Detección de caras

Lección 11.2

Dr. Pablo Alvarado Moya

MP6127 Visión por Computadora
Programa de Maestría en Electrónica
Énfasis en Procesamiento Digital de Señales
Escuela de Ingeniería Electrónica
Tecnológico de Costa Rica

I Cuatrimestre 2013

Contenido

Detección

- Tarea: encontrar objetos de una categoría en imagen
- Paso previo a reconocimiento
- Ampliamente utilizada: detección de **caras humanas**

Detección de caras

- Gran variedad de métodos propuestos
- Reto: debe ser ¡muy rápido!
- Tres tipos de técnicas
  1. basadas en características
  2. basadas en plantillas
  3. basadas en apariencia

Detección de caras

- Gran variedad de métodos propuestos
- Reto: debe ser ¡muy rápido!
- Tres tipos de técnicas
  1. basadas en características
     buscan patrones característicos como ojos, nariz y boca, y luego verifican plausibilidad geométrica.
  2. basadas en plantillas
  3. basadas en apariencia

Detección de caras

- Gran variedad de métodos propuestos
- Reto: debe ser ¡muy rápido!
- Tres tipos de técnicas
  1. basadas en características
  2. basadas en plantillas
     ajustan plantilla (AAM, ASM) que soporta variabilidad, pero requiere buena inicialización
  3. basadas en apariencia

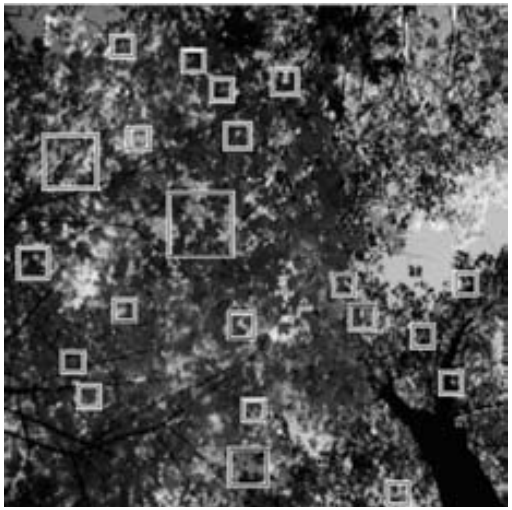
Detección de caras

- Gran variedad de métodos propuestos
- Reto: debe ser ¡muy rápido!
- Tres tipos de técnicas
  1. basadas en características
  2. basadas en plantillas
  3. basadas en apariencia

  barren imagen buscando candidatos factibles, proceso que se refina jerárquicamente con clasificadores cada vez más selectivos, pero más caros computacionalmente

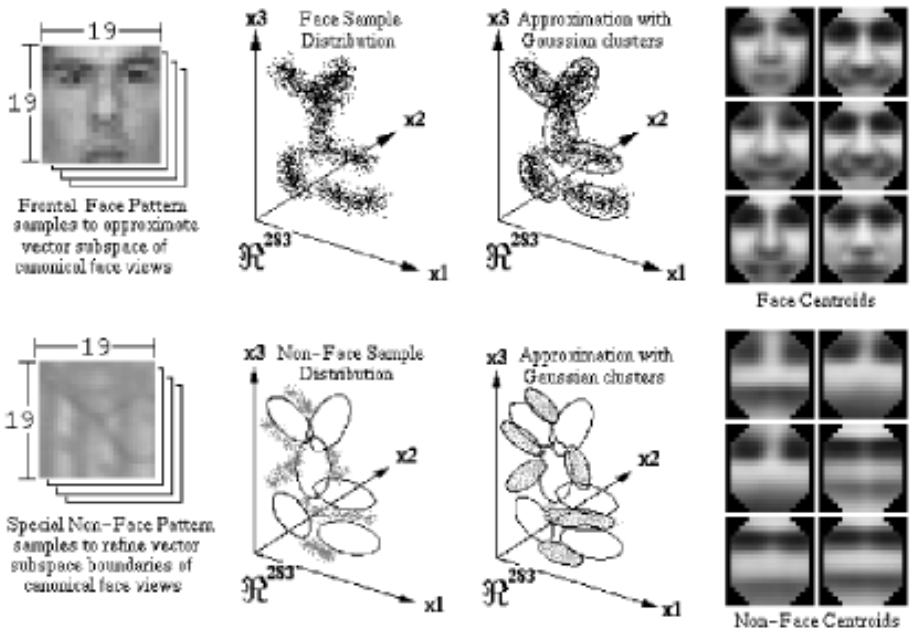
Detección basada en apariencia

- Uso de clasificadores implica cuerpos de entrenamiento de caras y no-caras
- Sintéticamente se rotan, reflejan, y desplazan datos
- Preprocesamiento necesario (iluminación, contraste, ruido . . .)
- Por ejemplo, Rowley et al. restan mejor ajuste lineal y luego ecualizan histograma



