Image Retrieval

Enver Sangineto

Dipartimento di Informatica e Automazione

Università degli Studi di Roma Tre

Cenni di Visual Retrieval
per il Corso di Seminari di Sistemi Informatici,
Università degli Studi di Roma Tre,
18 Dicembre 2005 — Roma.

Cos'è l'*Image Retrieval* (o *Visual Retrieval*)

Data la crescita di informazione visiva, siamo interessati a sistemi che recuperino immagini in base al loro contenuto percettivo.

Una prima soluzione consiste nell'affidarsi ai sistemi di retrieval testuale, confidando nelle informazioni testuali associate alle immagini.

Tali sistemi sono poco affidabili e richiedono un'indicizzazione manuale.

Nei sistemi di visual retrieval, il recupero di immagini, invece, si basa esclusivamente (o prevalentemente) sull’informazione visiva.

Visiva è la query dell’utente.

Visivo è il ”ragionamento”, ovvero i criteri di similitudine sui quali si basa il recupero e il ranking, o l’eventuale indicizzazione dell’archivio.

Tipi di retrieval

Una prima importante suddivisione dei sistemi di visual retrieval, riguarda il recupero di immagini fisse o di video.

Nel primo caso la query fornisce un esempio di cosa si cerca.

Nel secondo caso accade più spesso di incontrare sistemi semi-automatici che analizzano le sequenze video per estrarne informazioni utili al retrieval.

Gli elementi percettivi di base

Il recupero di immagini tramite il loro contenuto visivo (senza l'utilizzo di informazioni testuali) necessita del riconoscimento degli elementi percettivi indicati dall'utente, quali:

- il colore,
- la texture,
- la forma,
- le relazioni spaziali delle figure nell'immagine,
- la traiettoria

Image Retrieval tramite Forma

Un tipico sistema di recupero di immagini tramite somiglianza di forma presuppone:

- Una *query by sketch*: l'utente disegna una sagoma approssimativa di ciò che sta cercando nel database di immagini.
- La ricerca nel database avviene quindi tramite tecniche di *indicizzazione* e di *riconoscimento* provenienti dalla *Computer Vision*.
- L'output del sistema non si limita a dire cosa è stato o non è stato riconosciuto, ma deve fornire una lista di immagini ordinate per grado di somiglianza con la query.

