Corri cane, corri!

La scorsa primavera mi sono trovato a cena con il Prof. Antonio Grasselli, uno dei "padri fondatori" dell'informatica italiana, attualmente titolare di una cattedra di biomatematica alla facoltà di veterinaria dell'Università di Parma. Verso la fine della cena Antonio mi ha mostrato alcuni grafici che rappresentavano gli angoli sottesi alle gambe di un cane in corsa chiedendomi consiglio sul modo di individuarne un modello matematico. Dopo una buona cena mi sentivo in grado di analizzare qualunque cosa ed ho fatto subito l'incauta proposta di iniziare una collaborazione sull'argomento. L'articolo che segue è una presentazione abbastanza scherzosa dei primi risultati

di Francesco Romani

Le misure sul cane

Si prende un cane e, con il velcro, si fissano sulla pelle delle sfere foto-riflettenti in corrispondenza delle articolazioni. Si mette il cane su di un tappetino scorrevole e lo si convince a correre (con una bistecca?). Una forte luce illumina la scena ed una telecamera professionale riprende la scena a 50 fotogrammi al secondo. La ripresa viene inviata ad un dispositivo hardware specializzato che produce due sequenze di dati: le coordinate dei punti luminosi e gli angoli compresi tra coppie di arti consecutivi.

Tali dati possono venire ricombinati per costruire lo "scheletro" del cane in movimento (la cosiddetta *steak figure*). La **figura 1** come tutte le altre in questo articolo è stata generata con *Mathematica*.

Il problema consiste nell'analizzare il movimento del cane dal punto di vista matematico. Le possibili applicazioni di una simile analisi sono sia lo studio teorico dei movimenti del cane (e di tutti gli animali a cui si può applicare una simile tecnica), sia lo studio clinico di come corre "quel" cane (per esempio per vedere come sta guarendo dopo una operazione). Per entrambi gli scopi è utile disporre di funzioni matematiche che permettano di esprimere il movimento del cane normale (magari distinte in base alla razza, alla taglia, al tipo di corsa).

Un primo approccio, più corretto e impegnativo, è quello della scrittura di un modello matematico del moto del cane da cui sia possibile derivare un insieme di equazioni differenziali la cui soluzione dia le funzioni cercate.

Un altro approccio più semplice consite nell'interpolare e mediare i movimenti di cani reali alla ricerca di un'espressione approssimata delle stesse funzioni. Questa strada (che è quella che abbiamo seguito) è più semplice della precedente e può servire da introduzione ad uno studio più completo del problema.

Si deve tener presente che le misure che si ottengono mediante il video-sistema sono soggette ad errori. Innanzitutto, poiché la telecamera registra 50 immagini al secondo, vi è un errore di campionamento nella ricostruzione delle curve: non è quindi possibile analizzare eventi molto veloci (per esempio, un cavallo al galoppo). Inoltre, vi sono errori

