же определяем максимальное расстояние от точек контура до хорды.

$$ax + by + c = 0$$

- уравнение прямой линии. Если у нас есть концы прямой линии с координатами x_1, y_1, x_2, y_2 , то уравнение прямой линии через эти координаты может быть получено:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

$$a = y_2 - y_1$$

$$b = x_2 - x_1$$

$$c = y_1(x_2 - x_1) - x_1(y_2 - y_1)$$

Если нет непрерывных контуров - можно использовать преобразование Хаффа.

Как выделить траектории, описываемые кривыми второго порядка (кругами и эллипсами)? Нужно применить преобразование Хаффа для кругов и эллипсов.

$$y = y_0 + r \sin(\theta) \rightarrow y_0 = y - r \sin(\theta)$$

$$x = x_0 + r \cos(\theta) \rightarrow x_0 = x - r \cos(\theta)$$

Пространство четырехмерное. С ростом количества параметров размерность пространства признаков растет экспоненциально, поэтому искать кривые высоких порядков вычислительно сложно. Как решать переопределенные системы уравнений? Используется метод минимизации среднеквадратичной ошибки.

Пусть есть совокупность точек $\{x_i, y_i\}$
 $y' = f(x)$

Среднеквадратичная ошибка описывается уравнением:

$$LSE = \sum_{i=1}^N (f(x_i) - y_i)^2 \rightarrow \min$$

Приравняв производную этой функции нулю находим параметры в уравнении $c_1 x + c_0 = y$ нашей истинной прямой линии. Для этого ищем производные по этим параметрам, получаем систему уравнений, решая которую найдем истинные значения параметров.

Формальные грамматики

В формальной лингвистике алфавит - конечный набор символов. Предложение - цепочка конечной длины символов этого алфавита. Бывают пустые, не пустые. Язык - произвольное множество предложение в некотором алфавите. Грамматикой называется совокупность нетерминальных символов, терминальных символов, правила порождения и корневой символ. Например попытка формализовать простые предложения английского языка:

$$G = (V_N, V_T, P, S)$$

- Предложение the guy runs
 - Именная составляющая - нетерминальная the guy
 - Артикль the
 - Именная составляющая guy
 - Глагольная составляющая - нетерминальная
 - Глагольная составляющая runs

Далее предложение приобретает формальный вид

$$P = \{S \rightarrow aSb, S \rightarrow a\}$$

Терминальные символы определяют правила подстановки. Пример описания изображений при помощи формальных грамматик:

Квадрат:

Над (-, над (слева (|, |), над -))

Нетерминальный элемент над (слева (|, |), над -))

Проблема такого подхода в том, что мы пишем не только квадрат, но и любую другую совокупность - и | удовлетворяющих условиям. Туй и Гонсалес предложили описывать хромосомы алфавитом из элементов контура.

Лекция 10

Аппроксимация контура прямыми линиями

Что мы упустили в прошлый раз? Как оценить наилучшую прямую имея набор точек? По методу МНК, но вопрос какие взять оси. Можно грубо оценить наклон прямой и если больше 45 градусов то использовать расстояние по x , а если меньше, то стандартное по y , однако есть способ лучше:

1. оцениваем эллипс распределения точек;
2. методом главных компонент оцениваем направления главных осей этого эллипса;
3. зная направление второй оси мы будем знать в направлении, в котором нужно считать расстояние в МНК.

Варианты сегментации: по текстуре и по яркости.

- Яркость - скалярная величина, текстура - векторная.
- Яркость оценивается по одному пикселю, текстура - по нескольким окружающим пикселям.

Марковский процесс первого порядка – случайный процесс, каждое следующее состояние в котором зависит от своего состояния на предыдущем шаге. Двумерный марковский процесс - случайное поле, набор вариаций.

Как выбрать размер окошка, в котором считаются статистические характеристики, описывающие текстуры? Как компромисс между большим размером окна для выделения представительной статистики и малым размером, для разрешения деталей.

Методы текстурного анализа

Примитивные методы в малой степени учитывают изменения элементов текстуры по яркости или вдоль координатных осей. Умные описатели учитывают статистику вариации яркости и изменение формы текселов в различных направлениях, что позволяет восстановить наклоны рассматриваемых поверхностей.

Для описания статистических свойств текстур используются моменты случайно величины различного порядка. Моменты есть обычные, центральные и т.д. Вариации яркости без учета координат описываются моментами n -ого порядка такого вида:

$$\mu_n(z) = \sum_{l=0}^{L-1} (z_l - m)^n p(z_l) ,$$

где z - яркость, m - математическое ожидание, L - количество уровней яркости, $p(z_l)$ - вероятность, которая может быть посчитана по гистограмме, $\mu_0=1$, $\mu_1=0$.