Rowley, Baluja y Kanade, 1998

- Maldición de la dimensionalidad exige reducir dimensión de datos
- Por ejemplo, Sung y Poggio utilizan aglomeración con k-medias y ACP en cada conglomerado
- Usualmente "parches" de caras de 19×19 hasta 30×30

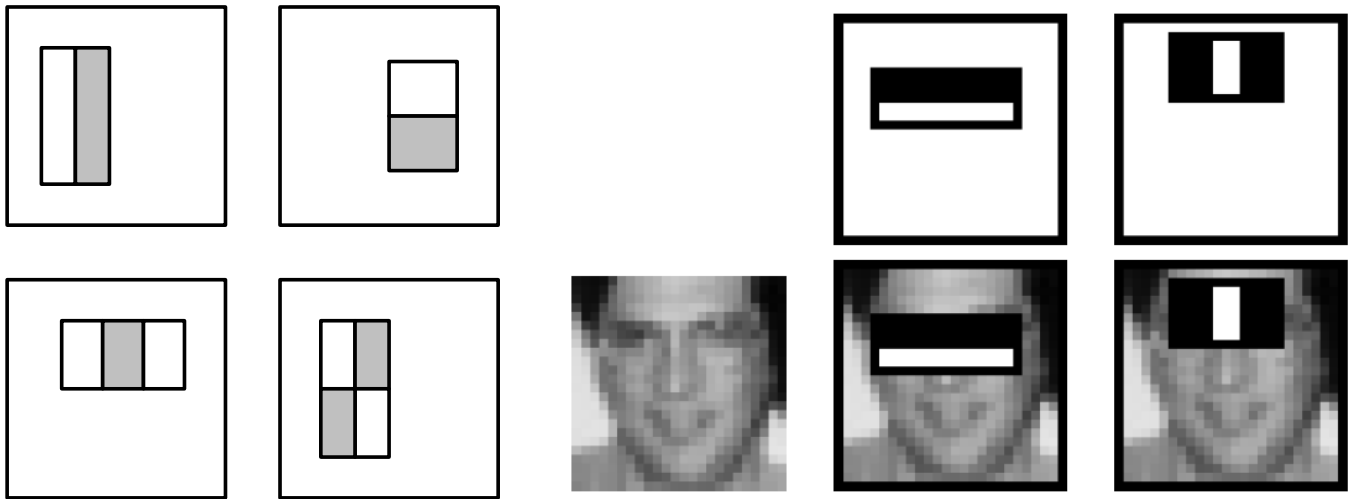


Sung y Poggio, 1998

- 6 conglomerados para cara y 6 para no-cara
- Distancia de patrón en estudio a cada conglomerado forma descriptor
- Algunos autores usan redes como MLP, clasificadores estadísticos como SVM, boosting, etc.

Detector de Viola y Jones

- Método de Viola y Jones (IJCV, 52(2), 2004) es el más usado
- Método es rápido y robusto
- Utiliza imágenes integrales (usadas también en SURF) para calcular rápidamente características rectangulares de 2, 3 ó 4 rectángulos:



- Entrenamiento busca cuáles características discriminan
- Se usa *boosting*

Boosting

- **Boosting** implica combinar varios clasificadores débiles en uno más fuerte:

$$h(\underline{\mathbf{x}}) = \text{signum}\left[\sum_{j=0}^{m-1} \alpha_j h_j(\underline{\mathbf{x}})\right]$$

- Viola y Jones usan AdaBoosting (adaptive boosting)
- Los valores de los descriptores provienen de la umbralización de las características f_j (rectángulos)

$$h_j(\underline{\mathbf{x}}) = a_j[f_j < \theta_j] + b_j[f_j \geq \theta_j] = \begin{cases} a_j & \text{si } f_j < \theta_j \\ b_j & \textit{resto} \end{cases}$$

- Factores a_j y b_j se elijen simétricos, usualmente $a_j = -s_j$ y $b_j = s_j$ con $s_j = \pm 1$.