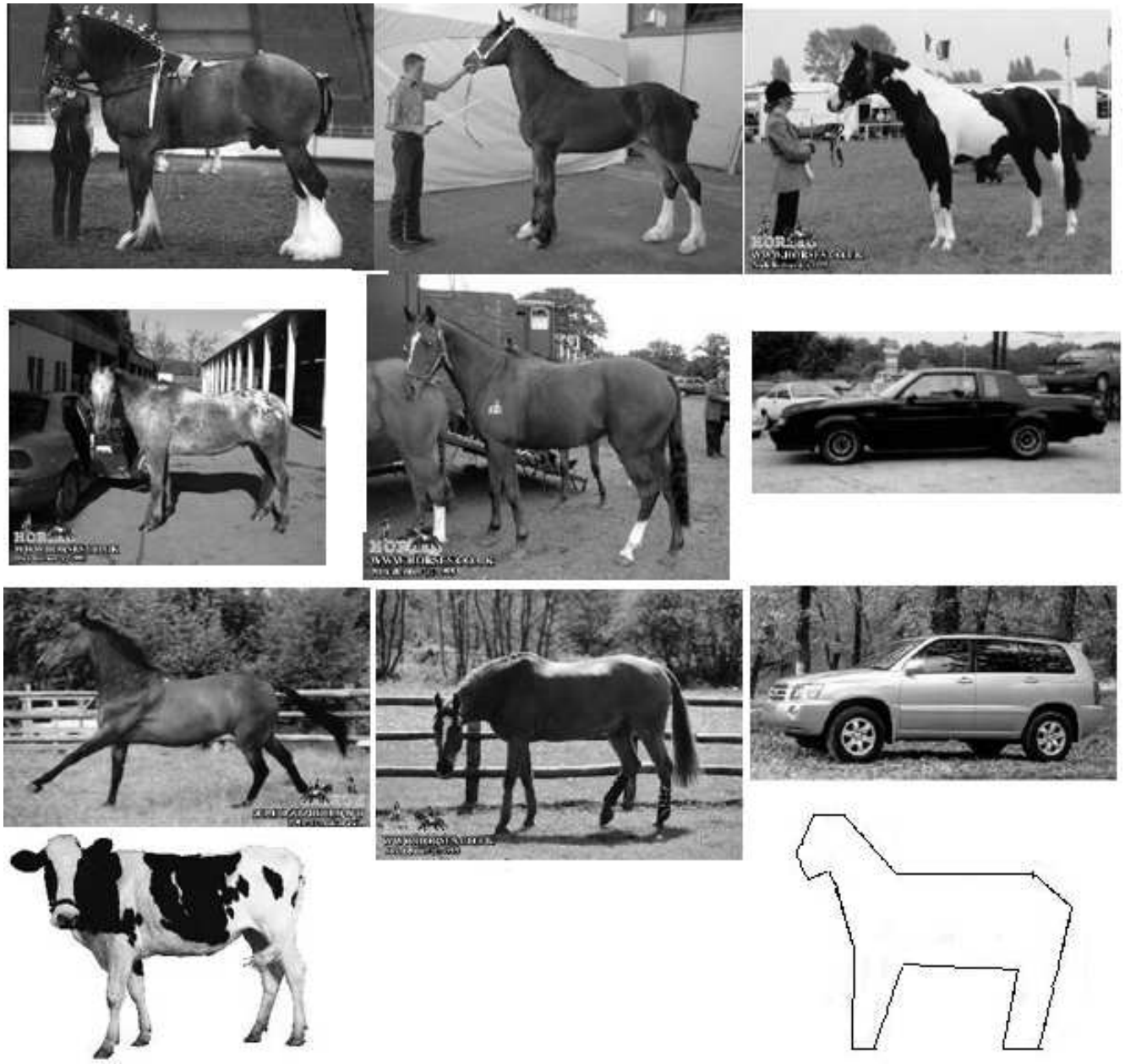

Figura 1:

Tecniche di riconoscimento peculiari dell'Image Retrieval

I due filoni principali sono:

- Approccio *statistico*
- *Template Matching Deformabile*

Approccio statistico

Si scelgono una serie di n caratteristiche (*features*) dell'apparenza di un oggetto che siano facilmente misurabili.

Generalmente si tratta di statistiche *globali* sui pixel appartenenti all'interno o ai bordi della sagoma dell'oggetto O.

O è descritto tramite un punto in $\mathbb{R}^n$ dato dal vettore $V(O)$ delle n misurazioni delle sue features.

Il database viene ordinato in base alla distanza tra lo sketch S e le immagini O_i in esso contenute:

$dist(V(O_i), V(S))$.

Esempi tipici di features

L’area della figura, cioè il numero dei suoi pixel.

La compattezza della regione, definita come il rapporto tra il quadrato del perimetro e l’area.

L’allungamento (*elongatedness*) della regione, definito come il rapporto tra la lunghezza della corda di lunghezza massima e la lunghezza della corda ad essa perpendicolare.

I *coefficienti dell'espansione in serie di Fourier* ottenuti interpretando la sagoma come fosse una funzione sinusoidale. La sagoma, infatti, è una curva chiusa di perimetro L, e un suo generico punto può essere descritto dalle coordinate $x(l)$ e $y(l)$, $0 \leq l \leq L$, da cui:

$$\phi(l) = x(l) + jy(l). \tag{1}$$

$\phi(l)$ è una funzione periodica la cui espansione tramite serie di Fourier fornisce i coefficienti della codifica.

I *momenti digitali*. Se f è la funzione binaria di un'immagine, una qualsiasi figura può essere espressa come:

$$S = \{(x, y)|f(x, y) = 1\}. \tag{2}$$

Per ogni paio di interi non negativi (j, k), il *momento digitale* (j, k)-esimo di S è dato da:

$$M_{jk}(S) = \sum_{(x,y)\in S} x^j y^k. \tag{3}$$

È facile constatare che $M_{00}(S)$ corrisponde all'area di S.

Il *centro di gravità* o *centroide* di S può essere rappresentato mediante le seguenti coordinate:

$$\overline{x} = \frac{M_{10}(S)}{M_{00}(S)} \tag{4}$$

$$\overline{y} = \frac{M_{01}(S)}{M_{00}(S)}. \tag{5}$$

A partire da queste definizioni vengono costruiti altri momenti con caratteristiche di invarianza rispetto a traslazioni, rotazioni e/o scalamenti.

Esempio: Recupero di immagini di volti umani

Da un archivio di immagini ($I^1, I^2, ...$), ognuna rappresentante il volto di un individuo (in ”formato tessera”), voglio recuperare, se esiste, l’immagine dell’individuo a cui appartiene una data, nuova immagine in input (Q).

Anzitutto ogni immagine (del database o di query) processata dal sistema deve essere analizzata al fine di estrarne la parte rappresentante il volto (*face detection*).