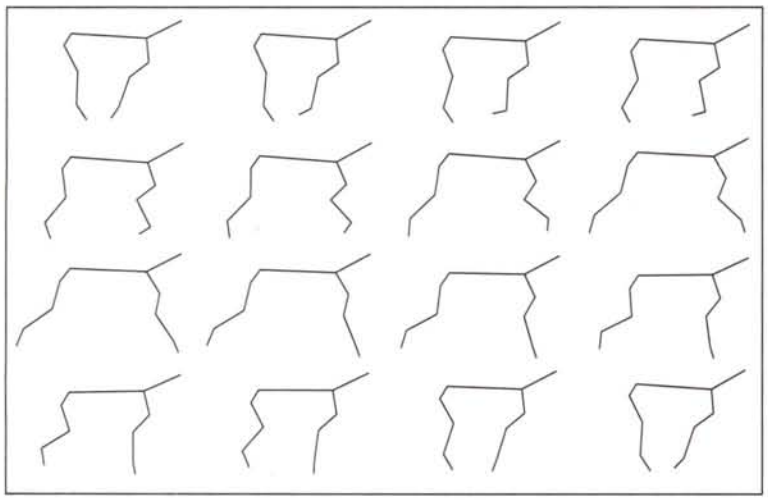

Figura 1

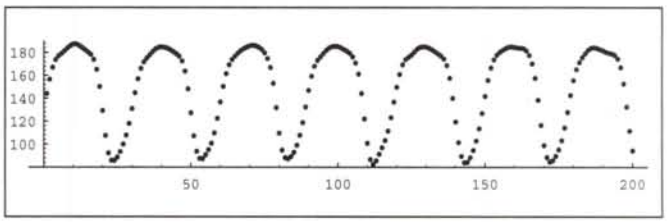

Figura 2

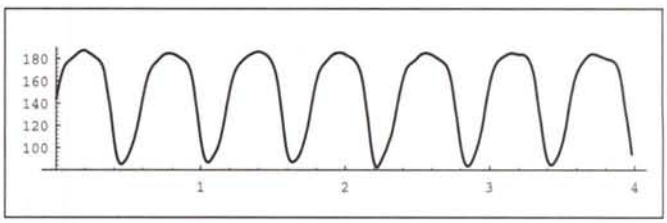

Figura 3

dovuti al sistema automatico di rilevamento delle marche, che però sono modesti se le marche sono bene illuminate e di dimensioni opportune. Gli errori più importanti sono sicuramente quelli dovuti al movimento della pelle su cui sono applicate le marche. Una ulteriore causa di variabilità dei risultati deriva dalla posizione delle marche: si può sperimentare che se le marche vengono posizionate due volte sullo stesso animale, nonostante l'operazione venga effettuata dallo stesso operatore, si ottengono risultati diversi.

Primo trattamento dei dati

L'idea base della nostra analisi è quella di determinare un insieme di funzioni che rappresentano nel modo migliore il movimento del cane. Ignorando la testa i dati grezzi sono 20 sequenze di 200 numeri che rappresentano le coordinate. Applicando le regole della trigonometria si ottengono 6 sequenze di 200 numeri che rappresentano i 6 angoli. Un grafico di questi si presenta come in **Figura 2.** Il primo passo è trasformare questi insiemi di valori discreti in un insieme di curve. *Mathematica* fornisce l'interpolazione con le spline cubiche che permette di approssimare la curva più "dolce" che passa per quei punti. Se **xx** è l'insieme delle ascisse e **yy** quello delle corrispondenti ordinate la funzione

In[1]:=
```
Interpolation[Transpose[{xx,yy}]]
```

rappresenta la spline cubica che passa per quei punti.

La **Figura 3** mostra uno degli angoli dopo la cura.

Determinazione del periodo

L'insieme dei dati raccolti copre circa 4 secondi e in questo tempo il cane riesce ad effettuare più di una serie completa di movimenti (circa 7 nel nostro esempio): è dunque importante determinare il periodo medio in modo da approssimare uno e uno solo dei cicli. Dopo una prima grossolana approssimazione basata sulla velocità del tappetino si innesca un ciclo di minimizzazione dell'area compresa tra due spezzoni di curva che vengono fatti scivolare tra loro fino ad ottenere una buona sovrapposizione. Questa fase viene realizzata sfruttando il minimizzatore di *Mathematica* **FindMinimum.** Le **Figure 4** e **5** mostrano due momenti di questa sovrapposizione, i due numeri nel titolo del grafico rappresentano il periodo stimato e il valore attuale dell'area che viene minimizzata.

Interpolazione trigonometrica

Quando si dispone del periodo t è possibile effettuare un nuovo campionamento questa volta con intervalli equidistanti sottomultipli di t (il periodo stimato) e per un numero intero di periodi (6 nell'esempio che stiamo trattando). In questo modo si ottiene un nuovo insieme di valori discreti che ben si presta ad essere sottoposto alla analisi di Fourier senza grossi problemi.

La trasformata di Fourier dei nuovi dati campionati rappresenta uno sviluppo in serie di funzioni trigonometriche che interpola i nostri dati con un periodo pari a $6t$. Prendendo un valore ogni 6 del risultato della trasformata si effettua una "decimazione in frequenza" ovvero si ottiene una funzione trigonometrica del tipo

(1)

$$\frac{\alpha_0}{2}+\frac{\alpha_n}{2}\cos\left(\frac{2\pi n x}{t}\right)+ \\ +\sum_{j=1}^{n-1}\left(\alpha_j\cos\left(\frac{2\pi j x}{t}\right)+\beta_j\sin\left(\frac{2\pi j x}{t}\right)\right).$$

che approssima il movimento del cane su un periodo t facendo una sorta di media tra i vari periodi.