Дисперсия - момент второго порядка. Вместо дисперсии иногда используют описатель вида

$$R = 1 - \frac{1}{1 + \sigma^2(z)}$$

- нормированная дисперсия.

$$\sigma_{\text{норм}}^2(z) = \frac{\sigma^2}{(L-1)^2} \quad \text{- нормировка к количеству уровней яркости.}$$

Третий момент характеризует асимметрию гистограмм яркостей. Четвертый момент характеризует остроту функции распределения яркостей. Для более высоких моментов таких характеристик не придумали.

Из характеристик связанных с вероятностью можно увидеть вычислить энтропию, характеризующую меру беспорядочности распределения:

$$e = - \sum_{i=0}^{L-1} p(z_i) \cdot \log p(z_i)$$

$$u = \sum_{i=0}^{L-1} p^2(z_i)$$

Это были примитивные заимствованные из статистической радиотехники методы. Период и фаза синусоиды всегда содержит информацию о координатах, поэтому статистика синусоид - более умный подход. Для этого рассматривают пространственные Фурье-спектры изображений.

Иногда рассматривают изображения в полярных координатах. Метод Волда разделяют периодические и непериодические составляющие и периодические и хаотические части спектра анализируются отдельно. Есть искажения изображений, которые нельзя компенсировать не в спектральной области.

Способ описания текстуры - форма ее автокорреляционной функции:

$$C(\Delta x, \Delta y) = \frac{\sum_x \sum_y I(x, y) \cdot I(x + \Delta x, y + \Delta y)}{\sum_x \sum_y I^2(x, y)}$$

Автокорреляция характеризует грубость текстуры (преобладание низких или высоких частот) и ее периодичность. Автокорреляционная функция - аналог обратного преобразования Фурье от квадратов амплитуд спектральных.

Полная система признаков Харалика - 28 компонентов. Это избыточно, нам было достаточно 10. Какие есть глобальные недостатки у метода Харалика?

1. Высокая вычислительная сложность (из-за этого приходится уменьшать число градаций яркости).
2. Требуется большой размер окошка (25-30 пикселей). Мелкие детали не разобрать.

Следующая система признаков Multi resolution cell autoregression (MRSAR).

Центральный пиксель окошечка вычисляется как авторегрессия восьми ряд расположенных. В четырех направлениях происходит аппроксимация по соседям второго, третьего и более порядков. Таким образом получают вектор признаков, описывающий текстуру в центральном пикселе через его соседей. Этот вектор характеризует грубость текстуры, коррелированность текстуры и другие свойства. Этот метод работает немного лучше, чем метод Харалика в основном для случайных яркостных полей. Этот метод требует меньшего окошка для усреднения, чем метод Харалика.

В Государственном оптическом институте Тетериным и Павловой был разработан метод текстурной сегментации The "sequents" method, основанный на анализе последовательностей центрированной (с вычтенным средним значением) яркости на изображении. Метод строит признаки, инвариантные к вращению картинки.

Методы описания текстуры по статистике границ. Тех кто описывает текстуры интересуют не сами тексели а границы элементов текстуры. Изображение внутри сканирующего окошечка делятся на клеточки и в каждой клеточке определяется доминирующее направление выделенной границы. Напоминает Dense SIFT. Дополнительно используется информация об интенсивности градиента. Изображение внутри окошечка характеризуется гистограммой направлений и интенсивностей градиента. Таким образом, получают вектор признаков текстуры, как вектор признаков гистограмм.

Так же текстуру можно описывать набором коэффициентов вейвлет-разложения (например с использованием вейвлетов Габора). Искусственные регулярные текстуры типа платья в горошек могут быть описаны регулярными грамматиками.

Лекция 11

Аффинный SIFT

Дифференцирующие фильтры: Превитт, Собель, Робертс, Моменты Майстры и Хо инвариантны к зеркальному отражению (однако это нечастное преобразование и им пренебрегают).

Наш SIFT позволяет компенсировать преобразование масштабирования, вращения и сдвига, однако не умеет хорошо компенсировать ракурсные преобразования (компенсирует только частично при помощи постарения векторов градиента в мелкой сетке в локальной окрестности).