Recupero di immagini di volti umani: Rappresentazione delle Features

Features: il valore di grigio dei pixel dell'immagine.

Se I è un generica immagine processata dal sistema (o una sua sotto-finestra) dalle dimensioni $n \times m$, allora I può essere rappresentata dalla concatenazione delle sue n righe:

$V(I) = (I_{11}, I_{12}, ..., I_{1m}, I_{21}, I_{22}, ...I_{nm})^T$.

$V(I)$, a sua volta, può essere rappresentato da un punto in $\mathbb{R}^{n*m}$.

Il sistema restituisce l'immagine più vicina al punto $V(Q)$ (*nearest neighbour*).

Recupero di immagini di volti umani: Compressione

Di solito $\mathbb{R}^{n*m}$ è uno spazio troppo grande e viene proiettato in uno spazio di dimensioni ridotte detto *face space*.

Tale proiezione avviene esaminando un insieme di immagini d'esempio (*training set*) al fine di stabilire quali sono le informazioni statistiche maggiormente discriminanti.

La *Principal Component Analysis* (PCA) è l'esempio più noto di tecnica di compressione.

Il vettore di features $V_{1 \times nm}$ viene rappresentato da un secondo vettore W di dimensioni $N \times 1$, con $N << nm$.

W, pur essendo più piccolo rispetto a V, viene costruito in modo da mantenere la maggior parte dell'informazione statisticamente rilevante.

Il feature space di dimensione N viene costruito esaminando il training set e scegliendo una nuova base ortonormale B di $\mathbb{R}^{n*m}$ tale che i vettori di B indichino, in ordine decrescente, le direzioni di massima varianza nel training set.

Una base di dimensione N viene scelta selezionando i primi N elementi di B.

Recupero di immagini di volti umani: Machine Learning

Se si suppone che per ogni individuo umano rappresentato nel database di volti siano disponibili più di un'immagine, è allora possibile pensare di adottare tecniche di *Machine Learning statistico* per permettere al sistema di *generalizzare* l'apparenza di ogni individuo.

Supponiamo che la persona P_1 sia rappresentata dalle immagini $I_{i_1}, I_{i_2}, ...$ del database e, quindi, dai punti $W_{i_1}, W_{i_2}, ...$ del face space.

Le *Reti Neurali*, le *Support Vector Machine* o altre tecniche di apprendimento supervisionato possono pemettere al sistema di costruire le classi $C_1, C_2, ...$, corrispondenti alle persone $P_1, P_2, ...$

Ogni classe è ottenuta raggruppando opportunamente i punti del face space ed è rappresentata da una regione multidimensionale (non necessariamente connessa) del face space.

La regione R_1 corrispondente alla persona P_1 è una generalizzazione degli esempi $W_{i_1}, W_{i_2}, \ldots$ ed è una rappresentazione delle possibili apparenze di P_1.

Una *nuova* immagine Q rappresentante una delle persone note viene assegnata alla persona P_k se $W(Q) \in R_k$.

Indexing

L'indicizzazione di un database secondo criteri che rendano efficiente il recupero on-line delle informazioni in esso contenute è sempre un'operazione importante se le dimensioni del database sono critiche.

I sistemi di Image Retrieval permettono l'indexing raramente.

Gli approcci statistici sono un'importante eccezione.

Se i dati possono essere rappresentati vettorialmente è infatti possibile organizzare il database sfruttando le relazioni spaziali tra i vettori.

Le tecniche più note si basano su strutture dati ad albero quali: il *k-d-Tree* o l'*R-Tree*.

Indexing: il k-d-Tree

Il k-d-Tree è una generalizzazione dell'albero di ricerca binario in k dimensioni.

Supponiamo di voler indicizzare un insieme di vettori di features $V_1, ..., V_N$, tali che $V_i \in \mathbb{R}^k$ e $V_i = (v_1^i, ..., v_k^i)^T$.

Scelgo anzitutto la prima feature, corrispondente alla prima componente di ogni vettore e trovo il valore f_1, mediano rispetto a $v_1^1, v_1^2, ..., v_1^N$.

Indexing: il k-d-Tree

La radice dell'albero conterrà f_1 e sarà associata all'elemento $V_j = (f_1, ...)$.

Nel sottoalbero a sinistra memorizzerò i vettori V_i tali che $v_1^i \leq f_1$ e nel sottoalbero destro gli altri.

A livello 1 scelgo la feature numero 2 e calcolo f_2, e così via.