È possibile dimostrare che tra tutti i polinomi trigonometri della forma (1) quello ottenuto in questo modo è quello che minimizza la somma dei quadrati degli scarti tra i dati cam-

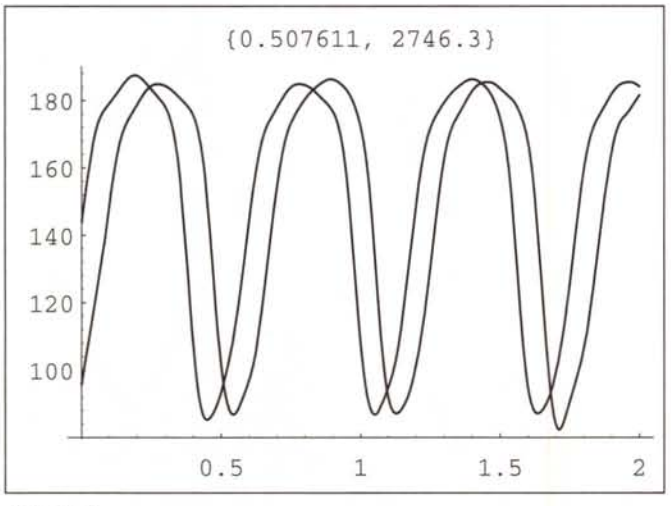

Figura 4

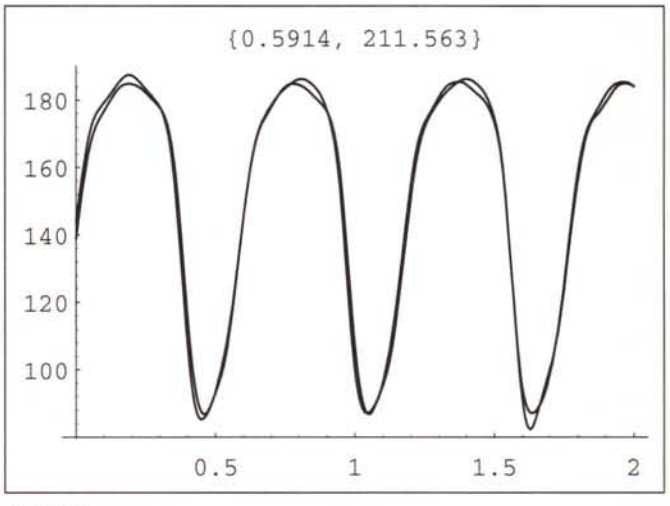

Figura 5

pionati e la funzione approssimante.

La **Figura 6** mostra le curve risultanti confrontate con l'insieme dei punti misurati effettivamente per tutti i 6 angoli. La scala delle ascisse rappresenta il tempo modulo t riportato all'intervallo (0,1).

Alla ricerca del cane "medio"

Applicando il metodo sopra descritto si ottengono, per ogni misura, 20 funzioni trigonometriche che descrivono le coordinate e 6 funzioni che descrivono gli angoli. La normalizzazione del periodo su (0,1) permette di confrontare (ad occhio) queste misure per cani diversi e per velocità diverse dello stesso cane.

Qualora si dispongano di più misure sullo stesso cane e alla stessa velocità del tappetino è possibile mediare per ottenere delle funzioni che definiscono un andamento medio di quel cane a quella velocità.

L'operazione va fatta sulle coordinate che rappresentando dei punti in uno spazio vettoriale (il piano cartesiano) soddisfano le proprietà di linearità necessare per fare la media. Per poter sommare le funzioni relative a cani diversi è necessario metterle prima in fase, ovvero farle tutte partire dallo stesso momento della corsa; questa operazione che potrebbe sembrare difficile è invece semplicissima, basta applicare ad ogni cane un ritardo temporale tale che nella equazione (1) si annulli il termine β_1.

Questo ritardo temporale si ricava direttamente dall'argomento del numero complesso $\alpha_1+i\beta_1$. Questa fasatura è fatta automaticamente per tutti i cani al momento della interpolazione trigonometrica.

Riassumendo, se disponiamo di più misure sullo stesso cane e alla stessa velocità del tappetino

- si fa la media delle funzioni che rappresentano le coordinate;
- si calcolano gli angoli corrispondenti alle coordinate medie;
- si calcola lo scarto quadratico medio dei dati in esame.

Nella **Figura 7** si vedono gli angoli medi di 6 misure con tre posizionamenti diversi delle marche. Le curve rosse sono le misure originali (interpolate) e l'area in colore rappresenta la media ± lo scarto quadratico medio.

Ossa di gomma?

L'ultimo problema che resta da affrontare è certamente il più difficile da risolvere.