Что делать, чтобы хорошо компенсировать ракурсные преобразования при помощи SIFT??? В аффинном SIFT выполняется перебор всех вариантов расположения камеры (с определенным шагом) в полусфере, расположенной перед наблюдаемой сценой. Для каждого такого положения происходит трансформация эталонных дескрипторов (трансформация изображений, по которым строятся дескрипторы) для данного ракурса и сравнение этих дескрипторов с дескрипторами, сформированными видеокамерой, которое рассматривает изображение. Такое преобразование позволяет компенсировать ракурсные преобразования (таким образом SIFT умеет все, кроме проективных преобразований и зеркального отражения).

В связи с тем, что в SIFT не закладывается модель геометрических преобразований в виду сложности дескрипторов, SIFT позволяет работать с трехмерными сценами, в отличие от метода универсальных структурных элементов. Аффинный SIFT не работает на неровных объемных поверхностях. В определенных статьях SIFT применяют для сопоставления аэрокосмических снимков, но это не работает, когда изображения подвержены сильным изменениям.

Методы улучшения изображения/Image enhanced techniques

Look up table используется для преобразования яркости пикселей. При помощи такой таблицы можно преобразовывать яркости самым замысловатым способом. (По сути это табличное представление дискретной функции). *Gamma*-преобразование (Гамма-коррекция). Наклон передаточной функции дает контраст в определенных участках. При бесконечном наклоне можно получить бинаризацию.

Эквализация гистограмм делает величину яркостей равновероятной по всему диапазону яркостей (это не растягивание гистограммы!!!)

$r \in [0, 1]$ - первоначальный диапазон яркостей

Мы хотим придумать функцию, которая выполнит эквализацию гистограммы.

$$S = T(r)$$

$S \in [0, 1]$ (мы хотим что диапазон остался таким же)

Эта функция должна быть монотонна (без отрицательных наклонов). Обратное преобразование будет записываться как

$$r = T^{-1}(S)$$

(если функция будет не монотонна - обратное преобразование будет неоднозначно).

$$p_s(S) = p_r(r) \left| \frac{dr}{dS} \right|$$

Для выполнения эквализации предлагают использовать следующую функцию преобразования:

$$S = T(r) = \int_0^r p_r(w) dw$$

$$\frac{dS}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr} \left[\int_0^r p_r(w) dw \right] = p_r(r)$$

Отсюда:

$$p_s = p_r(r) \left| \frac{dr}{ds} \right| = p_r(r) \left| \frac{1}{p_r(r)} \right| = 1,$$

для любого $S \in [0, 1]$.

Импульсный шум удаляется при помощи медианного фильтра. Для подчеркивания высоких частот складывают результат дифференцирования изображения с исходным изображением.

Фильтры в частотной области

Гомоморфный фильтр

Почему не сделать идеальный фильтр (с резким обрезанием границ)?

1. Спектр такого фильтра - sinc-функция и из-за обрезания sinc мы получим эффект Гиббса на границах.
2. Железка, которая выполняет аналоговую фильтрацию или компьютерную модель сложно сделать, ибо необходимо использовать дифференциальное уравнение очень большого порядка.

Реально применяемые фильтры: фильтр Баттерворта, фильтр Гаусса, фильтры Чебышева. Фильтры: высокочастотные, низкочастотные, полосовые и режекторные. Если известна спектральная полоса шума - всегда можно поставить полосовой фильтр на эту полосу и удалить этот шум.

Если у нас расфокусированное или смазано изображение:

Фильтрация Винера

$$g(x, y) = f(x, y) * h(x, y)$$

- яркость искаженной картинки - свертка исходной картинки с некоторой функцией

Чтобы сделать противосвертку нужно знать прямой фильтр. Фильтрация в дискретном случае может быть описана как матричная операция, а матрица не всегда имеет обратную матрицу.

$$G(w_x, w_y) = H(w_x, w_y) \cdot F(w_x, w_y)$$

Оценка спектра восстановленной картинки:

$$\hat{F}(w_x, w_y) = \frac{H(w_x, w_y) \cdot G(w_x, w_y)}{H^{-1}(w_x, w_y)}$$

Все было бы прекрасно, но есть еще дополнительный шум:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

$$G(w_x, w_y) = H(w_x, w_y) \cdot F(w_x, w_y) + N(w_x, w_y).$$

$$\hat{F}(w_x, w_y) = \frac{H(w_x, w_y) \cdot F(w_x, w_y) + N(w_x, w_y)}{H^{-1}(w_x, w_y)}$$