Boosting

- **Boosting** implica combinar varios clasificadores débiles en uno más fuerte:

$$h(\underline{\mathbf{x}}) = \text{signum}\left[\sum_{j=0}^{m-1} \alpha_j h_j(\underline{\mathbf{x}})\right]$$

  Clasificador débil es aquel solo un poco mejor que un clasificador aleatorio
- Viola y Jones usan AdaBoosting (adaptive boosting)
- Los valores de los descriptores provienen de la umbralización de las características f_j (rectángulos)

$$h_j(\underline{\mathbf{x}}) = a_j[f_j < \theta_j] + b_j[f_j \geq \theta_j] = \begin{cases} a_j & \text{si } f_j < \theta_j \\ b_j & \textit{resto} \end{cases}$$

- Factores a_j y b_j se elijen simétricos, usualmente $a_j = -s_j$ y $b_j = s_j$ con $s_j = \pm 1$.

Boosting

- **Boosting** implica combinar varios clasificadores débiles en uno más fuerte:

$$h(\underline{\mathbf{x}}) = \text{signum}\left[\sum_{j=0}^{m-1} \alpha_j h_j(\underline{\mathbf{x}})\right]$$

- Viola y Jones usan AdaBoosting (adaptive boosting)
- Los valores de los descriptores provienen de la umbralización de las características f_j (rectángulos)

$$h_j(\underline{\mathbf{x}}) = a_j[f_j < \theta_j] + b_j[f_j \geq \theta_j] = \begin{cases} a_j & \text{si } f_j < \theta_j \\ b_j & \textit{resto} \end{cases}$$

- Factores a_j y b_j se elijen simétricos, usualmente $a_j = -s_j$ y $b_j = s_j$ con $s_j = \pm 1$.

1. Entrada: N patrones positivos y negativos de entrenamiento: $\{(\underline{\mathbf{x}}_i, y_i)\}$, con $y_i = 1$ cara (positivos) e $y_i = -1$ no-cara (negativos)
2. Inicialice pesos para datos de primer clasificador débil $w_{i,1} \leftarrow \frac{1}{N}$. (Viola y Jones usan N_1 para datos positivos y N_2 para negativos)
3. Para cada fase de entrenamiento $j = 1 \ldots M$
    1. Renormalice los pesos para que sumen 1
    2. Seleccione el clasificador $h_j(\underline{\mathbf{x}}; f_j, \theta_j, s_j)$ que minimiza el error de clasificación ponderado:

$$e_j = \sum_{i=0}^{N-1} w_{i,j} e_{i,j}$$

$$e_{i,j} = 1 - \delta(y_i, h_j(\underline{\mathbf{x}}_i; f_j, \theta_j, s_j))$$

3. Calcule tasa de error modificada β_j y ponderación del clasificador α_j

$$\beta_j = \frac{e_j}{1 - e_j} \qquad \text{y} \qquad \alpha_j = -\log \beta_j$$

4. Actualice los pesos de acuerdo a los errores de clasificación $e_{i,j}$

$$w_{i,j+1} \leftarrow w_{i,j} \beta_j^{1-e_{i,j}}$$

(baja peso para patrones correctamente clasificados)

4. El clasificador final resulta de

$$h(\underline{\mathbf{x}}) = \text{signum} \left[\sum_{j=0}^{m-1} \alpha_j h_j(\underline{\mathbf{x}}) \right]$$

AdaBoost

- Clasificadores débiles se entrenan secuencialmente
- Entrenamiento es proceso costoso computacionalmente
- Uso del resultado es eficiente
- Tasas de 15 cuadros por segundo posibles con resolución VGA
- Puede aumentarse confiabilidad de detección con otras pistas (color, posición, etc.)

Otros tipos de detecciones

- Detección de peatones
- Detección de vehículos

Enlaces

- Tom Neumark presents on Facial Detection
- Michal Hruby: detección con OpenCL (GPU)

Resumen

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, LTI-Lib-2, GNU-Make, Kazam, Xournal y Subversion en GNU/Linux

Este trabajo se encuentra bajo una Licencia Creative Commons Atribución-NoComercial-LicenciarIgual 3.0 Unported. Para ver una copia de esta Licencia, visite http://creativecommons.org/licenses/by-nc-sa/3.0/ o envíe una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

© 2013 Pablo Alvarado-Moya Escuela de Ingeniería Electrónica Instituto Tecnológico de Costa Rica