Il movimento della pelle durante la corsa fa sì che la posizione delle marche non coincida con il vertice della articolazione e se si calcola la distanza tra due marche consecutive durante la corsa si vede che questa è ben lungi dall'essere costante. In altre parole le ossa del nostro cane trigonometrico si allungano e si accorciano come fossero di gomma.

La soluzione cruenta al problema, che consiste nel trapanare le ossa del cane e impiantarvi dei fotoemettitori, deve essere scartata per una infinità di ragioni che saranno ovvie

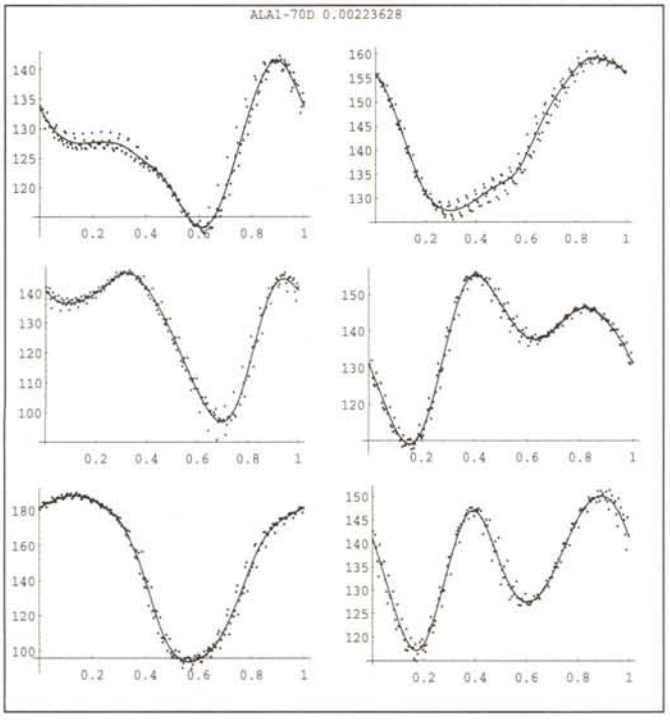

Figura 6

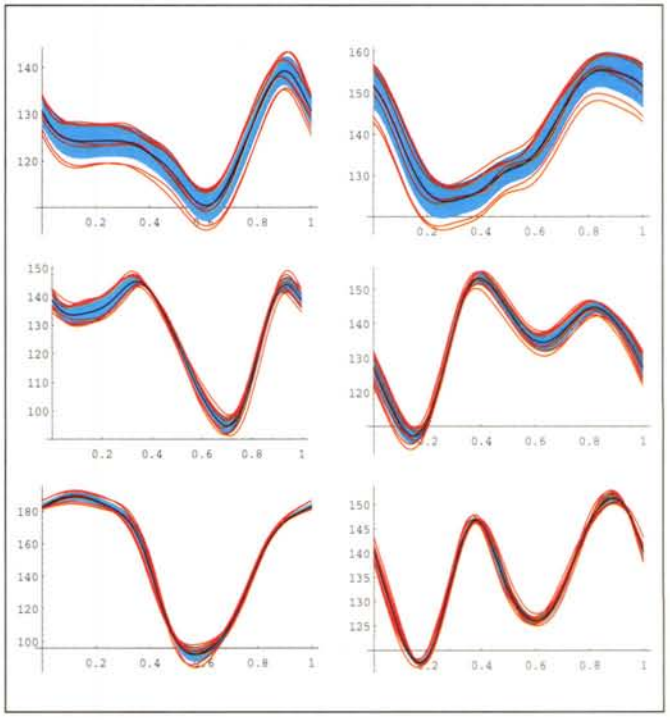

Figura 7

Attenzione allo stile

Riporto qui di seguito un semplice esempio (tratto da *MathUser* dell'autunno 1995) che mostra come uno stile di programmazione più *mathematico* possa drammaticamente ridurre i tempi di esecuzione. Il problema in esame è quello di aggiungere 1 a tutti gli elementi di una lunga lista. Costruiamo dapprima la lista ed effettuiamo la somma in vari modi. I tempi sono stati misurati su di un Macintosh 7100 con PPC 610 a 80 MHz.

Generiamo una lista di 20.000 numeri reali.

In[1]:=
```
lista=Table[Random[], {20000}];
Short[lista]
```

Out[2]//Short=
```
{0.456611,0.519746,<<19997>>,0.406799}
```

Il primo approccio usa il costrutto **For** (è la prima volta che lo adotto in un programma *Mathematica* e ne sconsiglio vivamente l'uso a tutti). Si crea una lista vuota e vi si mettono uno ad uno i nuovi elementi. Il risultato è scoraggiante.