и с добавлением шума метод перестает работать. В более простом случае эта проблема решается фильтрацией Винера, а в плохом случае используется регуляризация Тихонова. Параметры регуляризации сложно подобрать, а фильтрация Винера работает хорошо. Фильтр Винера основан на минимизации среднеквадратичной ошибки и записывается следующим образом:

$$\hat{F}(w_x, w_y) = \left[\frac{H^*(w_x, w_y)S(w_x, w_y)}{S_f(w_x, w_y)|H(w_x, w_y)|^2} + S_n(w_x, w_y) \right] G(w_x, w_y) =$$

$$= \left[\frac{1}{\frac{H(w_x, w_y)}{|H(w_x, w_y)|^2} + S_n(w_x, w_y)/S_f(w_x, w_y)} \right]^{2G} (w_x, w_y) \cdot$$

Остается вопрос - откуда взять спектр шума. По этой причине здесь вводят настраиваемый параметр для отношения $S_n(w_x, w_y)/S_f(w_x, w_y)$ и подбором этого параметра получают хорошее решение и метод получается не так чувствителен к изменяем параметров.

Есть так называемый спекл-шум, который возникает при когерентном освещении и геометрическими фильтрами получается убрать этот шум везде, кроме границ (там где была статистически однородная яркость - работает хорошо). Так же используют морфологическую обработку изображений (эрозия, дилатация и т.д.). Еще бывает дисторсия (подушка-образная или точка-образная)

$$\vec{R} = b \cdot \vec{r}$$

Если присутствует дисторсия третьего порядка, то радиус вектор:

$$\vec{R} = b \vec{r} + F_3 r^2 \vec{r}$$

Они бывают только нечетных порядков! Для высших порядков:

$$\vec{R} = b \vec{r} + F_3 r^2 \vec{r} + F_5 r^4 \vec{r} \dots$$

Дисторсия возникает в дешевых объективах и компенсируется при помощи обратного проецирования методом Кордано. Оно нужно для определения размера картинке со скомпенсированной дисторсией. Если краями картинке немного пожертвовать, то даже не нужно будет решать уравнения третьей степени.

Есть еще хроматические aberrации, возникающие из-за различного преломления волн с различными частотами. Получается, что границе светлого и черного мы видим радугу (продольная aberrация). Для того, чтобы от него избавиться делают оптические элементы, состоящие из разных материалов. Почти удастся такую aberrацию компенсировать.

При данной фокусировке объектива мы имеем определенную резкость и она хорошая только по-середине. На остальных участках фокусировка менее резкая. Авторы метода предложили взять все три картинки (красную, синюю и зеленую, которые по разному сфокусированы). Человеческий глаз не очень чувствителен к дефектам цветовой компоненты (за счет этого jpeg замечательно работает). Возьмем пространственные частоты спектров и перенесем верхние частоты хорошо сфокусированного канала перенесем в другие каналы.

Современные фотоаппараты позволяют сделать серию снимков с разной экспозицией для коррекции яркости при съемке против света.

Рассмотрим прикладные задачи: каким образом решаются проблемы дополненной реальности? Технологию анализа жестов для решения задач управления оборудованием. Интерактивные доски и т.д.

В пространстве rgb цвет человеческой кожи расположен на "сосиске" вдоль планковской кривой, соответствующие температуре освещения сцены. Границы этой кривой хорошо аппроксимируются параболой. В зависимости от кровообращения кожи и количества меланина цвет будет в разных местах этой "сосиски". Область поиска можно ограничить этой зоной.

Метод оказывался чувствительным к цвету освещения. Если освещение неравномерное наиболее интенсивным является признак цвета в пространстве HSV. По компоненте H кожа хорошо отличается от фона. По компоненте S - хуже. Проблема в том, что блики кожи не учитываются. Так же используется нормализация яркости.

Обратное проецирование гистограммы - метод в распознавании цвета. В результате

получают аналог вероятностной оценки.

Распознавание простых жестов может быть сделано по выделенным прямым линиям в контуре ладони, однако сложные жесты так не распознать, т.к. пальцы начинают друг друга загромождать, отбрасывать дополнительные тени и т.д. Для распознавания сложных жестов применяется механическая модель руки. Всего в ладони 28 подвижных костей, поэтому полный перебор всех положений медленен. Для ускорения процесса - ладонь распознается в эталонном положении, а потом покадрово анализирует возможные области затенения и возможные взаимные положения пальцев, в которые можно прийти из распознанного положения.

Применение распознавания жестов. Под Linux есть разработанная программа, которая умеет моделировать механические движения пальцев руки.

Где используются беспилотные аппараты?

- Военная промышленность
 - Роботы в виде плюшевых мишек для спасения раненых на поле боя
 - Беспилотные летательные аппараты
 - Крылатые ракеты
 - Наземные роботы
 - Под- и надводные роботы
- Индустрия развлечений и бытовая техника

Чамферное преобразование широко используется в военной технике для контурного сопоставления.