Dopo aver utilizzato la feature k-esima, ritorno ciclicamente a consisderare la prima feature, finchè tutti gli elemnti $V_1, ..., V_N$ sono stati assegnati a qualche nodo dell'albero.

Vantaggi e svantaggi dell'approccio statistico

Permette l'*indexing* del database.

Permette eventualmente di utilizzare tecniche di *Machine Learning*.

Scarso potere discriminante.

Non permette che l'oggetto d'interesse sia occluso o in contatto con altri oggetti nella scena.

Template matching Deformabile

Gli approcci di questo filone si basano sul tentativo di far ”combaciare” (*allineare*) lo sketch disegnato dall’utente con (una porzione de) l’immagine attualmente analizzata dal sistema.

Tale tentativo di allineamento avviene deformando progressivamente lo sketch iniziale per adattarlo come se fosse un “elastico” alle silhouette delle immagini in memoria.

Il processo iterativo termina o quando si raggiunge una sovrapposizione accettabile tra lo sketch deformato e l’immagine o quando il grado di deformazione applicato allo sketch supera un certo valore massimo.

Esempio: Elastic Template Matching (Del Bimbo-Pala)

Le immagini vengono inserite nel database del sistema nella forma contenente solo gli edge.

L'utente disegna il suo sketch usando un tool grafico e la sagoma finale viene rappresentata con una spline codificata mediante i suoi punti di controllo: $P = (p_1, ..., p_n)$, $p_i = (x_i, y_i)$.

Se il matching tra lo sketch e l'immagine candidata è elevata, la procedura termina qui.

Altrimenti, i vari p_i vengono "perturbati" in modo da modificare lo sketch e re-iterare la comparazione.

Più esattamente, il grado di matching tra lo sketch e l'immagine è definito da:

$$M(P) = C(P) - D(P),$$

Dove C e D sono delle funzioni, rispettivamente, del grado di sovrappozione e di deformazione dello sketch e:

$$D(P) = S(P) + B(P),$$

essendo S e B funzioni del grado di tensione e di curvatura dello sketch.

Ricerca dei massimi di M

Cerco un massimo (locale) di M trovando i punti in cui il gradiente ($\nabla M(P)$) si annulla.

$\nabla M(P) = 0$ può essere risolto analiticamente quando ∇M è sufficientemente semplice.

Altrimenti posso ricorrere ad un metodo iterativo.

Metodi iterativi

Si parte da una soluzione valida P_0.

Si procede ”perturbando” P:

1. $P' := P + \delta P$
2. $P := P'$ se $M(P') > M(P)$.

Metodo del gradiente ascendente

Si modificano progressivamente i p_i nella direzione di crescita massima ottenuta derivando M rispetto a P:

$$P(k+1) = P(k) + \nabla M(P(k)).$$

Conclusioni sul tamplate matching elastico

Qualsiasi sia la tecnica di ricerca dei massimi adottata, la necessità di una soluzione iniziale P_0 (lo sketch sovrapposto ”manualmente” sull’immagine) porta ad una mancata indipendenza da roto-traslazioni e scalamenti.

Ciò costringe a fornire le immagini del database di una *segmentazione* manuale di tutti i possibili oggetti di interesse (ad esempio tramite il *minimo rettangolo includente*), oppure a iterare il metodo per valori iniziali diversi di P_0.

Alcuni libri a carattere introduttivo

Image Retrieval: [2].

Libri introduttivi per la Computer Vision: [1, 7, 6, 4, 8, 3, 5] (si consiglia [5]).

[1] BALLARD, D. H., AND BROWN, C. M. *Computer Vision*. Prentice-Hall, 1982.

[2] DEL BIMBO, A. *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc. San Francisco, California, 1999.

[3] EDELMAN, S. *Representation and recognition in vision*. MIT Press. Cambridge, Massachusetts, 1999.

[4] FAUGERAS, O. *3–D Computer Vision, A Geometric Viewpoint*. The MIT Press–Cambridge Massachusetts–London, England, 1996.

[5] FORSYTH, D. A., AND PONCE, J. *Computer Vision: A Modern Approach*. Prentice Hall, 14 August, 2003, ISBN: 0130851981, 2003.

[6] HARALICK, R. M., AND SHAPIRO, L. G. *Computer and Robot Vision*. Vol 2. Addison–Wesley Publishing Company, 1993.

[7] SHAPIRO, L., AND STOCKMAN, G. *Computer Vision*. Prantice hall, 2001.

[8] ULLMAN, S. *High–level Vision. Object Recognition and Visual Cognition*. A Bradford Book. The MIT Press Cambridge, Massachusetts, 1996.