In[3]:=
```
Timing[
 somma1={};
 For[i=1,
     i<=Length[lista],
     i++,
     AppendTo[somma1,lista[[i]]+1]]]
```

Out[3]=
```
{550.2 Second, Null}
```

Non si ottiene nulla di meglio a sommare uno a tutti gli elementi della lista.

In[4]:=
```
Timing[
 somma2=lista;
 Do[somma2[[i]]=somma2[[i]]+1,
    {i,Length[somma2]}];]
```

Out[4]=
```
{592.417 Second, Null}
```

Un tempo di esecuzione 100 volte migliore si ottiene creando con **Table** una nuova lista con la proprietà voluta.

In[5]:=
```
Timing[
 somma3=Table[lista[[i]]+1,
  {i,Length[lista]}];]
```

Out[5]=
```
{5.4 Second, Null}
```

Si può anche definire una funzione ed applicarla a tutti gli elementi.

In[6]:=
```
Timing[
 f[x_]:=x+1;
 somma4=Map[f,lista];]
```

Out[7]=
```
{5. Second, Null}
```

Ma il modo più semplice e diretto si ottiene ricordando che **Plus** possiede l'attributo **Listable**.

In[8]:=
```
Timing[somma5=lista+1;]
```

Out[8]=
```
{2.9 Second, Null}
```

Verifichiamo infine di avere ottenuto sempre lo stesso risultato.

In[9]:=
```
somma1==somma2==somma3==
 somma4==somma5
```

Out[9]=
```
True
```

anche ai lettori meno sensibili.

Invece agire sul cane trigonometrico non crea nessun problema di coscienza ma solo un problema di tempo macchina: l'idea consiste nell'esplorare lo spazio dei cani alla ricerca di un insieme di funzioni di movimento che siano vicine a quelle originarie ma con una minore variazione della lunghezza delle ossa.

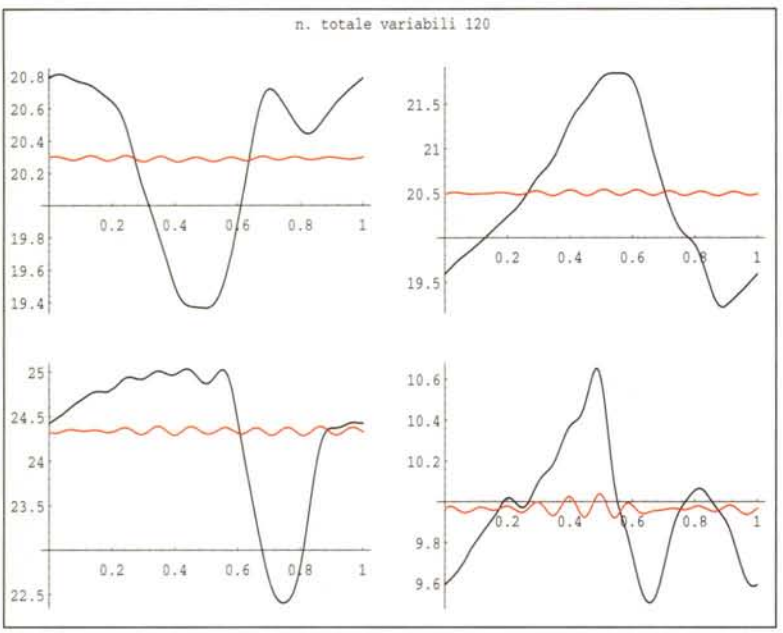

Figura 8

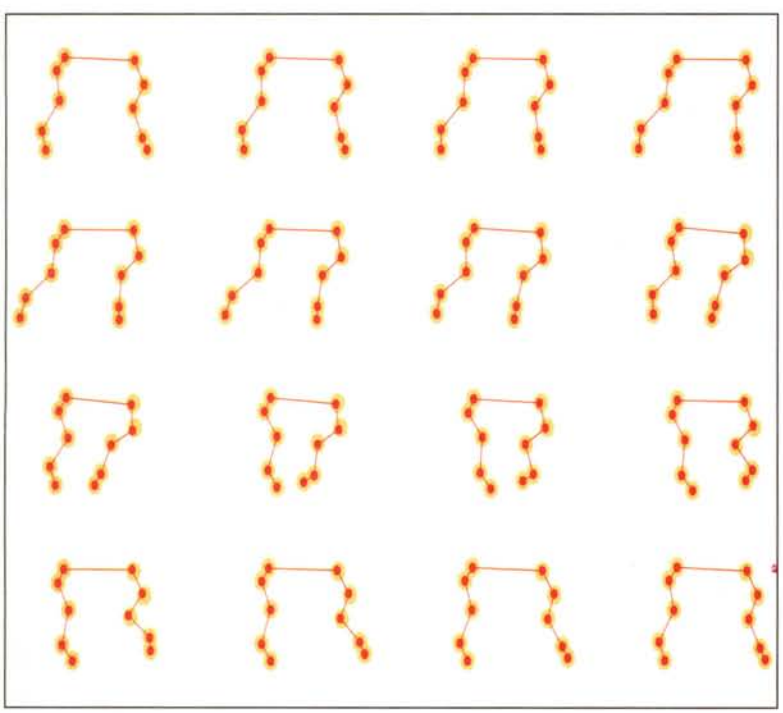

Figura 9

Il primo passo consiste nello sviluppare per via trigonometrica la funzione che esprime la lunghezza di un osso in funzione dei coefficienti di Fourier che determinano le coordinate.

Considerando questi coefficienti come variabili si potrebbe imporre che la lunghezza delle ossa fosse costante ovvero che il suo sviluppo in serie fosse composto dal solo termine costante.

Purtroppo una soluzione esatta non è determinabile con un numero finito di coefficienti e si è preferito considerare come variabili solo alcuni di essi e cercare di minimizzare la somma dei quadrati dei termini che avrebbero dovuto essere zero.

Combinando questa somma di quadrati con alcuni termini aggiuntivi si ottiene un polinomio di quarto grado in un centinaio di variabili i cui punti di minimo relativo approssimano dei cani vicini a quello sperimentale ma con una minore variazione della lunghezza delle ossa.

Nella **Figura 8** si vede la funzione lunghezza per le quattro ossa di una gamba prima (in nero) e dopo la cura (in rosso). In questo caso è stata minimizzata con *Mathematica* una funzione di 120 variabili.

Infine nella **Figura 9** si vedono 16 fotogrammi della corsa del cane prima (banana) e dopo la cura (in rosso).

Si noti che a fronte di una notevole riduzione della "elasticità" delle ossa il moto apparente del cane è rimasto invariato.

Conclusioni

Abbiamo visto un esempio di una applicazione scientifica abbastanza singolare realizzata interamente con *Mathematica*.

Gli stessi principi possono venire applicati ad altri problemi di approssimazione.

I dettagli matematici verranno descritti in una pubblicazione sottomessa ad una rivista specializzata (sarò lieto di fornire su richiesta maggiori dettagli).

MC

Francesco Romani è raggiungibile tramite Internet all'indirizzo romani@di.unipi.it

SOUND BLASTER 32 PLUG AND PLAY

...PER CHI VUOLE SCOPRIRE IL SUONO ALLO STATO PURO!

La nuova scheda Sound Blaster 32 Plug and Play di Creative Labs è facilissima da installare; basta inserirla nel computer e il gioco è fatto. Difficile immaginare qualcosa di più semplice! Ma non è tutto. La qualità e la purezza del suono della scheda raggiungono livelli inimmaginabili, se non in un auditorio. E allora, che cosa aspettate a comporre, orchestrare, arrangiare e creare musica a vostro piacimento? I vostri video giochi, poi, saranno accompagnati da un suono talmente realistico da farvi dimenticare che non si tratta della realtà! Con l'espandibile scheda Sound Blaster 32 Plug and Play di Creative Labs riceverete anche altri software, come quello per gli effetti sonori tridimensionali di Creative Labs che vi consente di sbizzarrirvi nella sonorizzazione delle attività musicali e dei giochi.

Con la nuova scheda Sound Blaster 32 Plug and Play, Creative Labs si afferma ancora una volta come lo standard per eccellenza!

CREATIVE

CREATIVE LABS

Il risveglio dei sensi

** Per DOS, Windows 3.1 e Windows 95.*

Se desiderate ricevere la documentazione completa, inviate il tagliando a **C2SI (Creative Labs) - Caselle Postale - Cascine Vicca 10090 - Italy.**

Cognome _____ Nome _____

Indirizzo _____

C.A.P. |_|_|_|_|_| Città _____